Title

Multi-Target Tracking in Surveillance Cameras

Author

Chen, Xiaojing

Publication Date

2015

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Multi-Target Tracking in Surveillance Cameras

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Xiaojing Chen

December 2015

Dissertation Committee:

Dr. Bir Bhanu, Chairperson
Dr. Chinya V. Ravishankar
Dr. Stefano Lonardi
Dr. Vagelis Hristidis

The Dissertation of Xiaojing Chen is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

Upon the completion of this work, I own my gratitude to a great number of people. I would first like to express my deepest gratitude to my advisor, Dr. Bir Bhanu, for his support and guidance during my PhD study. I would like to thank my committee members, Dr. Chinya Ravishankar, Dr. Stefano Lonardi and Dr. Vagelis Hristidis for their constructive comments to improve this work. I would also like to thank Dr. Tao Jiang and Dr. Subir Ghosh who were in my oral qualifying exam committee and gave me insightful feedback and suggestions.

I would like to thank Dr. Zhixing Jin, Dr. Songfan Yang, Dr. Linan Feng, Dr. Mehran Kafai, Dr. Yiming Li, Dr. Yu Sun, Dr. Ninad Thakoor, Dr. Zhen Qin, Dr. Bing Hu, Dr. Shu Zhang and all of my other current and former colleagues, friends at University of California, Riverside and other places who have offered me help, support, and joy. Special thanks go to my husband and colleague, Dr. Le An, who not only accompanied but also participated in every journey of my life from Europe to the United States, without him I would not have gained so much extraordinary and valuable experience. I would also like to thank all of my family members, including Xizi Chen, who gave me support and encouragement as they always do.

The materials of some chapters in this dissertation have appeared in "An Online Learned Elementary Grouping Model for Multi-target Tracking" © 2014 IEEE by X. Chen *et al* and "Multi-Target Tracking in Non-overlapping Cameras Using a Reference Set" © 2014 IEEE by X. Chen *et al.*

I dedicate this Dissertation,
To my father, Heyu Wang, and my mother, Guangyan Chen,
for their unconditional love, support, sacrifice, and encouragement.
To my husband, Le An, for his support, understanding, and endless love.
Without you, I would not have gone so far.

ABSTRACT OF THE DISSERTATION

Multi-Target Tracking in Surveillance Cameras

by

Xiaojing Chen

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2015
Dr. Bir Bhanu, Chairperson

As the number of surveillance cameras deployed in public areas increasing rapidly, automatic multi-target tracking in both a single camera and multiple non-overlapping cameras have been receiving great interest. The goal of multi-target tracking is to recover the trajectories of all moving targets while maintain their identities consistent. Although this problem has been studied for several years, there still remain many challenges, such as illumination and appearance variation, occlusion, sudden change in motion, and unpredictable motion across cameras. Driven by necessity for multi-target tracking in surveillance cameras, in this dissertation, we proposed several tracking methods.

First, we designed a framework for multi-target tracking in a single camera. Unlike previous methods that only rely on low-level information, and consider each target as an independent agent, in this dissertation, an online learned social grouping behavior model is used to provide more robust tracklets affinities. A disjoint grouping graph is used to encode social grouping behavior of pairwise targets, where each node represents an elementary group of two targets, and two nodes are connected if they share a common target. Probabilities of the uncertain target in two connected nodes being the same person are inferred from each edge of the grouping graph. Second, a novel reference set based appearance model is developed to improve multi-target tracking across cameras. A reference set is constructed for a pair of cameras, containing subjects appearing in both camera views. For track association, instead of directly comparing the appearance of two targets in different camera views, they are compared indirectly via the reference set. Third, we extend the single camera multi-target tracking framework with social grouping behavior to a network of non-overlapping cameras. The tracking problem is formulated using an online learned Conditional Random Field (CRF) model that minimizes a global energy cost. During intra-camera tracking, track associations that maintain single camera grouping consistencies are preferred.

To validate the proposed methods in this dissertation, extensive experiments on several datasets are conducted. Results show that each of the aforementioned method achieves state-of-the-art performance in various multi-target tracking tasks.

Contents

List of Figures **x**

List of Tables **xiv**

1 Introduction **1**

2 Multi-person Tracking by Online Learned Grouping Model with Non-linear Motion Context **3**

2.1 Introduction . . . 3

2.2 Related Work and Contributions . . . 6

2.2.1 Related Work . . . 6

2.2.2 Contributions of This Chapter . . . 9

2.3 Technical Approach . . . 9

2.3.1 Tracking Framework with Grouping . . . 9

2.3.2 Learning of the Elementary Groups . . . 11

2.3.3 Group Tracking . . . 13

2.3.4 Creation of Virtual Nodes . . . 17

2.3.5 Inference from the Grouping Graph . . . 17

2.4 Experiments . . . 19

2.4.1 Implementation Details . . . 21

2.4.2 Results on CAVIAR Dataset . . . 23

2.4.3 Results on TownCentre Dataset . . . 24

2.4.4 Results on PETS2009 Dataset . . . 24

2.4.5 Results on UNIV Dataset . . . 27

2.4.6 Computational Time . . . 27

2.5 Conclusions . . . 29

3 Multi-Target Tracking in Non-overlapping Cameras Using a Reference Set **30**

3.1 Introduction . . . 30

3.2 Related Work and Contributions . . . 33

3.2.1 Related Work . . . 33

3.2.2 Contributions of This Chapter . . . 34

3.3 Technical Approach . . . 35

3.3.1 Formulation of the Multi-Camera Tracking Problem 35
3.3.2 Reference Set Based Appearance Model for Across Camera Tracks . 36
3.3.3 Track Division 39
3.3.4 Appearance Model for Within Camera Subtracks 40
3.4 Experiments 42
3.4.1 Implementaion Details 42
3.4.2 Baseline Models and Metrics 43
3.4.3 Evaluation of Feature Learning Algorithm 43
3.4.4 Results on MultiCam dataset 44
3.4.5 Results on VideoWeb dataset 47
3.4.6 Reference Set Analysis 52
3.4.7 Feature Discriminality Analysis 53
3.5 Conclusions 55

4 Integrating Social Grouping Behavior for Multi-target Tracking Across Cameras in a CRF Model 56

4.1 Related Work and Contributions 60
4.1.1 Related Work 60
4.1.2 Contributions of This Chapter 61
4.2 Technical Approach 62
4.2.1 CRF Model for Inter-camera Tracking 62
4.2.2 CRF Graph Creation 63
4.2.3 Unary Energy Functions 65
4.2.4 Pairwise Energy Functions 68
4.2.5 Energy Minimization Algorithm 71
4.3 Experiments 73
4.3.1 Datasets 73
4.3.2 Evaluation Metrics 76
4.3.3 Experimental Results 77
4.4 Conclusions 78

5 Conclusions 82

Bibliography 84

List of Figures

2.1 Examples in which grouping information is helpful under the challenging conditions for tracking in a video. The same color indicates the same target. Note that for both targets with bounding boxes there are significant appearance and motion changes due to occlusions and cluttered background. Images are from CAVIAR dataset [1] 5
2.2 Overview of the elementary grouping model. 6
2.3 Block diagram of our tracking system. After initial tracklets are generated by linking detection responses, confident tracklets are selected to form elementary groups. The relationships between elementary groups are identified by group tracking with non-linear motion context. Then a disjoint grouping graph is constructed, from which high level information (i.e., grouping behavior) is extracted. Finally, tracklet association is carried out based on affinity model that combines both high level and low level information. Tracklets with the same color contain the same target. For the legends in this figure please see the box in the upper right hand side. Best viewed in color. 7
2.4 Examples of generating incorrect elementary groups if the distances are not normalized. 13
2.5 An example of estimating motion affinity using the non-linear motion map. 16
2.6 Inference for each edge in the grouping graph in Fig. 2.3: (a) edge between g_1 and g_2, (b) edge between g_2 and v_1, (c) edge between g_2 and v_2, (d) edge between g_4 and g_5 (see Fig. 2.3 for group annotations). Black solid line represents interpolation between the two tracklets that need inference, black dashed line is the group mean trajectory, and colored dotted line indicates a virtual tracklet. Best viewed in color. 18
2.7 An example of multiple inferences related to the same two tracklets. According to the proposed elementary grouping model, a grouping graph (shown on the right) is created based on the input tracklets (shown on the left). Thus, inferences based on the edge between node g_1 and g_2 and the edge between node g_3 and g_4 are all related to tracklets T_2 and T_3. 19
2.8 Examples of tracking results of our approach on CAVIAR dataset. The same color indicates the same target. Best viewed in color. 25

2.9 Examples of tracking results of our approach on TownCentre dataset. With grouping information, targets (199 and 201) pointed by arrows are correctly tracked under frequent occlusions. The same color indicates the same target. Best viewed in color. 26

2.10 Examples of tracking results of our approach on PETS2009 dataset. Track targets (47, 51, 69) with non-linear motion successfully. Best viewed in color. 26

2.11 Examples of tracking results of our approach on UNIV dataset. Using only the grouping model, we correctly tracked targets (1, 2, 3, 4) in situations where the group split and merge occur. The same color indicates the same target, best viewed in color. 28

2.12 Examples of tracking results from referenced models. First row, Our Model (linear) on PETS2009 dataset. Targets (48, 72) cannot be correctly tracked, as tracklet associations generating non-linear motion pattern are penalized when only linear motion model is used. Second row, SGB model on UNIV dataset. Trajectories of targets (1, 2) cannot be fully recovered, because SGB model is not able to link tracklets that are not assigned to the same group. Best viewed in color. 28

3.1 Sample frames from each camera view of the MultiCam dataset. Bounding boxes with the same color indicate the same target. Note that illumination may change drastically within camera and across cameras. As a result, the appearance of the same target may vary significantly. 31

3.2 Illustration of the reference set based appearance model. For a pair of cameras C_i and C_j, a reference set $RefSet_{ij}$ (the middle part) is constructed containing a number of reference subjects appearing in both C_i and C_j. When comparing track T_1 in C_i with tracks T_2 and T_3 in C_j using their color histograms directly, T_3 is more likely to be matched with T_1. Even though they contain totally different targets, the significant illumination change in C_i makes T_1 looks much darker than its actual appearance. Instead of comparing the tracks directly, each input track is described by all the reference subjects. The description is a vector of similarities ordered by the identities of reference subjects, and each similarity is generated by comparing the input track with one reference subject. The right part of this figure shows the similarity plots obtained by comparing T_1, T_2, T_3 with R_1, R_2, and R_3, respectively. Note that both the input and the reference subjects have multiple appearance instances (only three instances are shown for illustration purpose) that cover the appearance changes of corresponding targets in a particular camera. This indirect match enables us to handle within camera illumination and pose variation. After representing T_1, T_2 and T_3 using the reference set (the right part), it is clear that T_1 is more similar to T_2 than to T_3. 37

3.3 An example of track division (only detections on key frames are shown). Detections in the same subtrack have higher appearance similarities as compared to the detections in other subtracks. 39

3.4 Features (HSV color histogram, LBP, HOG) are extracted from different locations of the detection response: torso, legs and body. The torso part is the upper half of the detection and the legs part is the lower half of the detection. 41
3.5 Detection examples of participants that appear in both the reference set and the test set for MultiCam dataset. 44
3.6 Topology for cameras used in MultiCam dataset. 45
3.7 Comparison of the proposed method and other baseline models on MultiCam dataset. The minus sign (-) indicates no camera topology knowledge is used for linking cost computation. 47
3.8 Topology for cameras used in VideoWeb dataset. 48
3.9 Detection examples of participants that appear in both the reference set and the test set for VideoWeb dataset. The "None" box indicates the corresponding participant (P4) generates no track in camera 20 for the reference set. 49
3.10 Comparison of the proposed methods and other baseline models on VideoWeb dataset. The minus sign (-) indicates no topology knowledge is used for linking cost computation. 49
3.11 Example tracking results on MultiCam dataset. The first and the third columns are the results obtained by using BTF in [63], the second and the fourth columns are the results by the proposed method using reference set based appearance model with weighted features (RefSet2). With the reference set, our method is able to match most of the targets even with the presence of drastic within camera and across camera illumination variations. The method in [63] fails to associate tracks that contain the same target in these challenging situations. Best viewed in color. 50
3.12 Example tracking results on VideoWeb dataset. The first and the third columns are the results obtained by using BTF in [63], the second and the fourth columns are the results by the proposed method using reference set based appearance model with weighted features (RefSet2). Best viewed in color. 51
3.13 Feature discriminality analysis for MultiCam dataset. Each column represents the RMSE (representing discriminality) when the corresponding feature is *removed* from feature pool of the appearance model. 54
3.14 Feature discriminality analysis for VideoWeb dataset. Each column represents the RMSE (representing discriminality) when the corresponding feature is *removed* from the appearance model. 55

4.1 Tracking results of our proposed model on Dataset4. Bounding boxes with the same color indicate the same person, and the dashed lines illustrate the trajectories generated by targets walking across different cameras. 57
4.2 Block diagram of our tracking system resented with a simple illustrative example. Blocks shown in red contain novelty part of this chapter. Tracklets with the same color contain the same target. Best viewed in color. For the legends in this figure please see the box in the upper right hand side. 59

4.3 An example of applying the color transfer method on images obtained by two different cameras (one outdoor, one indoor). The full images captured by each camera are shown in the first row. The person appears in both cameras and its corresponding HSV color histograms are presented in the second row. It is obvious that the person in the transformed image is more alike to the person in the target image based on HSV color histograms. . . 67

4.4 Local patches with various scales are defined at different locations of a detection. Patches 1 to 6 have the same size and are served as basic patches, patches 7 to 14 are different combinations of basic patches, and patch 15 captures the middle third region of a detection. 70

4.5 Camera topology for Dataset1 and Dataset2. Cam1 and Cam2 are outdoor cameras, and Cam3 is a indoor camera. 74

4.6 Camera topology for Dataset3. Cam1 to Cam4 are all indoor cameras. . . . 74

4.7 Camera topology for Dataset4. Cam1 to Cam5 are all outdoor cameras. . . 75

4.8 A visual comparison of our model (the first row) and Baseline1 (the second row) on Dataset1. It is observed that Baseline1 mistakenly identifies a new target in Camera 3 (the one pointed by arrow) as target 3, while our model avoid this error by maintaining the group consistency between target 3 and 4. Bounding box with the same color indicates the same target. Best viewed in color. 79

4.9 A visual comparison of our model (the first row) and Baseline2 (the second row) on Dataset2. In the result of our model, target 28 and 29 are correctly tracked in all cameras. But their IDs are switched in Camera 3 in the result of Baseline2, due to the lack of local appearance consistency. Bounding box with the same color indicates the same target. Best viewed in color. 80

4.10 Sample tracking results of our proposed method on Dataset3. In the first row, by taking advantage of the grouping information, target 55 and 56 are successfully tracked in all cameras, even under significant within and across camera illumination changes. In the second row, target 162 in Camera 4 is not correctly link to the same target (target 175) in Camera 3. This target is severely occluded by target 161 in Camera 3, even with group information we are unable to link them, as such association does not maintain appearance consistency. Bounding box with the same color indicates the same target. Best viewed in color. 81

4.11 Sample tracking results of our proposed method on Dataset4. Target 30 is correctly tracked in all cameras. However, target 36 in Camera 3 is mistakenly linked to another target in Camera 4 (pointed by green arrow). Since both of them form an elementary group with target 30, and are visually very similar. Bounding box with the same color indicates the same target. Best viewed in color. 81

List of Tables

2.1 Comparison of tracking results on CAVIAR dataset. The number of trajectories in the ground-truth (GT) is 75. 23

2.2 Comparison of tracking results on TownCentre dataset. The number of trajectories in the ground-truth (GT) is 220. 24

2.3 Comparison of tracking results on PETS2009 dataset. The number of trajectories in the ground-truth (GT) is 74. 26

2.4 Comparison of tracking results on UNIV dataset. The number of trajectories in the ground-truth (GT) is 40. 27

3.1 RMSE comparison of different feature learning algorithms on MultiCam and VideoWeb datasets. 44

3.2 Tracking results with different reference set sizes. "N" is the number of reference subjects in the original reference set. "Match" and "Error" stand for match rate and error rate. 52

3.3 Tracking results with and without participants appearing in both reference and test sets. The asterisk sign (*) indicates dataset without identity overlap in reference and test sets. 53

4.1 Specifics for each sub-dataset in the NLPR_MCT dataset. 75

4.2 The number of cross camera true positive in each sub-dataset. 76

4.3 Comparison of cross camera tracking results on the NLPR_MCT dataset. . 77

Chapter 1

Introduction

For multi-target tracking in a single camera, most existing data association-based tracking approaches only use low-level information (e.g., time, appearance, and motion) to build the affinity model, and consider each target as an independent agent. In Chapter 2, we introduce a novel approach to learn possible elementary groups (groups that contain only two targets) online for inferring high-level context that can be used to improve multi-target tracking in a data-association based framework. Social grouping behavior of pairwise targets is first learned from confident tracklets and encoded in a disjoint grouping graph. Relationships between elementary groups are discovered by group tracking, and a non-linear motion map is used for explaining non-linear motion pattern between elementary groups. The proposed method is efficient, able to handle group split and merge, and can be easily integrated into any basic affinity model. The approach is evaluated on four datasets, and it shows significant improvements compared with state-of-the-art methods.

Tracking multiple targets across non-overlapping cameras aims at estimating the trajectories of all targets, and maintaining their identity labels consistent while they move from one camera to another. As the observations of the same targets are often separated by time and space, there might be significant appearance change of a target across camera views caused by variations in illumination conditions, poses, and camera imaging characteristics. Consequently, the same target may appear very different in two cameras. Therefore, associating tracks in different camera views directly based on their appearance similarity is difficult and prone to error. In most previous methods the appearance similarity is computed either using color histograms or based on pre-trained Brightness Transfer Function (BTF) that maps color between cameras. In Chapter 3, a novel reference set based appearance model is proposed to improve multi-target tracking in a network of non-overlapping cameras. Contrary to previous work, a reference set is constructed for a pair of cameras, containing subjects appearing in both camera views. For track association, instead of directly comparing the appearance of two targets in different camera views, they are compared indirectly via the reference set. Besides global color histograms, texture and shape features are extracted at different locations of a target, and AdaBoost is used to learn the discriminative power of each feature. The effectiveness of the proposed method over the state-of-the-art on two challenging real-world multi-camera video datasets is demonstrated by thorough experiments.

Matching targets from different cameras can be very challenging, as there might be significant appearance variation and the blind area between cameras makes target's motion less predictable. Unlike most existing methods that only focus on modeling appearance and spatial-temporal cues for intra-camera tracking, Chapter 4 presents a novel online learning approach that further considers integrating high-level contextual information into the tracking system. The tracking problem is formulated using an online learned Conditional Random Field (CRF) model that minimizes a global energy cost. Besides low-level information, social grouping behavior is explored in order to maintain target identities as they move across cameras. In the proposed method, pair-wise grouping behavior is first learned within each camera. During intra-camera tracking, track associations that maintain single camera grouping consistencies are preferred. In addition, we introduce an iterative algorithm to find good solution for the CRF model. Comparison experiments on several challenging real-world multi-camera video sequences show that the proposed method is effective and outperforms the state-of-the-art approaches.

Each chapter in this dissertation stands alone as a complete description of each aforementioned method.

Chapter 2

Multi-person Tracking by Online Learned Grouping Model with Non-linear Motion Context

2.1 Introduction

Automatic tracking of multiple targets simultaneously in real-world scenes has been an active research topic in computer vision for many years, as it is crucial for many industrial applications and high level analysis, such as visual surveillance, human-computer interaction, and anomaly detection. The goal of multi-target tracking is to recover trajectories of all targets while maintaining consistent identity labels. There are many challenges for this problem, such as illumination and appearance variation, occlusion, and sudden change in motion [11, 121]. As great improvement has been achieved in object detection, data association-based tracking (DAT) has become popular recently [53, 98, 127, 95, 94]. In the DAT framework, often a pre-learned detector is applied on each frame to produce detection responses of all targets, and short-term tracking results (i.e., tracklets) are generated by associating responses from consecutive frames that have high probability to contain the same target. These tracklets are further linked to produce long-term tracking results. An affinity model integrating multiple visual cues (appearance and motion information) is formulated to find the linking probability between tracklets, and the global optimal solution is often obtained by solving the maximum a posteriori (MAP) problem using various optimization algorithms.

Although much progress has been made in building more discriminative appearance and motion models, problems such as identity switch and track fragmentation still exist in current association based tracking approaches, especially under challenging conditions where appearance or motion of the target changes abruptly and drastically, as shown in Fig. 2.1. The goal of association optimization is to find the best set of associations with the highest probability for all targets, which makes it not necessarily capable of linking each of the difficult tracklet pairs. In this chapter, we explore high level contextual information, i.e., social grouping behavior, for associating tracklets that are very challenging by using only lower level features (time, appearance, and motion).

When there are only a few interactions and occlusions among targets, DAT achieves robust performance. Discriminative descriptors of targets are usually generated using appearance and motion information from tracklets. Appearance model often uses global or part-based color histograms to match tracklets, and a linear motion model that assumes all targets maintain constant speed without motion direction change is often adopted to constrain motion smoothness of two tracklets. However, these low level descriptors generally fail to associate tracklet pairs with long time gap. This is because the appearance of a target might change drastically due to heavy occlusion, and the linear motion model is unreliable for predicting location of a target after a large time interval.

Nevertheless, there is often other useful high level contextual information in the scene which can be effectively used to mitigate the aforementioned shortcomings. For instance, sociologists have found that up to 70% of pedestrians tend to walk in groups in a crowd, and people in the same group are more likely to have similar motion pattern and be spatially close to each other for better group interaction [88]. Moreover, pedestrians in the crowd often either consciously or unconsciously follow other individuals with similar destination to facilitate navigation [52]. It is also observed in many real world surveillance videos that if two people are walking together at certain time then it is very likely that these two people will still walk together after a short time period.

Based on the above observations, we propose *an elementary grouping model with non-linear motion context to compensate the errors caused by using basic appearance model and linear motion model.* A grouping graph is constructed based on input tracklets with

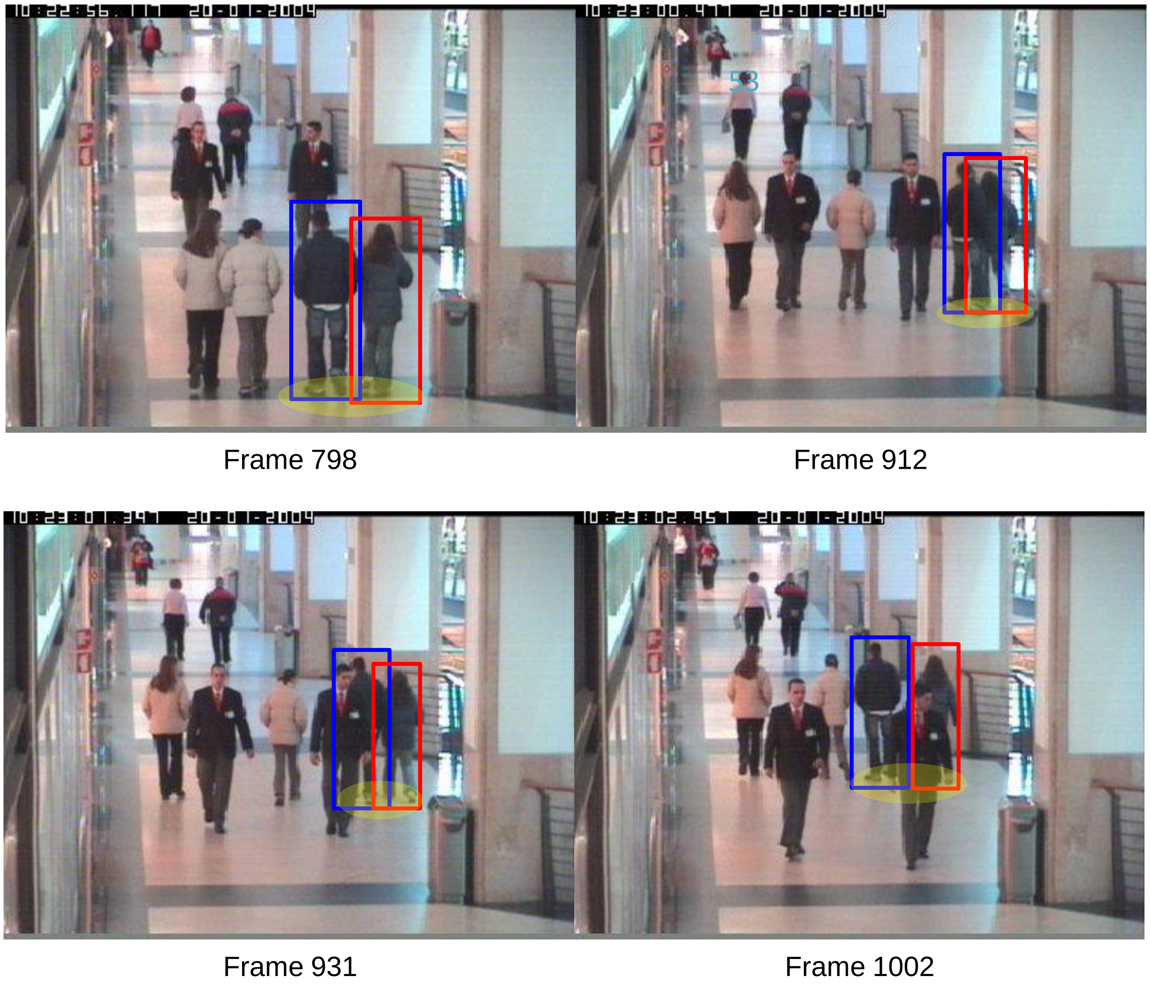


Figure 2.1: Examples in which grouping information is helpful under the challenging conditions for tracking in a video. The same color indicates the same target. Note that for both targets with bounding boxes there are significant appearance and motion changes due to occlusions and cluttered background. Images are from CAVIAR dataset [1]

high confidence, where each node represents a pair of tracklets that form an elementary group (a group of two targets) and each edge indicates that the connected two nodes (two elementary groups) have at least one target in common. The group trajectories of any two linked nodes are used to estimate the probability of the other target in each group being the same person. Neighboring tracklets that have time overlap and similar motion pattern are possible candidates for elementary groups. Relationships between elementary groups are further discovered with the help of group tracking, in which a non-linear motion map is

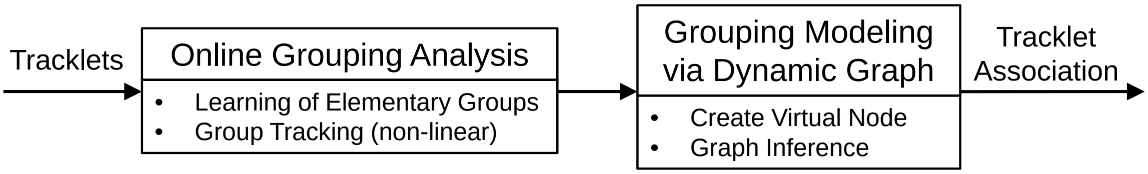

Tracklets
Online Grouping Analysis
• Learning of Elementary Groups
• Group Tracking (non-linear)
Grouping Modeling via Dynamic Graph
• Create Virtual Node
• Graph Inference
Tracklet Association

Figure 2.2: Overview of the elementary grouping model.

used to explain large time gap between two elementary groups. The elementary grouping model is summarized in Fig. 2.2.

The size of a group may change dynamically as people join and leave the group, but a group of any size can always be considered as a set of elementary groups. Therefore, focusing on finding elementary groups instead of the complete group makes our approach capable of modeling flexible group evolution [49] in the real world. Note that the social group in this chapter refers to a number of individuals with correlated movements and does not indicate a group of people who know each other.

The rest of the chapter is organized as follows: Section 2.2 discusses related work and contributions of this paper; the proposed elementary grouping model is described in Section 2.3; experiments are presented in Section 2.4; and Section 2.5 concludes this paper.

2.2 Related Work and Contributions

2.2.1 Related Work

Traditional filtering-based multi-target tracking methods process videos on a frame-by-frame basis, which are more suitable for time-critical applications [19, 66]. However, such greedy methods tend to get stuck at a local optimum, with the possible solution space growing exponentially in the presence of observation gaps. Recently, the focus of multi-target tracking has shifted to robust DAT schemes, due to their global reasoning ability of the solution space. With a deferred global inference, DAT is more robust against observation gaps resulting from heavy interactions and occlusions [83].

Huang *et al.* [57] first propose to hierarchically associate detection responses for multi-person tracking. Since then, most follow-up works focus on designing features for more reliable association scores or developing effective optimization schemes. In the first

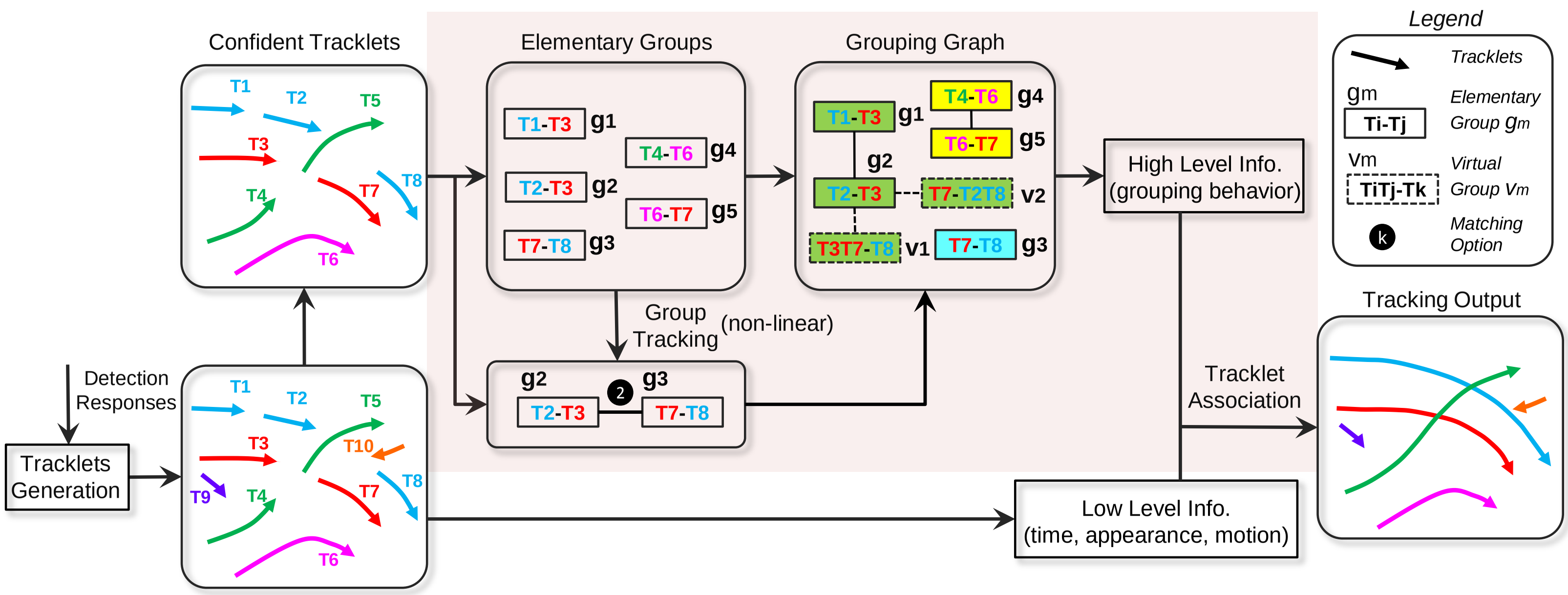


Figure 2.3: Block diagram of our tracking system. After initial tracklets are generated by linking detection responses, confident tracklets are selected to form elementary groups. The relationships between elementary groups are identified by group tracking with non-linear motion context. Then a disjoint grouping graph is constructed, from which high level information (i.e., grouping behavior) is extracted. Finally, tracklet association is carried out based on affinity model that combines both high level and low level information. Tracklets with the same color contain the same target. For the legends in this figure please see the box in the upper right hand side. Best viewed in color.

regime, affinity scores are generally extracted from appearance information such as color histograms and motion features such as motion smoothness. Global appearance constraints are exploited to prevent identity switches in multi-target tracking [13]. Part-based appearance models have been applied in multi-target tracking to mitigate occlusions [105]. For optimization, bipartite matching via the Hungarian algorithm is among the most popular and simplest algorithms [94, 57]. A lot of other optimization frameworks have been proposed, such as K-shortest path [16], set-cover [118], Linear Programming [65], and Quadratic Boolean Programming [75].

Most of the work only considers pairwise similarities, without referring to high level contextual information. Thus, problems such as possible abrupt motion changes cannot be properly accounted for. Yang *et al.* [82] use a Conditional Random Field (CRF) for tracking while modeling motion dependencies among associated tracklet pairs. Butt *et al.* [22] carry out a Lagrangian relaxation to make higher-order reasoning tractable in the min-cost flow framework. These methods focus on higher-order constraints such as constant velocity. However, both of them [82, 22] concentrate on individuals and may fail in real-world scenarios, in which individuals may possess a lot of freedom.

In this chapter, we focus on utilizing social grouping information for more natural **high-level contextual constraints**. Social factors have attracted a lot of attentions in multi-target tracking recently, since they are complementary to unreliable visual features and are motivated by sociology research. Pellegrini *et al.* [92] propose a more effective dynamic model by leveraging nearby people's positions. Brendel *et al.* [21] also consider nearby tracks as contextual constraints. Alahi *et al.* [5] study large-scale crowd destination forecasting with social context. Pellegrini *et al.* improve trajectory prediction accuracy by inferring pedestrian groups [93]. In the DAT context, Qin *et al.* [97] seek the consistency of trajectories in both tracklet association space and tracklet group assignment space based on visual and grouping cues. They use gradient-based optimization and K-means clustering with multiple random initializations. Bazzani *et al.* [12] consider joint individual-group tracking, with a decentralized particle filter sampling in both individual and group spaces. Yan *et al.* [124] explicitly consider group structures to improve tracking consistency across time. Compared to these methods, our approach is deterministic with a closed-form solution. Furthermore, the previous work assumes a static group structure or a fixed number of groups, while our grouping scheme is more flexible by using elementary groups and allows for more local refinements.

2.2.2 Contributions of This Chapter

The contributions of this chapter include:

- An approach estimating elementary groups online is proposed, which infers grouping information to adjust the affinity model for data association-based tracking. This approach is independent of detection methods, affinity models, and optimization algorithms.
- A motion model that takes advantage of nearby non-linear motion patterns is integrated into group tracking. It enables the proposed method to explain reasonable non-linear motions of targets.
- The proposed approach based on elementary grouping is simple and computationally efficient, while it is effective and robust.
- Four real-world surveillance datasets are used for evaluation and extensive experiments are carried out to validate the effectiveness of the proposed method.

2.3 Technical Approach

In this section, we introduce how the elementary grouping model is integrated into the basic tracking framework for tracklet association. An overview of the proposed method is presented in Fig. 2.3.

2.3.1 Tracking Framework with Grouping

Given a video sequence, a human detector is first applied to each frame to obtain detection responses. Finding the best set of detection associations with the maximum linking probability is the aim of detection-based tracking. In an ideal association, each disjoint string of detections should correspond to the trajectory of a specific target in the groundtruth. However, object detector is prone to errors, such as false alarms and inaccurate detections. Also, directly linking detections incur a high computational cost. In order to generate a set of reliable tracklets (trajectory fragments), therefore, it is a common practice to pre-link detection responses that have high probability to contain the same person. Next, a global optimization method is employed to associate tracklets according to multiple cues. Finally, missed detections are inserted by interpolation between the linked tracklets. Detections that do not belong to any tracklet or tracklets that are too short are considered as false alarms and removed from the final results.

A mathematical formulation of the tracking problem is given as follows. Suppose a set of tracklets $\mathcal{T} = \{T_1, .., T_n\}$ is generated from a video sequence. A tracklet T_i is a consecutive sequence of detection responses or interpolated responses that contain the same target. The goal is to associate tracklets that correspond to the same target, given certain spatial-temporal constraints. Let association a_{ij} defines the hypothesis that tracklet T_i and T_j contain the same target, assuming T_i occurrs before T_j. A valid association matrix A is defined as follows:

$$A = \{a_{ij}\}, a_{ij} = \begin{cases} 1, & \text{if } T_i \text{ is associated to } T_j, \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$
$$\text{s.t. } \sum\nolimits_{i=1}^{n} a_{ij} = 1 \text{ and } \sum\nolimits_{j=1}^{n} a_{ij} = 1.$$

The constraints for matrix A indicate that each tracklet should be associated to and associated by only one other tracklet (the initial and the terminating tracklets of a track are discussed in Section 2.4.1).

We define S_{ij} as the basic cost for linking tracklet T_i and T_j based on low level information (time, appearance, and motion). It is computed as the negative log-likelihood of T_i and T_j being the same target (explained in detail in Section 2.4.1). Note that $S_{ij} = \infty$ if T_i and T_j have overlap in time.

Let Ω be the set of all possible association matrices, the multi-target tracking can be formulated as the following optimization problem:

$$A^* = \arg\min_{A \in \Omega} \sum_{ij} a_{ij} S_{ij}. \tag{2.2}$$

This assignment problem can be solved optimally by the Hungarian algorithm in polynomial time. In order to reduce computational cost, the video is segmented by a pre-defined time sliding window, which is fixed to be 12 seconds long. Tracklet association is carried out in each time sliding window. There has to be a 50% overlap between two neighboring time windows. To handle association conflicts in the overlapping part of two windows, we use a method similar to [82]. More specifically, the overlapped part is evenly divided into two parts. In the first half, tracking results produced by the previous time window is kept, while in the second half, original input tracklets are used despite the association results from the previous time window.

As low level information is not sufficient to distinguish targets under challenging situations, we consider to integrate high-level information from social grouping behavior into the cost matrix to regularize the solution. However, group configuration is often not known *a priori*. Also, it is not fixed for the entire video, as people might change groups. Therefore, we propose elementary groups that are learned and updated "online", during the tracking process to provide useful social grouping information while maintaining the flexibility of the group structure. Two tracklets T_i and T_j are likely to correspond to the same target if they satisfy the following constraints: 1) each of them forms an elementary group with the same tracklet, namely, the same target; 2) the trajectory obtained by linking T_i and T_j has a small distance to the group mean trajectory. The first constraint is based on the observation that if two people are walking together for a certain time, then there is high probability that they will still walk together after a short time period. The second constraint prevents us from linking wrong pair of tracklets. Let P_{ij} be the inferred high level information for T_i and T_j, the tracklet association problem can be refined as:

$$A^* = \underset{A \in \Omega}{\arg\min} \sum_{ij} a_{ij}(S_{ij} - \alpha P_{ij}), \tag{2.3}$$

where α is a weighting parameter. It is selected by coarse binary search in only one time window and kept fixed for all the others.

In the following, we introduce an online method for group analysis and obtain P_{ij} by making inferences from the grouping graph.

2.3.2 Learning of the Elementary Groups

In this part, we explain how the nodes (elementary groups) of the grouping graph are created. A set of tracklets is generated after low level association, but only confident tracklets are considered for grouping analysis, as there might be false alarms which may lead to incorrect associations in the input tracklets. Based on the observation that inaccurate tracklets are often the short ones, we define a tracklet as confident if it is long enough (e.g., it exists for at least 10 frames).

Two tracklets T_i and T_j form an elementary group if they have the following properties: 1) T_i and T_j have overlap in time for more than l frames (l is set to 5 in our experiments); 2) they are spatially close to each other; 3) they have similar velocities. Mathematically, we use G_{ij} to denote the probability of T_i and T_j forming an elementary group:

$$G_{ij} = P_t(T_i, T_j) \cdot P_d(T_i, T_j) \cdot P_v(T_i, T_j), \tag{2.4}$$

where $P_t(\cdot)$, $P_d(\cdot)$ and $P_v(\cdot)$ are the grouping probabilities based on overlap in time, distance and velocity, respectively. Their definitions are given in Eq. (2.5), Eq. (2.6), Eq. (2.7).

$$P_t(T_i, T_j) = \frac{L_{ij}}{L_{ij} + l}, \tag{2.5}$$

$$P_d(T_i, T_j) = \frac{1}{L_{ij}} \sum\nolimits_{n=1}^{L_{ij}} (1 - \frac{2}{\pi} \arctan(dist_n)), \tag{2.6}$$

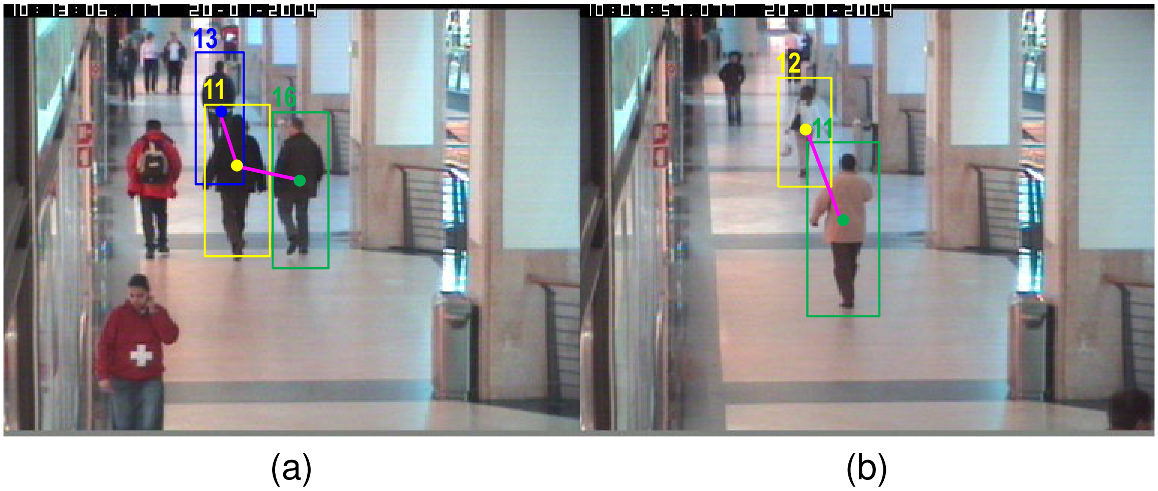
$$P_v(T_i, T_j) = \frac{cos\theta + 1}{2}, \tag{2.7}$$

where L_{ij} is the length of overlapped frames for T_i and T_j, $dist_n$ is the normalized center distance for T_i and T_j on the n^{th} overlapped frame, and θ is the angle between the average velocities of the two tracklets during the overlapped frames. In our experiments, $dist_n$ is set as follows:

$$dist_n = ratio_n \cdot d / 0.5(width_i + width_j), \tag{2.8}$$

where $ratio_n$ is the size of the larger target over the size of the smaller target, d is the Euclidean distance between the two object centers, and $0.5(width_i + width_j)$ is the smallest distance in the image space for two people that walk side by side. The term $ratio_n$ prevents tracklets as shown in Fig. 2.4 to be considered as a group, where the distance in the image space is small while the distance in the 3D space is quite large.

We create a node for each pair of tracklets that have non-zero grouping probability G. Thus, each node contains two tracklets/targets and is associated with a probability G, its value indicates the similarity of motion patterns for these two tracklets during their co-existence period.



(a) (b)

Figure 2.4: Examples of generating incorrect elementary groups if the distances are not normalized.

Note that if two tracklets form an elementary group, their group mean trajectory is obtained by computing the mean position using only their overlapping parts, as the grouping is only meaningful for the overlapped time period. For example, if T_a and T_b are in the same elementary group, this only indicates that T_a and T_b have similar motion patterns for the period that they have time overlap. During the non-overlapping period, T_a may form elementary groups with other tracklets/targets that are even in a different group than the group of T_b. Such property makes the elementary group flexible to handle group split and merge.

2.3.3 Group Tracking

The relationship between two elementary groups is identified by group tracking. Inspired by association-based multi-target tracking, we define our group tracking as a problem of finding globally optimal associations between elementary groups based on the three most commonly used features: time, appearance, and motion. More specifically, given a set of elementary groups, we compute the linking cost for any two groups and obtain the association results by finding the association set with the minimum total cost.

Let $\{T_1^{g_i}, T_2^{g_i}\}$ denote the two tracklets in an elementary group g_i. Given two elementary groups g_i and g_j, assuming g_i starts before g_j, their linking cost is $C^g(g_i, g_j) =$

$C_t^g(g_i, g_j) + C_{appr}^g(g_i, g_j) + C_{mt}^g(g_i, g_j)$, where $C_t^g(\cdot)$, $C_{appr}^g(\cdot)$, and $C_{mt}^g(\cdot)$ are linking costs based on time, appearance, and motion, respectively. Similar to Eq. (2.2), let Φ be the set of all possible group association matrices, then the group tracking can be formulated as the following optimization problem:

$$A^{g*} = \underset{A^g \in \Phi}{\arg\min} \sum_{ij} a_{ij} C^g(g_i, g_j). \tag{2.9}$$

Hungarian algorithm is used to solve this assignment problem.

Time Model for Group Tracking

For the linking cost based on time, we defined it as:

$$C_t^g(g_i, g_j) = \begin{cases} 0, & g_i \text{ is not overlapped with } g_j, \\ \infty, & \text{otherwise,} \end{cases} \tag{2.10}$$

where the non-overlapping constraint means any tracklet in g_i has no time overlap with any tracklet in g_j.

If g_i and g_j contain the same two targets, there are only two matching possibilities: 1) $T_1^{g_i}$ and $T_1^{g_j}$ are the same target, $T_2^{g_i}$ and $T_2^{g_j}$ are the same target; 2) $T_1^{g_i}$ and $T_2^{g_j}$ are the same target, $T_2^{g_i}$ and $T_1^{g_j}$ are the same target. We explain in detail for matching option 1), note that the computation for matching option 2) is similar. For each matching option, we compute the linking cost based on appearance and motion, and use the one with the smaller sum for $C_{appr}^g(g_i, g_j) + C_{mt}^g(g_i, g_j)$. Also, the matching option is recorded for each group association.

Appearance Model for Group Tracking

Let $S(\cdot)$ be the appearance similarity for two tracklets, the group linking cost based on appearance is defined as:

$$C_{appr}^g(g_i, g_j) = -ln(S(T_1^{g_i}, T_1^{g_j}) + S(T_2^{g_i}, T_2^{g_j})). \tag{2.11}$$

As there might be appearance variations in a single tracklet due to occlusion and lighting changes, it is hard to generate features that can robustly represent the appearance of a target. In order to more reliably compute the similarity between two tracklets, we adopt the modified Hausdorff metric [41] which is able to compute the similarity of two sets of images. Given a tracklet T_i that has length m_i, let $T_i = \{d_1^i, d_2^i, ..., d_{m_i}^i\}$ where d_x^i is the x^{th} estimation of T_i, then $S(\cdot)$ is defined as:

$$S(T_i, T_j) = \min(\frac{1}{m_i} \sum_{d_x^i \in T_i} s(d_x^i, T_j), \frac{1}{m_j} \sum_{d_y^j \in T_j} s(d_y^j, T_i)), \tag{2.12}$$

where $s(d, T) = \max_{d' \in T}(s_{cos}(d, d'))$ is the Hausdorff similarity between an estimation and a tracklet. A modified cosine similarity measure [78] $s_{cos}(\cdot)$ is used to compute the similarity between two estimations, which is defined as

$$s_{cos}(u, v) = \frac{|u^T \cdot v|}{\|u\| \, \|v\| \, (\|u - v\|_p + \epsilon)}, \tag{2.13}$$

where u, v are the feature descriptors from two images, $\|\cdot\|_p$ is the l_p norm (we set $p = 2$), and ϵ is a small positive number to avoid dividing by zero. In our experiments, we use the concatenation of HSV color histogram and HOG features as the feature descriptors.

Motion Model for Group Tracking

We measure the motion affinity of two elementary groups by the motion smoothness between the group mean trajectories of the two corresponding elementary groups. The motion cost for linking two group mean trajectories is defined as the negative logarithm of the motion affinity:

$$\begin{aligned} C_{mt}^g(g_i, g_j) = &- ln(G(f_{predict}(g_i, +\Delta t) - p_{head}^{g_j}, \Sigma_p) \\ &\cdot G(f_{predict}(g_j, -\Delta t) - p_{tail}^{g_i}, \Sigma_p)), \end{aligned} \tag{2.14}$$

where $G(\cdot)$ is a zero mean Gaussian distribution, Δt is the time gap between g_i and g_j, $f_{predict}(g_i, \pm\Delta t)$ gives the location prediction for the group mean trajectory of g_i after $(+)$ or before $(-)$ Δt, p_{head} and p_{tail} are the head and tail locations for a group mean trajectory.

In most previous tracking frameworks [121][97][14], targets are commonly assumed to maintain linear motion pattern. Thus, $f_{predict}(g_i, +\Delta t) = p_{tail}^{g_i} + v_{tail}^{g_i}\Delta t$ and $f_{predict}(g_j, -\Delta t) = p_{head}^{g_j} - v_{head}^{g_j}\Delta t$. However, in real world scenarios, it is common to observe several non-linear motion patterns in the scene. In order to produce more robust

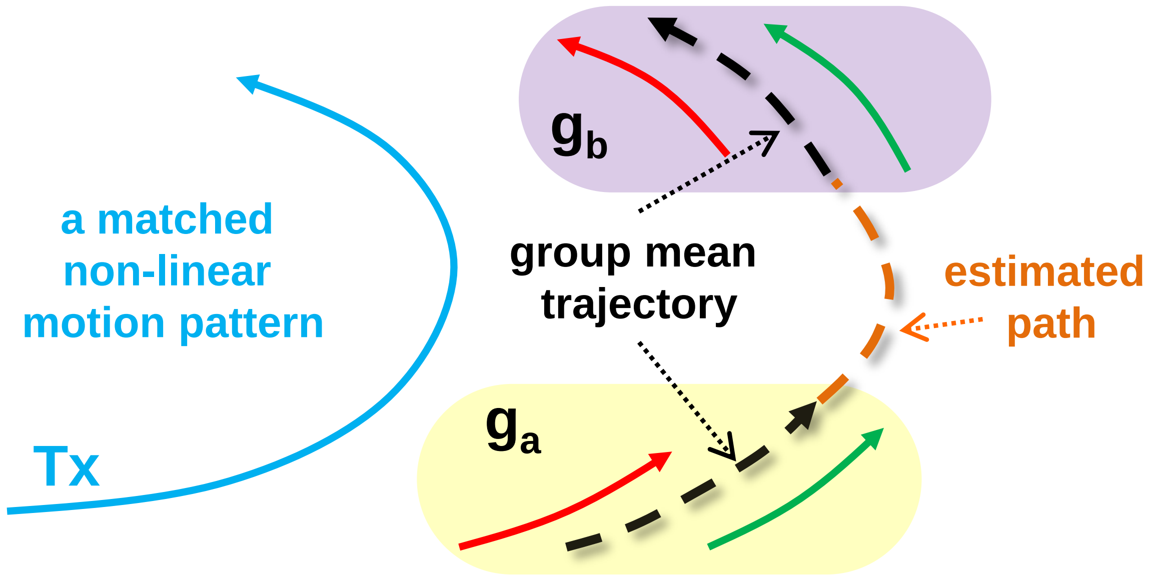


Figure 2.5: An example of estimating motion affinity using the non-linear motion map.

motion affinity for two elementary groups, we use the non-linear motion map [127] to explain large non-linear time gaps between group mean trajectories. Note that in [127] the non-linear motion map is directly used to estimate the motion affinity of two tracklets, whereas we use it for explaining non-linear gap between two elementary groups.

The non-linear motion map M is a set of all existing non-linear tracklets in current time sliding window, and the tracklets are selected only from the confident ones. An example of estimating motion affinity between g_a and g_b using a non-linear motion pattern T_x in the motion map is illustrated in Fig. 2.5. The tracklet $T_x \in M$ is a non-linear motion pattern that has co-existed in time with both g_a and g_b and is a *matched tracklet* for the group mean trajectories of g_a and g_b. T_x is a matched tracklet indicates that it is spatially close to the elementary group and has similar motion direction as the elementary group. Then a quadratic curve that best fits positions at the tail part of g_a and the head part of g_b is estimated to fill the path between g_a and g_b. Therefore, each group association has a specific quadratic function for its non-linear motion estimation. The estimated path is only valid if T_x is a matched tracklet for it. The motion cost for linking g_a and g_b based on non-linear prediction of locations can be computed according to Eq. (2.14).

For each pair of elementary groups, both linear and non-linear motion models are used, and the score with a lower cost is selected. Note that when only linear motion model is used, any trajectory not following the pattern is penalized. With the non-linear motion model, we are able to explain non-linear motion in the scene without producing extra penalties for individuals who do not follow a linear motion pattern.

2.3.4 Creation of Virtual Nodes

Our goal is to encode grouping structure of the tracklets by the elementary grouping graph. With elementary groups as nodes of the graph, we define an edge between two nodes indicating the existence of at least one common target in the corresponding two elementary groups. For simple cases where two nodes have one tracklet in common, we link these two nodes directly, such as nodes g_1 and g_2, g_4 and g_5 shown in Fig. 2.3. For difficult cases where there are four different tracklets in two nodes, we use the results of group tracking to find their relationship.

Suppose g_i and g_j are associated by group tracking, namely, these two elementary groups contain the same two targets. We create two virtual nodes v_p and v_q, set their grouping probability G to be the same as that of node g_j, and build edges between g_i and the virtual nodes. Note that the virtual nodes can also be added in the other way (i.e., set G to be the same as g_i and link the virtual nodes to g_j), but these two options are exclusive to each other. Each virtual node also contains two tracklets, one is a virtual tracklet generated by linking a pair of matched tracklets in g_i and g_j, the other is the tracklet left in g_j. An example of virtual node creation is presented in Fig. 2.3. Based on the association of g_2 and g_3, two virtual nodes v_1 and v_2 are created and connected to g_2. Two virtual nodes are used since there are two pairs of tracklets that need inference (edge for g_2 and v_1 indicates inference for T_2 and T_8; edge for g_2 and v_2 indicates inference for T_3 and T_7). In the following, we show that by using the virtual node inference can be easily done.

2.3.5 Inference from the Grouping Graph

In the grouping graph, each node is an elementary group and each edge indicates that the two connected elementary groups have one target in common. According to the observation that two people walk together at certain time are likely to walk together after a short period, given two directly connected groups, we can infer the probability of the

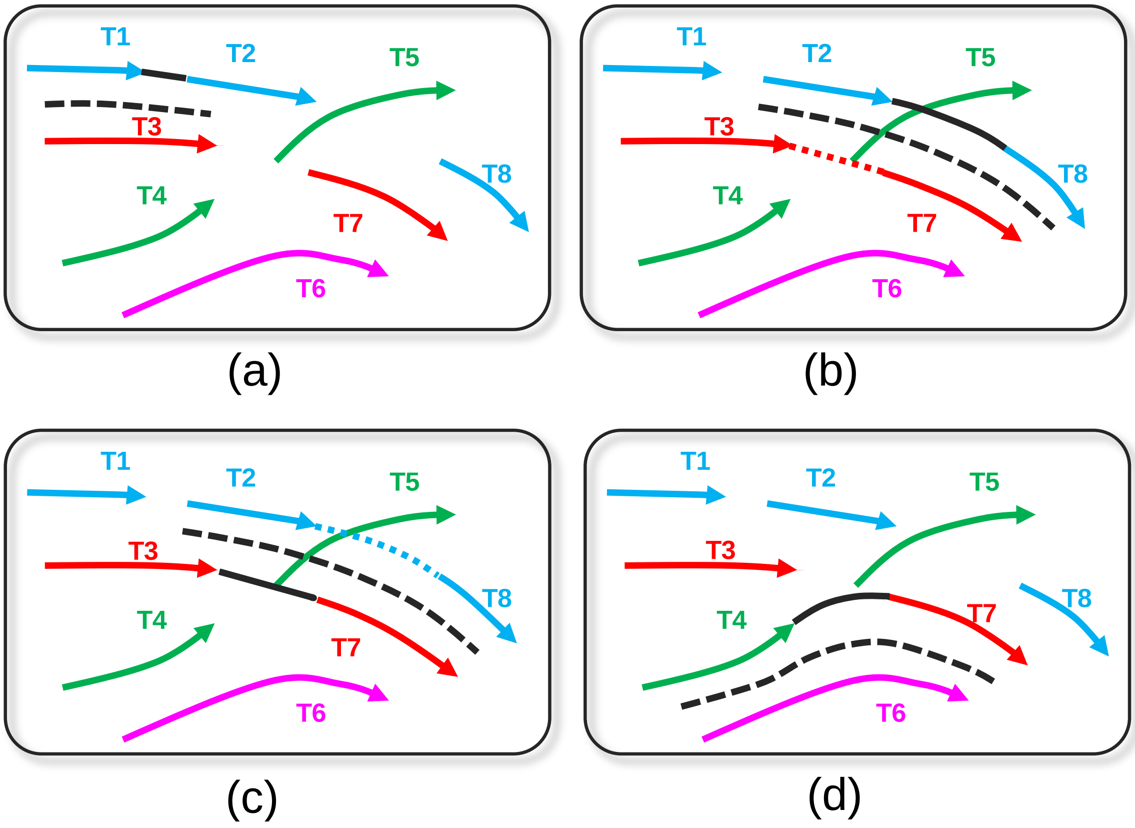


Figure 2.6: Inference for each edge in the grouping graph in Fig. 2.3: (a) edge between g_1 and g_2, (b) edge between g_2 and v_1, (c) edge between g_2 and v_2, (d) edge between g_4 and g_5 (see Fig. 2.3 for group annotations). Black solid line represents interpolation between the two tracklets that need inference, black dashed line is the group mean trajectory, and colored dotted line indicates a virtual tracklet. Best viewed in color.

uncertain target in each group to be the same.

Suppose there is an edge between nodes g_i and g_j in the grouping graph, assuming $T_1^i = T_1^j = T_k$, $T_2^i = T_l$, and $T_2^j = T_m$ without loss of generality, the probability of T_2^i and T_2^j contain the same target is defined as follows:

$$p_{lm} = 0.5(G_{kl} + G_{km}) \times TSimi(T_{\{l,m\}}, G_{\{k,l,m\}}), \tag{2.15}$$

where $TSimi(T_{\{l,m\}}, G_{\{k,l,m\}})$ is the trajectory similarity between trajectory $T_{\{l,m\}}$ (created by linking T_l and T_m) and the group mean trajectory $G_{\{k,l,m\}}$ (created by computing the mean position of T_k and $T_{\{l,m\}}$). We define the trajectory similarity as follows:

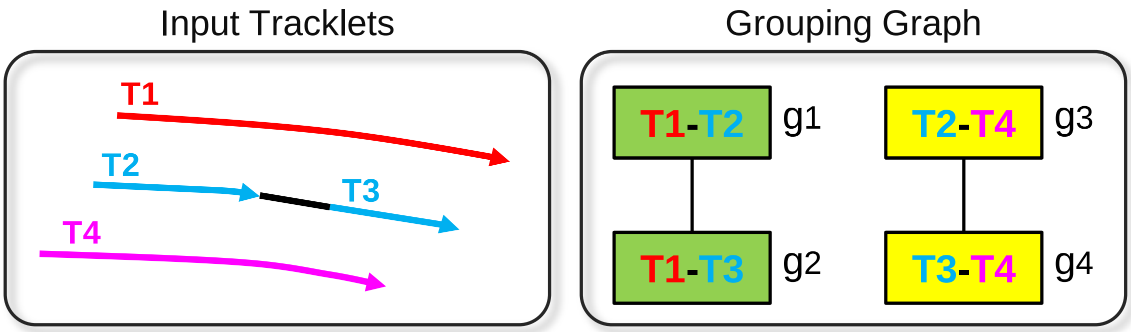


Figure 2.7: An example of multiple inferences related to the same two tracklets. According to the proposed elementary grouping model, a grouping graph (shown on the right) is created based on the input tracklets (shown on the left). Thus, inferences based on the edge between node g_1 and g_2 and the edge between node g_3 and g_4 are all related to tracklets T_2 and T_3.

$$TSimi(T, G) = 1 - \frac{2}{\pi}\arctan(Dist), \tag{2.16}$$

where $Dist$ is the average Euclidean distance of trajectory T and group mean trajectory G.

For edges connecting two normal nodes and edges connecting to one virtual node, the same inference function can be used. The only difference is that the latter uses one virtual tracklet and two normal tracklets as input. Examples of making inference for a grouping graph are shown in Fig. 2.6. Note that there might be multiple inferences related to the same two tracklets, as the same tracklet may be contained in multiple elementary groups, as shown in Fig. 2.7. Therefore, P_{ij} in Eq. (2.3) is the sum of all inferences that relate to T_i and T_j:

$$P_{ij} = \sum p_{ij}. \tag{2.17}$$

A summary of the proposed elementary grouping model is shown in Algorithm 1.

2.4 Experiments

We evaluate our approach on four datasets: the CAVIAR dataset [1], the TownCentre dataset [14], the PETS2009 dataset [45], and the UNIV dataset [44]. The popular

Algorithm 1 Learning algorithm for elementary grouping model

```
Input: Tracklet set $T = \{T_1, .., T_n\}$
Output: Inference matrix $P$, where $P_{ij}$ is the inference for $T_i$ and $T_j$
$P \leftarrow$ empty set, $Nodes \leftarrow \emptyset$, $Edges \leftarrow \emptyset$
for $i = 1, ..., n$ do
    for $j = i + 1, ..., n$ do
        if $T_i$ and $T_j$ are confident tracklets then
            $G_{ij} = P_t(T_i, T_j)P_d(T_i, T_j)P_v(T_i, T_j)$
            if $G_{ij} > 0$ then
                Create node $g = \{T_i, T_j\}$
                $Nodes = Nodes \cup \{g\}$
for $i = 1, ..., size(Nodes)$ do
    for $j = i + 1, ..., size(Nodes)$ do
        if $\exists T \in g_i$, $T = T_1^{g_j}$ or $T = T_2^{g_j}$ then
            Create an edge $e_{\{g_i, g_j\}}$ for $g_i$ and $g_j$
            $Edges = Edges \cup \{e_{\{g_i, g_j\}}\}$
Update $Nodes$ and $Edges$ according to group tracking
for all $e \in Edges$ do
    Compute $p_{xy}$ for the corresponding tracklet pair using Eq. (2.15)
    Update $P$: $P_{xy} = P_{xy} + p_{xy}$
```

evaluation metrics defined in [77] and the CLEAR MOT metrics defined in [17] are used for performance comparison:

- *GT* the number of trajectories in the ground-truth.

- *MT* the ratio of mostly tracked trajectories, which are successfully tracked for more than 80% of the time.

- *ML* the ratio of mostly lost trajectories, which are successfully tracked for less than 20% of the time.

- *Frag* fragments, the number of times that a ground-truth trajectory is interrupted.

- *IDS* ID switches, the number of times that a tracked trajectory changes its matched id.

- *FP* false positive, the number of tracker hypotheses for which no real object exists.

- *FN* false negative, the number of times that targets have no matched hypothesis.

- *MOTA* multiple object tracking accuracy, a combined measure which takes into account false positives, false negatives and identity switches.

- *MOTP* multiple object tracking precision, measures the alignment of tracking results with respect to ground-truth.

The following tracking approaches are tested:

- *Our Model (non-linear)*: the proposed elementary grouping model with non-linear motion context for group tracking.
- *Our Model (linear)*: the proposed elementary grouping model with only linear motion model for group tracking.
- *Baseline Model 1*: the basic affinity model.
- *Baseline Model 2*: the proposed elementary grouping model without group tracking.
- *SGB*: the Social Grouping Behavior model [97].

For a fair comparison, the same input tracklet set, ground-truth, as well as basic affinity model are used for all the methods. All the results for the SGB model are kindly provided by the authors of [97]. Both quantitative comparisons with the state-of-the-art methods and visual results of our approach are presented.

2.4.1 Implementation Details

Tracklets generation: Two different ways of generating tracklets are employed to demonstrate that the proposed grouping model can be easily integrated into any DAT based tracking system, regardless of the method used to extract the initial tracklets. In the first method, targets on each frame are detected using the discriminatively trained deformable part-based models [42]. We apply a nearest neighbor detection association method similar to [94] to generate the initial tracklets. For each unassociated detection a Kalman filter based tracker is initialized with position and velocity states. A detection A is associated to a detection B in the next frame if B has the minimum distance to the predicted location and overlaps at least 50% (measured as $size(A \cap B)/size(A \cup B)$) in size with detection

A. Then the corresponding Kalman filter is updated with the newly associated detection. The tracker terminates if no association is found for more than two consecutive frames, or a detection is associated by multiple trackers.

In the second method, the popular HOG based human detector [34] is used. Tracklets are generated by connecting detections in consecutive frames that have high similarity in appearance and have large overlap in size. A simple two-threshold strategy [57] is used to generate reliable tracklets. In our experiments, two detections are connected if and only if: 1) their affinity is higher than 90%; 2) their affinity is at least 20% larger than the affinities of any other alternatives.

Basic affinity model: In order to produce reasonable basic affinity for a pair of tracklets, three commonly used features are adopted: time, appearance and motion. The basic affinity P_{basic} for two tracklets T_i and T_j is defined as

$$P_{basic}(T_i, T_j) = f_t(T_i, T_j) \cdot f_{appr}(T_i, T_j) \cdot f_{mt}(T_i, T_j). \tag{2.18}$$

The time affinity model f_t assigns zero affinity to tracklet pairs whose time gap is greater than a pre-defined threshold GAP, it is defined as

$$f_t(T_i, T_j) = \begin{cases} 0, & \text{if } Gap_{ij} > GAP, \\ 1, & \text{otherwise.} \end{cases} \tag{2.19}$$

The appearance affinity model f_{appr} is based on the Bhattacharyya coefficient of two average HSV color histograms. For the motion affinity model f_{mt}, the same method as shown in Eq. (2.14) with linear motion for $f_{predict}$ is used to measure the motion smoothness of two tracklets in both forward and backward directions. Given $P_{basic}(T_i, T_j)$, the basic cost S_{ij} in Eq. (2.2) is computed as $S_{ij} = -ln(P_{basic}(T_i, T_j))$.

Cost matrix S: Due to the constraints in Eq. (2.1), the traditional pairwise assignment algorithm is not able to find the initial and the terminating tracklets. Therefore, instead of using the cost matrix S ($n \times n$) directly, we use the augmented matrix ($2n \times 2n$) proposed in [97] as the input for the Hungarian algorithm. This enables us to set a threshold for association, a pair of tracklets can only be associated when their cost is lower than the threshold. In our experiments, the threshold is set to $-ln0.5$ for all datasets.

Table 2.1: Comparison of tracking results on CAVIAR dataset. The number of trajectories in the ground-truth (GT) is 75.

Method	MT	ML	Frag	IDS	FP	FN	MOTA	MOTP	Time
Baseline Model 1	74.7%	6.7%	11	12	1459	10827	79.2%	78.8%	1.5s
Baseline Model 2	78.7%	6.7%	10	8	1535	9134	82.0%	81.7%	4.2s
SGB Model [97]	89.3%	2.7%	7	5	1597	8497	83.0%	82.1%	50s
Our Model (linear)	90.7%	2.7%	6	5	1668	8081	83.5%	82.0%	4.6s
Our Model (non-linear)	90.7%	2.7%	6	5	1668	8081	83.5%	82.0%	6.1s

2.4.2 Results on CAVIAR Dataset

The videos in the CAVIAR dataset are acquired in a shopping center where frequent interactions and occlusions occur and people are more likely to walk in groups. We select the same set of test videos as in [97], which are the relatively challenging ones in the dataset. We generate input tracklets using the first method described in Section 2.4.1. The comparative results are shown in Table 2.1. Our proposed models (both linear and non-linear) achieve the best overall tracking accuracy (MOTA) with the high tracking precision (MOTP) as compared to the other alternatives. It is observed that the basic affinity model (Baseline Model 1) can produce reasonable tracking results, and the performance is further improved by integrating high-level grouping information (Baseline Model 2, Our Model (linear), and Our Model (non-linear)). Both linear and non-linear versions of our model have comparable or better performances in most metrics as compared to the SGB model (e.g., better results in MT and Frag, the same results in ML and IDS), but with much less computational time. The comparisons between Baseline Model 2 and Our Model (both linear and non-linear) demonstrate the importance of group tracking, as they reveal more grouping information. Since most pedestrians in the videos are walking linearly along a corridor in this dataset, there is barely any non-linear context in the scene. Therefore, the linear and non-linear versions of our model have the same performance (except computational time) on this dataset. Sample tracking results are shown in Fig. 2.8.

Table 2.2: Comparison of tracking results on TownCentre dataset. The number of trajectories in the ground-truth (GT) is 220.

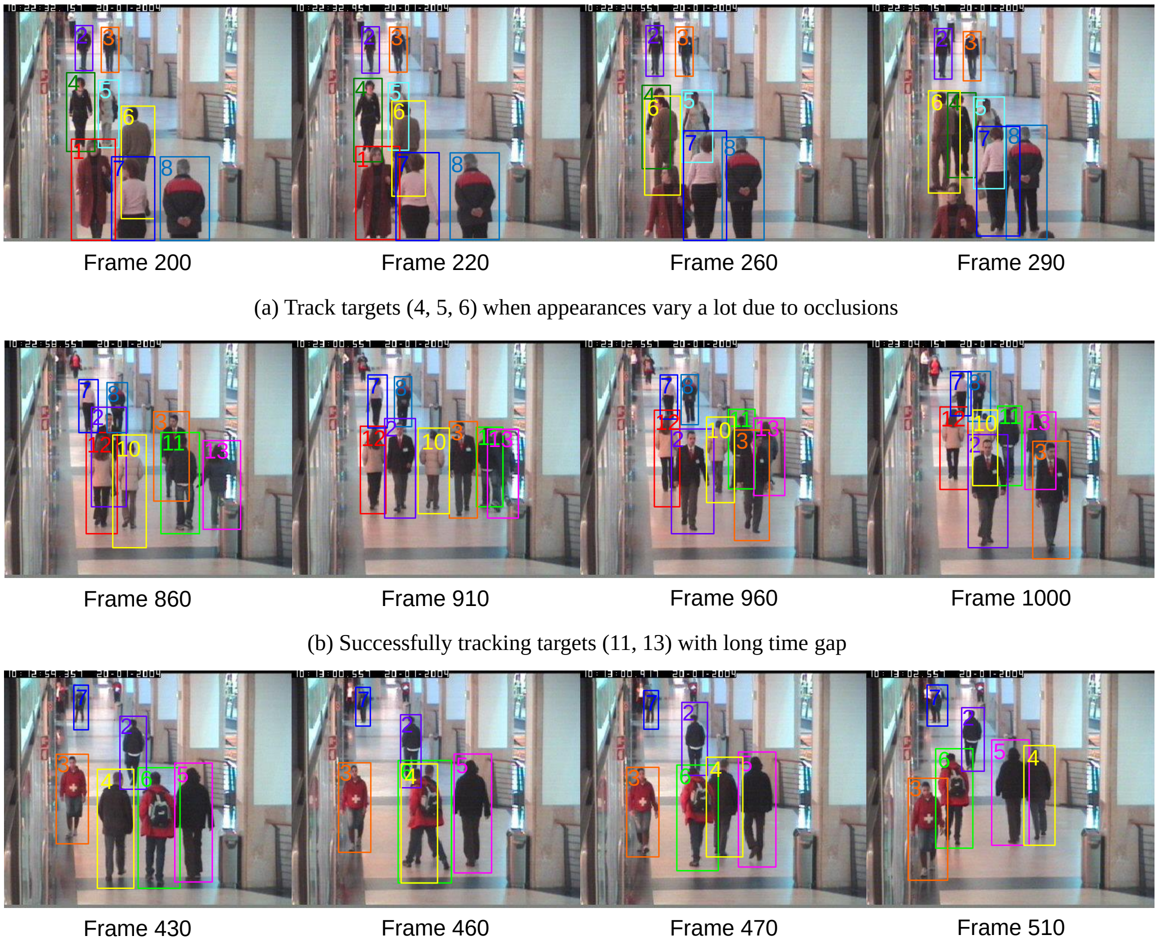
Method	**MT**	**ML**	**Frag**	**IDS**	**FP**	**FN**	**MOTA**	**MOTP**	**Time**
Baseline Model 1	76.8%	7.7%	37	60	2746	28493	56.1%	68.8%	350s
Baseline Model 2	78.6%	6.8%	34	46	3155	22236	64.3%	71.3%	457s
SGB Model [97]	83.2%	5.9%	28	39	4387	15871	81.8%	69.7%	4861s
Our Model (linear)	85.5%	5.9%	26	36	4105	14804	73.4%	69.2%	465s
Our Model (non-linear)	86.4%	5.9%	25	36	4938	13910	73.5%	69.2%	505s

2.4.3 Results on TownCentre Dataset

The TownCentre dataset has one high-resolution video which captures the scene of a busy street. There are 220 people in total, with an average of 16 people visible per frame. We test all models using the first 3 minutes of the video, and generate input tracklets using the second method described in Section 2.4.1. The comparative results are shown in Table 2.2. Similar to the observations from Table 2.1, Table 2.2 suggests that the performance of our method is consistent on both datasets. As there are some non-linear motion in this dataset, the tracking performance is slightly improved by the incorporation of non-linear context. Sample tracking results are shown in Fig. 2.9.

2.4.4 Results on PETS2009 Dataset

We select sequence *S2L2* in the PETS2009 dataset to evaluate the performance of the proposed method. This sequence captures the outdoor scene of a campus from an elevated viewpoint. Unlike the widely used sequence *S2L1*, sequence *S2L2* is more challenging as it has higher crowd density (up to 33 targets per frame) and includes many non-linear motion patterns. A rectangular area is defined in the world coordinates and used as the boundary of the tracking area (as shown in Fig. 2.10), trajectories outside the area are excluded from our solution. The first method described in Section 2.4.1 is used to generate input tracklets. The comparative results are shown in Table 2.3. We can see that when many non-linear walking patterns present in the dataset, significant improvements are achieved by integrating non-linear motion context into the tracking system. Our model



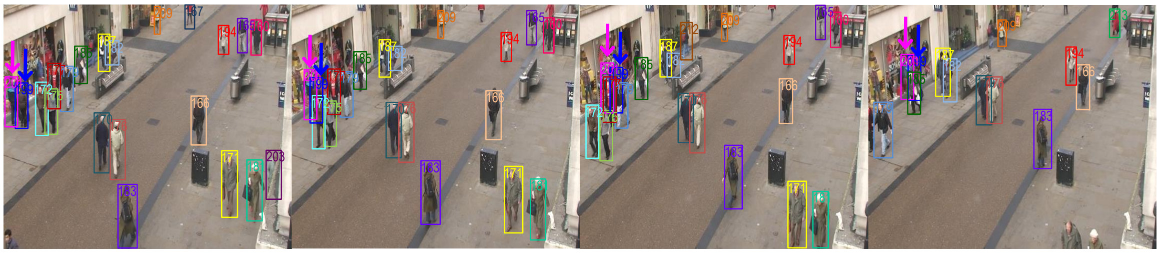
(a) Track targets (4, 5, 6) when appearances vary a lot due to occlusions

(b) Successfully tracking targets (11, 13) with long time gap

(c) Track targets (4, 5, 6) when sudden motion change and occlusion happen

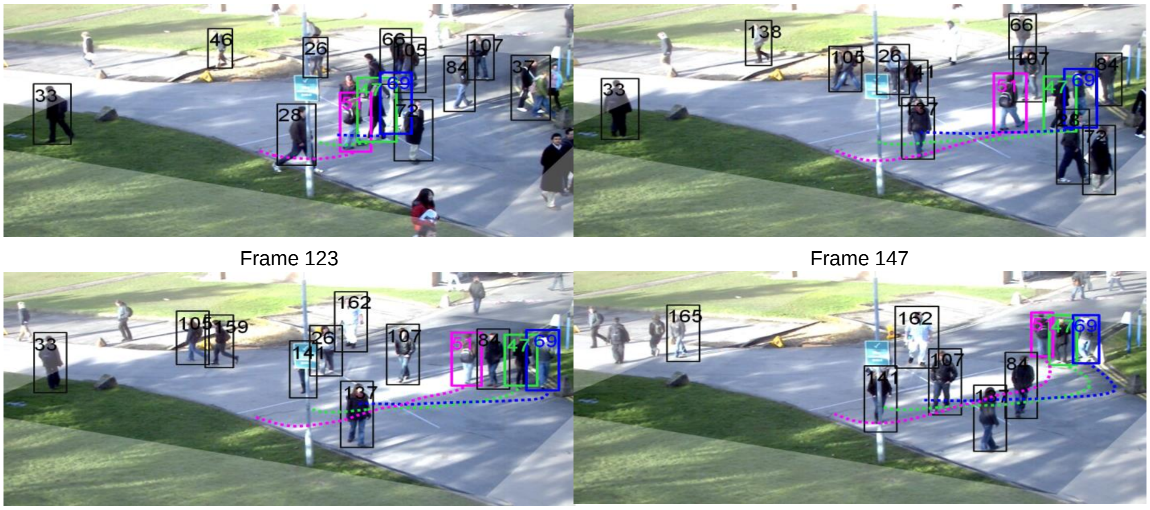
Figure 2.8: Examples of tracking results of our approach on CAVIAR dataset. The same color indicates the same target. Best viewed in color.

with non-linear context gives the best MOTA and has a higher MT (33.8%) and a lower ML (35.1%) compared to the SGB method (MT: 23%, ML: 41.9%) and Our Model (linear) (MT: 28.4%, ML: 44.6%) that only consider linear motion during grouping. Also the number of fragments and ID switches are greatly reduced when social grouping and non-linear context are employed. Sample tracking results of the proposed method with non-linear motion context are shown in Fig. 2.10. In the first row of Fig. 2.12 we present tracking examples of our method with linear motion model on the same sample sequence as shown in Fig. 2.10.



Frame 3380 Frame 3400 Frame 3420 Frame 3460

Figure 2.9: Examples of tracking results of our approach on TownCentre dataset. With grouping information, targets (199 and 201) pointed by arrows are correctly tracked under frequent occlusions. The same color indicates the same target. Best viewed in color.



Frame 123 Frame 147

Frame 161 Frame 184

Figure 2.10: Examples of tracking results of our approach on PETS2009 dataset. Track targets (47, 51, 69) with non-linear motion successfully. Best viewed in color.

Table 2.3: Comparison of tracking results on PETS2009 dataset. The number of trajectories in the ground-truth (GT) is 74.

Method	**MT**	**ML**	**Frag**	**IDS**	**FP**	**FN**	**MOTA**	**MOTP**	**Time**
Baseline Model 1	14.9%	64.9%	120	88	271	5414	32.4%	60.5%	297s
Baseline Model 2	21.6%	50%	104	102	436	4773	37.8%	59.7%	381s
SGB Model [97]	23%	41.9%	95	91	691	3828	46.0%	59.9%	4962s
Our Model (linear)	28.4%	44.6%	93	97	683	3987	44.1%	58.8%	477s
Our Model (non-linear)	33.8%	35.1%	79	89	729	3081	54.3%	60.1%	612s

Table 2.4: Comparison of tracking results on UNIV dataset. The number of trajectories in the ground-truth (GT) is 40.

Method	MT	ML	Frag	IDS	FP	FN	MOTA	MOTP	Time
SGB Model [97]	75%	5%	38	7	213	443	96.7%	82.9%	47s
Our Model (linear)	87.5%	5%	26	5	224	287	97.4%	83.1%	3.9s
Our Model (non-linear)	87.5%	5%	26	5	224	287	97.4%	83.1%	4.2s

2.4.5 Results on UNIV Dataset

To further evaluate the effectiveness of the proposed method in handling dynamics of social groups (e.g., group merge and split), four video sequences are collected from an elevated viewpoint that allows the capture of rich group evolving scenarios. Each video is about 30 seconds long with an average of 9 pedestrians visible in each frame, some sample frames are shown in Fig. 2.11. The input tracklets for this dataset are produced using the second method described in Section 2.4.1. Multi-target tracking is carried out using only the grouping information, namely, the linking costs for tracklet pairs are based only on P_{ij} in Eq. (2.3). The comparative results are shown in Table 2.4. Our model with both linear and non-linear motion have the same performance, as this dataset contains little non-linear motion pattern. Compared to the SGB model which assumes a fixed number of groups in the scene, our grouping model improves MT by 12.5%, reduces the fragments by 31.5%, and also achieves higher MOTA and MOTP. The results imply that our grouping model is better at handling group dynamics in the scence, as it focuses on analyzing elementary groups instead of the complete groups. Sample tracking results of the proposed method are shown in Fig. 2.11. In the second row of Fig. 2.12 we show tracking examples of SGB model on the same sample sequence as shown in Fig. 2.11.

2.4.6 Computational Time

The computational time is greatly affected by the number of targets in a video and the length of the video. All methods are implemented in Matlab without code optimization or parallelization and tested on a PC with 3.0 GHz CPU and 8 GB memory. The average computational times for all the datasets are shown in the last columns in Table 2.1-2.4. Note that the computational times for object detection, tracklet generation, and appearance and

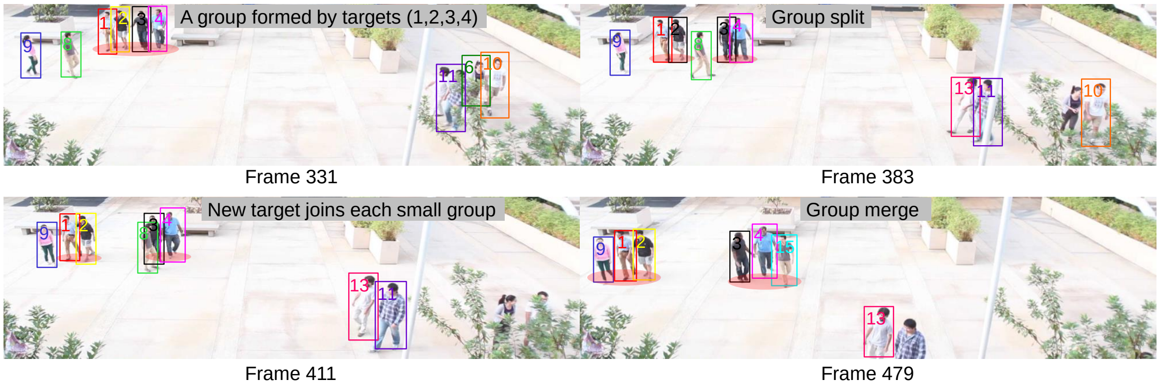

A group formed by targets (1,2,3,4)
Group split
Frame 331
Frame 383
New target joins each small group
Group merge
Frame 411
Frame 479

Figure 2.11: Examples of tracking results of our approach on UNIV dataset. Using only the grouping model, we correctly tracked targets (1, 2, 3, 4) in situations where the group split and merge occur. The same color indicates the same target, best viewed in color.

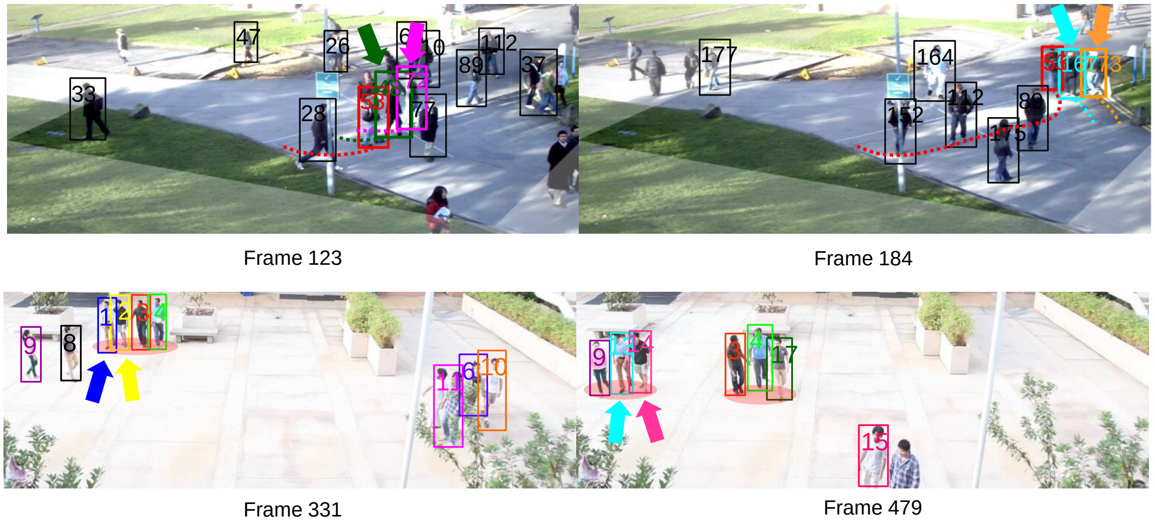


Figure 2.12: Examples of tracking results from referenced models. First row, Our Model (linear) on PETS2009 dataset. Targets (48, 72) cannot be correctly tracked, as tracklet associations generating non-linear motion pattern are penalized when only linear motion model is used. Second row, SGB model on UNIV dataset. Trajectories of targets (1, 2) cannot be fully recovered, because SGB model is not able to link tracklets that are not assigned to the same group. Best viewed in color.

motion feature extraction are not included in the above estimates of computational time. It is clear that Our Models (both linear and non-linear) improve the computational efficiency by an order of magnitude compared with the SGB model which also uses social grouping information in tracking. For the relatively short videos (30 to 66 seconds) in CAVIAR and UNIV dataset, our approach takes 292 fps for the linear version and 235 fps for the non-linear version on average. For the video in TownCentre (3 minutes) the computational time is 10 fps for the linear version and 9 fps for the non-linear version. When our approach is applied on the high crowd density video in PETS2009, the computational time is 0.9 fps for the linear version and 0.7 fps for the non-linear version. It is observed that integrating non-linear context into the motion model increases the computational cost, but still our model is significantly more efficient than the SGB model and produces better tracking results.

From a theoretical perspective, the optimization of SGB is a gradient-based iterative method. To compute the gradient, an alternative approach involving the Hungarian algorithm and K-means clustering is applied. K-means clustering needs multiple initial starts to reach a reasonable local optimum, which leads to high computational cost. Our solver, on the other hand, has a closed form solution based only on the deterministic Hungarian algorithm and thus can be computed much more efficiently.

2.5 Conclusions

In this work we have presented an online approach that integrates high level grouping information into the basic affinity model for multi-target tracking. The grouping behavior is modeled by a novel elementary grouping graph, which not only encodes the grouping structure of tracklets but is also flexible to cope with the evolution of group (i.e., group split and merge). We have used non-linear motion context explicitly for discovering relationships between elementary groups. Experimental results on four challenging datasets demonstrated the superior tracking performance by integrating elementary grouping information. As compared to the state-of-the-art social grouping model, our approach provides better performance in a more computationally efficient manner. However, if there is not much grouping or all the targets follow a linear motion pattern in the input video, the integration of the elementary grouping model will have limited improvement on the tracking performance. Possible future work would be extending the elementary grouping model to multi-person tracking in multiple cameras.

Chapter 3

Multi-Target Tracking in Non-overlapping Cameras Using a Reference Set

3.1 Introduction

As the demand for surveillance cameras at public areas (e.g., airports, parking lots, and shopping malls) is rapidly growing, a major effort has been underway in the vision community to develop effective and automated surveillance and monitoring systems [102, 56, 117, 106, 98]. In most cases, it is not feasible to use a single camera to cover a complete area of interest, and using multiple cameras with overlapping field-of-views (FOVs) has high cost in both economical and computational aspects. Therefore, camera networks with non-overlapping FOVs are preferred and widely adopted in real world applications.

Multi-target tracking is an extensively exploited topic in the surveillance domain, as it is the foundation for many higher level applications, such as anomaly detection, activity detection and recognition [145], and human behavior understanding [24]. The goal of multi-target tracking is to estimate the trajectories of all moving targets and keep their identities consistent from frame to frame. In single camera tracking, successive observations of the same target often have a large proximity in appearance, space and time [97, 28]. However, it is not the case for tracking people across cameras with non-overlapping FOVs. The appearance of the same target may have a large difference even in two adjacent cameras

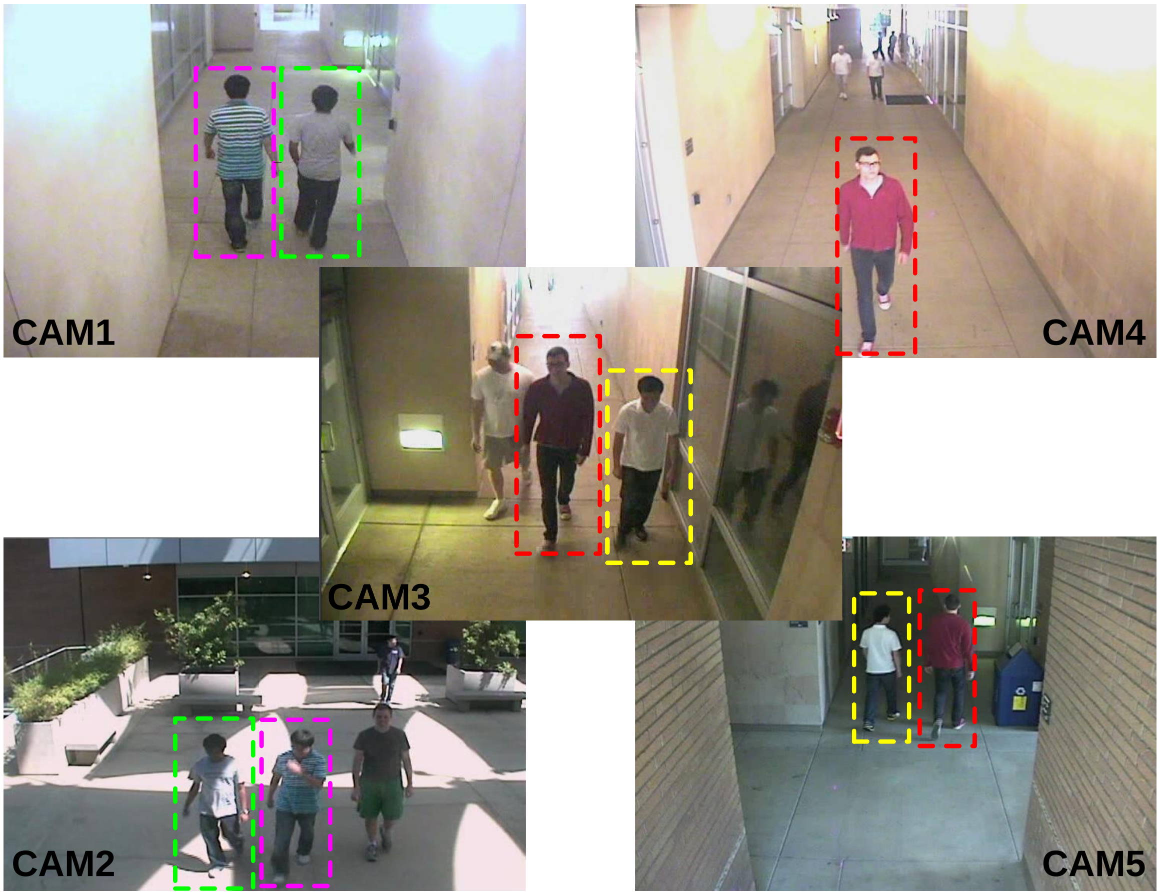


Figure 3.1: Sample frames from each camera view of the MultiCam dataset. Bounding boxes with the same color indicate the same target. Note that illumination may change drastically within camera and across cameras. As a result, the appearance of the same target may vary significantly.

due to a sudden change in illumination (e.g., from outdoor to indoor). Other aspects, such as variations in pose (e.g., frontal view to rear view) and camera imaging conditions (e.g., low resolution and noise) further complicate the tracking task in multiple cameras. In Fig. 3.1 some sample frames are shown in which the appearance of the same target in different camera views differs significantly.

A possible way to tackle the appearance difference in multiple cameras is to learn a Brightness Transfer Function (BTF) [50, 96, 63, 30, 40, 31] that is a mapping of color models between a pair of cameras. However, BTF is not suitable for a camera network that has a large *within* camera illumination change. For example, suppose camera i and camera j both have dark and bright regions in their camera views. A BTF that is able to map colors in dark region of camera i (low brightness) to colors in bright region of camera j (high brightness) will not work well for mapping colors in bright region of camera i (high brightness) to dark region of camera j (low brightness).

To address this problem, we propose a novel reference set based appearance model to estimate the similarity of multiple targets in different cameras. Given the intra-camera tracking results of all involved cameras, the goal of multi-target tracking across cameras is to associate tracks in different cameras that contain the same person. Our method is inspired by the recent advances in face verification/recognition [103, 133, 8] and person re-identification [9] in which an external reference set or a library is used to facilitate the matching process of the same objects imaged under different conditions. The reference set contains the appearance of individuals in different camera views under different imaging conditions. Namely, there are multiple appearance instances for each individual in the reference set. During tracking, instead of comparing the appearance of two targets directly, targets from different cameras are compared to the individuals in the reference set. The individuals in the reference set act like basis functions and for a given target, its similarity to each of the individuals in the reference set are used as its new feature representation instead of the original low level color or texture features.

In order to create a comprehensive representation for each target, besides color features, we also extract shape and texture features from different locations on the body of a target. The discriminative power for each feature is learned using the reference set, and features with high discriminative power contribute more to the similarity score.

The rest of this chapter is organized as follows: an overview of the related work and contributions of this chapter are provided in Section 3.2. Section 3.3 describes the proposed reference set based appearance model for multi-target tracking across non-overlapping cameras. Experimental results are shown in Section 3.4. Finally, Section 3.5 concludes this chapter.

3.2 Related Work and Contributions

3.2.1 Related Work

In general, methods for tracking multiple targets in multiple cameras can be categorized into two groups according to the structure of camera networks: methods for overlapping FOVs and methods for non-overlapping FOVs. Techniques used for tracking in these two groups differ significantly. For instance, tracking in cameras with overlapping FOVs normally require explicit camera calibration [60, 25, 38, 18] while it is not a necessity for tracking with non-overlapping FOVs. As this chapter focuses on inter-camera tracking with non-overlapping FOVs, related work for tracking in overlapping camera views is not discussed.

To cope with the illumination change in different camera views, BTF has been studied extensively [50, 96, 63, 30, 40, 31]. An incremental unsupervised learning method is proposed in [50] to model color variations and posterior probability distributions of spatial-temporal links between cameras in parallel. The model becomes more accurate over time with accumulated evidence. In [96] a cumulative BTF is proposed to map color between different cameras and significant improvement over other BTF-based methods is reported. In [63] the inter-camera relationships is learned using multivariate probability density of space-time variables. It is shown that BTFs from one camera to another camera lie in a low dimensional subspace and this subspace is learned for appearance matching. In [30], BTFs are obtained from the overlapped area during tracking to compensate for the color difference between camera views. In addition, the perspective difference is compensated with tangent transfer functions (TTFs) by computing the homography between two cameras. In [40] different methods are compared to evaluate the color BTFs between non-overlapping cameras and experimental results show BTFs have limitations in people association when a new person enters in camera's FOV. In [31], to track people across non-overlapping cameras, a camera link model including BTF, transition time distribution, region mapping matrix/weight, and feature fusion weight is estimated in an unsupervised manner.

In [54] a combined maximum a posteriori (MAP) formulation is proposed to jointly model multi-camera reconstruction and global temporal data association, in which a flow graph is constructed to track objects. In [67] information from a crowd simulation is integrated into a multi-camera multi-target tracking framework to improve the tracking accuracy. In [84] a data association approach based on principal axis and a joint probabilistic model are applied for multi-object tracking in multi-cameras to overcome occlusion in camera views. In [85] a metric based on three performance indexes is developed to evaluate the performance of multi-camera tracking algorithm based on Rao-Blackwellized Monte Carlo data association (RBMCDA). In [112] a track-before-detect particle filter (TBD-PF) is used to increase track consistency against noisy data for multi-camera multi-target fusion and tracking. In [86] a modified Social Force Model (SFM) with a goal-driven approach for multi-camera tracking is proposed. This work takes into account key regions as potential intersections where people can change the direction of motion. In [29] inter-camera transfer models containing spatio-temporal cues and appearance cues are proposed, which are learned by a topology recovering method and a color characteristic transfer (CCT) method for tracking across non-overlapping cameras.

Recently, the reference-based idea has been used in different fields of computer vision, for example, face verification [103], face recognition [133], and person re-identification [9]. The reference-based framework is data-driven in which different entities to be matched are first described using the samples in the reference set and then reference-based descriptors are generated. Therefore, a direct comparison of objects (e.g., faces at different poses) is avoided. In [103], pose, illumination, and expression invariant face verification is achieved by using a library of faces in various appearances to describe a given face based on the insight that it is most meaningful to compare faces with the same imaging conditions. In [133] an "Associate-Predict" model is proposed which is built on a generic identity data set that contains multiple images with large intra-person variation. Given a face, it is first associated to alike identities in the data set and then its appearance under settings of another input face is predicted. In this way the intra-personal variation is handled. Recently, to improve person re-identification in different camera views, a reference set is used to generate reference-based descriptors for probe and gallery subjects, bypassing the need to direct compare the features from subjects with significant appearance change [9].

3.2.2 Contributions of This Chapter

The contributions and novelty of this chapter are:

- A reference set based appearance model is proposed to mitigate track association ambiguities caused by cross camera illumination and pose variations.

- Each track is divided into several subtracks based on time constraint and appearance similarity to provide multiple appearance instances of a target.

- Various appearance features are extracted from different locations of a target, and their discriminative powers are learned and used to build a robust appearance model.
- Two real-world surveillance datasets are used for evaluation and extensive experiments are carried out to validate the effectiveness of the proposed method.

3.3 Technical Approach

3.3.1 Formulation of the Multi-Camera Tracking Problem

Suppose we have m cameras $C_1, C_2, ..., C_m$ with non-overlapping FOVs. Given the tracking results in each single camera, we can generate a set $T = \{T_1, .., T_N\}$ that contains all within-camera tracks. A track T_i is a consecutive sequence of detections that contain the same target, in a time interval $[t_i^{begin}, t_i^{end}]$, and its corresponding camera is denoted as $C(T_i)$. The problem of tracking across cameras is to find out tracks that contain the same target, given certain spatio-temporal constraints. Let association a_{ij} define the hypothesis that track T_i and T_j contain the same target, with T_i occurring before T_j and $C(T_i) \neq C(T_j)$ (associating tracks that contain the same target in the same camera is not considered in this chapter). A valid association matrix A is defined as follows:

$$A = \{a_{ij}\}, a_{ij} = \{ \begin{array}{ll} 1 & \text{if } T_i \text{ is associated to } T_j \\ 0 & \text{otherwise} \end{array} \tag{3.1}$$

$$\text{s.t.} \sum_i a_{ij} = 1 \text{ and} \sum_j a_{ij} = 1$$

The constraints for matrix A indicate that each track should be associated to and associated by only one other track.

The cost S_{ij} for linking track T_i and T_j is based on time, appearance, and camera topology constraints, as defined below:

$$S_{ij} = Time(T_i, T_j) + Topo(T_i, T_j) + Appr(T_i, T_j) \tag{3.2}$$

where $Time(\cdot)$, $Topo(\cdot)$, and $Appr(\cdot)$ are the time, topology, and appearance models, respectively. The time model is defined as:

$$Time(T_i, T_j) = \{ \begin{array}{ll} 0 & \text{if } 0 < Gap_{ij} < GAP \\ \infty & \text{otherwise} \end{array} \tag{3.3}$$

where Gap_{ij} is the time difference between T_i and T_j, and only when Gap_{ij} is smaller than the pre-defined maximum allowed gap GAP the two tracks can be linked. The topology model is similar to the time model, which gives the restriction that T_i can be associated with T_j only when there is a path allowing people to walk between camera $C(T_i)$ and $C(T_j)$ without entering the view of any other cameras.

Let Ω be the set of all possible association matrices, the task of multi-target tracking in non-overlapping camera views is formulated as the following optimization problem:

$$A^* = \underset{A \in \Omega}{\arg\min} \sum_{ij} a_{ij} S_{ij} \tag{3.4}$$

This assignment problem can be solved by Hungarian algorithm [89] in polynomial time. In order to reduce the computational cost, a pre-defined time sliding window is used, and the association is carried out independently in each time sliding window. Instead of using the cost matrix S directly, we use the augmented matrix S' (details for the augmented matrix can be found in [97]) as the input for the Hungarian algorithm. This augmented matrix enables us to set a threshold for association, a pair of tracks can only be associated when their cost is lower than the threshold. In the following, we present the reference set based appearance model in detail.

3.3.2 Reference Set Based Appearance Model for Across Camera Tracks

The basic idea of reference set based appearance model is illustrated in Fig. 3.2. A reference set $RefSet_{ij}$ is constructed for a pair of cameras C_i and C_j. It contains a set of reference subjects $R = \{R_1, R_2, ..., R_n\}$ that appear in both C_i and C_j. The tracks for all the reference subjects that appear in C_i form $RefSet^i_{ij}$, and the tracks for all the reference subjects that appear in C_j form $RefSet^j_{ij}$, as shown in Fig. 3.2. Given two tracks T_p and T_q with T_p captured in the view of camera C_i and T_q captured in the view of camera C_j, the appearance similarity between these two tracks is not computed by comparing T_p and T_q directly. Instead, T_p is compared with all the tracks in $RefSet^i_{ij}$ and T_q is compared with all the tracks in $RefSet^j_{ij}$, and their similarities with the reference set are used to calculate

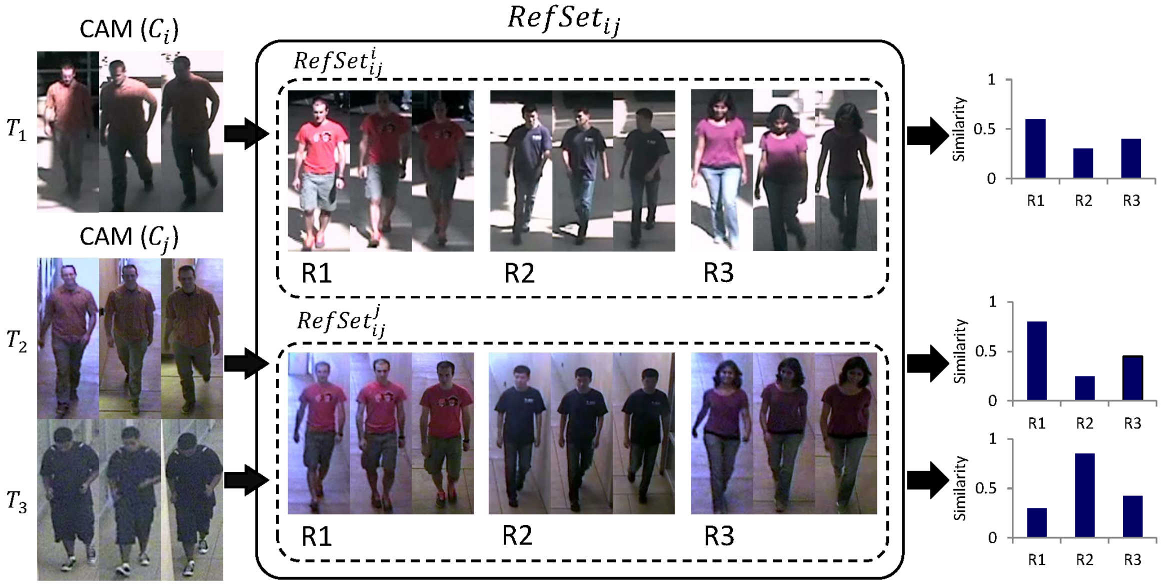


Figure 3.2: Illustration of the reference set based appearance model. For a pair of cameras C_i and C_j, a reference set $RefSet_{ij}$ (the middle part) is constructed containing a number of reference subjects appearing in both C_i and C_j. When comparing track T_1 in C_i with tracks T_2 and T_3 in C_j using their color histograms directly, T_3 is more likely to be matched with T_1. Even though they contain totally different targets, the significant illumination change in C_i makes T_1 looks much darker than its actual appearance. Instead of comparing the tracks directly, each input track is described by all the reference subjects. The description is a vector of similarities ordered by the identities of reference subjects, and each similarity is generated by comparing the input track with one reference subject. The right part of this figure shows the similarity plots obtained by comparing T_1, T_2, T_3 with R_1, R_2, and R_3, respectively. Note that both the input and the reference subjects have multiple appearance instances (only three instances are shown for illustration purpose) that cover the appearance changes of corresponding targets in a particular camera. This indirect match enables us to handle within camera illumination and pose variation. After representing T_1, T_2 and T_3 using the reference set (the right part), it is clear that T_1 is more similar to T_2 than to T_3.

the similarity of T_p and T_q. In other words, track T_p and T_q are compared with other tracks that undergo the same illumination conditions as T_p and T_q, and if they are the tracks of the same target, they should have high similarities with the same set of reference subjects. Otherwise, they are more likely to be the tracks that contain different targets.

In order to handle within camera illumination and pose variation, each track is further divided into several short subtracks (details for track division are presented in Section 3.3.3) such that detections in each subtrack are visually very similar. After track division, each subtrack is an appearance instance for the target under certain illumination condition. Features extracted from each detection in the subtrack are fused into a single set of features, which is used as one representation for the target contained in the subtrack. By this means, we generate multiple representations for each target that covers the appearance changes of that target in a certain camera.

To represent a track by its corresponding reference set, we need to formulate a way of comparing tracks that are obtained in the same camera. When comparing the similarity of two tracks T_a and T_b in the same camera, every subtrack in T_a is compared with every subtrack in T_b. Let t_a^k denotes the k-th subtrack in track T_a, $sim(t_x, t_y)$ be the similarity of two subtracks, and N_a and N_b be the number of subtracks in T_a and T_b, respectively. The similarity score for T_a and T_b is defined as follows:

$$Sim(T_a, T_b) = \frac{1}{N_a} \sum_{i=1}^{N_a} max(\{sim(t_a^i, t_b^j), j \in [1, N_b]\}) \tag{3.5}$$

Concretely, each t_a^i is compared with all subtracks in T_b, and the maximum score is used as the similarity between t_a^i and T_b. Similarity between T_a and T_b is the average of all these maximum scores. The appearance model used to compute $sim(t_a^i, t_b^j)$ is explained in detail in Section 3.3.4.

In the reference set, each reference subject may have several tracks in the same camera (e.g., walking towards and away from the camera). The similarity between a track T_i and a reference subject R_l is the maximum of the similarities of T_i and all the tracks for R_l. This lays the strength of our reference set based appearance model - tracks from different cameras that contain the same target under various pose and illumination conditions have a chance to get high similarity scores with similar reference subjects. In other words, each reference subject is an indirect feature that describes some characteristics of the target's appearance, and having the tracks in two different cameras compared to the same set of reference subjects enables us to better compare the similarity of these two tracks. Besides variation in illumination conditions, difference in poses are also taken care of by the presence of various appearance instances in each reference subject.

After comparing tracks T_p and T_q with each reference subject in its corresponding reference set, we get two vectors of similarities ordered by the identities of reference subjects, as shown in Fig. 3.2. Let $Ref_{ij}^i(T_p)$ and $Ref_{ij}^j(T_q)$ be the representations of T_p and T_q by the reference set $RefSet_{ij}$, the similarity of T_p and T_q is computed using cosine similarity. As it

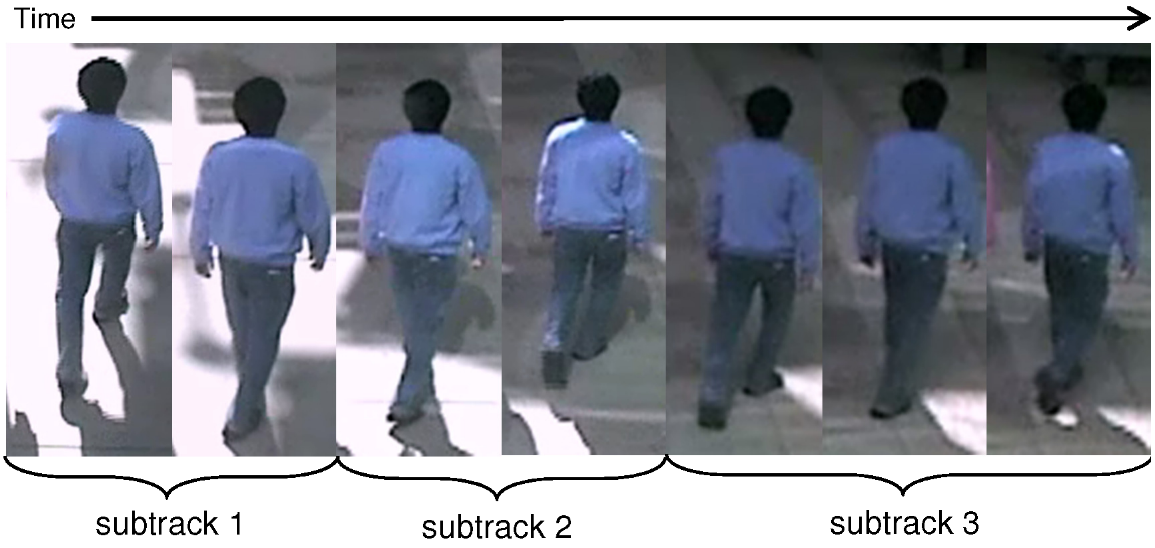


Figure 3.3: An example of track division (only detections on key frames are shown). Detections in the same subtrack have higher appearance similarities as compared to the detections in other subtracks.

is widely used in tracking, negative logarithm is applied to similarity/linking probability to obtain the linking cost, which is then minimized [28, 129, 53]. In order to get the appearance model, we use the negative logarithm function to calculate the cost, as defined in Eq. (3.6):

$$Appr(T_p, T_q) = -\ln(cos(Ref^i_{ij}(T_p), Ref^j_{ij}(T_q))) \tag{3.6}$$

where $cos(\cdot, \cdot)$ is the cosine similarity between two vectors. We also tested other similarity/distance measures (i.e., χ^2 distance, l_2 norm). Among them, cosine similarity and l_2 norm provide comparable performance and are better than χ^2 distance. As cosine similarity is computationally more efficient, it is chosen as the similarity measure in our experiments.

3.3.3 Track Division

In a track, the appearance of a target may vary with time (see Fig. 3.3), but the detection responses that are obtained in consecutive frames often possess high visual similarity. For efficient computation and to create concise representation of a track, we further divide each track into several subtracks and consider every subtrack as an appearance instance of a target. An example of track division is shown in Fig. 3.3

We assume that a target cannot have large pose variation in a very short period of time Δt (0.5s in our experiments). Track division starts from the beginning of a track and subtracks are generated one by one. Let len be the number of frames included in Δt (about 10 in our experiments). The first detection of a track is the appearance reference to form the current subtrack. Specifically, with respect to the detection in the first frame as the reference detection, the detections from the following frames within Δt are compared to the reference detection. As long as the detection similarities are above a pre-defined threshold (0.9 in our experiments), the corresponding frames are kept in this subtrack. Thus, the number of detections in a subtrack is smaller or equal to len. Once a detection's similarity to the reference detection is below the threshold or the number of detections in the current subtrack exceeds len, this detection becomes a new reference detection and detections in latter frames are compared to this reference detection to form a new subtrack. Here color histogram is used to measure the detection similarity.

3.3.4 Appearance Model for Within Camera Subtracks

To build a comprehensive and strong appearance representation, different local and global features are extracted to describe a tracked target. Three kinds of widely used appearance features: HSV color histograms [116], Local Binary Pattern (LBP) [81], and Histogram of Gradient (HOG) [82], are used to capture color, texture, and shape information of a target. Given a detection response, each feature is extracted at different scales and locations to increase the descriptive ability. Specifically, each detection is divided into an upper and a lower part with equal height to provide coarse representations of the torso and the legs of the contained target, as shown in Fig. 3.4. Therefore, nine feature descriptors (BodyHSV, BodyLBP, BodyHOG, TorsoHSV, TorsoLBP, TorsoHOG, LegsHSV, LegsLBP, and LegsHOG) are extracted from each detection response. As there are several detection responses in one subtrack, features of the same type are averaged to construct a concise representation for each subtrack.

Given two subtracks t_a and t_b, we can obtain a similarity score by comparing one of the nine appearance feature descriptors. Let x_i denotes a pair of subtracks (t_a, t_b), a feature vector $f(x_i)$ is generated by concatenating the nine similarity scores. We consider each element in this vector as input to a weak classifier. For color histograms and HOG

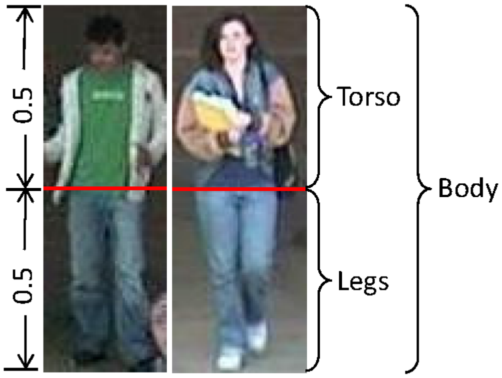


Figure 3.4: Features (HSV color histogram, LBP, HOG) are extracted from different locations of the detection response: torso, legs and body. The torso part is the upper half of the detection and the legs part is the lower half of the detection.

features, we use Bhattacharyya coefficient [33] to measure the similarity. For LBP features, χ^2 distance is used as measurement.

Our goal is to design a discriminative appearance model that gives high similarity for a pair of subtracks that contain similar target while assigning low similarity for two subtracks that contain dissimilar targets. Multiple feature learning algorithms are evaluated (see Section 3.4.3). Due to its superior performance, AdaBoost is selected to learn the appearance model for within camera subtracks, namely, $sim(\cdot)$ in Eq. 3.5. AdaBoost consists of a number of weak classifiers and adaptively learns a strong classifier that is a linear combination of all weak classifiers and minimizes the overall error. In our appearance model, the similarity computed from each feature is used in a weak classifier, and AdaBoost assigns a weighting parameter for each feature during the learning process. We formulate the learned appearance model as follows:

$$sim(t_a, t_b) = H(t_a, t_b) = \sum_{t=1}^{T} \alpha_t h_t(t_a, t_b) \tag{3.7}$$

where t indicates the iteration index, α_t is the weighting parameter and $h_t(t_a, t_b)$ is a weak classifier based on one of the features extracted from subtracks t_a and t_b.

For each camera, we train such a discriminative model using data in the reference set collected from the corresponding camera. A pair of subtracks $x_i = (t_x, t_y)$ is a positive sample if $t_x, t_y \in T_i$, and $t_x \neq t_y$, and it is a negative sample when $t_x \in T_i$, $t_y \in T_j$ and $T_i \neq T_j$. The feature of a training sample is an 9-dimensional vector as explained above.

Algorithm 2 Learning Feature Discriminality

```
Input:
    $\mathcal{S}^+ = \{(x_i, +1)\}$: positive samples
    $\mathcal{S}^- = \{(x_i, -1)\}$: negative samples
    $\mathcal{F} \ = \{f(x_i)\}$: feature pool
    $T$: number of iterations
    $K$: number of weak classifiers
Set $w_i = \frac{1}{2|\mathcal{S}^+|}$, if $x_i \in \mathcal{S}^+$; $w_i = \frac{1}{2|\mathcal{S}^-|}$, if $x_i \in \mathcal{S}^-$
Set $t = 1$, $k = 1$
for $t \leq T$ do
    for $k \leq K$ do
        $r = \sum_i w_i y_i h_k(x_i)$
        $\alpha_k = \frac{1}{2} ln(\frac{1+r}{1-r})$
    Choose $k^* = argmin_k \sum_i w_i exp[-\alpha_k y_i h_k(x_i)]$
    Set $\alpha_t = \alpha_{k^*}$ and $h_t = h_{k^*}$
    Update $w_i \leftarrow w_i exp[-\alpha_t y_i h_t(x_i)]$
    Normalize $w_i$
Output:
    $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$
```

We summarize the learning procedure in Algorithm 2.

3.4 Experiments

Compared to tracking in single camera, there are fewer publicly available datasets designed for real-world multi-camera tracking. In this work, we use two datasets, MultiCam dataset and VideoWeb dataset [37], to evaluate the performance of our proposed model.

3.4.1 Implementaion Details

Targets in each frame are detected via the discriminatively trained deformable part models [43]. We use the multi-target tracking method in [57] to produce reliable intra-camera tracks. It is a hierarchical association approach. First, tracklets are generated by connecting detections in consecutive frames that have high similarity in position, appearance and size using a two-threshold strategy. Then, these tracklets are further associated based on more complex affinity measures to recover the full trajectory of a target.

3.4.2 Baseline Models and Metrics

In this evaluation, our main focus is to associate tracks that contain the same target in different camera views given certain spatio-temporal constraints. We apply our reference set based appearance model with weighted features (RefSet2) on the test set. We introduce three baseline models for comparison: (1) using Bhattacharyya distance of holistic color histograms directly to measure the appearance similarity (Color); (2) generating appearance model based on the BTF model in [63] (BTF); (3) our proposed reference set based appearance model with only holistic color histograms as appearance feature (RefSet1).

For each model, various thresholds (ranging from 0.2 to 0.6) are tested for the augmented cost matrix, and the best result is chosen. Two metrics are used for evaluation:

$$ErrorRate = \frac{Error}{N_{result}}, \ MatchRate = \frac{Match}{N_{GT}} \tag{3.8}$$

where $Error$ and $Match$ are the number of incorrectly and correctly associated track pairs in the result. N_{result} and N_{GT} are the number of track associations in the result and the ground-truth, respectively.

3.4.3 Evaluation of Feature Learning Algorithm

In order to find a suitable learning algorithm to build discriminative appearance models for within camera subtracks, we compare the performance of multiple alternatives, including: AdaBoost [47], GentleBoost [47], LogitBoost [47], RUSBoost [104], and Multiple Kernel Learning (MKL) [115]. The reference set is used as the dataset to test each algorithm. Root Mean Squared Error ($RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}$) is used for performance evaluation, it measures the differences between the ground truth y_t and the prediction results $\hat{y}_t$ generated by the learned appearance model. The final result is the average of five-fold cross-validation. Comparison of different algorithms on MultiCam and VideoWeb datasets are shown in Table 3.1. As can be seen, AdaBoost gives the smallest error on both datasets. With respect to model training time, on MultiCam dataset, the boosting algorithms take less than 2 seconds, and the average training time for MKL is 181 seconds.

Table 3.1: RMSE comparison of different feature learning algorithms on MultiCam and VideoWeb datasets.

	AdaBoost	GentleBoost	LogitBoost	RUSBoost	MKL
MultiCam	**0.228**	0.303	0.235	0.249	0.240
VideoWeb	**0.387**	0.456	0.422	0.468	0.394

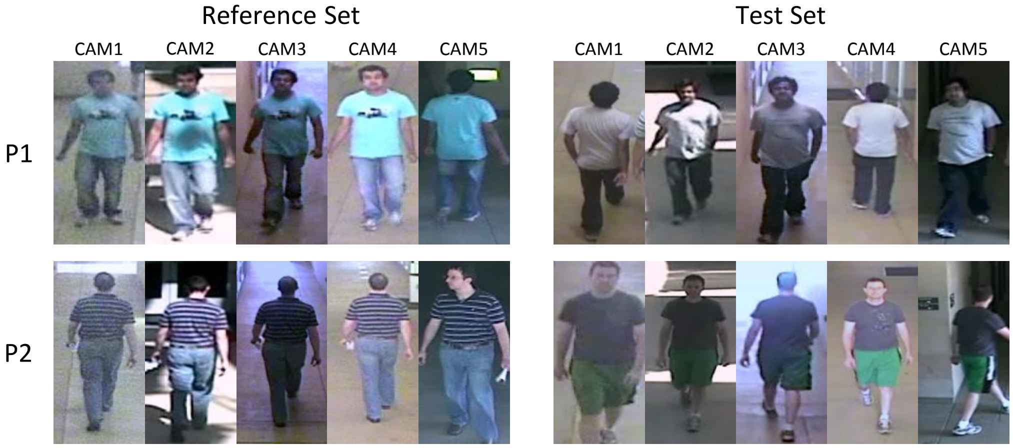


Figure 3.5: Detection examples of participants that appear in both the reference set and the test set for MultiCam dataset.

On the VideoWbe dataset, the training time for MKL is also about two orders of magnitude more than that of the boosting algorithms. Taking both performance and computational time into account, we select AdaBoost as the feature learning algorithm for the appearance model.

3.4.4 Results on MultiCam dataset

We use five cameras (four indoor and one outdoor) to build a real-world non-overlapping multi-camera network, the topology of this camera network is presented in Fig. 3.6 and sample frames from each camera are shown in Fig. 3.1. All the videos (five in total) are taken during the same time period and the length of each video is about 20 minutes. The resolution of each frame is 704 × 480 and the frame rate is 20fps. The number of participants involved in each video ranges from 7 to 10. We refer this dataset as the

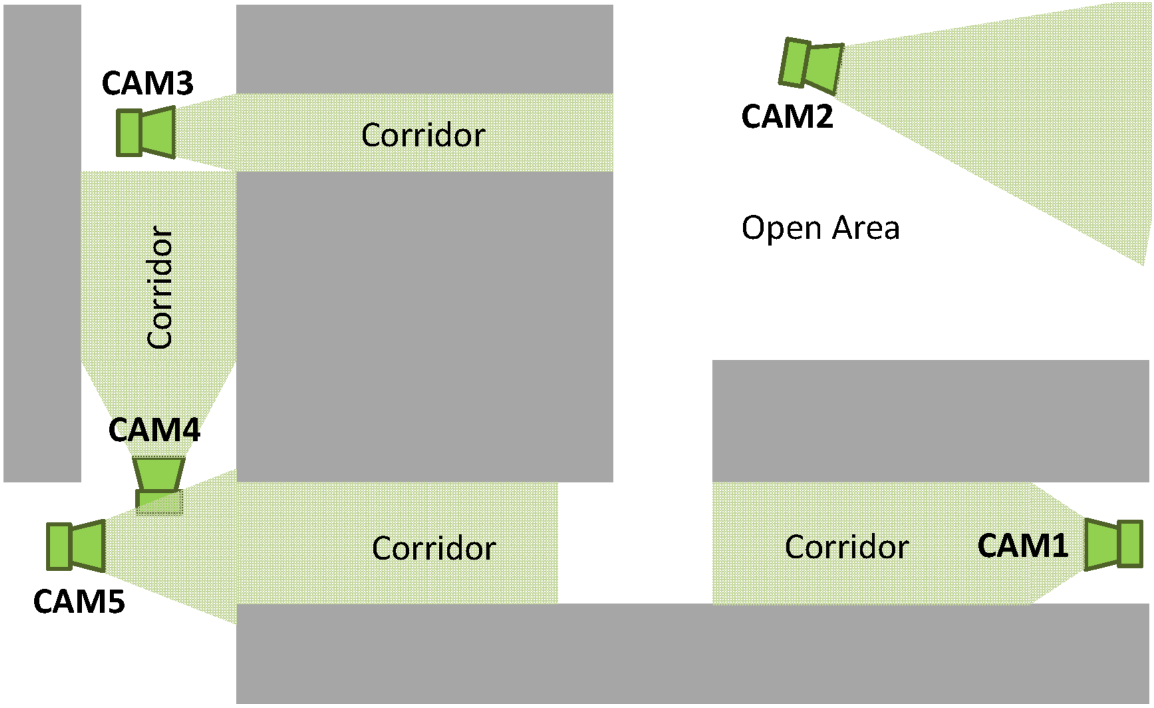


Figure 3.6: Topology for cameras used in MultiCam dataset.

MultiCam dataset in this chapter.

The setting of this dataset is very challenging for multi-camera tracking due to the following reasons:

1. The outdoor camera view contains drastic illumination changes, and there exists lighting variations for indoor camera views as well. This makes it unreliable to use a single transformation to map colors in a pair of cameras, such as BTFs [63].
2. The number of cameras involved in this dataset is greater than most of the previous work that normally use 2-3 cameras [96] [63].

In order to construct the reference set, another set of data is used. It is collected using the same camera network and under similar illumination condition but with participants either not included in the test set or included in the test set but with very different clothes. There are two participants that appear in both the reference set and the test set, as shown in Fig. 3.5. As the appearances of the same participant have a great difference even in the same camera, each of the trained appearance model classifies them as negative (i.e., two different people) with more than 90% confidence. The data collected for each reference subject contains the appearance change of the target under different illumination conditions and various poses. The number of participants involved in each reference set ranges from 9 to 11. We manually labeled the ground-truth which consists of 220 track associations (there are 368 single camera tracks in total).

A quantitative comparison between the proposed model and baseline models is presented in Fig. 3.7. It can be observed that when using the reference set based appearance model with weighted features, we achieve the highest match rate and the lowest error rate compared to all the baseline models. Compared with BTF, the RefSet2 model increases the match rate by 23% and reduces the error rate by 6%. Even with color histograms only, the reference set based appearance model (RefSet1) provides better performance than BTF in terms of both the error rate and the match rate. The comparison between RefSet1 and RefSet2 demonstrates that by using features of various types and extracted at different locations we can get more information than using global color hitograms only, as they capture the appearance information that is overlooked by color hitograms. It is worth noting that although the error rate is high even for RefSet1 and RefSet2 (more than 50%), these results are obtained by using appearance information as the only visual cue.

In addition, to evaluate the contribution of camera topology knowledge to the overall tracking system, we further conducted experiments in which the topology information is not used for computing the linking cost in Eq. 3.2. With such relaxation, more track pairs are included in each time sliding window as potential association candidates. Therefore, the error rates increase by at least 5% for all the methods, and the match rates decrease by 2.5% on average, as shown in Fig. 3.7. These results demonstrate the importance of camera topology information as prior knowlege for a tracking system, as it is helpful for mitigating unnecessary track association ambiguities ahead of time.

As another kind of clue, motion information plays an important role in multi-target tracking. For example, in a time sliding window, a track in CAM4 can be associated with tracks in both CAM3 and CAM5 based on the camera topology (Fig. 3.6). Given the knowledge that the target is walking away from CAM4, we can easily eliminate tracks in CAM5 from possible associations. When a motion model that measures the walking direction of a target is integrated into the tracking system (RefSet2+Motion), the error rate is greatly reduced to 31%. Also, with motion information our proposed method can

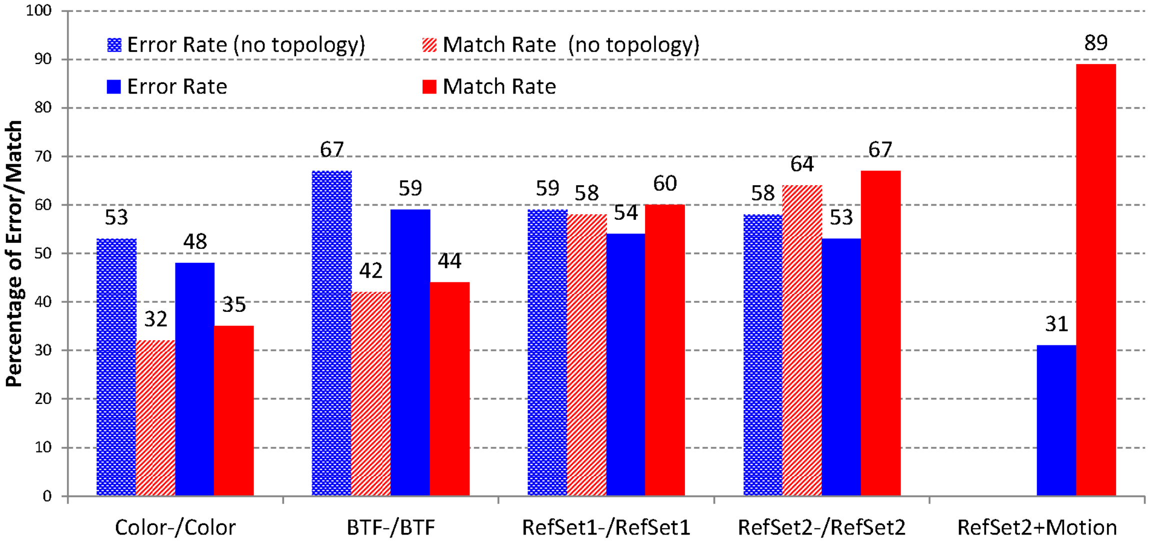


Figure 3.7: Comparison of the proposed method and other baseline models on MultiCam dataset. The minus sign (-) indicates no camera topology knowledge is used for linking cost computation.

correctly associate almost 90% track pairs. Comparison between BTF and RefSet2 on some challenging cases are shown in Fig. 3.11, which validates the robustness of our method.

3.4.5 Results on VideoWeb dataset

In order to further validate our method, we carried out experiments on a public dataset, the VideoWeb dataset [37]. Three cameras CAM20, CAM21, and CAM36 with disjoint views are selected to form the multi-camera network, the topology of which is shown in Fig. 3.8. Three sets of videos are selected as the test set, each videos is about 6 minutes, the resolution of a frame is 640 × 480, and the frame rate is 30fps. Under the same setting, videos from Day3 are used to generate the reference set and videos from Day2 are used to build the test set. Participants involved in Day2 are either not included in Day3 or they are included but with different clothes. There are four participants appearing in both videos from Day3 (reference set) and videos from Day2 (test set), as shown in Fig. 3.9. However, due to significant appearance differences, tracks in the reference set and tracks in the test set, even from the same target and captured by the same camera, are considered to contain two different people (with more than 85% confidence) according to the prediction by the trained appearance model. There are 10 participants involved in the test set, and the

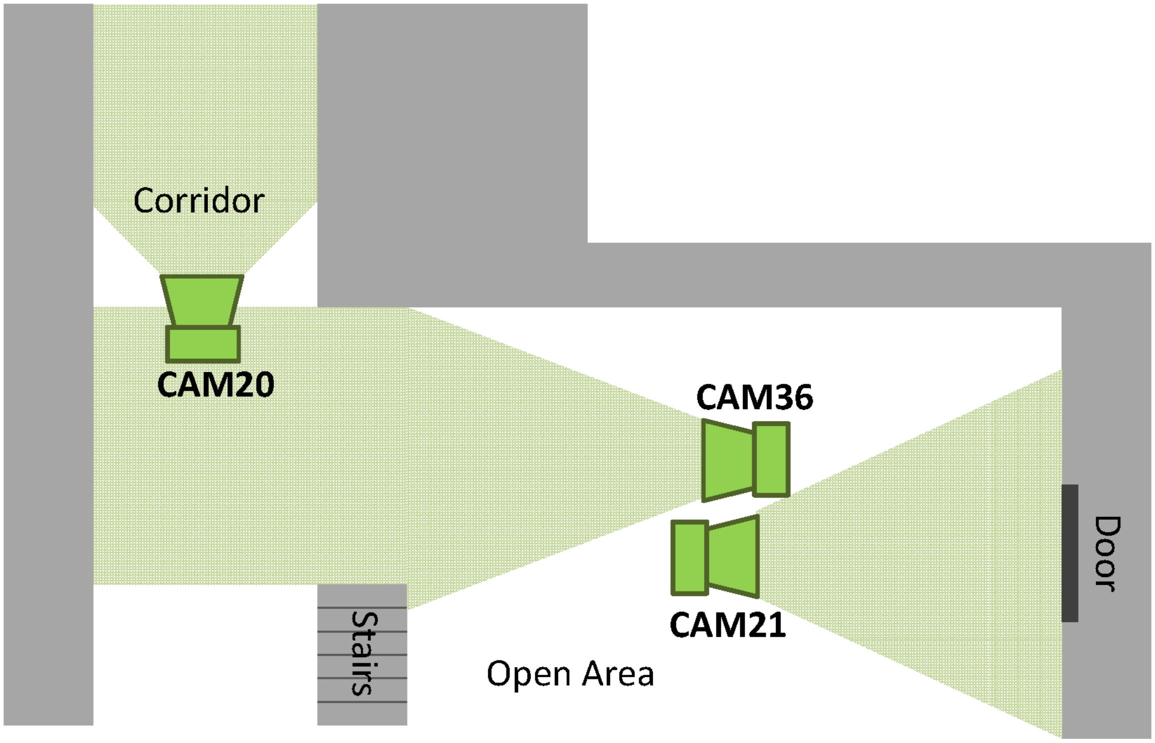


Figure 3.8: Topology for cameras used in VideoWeb dataset.

number of reference subjects in each reference set ranges from 9 to 12. We manually labeled the ground-truth which consists of 66 track associations (there are 222 single camera tracks in total).

A quantitative comparison between the proposed model and baseline models is presented in Fig. 3.10. Using the reference set based appearance model with weighted features (RefSet2) we obtain a match rate of 64% and an error rate of 44%, which is better than the performance of all the other baseline models. Comparison between RefSet1 and BTF (both of them use global color histograms only as appearance feature), further demonstrate the superiority of the reference set based appearance model as it provides a better method to handle track association ambiguities caused by illumination and pose variations across cameras.

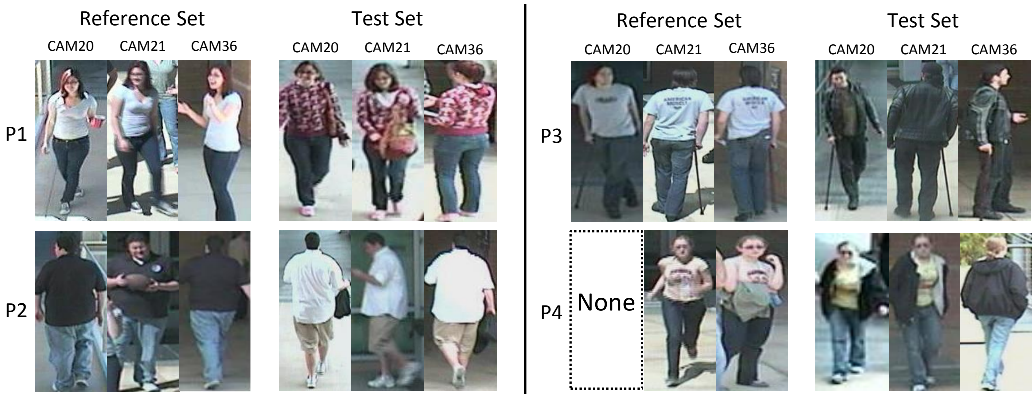


Figure 3.9: Detection examples of participants that appear in both the reference set and the test set for VideoWeb dataset. The "None" box indicates the corresponding participant (P4) generates no track in camera 20 for the reference set.

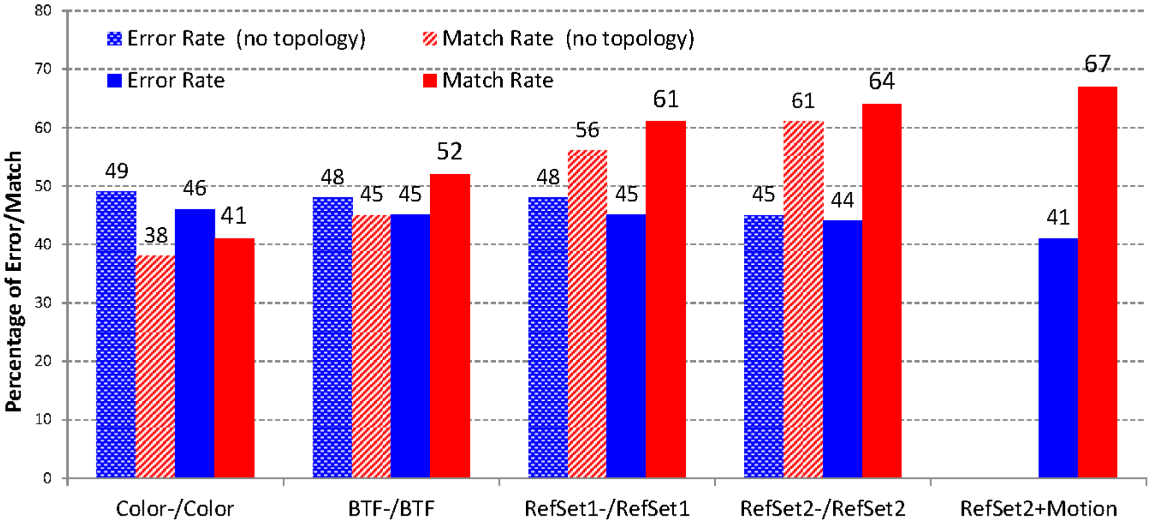
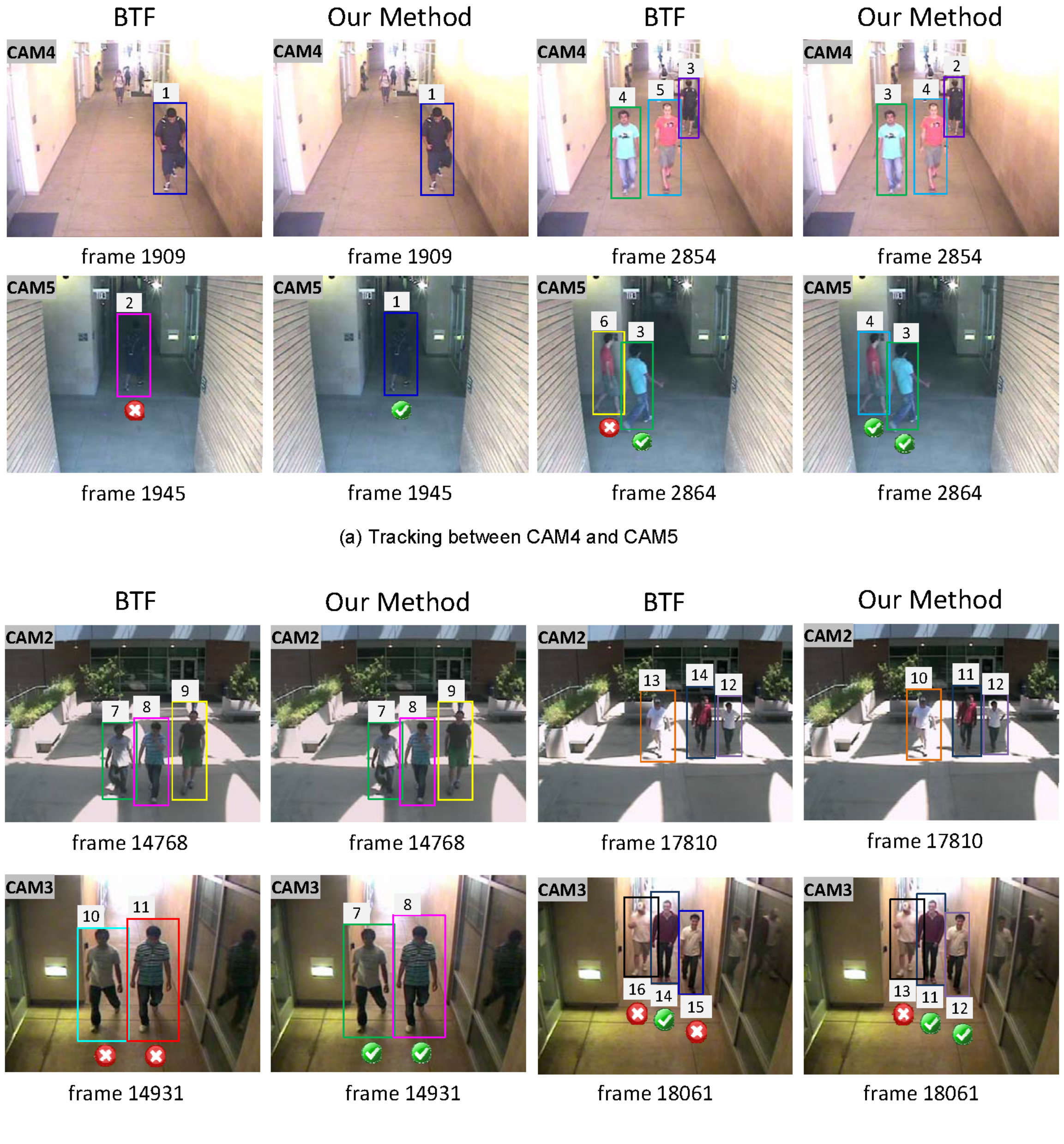


Figure 3.10: Comparison of the proposed methods and other baseline models on VideoWeb dataset. The minus sign (-) indicates no topology knowledge is used for linking cost computation.



(a) Tracking between CAM4 and CAM5

(b) Tracking between CAM2 and CAM3

Figure 3.11: Example tracking results on MultiCam dataset. The first and the third columns are the results obtained by using BTF in [63], the second and the fourth columns are the results by the proposed method using reference set based appearance model with weighted features (RefSet2). With the reference set, our method is able to match most of the targets even with the presence of drastic within camera and across camera illumination variations. The method in [63] fails to associate tracks that contain the same target in these challenging situations. Best viewed in color.

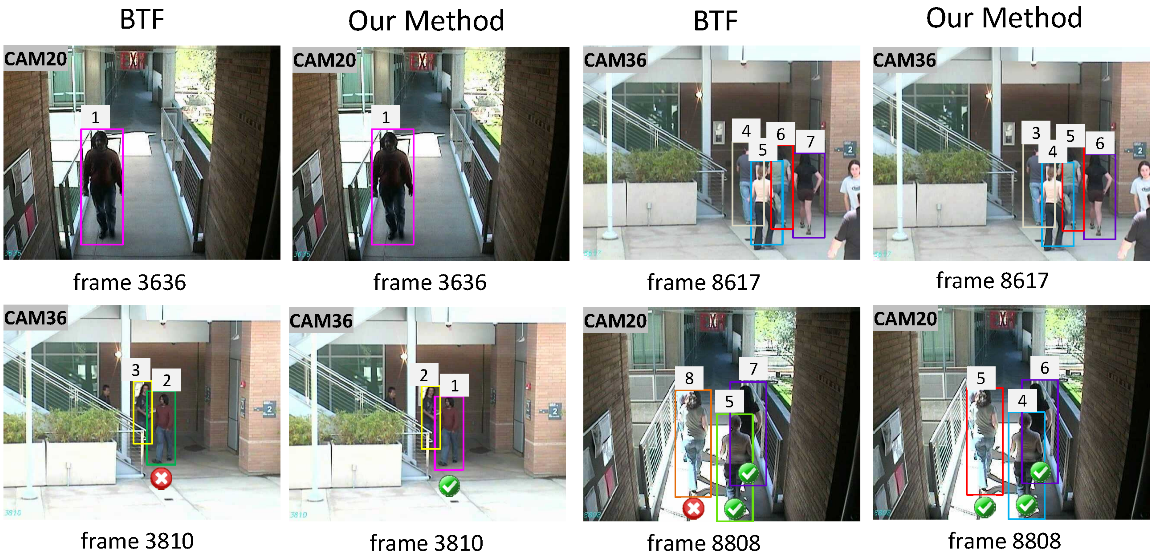


Figure 3.12: Example tracking results on VideoWeb dataset. The first and the third columns are the results obtained by using BTF in [63], the second and the fourth columns are the results by the proposed method using reference set based appearance model with weighted features (RefSet2). Best viewed in color.

We also tested the tracking system without using camera topology information on this dataset. Similar to the observations from Fig. 3.7, with no camera topology information as prior knowledge to reduce unnecessary track associations, higher error rates and lower match rates are observed, as illustrated in Fig. 3.10. However, the impact on the error rate is not as significant as it is on the MultiCam dataset, the error rates increase by at most 3% for all the methods on this dataset. This is probably due to the following two reasons: 1) this dataset has less number of cameras compared to the MultiCam dataset; 2) this dataset contains fewer scenarios in which participants exist in the FOVs of all the three cameras simultaneously. Therefore, the numbers of potential track associations generated by our tracking system with and without camera topology information would be close on this dataset.

Results from Fig. 3.7 and Fig. 3.10 suggest that the performance of our method is consistent on both datasets, which validate the robustness of the reference set based appearance model with weighted features. Note that the ViedoWeb dataset is originally designed for complex real-world activity recognition, participants in this dataset have more non-linear motion and heavy interactions than that in the MutliCam dataset. Therefore,

Table 3.2: Tracking results with different reference set sizes. "N" is the number of reference subjects in the original reference set. "Match" and "Error" stand for match rate and error rate.

RefSet size →	n = N		n = 2/3*N		n = 1/2*N	
Results →	Match	Error	Match	Error	Match	Error
MultiCam	0.67	0.53	0.45	0.49	0.27	0.39
VideoWeb	0.64	0.44	0.43	0.41	0.21	0.32

the overall tracking performance on this dataset is not as good as that on the MultiCam dataset. Also, non-linear motion and interactions among individuals make it difficult to predict accurate motion direction of a target. Thus, after integrating motion model with RefSet2, the improvement on both error rate and match rate is small. Comparison between BTF and RefSet2 on some challenging cases of VideoWeb dataset are illustrated in Fig. 3.12.

3.4.6 Reference Set Analysis

In Table 3.2 we evaluate the performance of our reference set based appearance model with reduced reference subjects. Each time, a subset of the original reference set is randomly selected as a new reference set. The reported results are the average of 10 runs. It is observed that as the size of the reference set reduces the match rate degrades. As less number of track associations are produced in the result, the error rate also decreases. Therefore, the results suggest that for small test sets (about 10 subjects) in order to get good performance from the reference set based appearance model, it is better to make the number of reference subjects comparable to the number of targets in the test set. However, as the size of reference set increases, more redundancy together with more diversity are introduced. Different methods can be used to select a subset from the entire reference candidate pool in order to maintain discriminality while reducing redundancy for better efficiency. For example, in [68] for face recognition, reference set selection is proposed from a low-rank decomposition point of view. In [51] for biometric pattern retrieval, rule-based methods are suggested for reference set selection, including max-variation, max-mean, and min-correlation.

Moreover, the appearances of subjects in a reference set should be as diverse as possible, so that each reference subject can be used to capture some unique characteristic

Table 3.3: Tracking results with and without participants appearing in both reference and test sets. The asterisk sign (*) indicates dataset without identity overlap in reference and test sets.

	GT	ErrorRate	MatchRate
MultiCam	220	0.31	0.89
MultiCam*	174	0.32	0.89
VideoWeb	66	0.41	0.67
VideoWeb*	43	0.43	0.65

of a target. If there are highly similar subjects in the reference set, there will be redundant information in the reference set based appearance descriptor. When such redundancy increases, the performance of the model will be adversely affected.

To evaluate the effect of having participants existing in both reference and test set, we removed the overlap and carried out experiments on both MultiCam and VideoWeb datasets, the results are shown in Table 3.3. As can be seen, the numbers of track associations in the ground-truth decreased as we removed some participants from the test sets, but there was no significant difference in both the error rate and the match rate for datasets with and without overlapping identities. The results justify the rationality of our experiments in Section 3.4.4 and 3.4.5, that is to say, having participants appearing in both reference and test sets but with very different clothes did not impact the performance of the proposed method greatly. This is because only the appearance of reference subjects matters, not the real identities of those particular subject.

3.4.7 Feature Discriminality Analysis

In addition, we further carried out experiments to analyze the discriminative power of all the nine features (HSV, LBP, and HOG extracted on body, torso and legs, respectively) used in the appearance model for within camera subtracks. For each feature, the RMSE obtained when that feature is "removed" from the appearance model is considered as its discriminality measurement. A more discriminative feature would produce a higher RMSE when it is discarded from the feature pool, therefore, it is more important for the learned appearance model. The experimental results for MultiCam and VideoWeb datasets are shown in Fig. 3.13 and Fig. 3.14, respectively. For both datasets, it is clear that HSV are

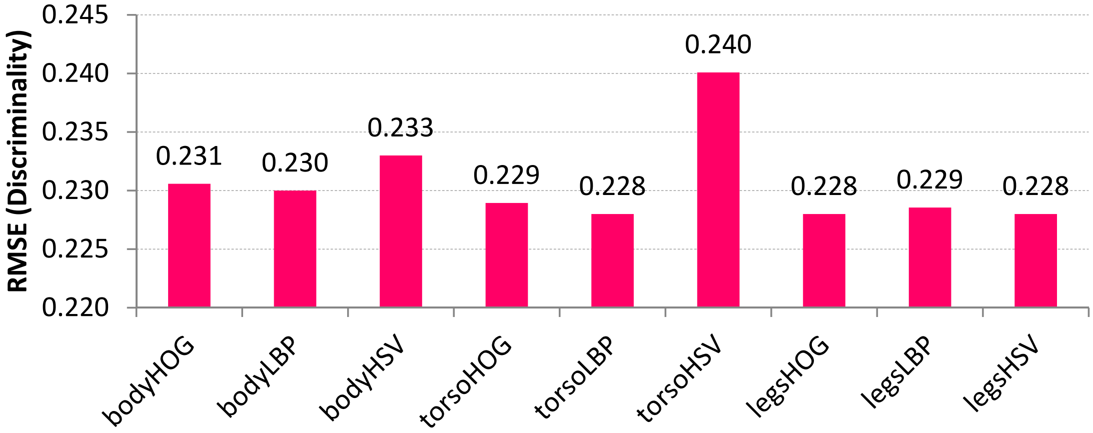


Figure 3.13: Feature discriminality analysis for MultiCam dataset. Each column represents the RMSE (representing discriminality) when the corresponding feature is *removed* from feature pool of the appearance model.

more discriminative than HOG and LBP, and torsoHSV is the most discriminative one. Also, legs carry less information for the appearance model compared to body and torso, probably because most of the participants are wearing jeans with similar color. Since HOG features are not pose invariant, discarding HOG features does not increase the RMSE in the VideoWeb dataset where participants interacted heavily, indicating that in this case HOG features do not have high discriminality. On the other hand, in the MultiCam dataset, we observe slightly higher RMSE for bodyHOG and torsoHOG, as participants in this datasets are less active and their poses remained relatively stable during the data capturing process. Moreover, the RMSE obtained by using all the nine features is 0.228 and 0.387 for MultiCam and VideoWeb datasets, respectively. These RMSE values are equal or smaller than the RMSE values obtained with one of the nine features removed from the feature pool. It is observed that the removal of some features, such as legsHSV and legsHOG, have no impact on the RMSE results. This is plausible since in practice often the upper body dress of the subject being tracked is more distinctive (e.g., shirts with various color and patterns) compared to the lower body dress (e.g., jeans) which is more uniform, as shown, for example, in Fig. 3.1. Although not effective on the datasets used in our experiments, these less discriminative features may contribute to the tracking accuracy when the appearance captured by these features is more discriminative.

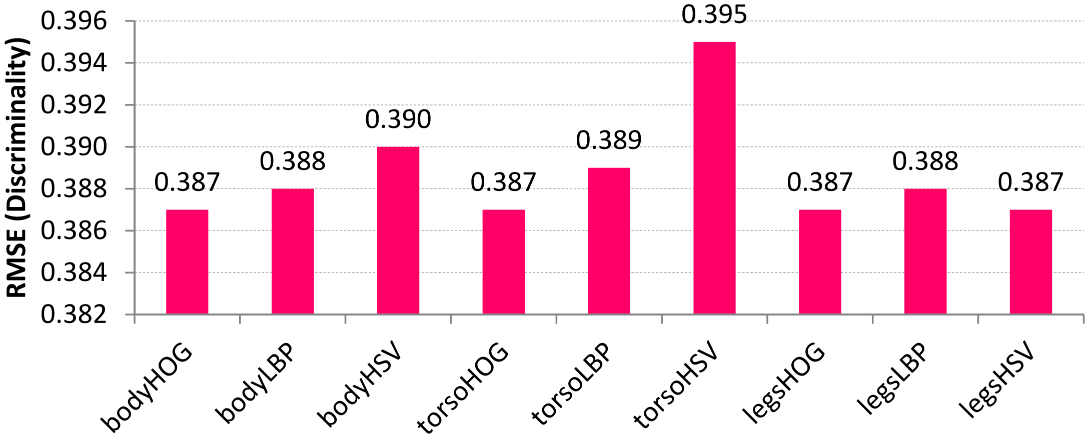


Figure 3.14: Feature discriminality analysis for VideoWeb dataset. Each column represents the RMSE (representing discriminality) when the corresponding feature is *removed* from the appearance model.

3.5 Conclusions

In this paper, we propose a novel reference set based appearance model with weighted features for multi-target tracking in a camera network with non-overlapping FOVs. In order to deal with track association ambiguities caused by illumination and pose variations across cameras, we generate multiple appearance instances for each track and make indirect comparison of two tracks obtained in different cameras by utilizing a reference set. The experimental results demonstrate the superiority of the combination of reference set based appearance model and weighted features over the baseline models on two challenging real-world video datasets. A future work would be testing the proposed reference set based tracking method on larger datasets with more analysis on reference subjects selection.

Chapter 4

Integrating Social Grouping Behavior for Multi-target Tracking Across Cameras in a CRF Model

With more and more surveillance cameras deployed at public places (e.g., airports, parking lots, and shopping malls) to monitor a large area, the demand for effective and automated surveillance and monitoring systems is rapidly growing [102, 56, 117, 106]. Since using multiple cameras with overlapping field-of-views (FOVs) is not cost-efficient in both economical and computational aspects, cameras with non-overlapping FOVs are widely used in real-world applications. Tracking multiple targets across non-overlapping cameras is of great importance, as it is crucial for many industrial applications and high level analysis, such as anomaly detection, crowd analysis, and activity detection and recognition. Although there have been some improvement in this area, it remains a less explored topic compared to single camera multi-target tracking.

The goal of multi-target tracking across non-overlapping cameras is to automatically recovery the trajectories of all targets and keep their identities consistent while they travel from one camera to another, as shown in Fig. 4.1. Compare to single camera tracking, where successive observations of the same target are likely to have a large similarity in appearance, space and time [28], tracking across non-overlapping cameras is a more challenging task due to the following factors.

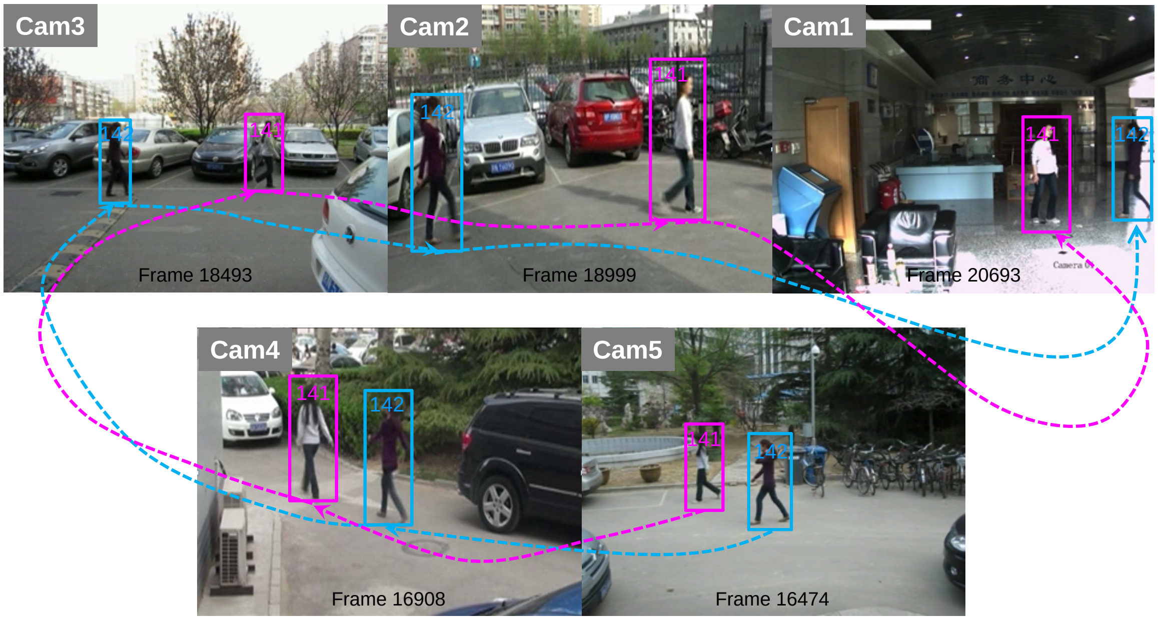


Figure 4.1: Tracking results of our proposed model on Dataset4. Bounding boxes with the same color indicate the same person, and the dashed lines illustrate the trajectories generated by targets walking across different cameras.

- Significant appearance variation. In multi-camera tracking, the observations of the same target in different cameras often have significant difference, caused by illumination variation, pose change, and difference in sensor characteristics.
- Less predictable motion. The open blind area between the FOVs of non-overlapping cameras makes the motion prediction for each target less reliable. When a target leaves the FOV of one camera, he may enter the FOV of another camera, or exit the region under monitoring in the blind area.

In most existing inter-camera multi-target tracking approaches, first intra-camera tracking is carried out in each camera to produce tracks of different targets, then inter-camera tracking is conducted in the form of track association so that consistent labeling of each target across cameras can be achieved. To match tracks from different cameras, prior work mainly rely on appearance and spatial-temporal cues. However, these low-level information is often unreliable especially for tracking in non-overlapping cameras, as discussed above. In this chapter, we further consider integrating high-level contextual information, i.e., social grouping behavior, to mitigate ambiguities in inter-camera tracking.

Sociologists have found that up to 70% of people tend to walk in groups in a crowd for better group interaction [88, 48]. In addition, the “leader-follower” phenomenon generally exists in reality, which means pedestrians are likely to follow other individuals with the same destination either consciously or unconsciously to facilitate navigation [52]. Therefore, when two people are observed walking together in one camera for some time, it is very likely that these two people will appear together in a neighboring camera, an example is shown in Fig. 4.1. Based on the above observations, we proposed an online learning approach for inter-camera tracking which in favor of track associations that maintain group consistency. Note that, we are not only focus on groups that are formed by people who know each other, but also interested in groups of individuals who have correlated movement.

We assume that the intra-camera tracking results of all involved cameras are given, and the topology graph of cameras is known. To associate tracks from different cameras that contain the same person, an online learned CRF model is used, as shown in Fig. 4.2. Track pairs that are linkable under certain spatial-temporal constraints form the nodes in the CRF model. Each node has a binary label (1 or 0) indicating whether the corresponding two tracks are linked or not in the final tracking result. A global appearance model is used to estimate the energy cost for each node. We use elementary groups proposed in [28] to analysis grouping status in each single camera. Two tracks form an elementary group if they have similar motion pattern and are temporally close to each other. Single camera grouping information is used to infer across camera grouping behavior. If two nodes in the CRF model contain at least one elementary group, an edge is created between them. Energy cost for each edge is estimated using the combination of both grouping and appearance information. For each track, we online learn a target-specific appearance model using AdaBoost. If two linked nodes not only have a high probability to maintain group consistency across cameras, but also have high appearance affinities according to target-specific appearance models, their corresponding edge will be assigned a small energy cost. Then the tracking task is formulated as an energy minimization problem, i.e., to find label assignment for the CRF graph that produces the smallest overall energy cost.

The rest of the chapter is organized as follows: Section 4.1 discusses related work and presents contributions of this chapter; the proposed CRF model and its corresponding approximation algorithm are described in Section 4.2; experiments are given in Section 4.3; and Section 4.4 concludes this chapter and provides possible future work.

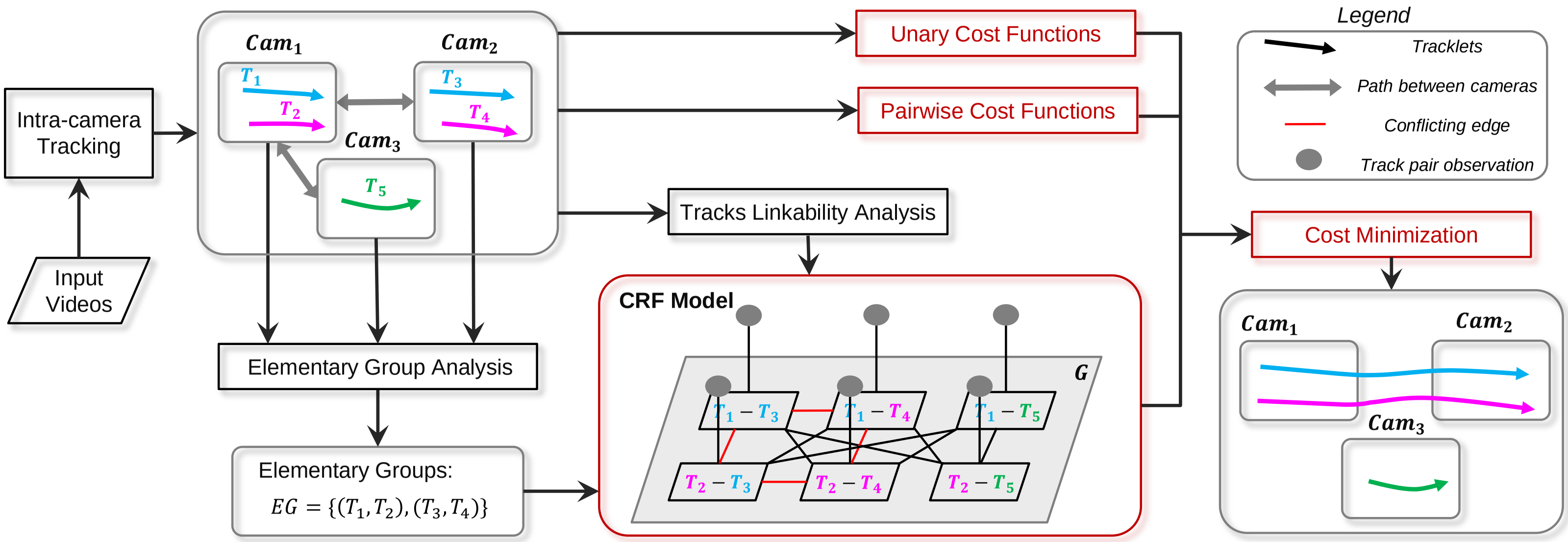


Figure 4.2: Block diagram of our tracking system resented with a simple illustrative example. Blocks shown in red contain novelty part of this chapter. Tracklets with the same color contain the same target. Best viewed in color. For the legends in this figure please see the box in the upper right hand side.

4.1 Related Work and Contributions

4.1.1 Related Work

Multi-target tracking across cameras has been an active topic in computer vision for many years, a recent comprehensive survey for this problem can be found in [117]. The inter-camera tracking is essentially a data association task, in which same subject's tracks are to be matched. Due to the illumination and pose change across cameras, such data association is inherently challenging.

Among various approaches for multi-target tracking, appearance cue is commonly utilized. To tackle illumination change, Brightness Transfer Functions (BTFs) have been exploited [50, 63, 96]. BTFs model color changes between a pair of cameras through mapping functions. Variations of BTFs include multi-variate probability density function [63], joint brightness and tangent functions [30], etc. Evaluations of different BTFs are performed in [40] and the findings suggest that under certain conditions, such as the entering of a new subject, BTFs are prone to error. Besides BTFs, color correction models can also be used for tracking objects [110, 55]. In general, learning BTFs or color correction model requires large amount of training data and these models may not be robust against drastic illumination changes across different cameras.

In addition, spatial-temporal cue can be combined with appearance cue to improve multi-target tracking performance. For example, Kuo *et al* [73] learned a discriminative appearance model in a Multiple Instance Learning (MIL) framework, which can effectively combine multiple descriptors and similarity measures. This appearance model is used in conjunction with spatio-temporal information for improved tracking accuracy. The work in [139] exploits spatio-temporal relationships between targets to identify group merge and split events with time. It is designed to simultaneously track individuals and groups in a camera network, which is important for the problem of tracking in a cluttered scene. In addition, both spatio-temporal context and relative appearance context can be used jointly for inter-camera multi-target tracking. For example, in [23], the spatio-temporal cue supports sample collection for appearance model learning, and the relative appearance context helps disambiguate people in proximity. An inter-camera transfer model, including both spatio-temporal and appearance cues, is proposed in [29]. Particularly, the spatio-temporal model is learned using an unsupervised topology recovering approach, and the appearance model is learned by modeling color changes across cameras.

Another recently popular research topic, person re-identification, is closely related to inter-camera multi-target tracking. Both problems aim to match observations of the same people across non-overlapping cameras. However, in most person re-identification work, only a single or multiple snapshots of people are to be matched. Therefore, contextual information is often not available for person re-identification problem. On the other hand, in an inter-camera tracking problem, each person is presented by a track, which is a string of detections extracted from consecutive frames. In order to handle the large intra-class variation in person re-identification, robust appearance models have been studied [74, 131, 79, 141]. Another way is to learn specialized distance metrics or feature transformations [143, 122, 7, 10, 113]. For training purpose, a training set with corresponding detection pairs, which share similar imaging conditions as the testing samples, is required.

While most of the previous works (e.g., [26, 29]) only consider pairwise relationships using global optimization techniques such as Hungarian algorithm, we employ CRF to simultaneously model both pairwise and higher order relationships for track association. Compared to person re-identification, in which only images of the subjects are matched, our framework is a dynamic system, meaning that the track association is executed to cover both spatial and temporal spans. Such system is more desirable for real-time tracking and monitoring in practical applications.

4.1.2 Contributions of This Chapter

The contributions of this chapter include:

- A novel CRF framework that combines social grouping behavior with traditionally used appearance and spatial-temporal cues for robust multi-target tracking across non-overlapping cameras.

- An online learning approach for modeling unary and pairwise energy costs in the CRF model. The proposed approach does not require a large training set with known correspondence between samples, and can be easily updated to adapt environmental changes.

- An effective approximation algorithm for the CRF model that produces good tracking results with low energy cost.

- Evaluation on four challenging real-world surveillance video sequences are used to validate the effectiveness of the proposed method.

4.2 Technical Approach

4.2.1 CRF Model for Inter-camera Tracking

In this section, we introduce how to formulate inter-camera tracking as a inference problem using the CRF framework. An outline of the proposed tracking system is illustrated in Fig. 4.2.

Given a set of tracks $T = \{T_1, T_2, ..., T_N\}$, which is the intra-camera tracking results of M non-overlapping cameras $Cam_1, Cam_2, ..., Cam_M$. Each track T_i is a string of detections that correspond to the same person and extracted from a set of continuous frames. The time interval for T_i is denoted as $[t_i^{begin}, t_i^{end}]$, and its corresponding camera is $Cam(T_i)$. The task of inter-camera multi-target tracking is to associate tracks from different cameras that contain the same person under certain spatial-temporal constraints. Since the CRF framework is capable of encoding relationship between observations, it is especially suitable for modeling contextual information in the scene.

We create a CRF graph $G = \{V, E\}$. Each vertice $v_i = (T_i^1, T_i^2)$ in V represents a linkable pair of tracks, assuming T_i^1 starts before T_i^2, and each edge $e_j = (v_j^1, v_j^2)$ in E indicates that the connected two vertices are correlated (detailed explanations for CRF graph creation is presented Section 4.2.2). Let $L = \{l_1, l_2, ..., l_m\}$ be a set of binary labels for all vertices, i.e., all possible track associations, with $l_i = 1$ indicating T_i^1 is associated with T_i^2 in the final tracking result, and $l_i = 0$ represents the opposite. During tracking, our goal is to find the label configuration L^* that maximizes the overall linking probability given T. Mathematically, the inter-camera track problem can be defined by the following optimization equation:

$$L^* = \arg\max_L P(L|T) = \arg\min_L \frac{1}{Z} exp(-\Psi(L|T)), \tag{4.1}$$

where Z is a normalization factor that does not depend on L, and $\Psi(\cdot)$ is a potential/cost function. We assume that the joint distributions of more than two associations have no contributions to the conditional probability $P(L|T)$, then

$$
\begin{aligned}
L^* &= \arg\min_L \Psi(L|T) \\
&= \arg\min_L (\sum_i U(l_i|T) + \sum_{i,j} B(l_i, l_j|T)),
\end{aligned} \tag{4.2}
$$

where $U(l_i|T)$ and $B(l_i, l_j|T)$ are the unary and pairwise energy functions and correspond to the node and edge costs in the CRF graph, respectively. Learning of the unary and pairwise costs are described in Section 4.2.3 and Section 4.2.4.

For efficiency, track association is not applied on the entire videos. Instead, a pre-defined time sliding window is used, and a CRF model is online learned for each sliding window. Moreover, in order to prevent impractical associations, a valid label set L needs to follow certain constraints. Let L^1 be the set of all labels that are assigned to 1, namely, $L^1 = \{l_i = 1\} \forall l_i \in L$. Similarly, L^0 corresponds to the set of labels assigned with 0. For a label l_k, with its corresponding vertice denoted as $v_k = \{T_k^1, T_k^2\}$, we use $C(l_k)$ to represent the set of its conflicting labels. A label l_x is conflicting to l_k, if its corresponding vertice $v_x = \{T_x^1, T_x^2\}$ falls into one of the following patterns: 1) $T_x^1 = T_k^1$ and $T_x^2 \neq T_k^2$; 2) $T_x^2 = T_k^2$ and $T_x^1 \neq T_k^1$. Then L is a valid label set, if

$$
\forall l_k \in L^1, C(l_k) \subset L^0 \tag{4.3}
$$

This constraint implies that each track can be associated to and associated by only one other track.

4.2.2 CRF Graph Creation

In the CRF graph, each vertice represents a pair of linkable tracks. Track T_i can be associated to T_j if they satisfy the following spatial-temporal constraints.

- Spatial constraints: First, T_i and T_j are captured in different cameras, namely, $Cam(T_i) \neq Cam(T_j)$. Second, according to the camera topology graph, linking T_i and T_j forms a feasible path allowing people to walk from $Cam(T_i)$ to $Cam(T_j)$ without entering the FOV of any other cameras.

- Temporal constraints: T_i starts before T_j. Let $Gap_{ij} = t_j^{begin} - t_i^{end}$ be the time gap between these two tracks, then $0 < Gap_{ij} < GAP$ should hold, where GAP is a threshold for maximum time gap between any two linkable tracks.

The spatial constraints enable us to focus only on inter-camera tracking, as well as eliminate those practically infeasible track associations. The temporal constraints prevent us from linking track pairs with time overlap, as one individual cannot appear at two different places at the same time. The threshold GAP avoids track pairs outside the time sliding window to be considered.

Given a set of tracks, the linkability of any two tracks is evaluated according the above spatial-temporal constraints. A set of vertices V is created, and each vertice in V denotes a pair of linkable tracks as

$$\begin{gathered} V = \{v_i = (T_i^1, T_i^2)\} \qquad (4.4) \\ \text{s.t. } T_i^1 \text{ can be linked to } T_i^2. \end{gathered}$$

In order to build edges between the vertices in the CRF graph, we first find elementary groups in each single camera. Elementary group is a flexible structure for within-camera grouping analysis [28]. An elementary group is a group including only two people that move with similar motion pattern and are temporally close to each other. Because the number of groups and the sizes of groups in the scene are unknown and may change over time, learning the complete group structure directly is quite challenging. Elementary group provides a simple but effective way for inferring useful group information, since a group of any size can be presented by a set of elementary groups. Note that, elementary group analysis is carried out in an online mode.

In a single camera, track T_i forms an elementary group with T_j if they have the following properties: 1) T_i and T_j co-exist for at least t seconds (t is set to 2 in our experiments); 2) the angle between the velocities of T_i and T_j is smaller than 45 degree. The first constraint guarantees that the two tracks in an elementary group are temporally close to each other. As we assume there is only a small variation in the walking speed of all pedestrians, two targets are considered as dynamically correlated if they walk toward approximately the same direction. Unlike [28], we relax the elementary group criterion by removing the spatially close constraint. Because [28] focuses on intra-camera tracking where spatial distance plays an important role, while this chapter deals with inter-camera tracking, where such information is less useful as tracks to be associated are inherently not close to each other. In addition, with the relaxed criterion, elementary groups can be constructed from more “leader-follower” instances, thus more contextual information can be obtained.

Let $EG = \{g_i = (T^1_{g_i}, T^2_{g_i})\}$ be the set of elementary groups found in all cameras. An edge is created for two vertices $v_i = (T^1_i, T^2_i)$ and $v_j = (T^1_j, T^2_j)$, if at least one elementary group can be formed by the four involved tracks. Mathematically, we define a set of edges E for the CRF graph as:

$$\begin{aligned} E = \{(v_i, v_j)\}\ \forall v_i, v_j \in V \\ \text{s.t. } (T^1_i, T^1_j) \in EG \text{ or } (T^2_i, T^2_j) \in EG. \end{aligned} \tag{4.5}$$

Moreover, edges are divided into conflicting ones and non-conflicting ones. A conflicting edge means that the connected two vertices can not be assigned with label 1 at the same time, in order to guarantee a valid label set. Note that, edges are created between vertices containing targets with the same motion direction, e.g., from Cam_1 to Cam_2. During tracking, the set of track pairs that maintains the overall group consistency are more likely to be associated. In the example shown in Fig. 4.2, two elementary groups (T_1, T_2) and (T_3, T_4) are found based on all the input tracks. Therefore, if we know T_1 and T_3 have a high probability to be associated, then the probability for linking T_2 and T_4 should be increased, as the same group of people are likely to re-appear together in a neighboring camera. Besides overall group consistency, the associated tracks should also keep appearance consistency based on online learned target-specific appearance models. Both group and appearance consistency are estimated by online learned pairwise costs (see Section 4.2.4).

4.2.3 Unary Energy Functions

Unary energy functions in Eq. 4.2 evaluate the energy cost for linking two tracks. The cost is defined as the negative log-likelihood of two tracks being the same target according to a global appearance model $P_{app_1}(\cdot)$,

$$U(l_i = 1|T) = -lnP_{app_1}(T^1_i, T^2_i|T). \tag{4.6}$$

Track Division

In a track, detections from adjacent frames often have high appearance similarity. In order to reduce redundancy and create concise and robust representation, for each track we combine visually similar detections into a subtrack and consider each subtrack as an appearance instance of a target, as used in [26].

More specifically, given a track, its first detection is used as a reference detection for its first subtrack. Following detections that have high appearance similarity (≥ 0.9) compared to the reference detection are included into the first subtrack. When a detection's similarity to the reference detection is below 0.9, this detection is considered as the reference detection for the next subtrack. The process continues until we reach the end of the track. Additionally, we set the maximal length of a subtrack to 20 frames (about 1 second), to ensure there is no large pose variation for detections contained in the same subtrack.

Color Transfer

In order to compute appearance similarity of tracks from different cameras, we first need to handle appearance variance across cameras. In this chapter, we adopted the color transfer method proposed in [101, 29] as a pre-pocessing step to normalize color between different cameras. Given two images, the color transfer method achieves color normalization by imposing the color characteristics of one image (target image) onto the other (source image), as shown in Fig. 4.3. In our experiments, the first full image from one camera is used as the target image, and images from other cameras are considered as source images.

As correlations exist among the three different color channels of the RGB color space [101], to change the color of one pixel, the values of this pixel in all channels must be modified. Such correlations are undesirable for color transfer. Therefore, images are transferred from the original RGB color space to the $l\alpha\beta$ color space, where there is little correlation between different color channels. Then, the target image is transformed according to the color characteristics exacted from the source image, as follows:

$$\begin{aligned} l^* &= \frac{\sigma_t^l}{\sigma_s^l}(l_s - m_s^l) + m_t^l, \\ \alpha^* &= \frac{\sigma_t^\alpha}{\sigma_s^\alpha}(\alpha_s - m_s^\alpha) + m_t^\alpha, \\ \beta^* &= \frac{\sigma_t^\beta}{\sigma_s^\beta}(\beta_s - m_s^\beta) + m_t^\beta, \end{aligned} \tag{4.7}$$

where l, α, and β represent the pixel value in a corresponding color channel, m and σ denote the mean and standard deviation of one image. Target and source images are indexed by

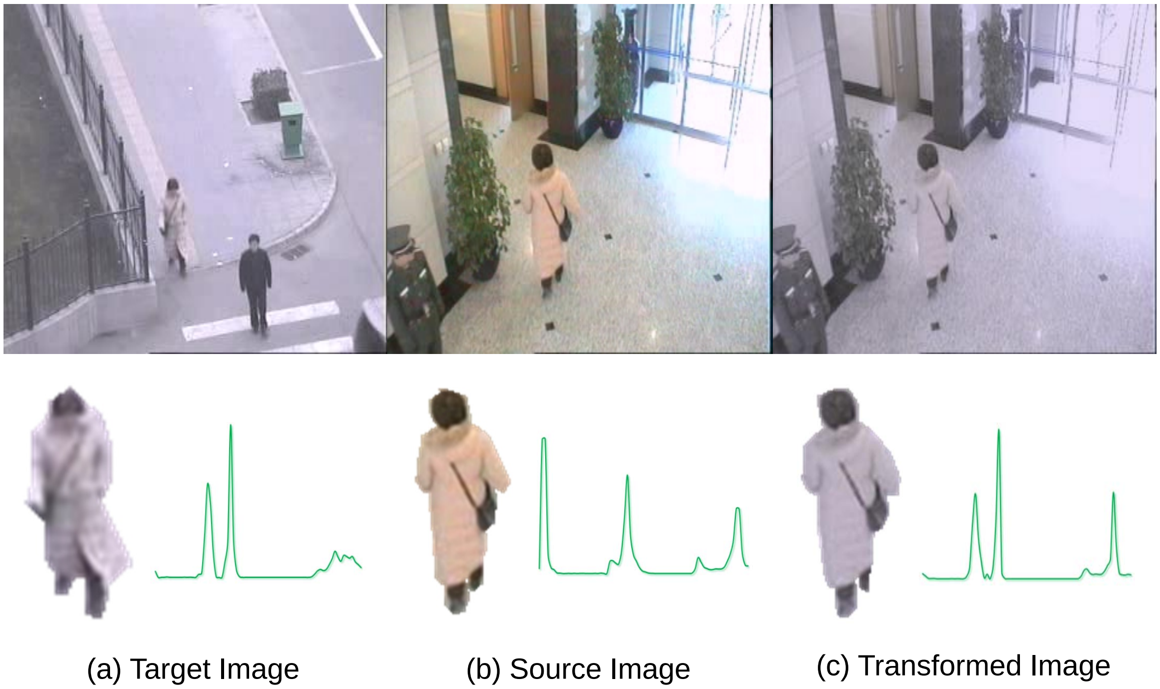


Figure 4.3: An example of applying the color transfer method on images obtained by two different cameras (one outdoor, one indoor). The full images captured by each camera are shown in the first row. The person appears in both cameras and its corresponding HSV color histograms are presented in the second row. It is obvious that the person in the transformed image is more alike to the person in the target image based on HSV color histograms.

subscript t and s, respectively. $[l^*, \alpha^*, \beta^*]$ is the representation of the transformed image in the $l\alpha\beta$ color space. After color transformation, the transformed image is converted back to the RGB color space from the $l\alpha\beta$ color space.

Given two tracks T_i and T_j with $Cam(T_i) \neq Cam(T_j)$, HSV color histograms are extracted from each detection. The average of HSV color histograms from the same subtrack is regarded as appearance descriptor for the target contained in the track. The global appearance model for T_i and T_j is defined as

$$P_{app_1}(T_i, T_j|T) = \frac{1}{R}\sum_{n=1}^{R} BC(d_n^i, d_n^j), \tag{4.8}$$

where d_n^i is the nth subtrack in track T_i. $BC(\cdot)$ is the Bhattacharyya Coefficient [33], it is used as a measure for the appearance similarity of two subtracks. R subtracks are randomly selected from each track, and their average similarity is used as the similarity for T_i and T_j.

4.2.4 Pairwise Energy Functions

The pairwise energy functions are formulated according to global grouping cues and target-specific appearance cues, as defined in Eq. 4.9,

$$B(l_i, l_j|T) = -ln(P_{group}(l_i, l_j|T) \times P_{app_2}(l_i, l_j|T)), \tag{4.9}$$

where P_{group} is the probability of maintaining group consistency for a specific assignment of (l_i, l_j), and P_{app_2} is the probability of keeping appearance consistency based on the value of l_i and l_j. Details for P_{group} and P_{app_2} are presented in the following parts.

Group Consistency

According to the observation that two people walking together for a certain time in one camera are likely to re-appear together in a neighboring camera, given the labels of two connected vertices in the graph, we can infer its probability of maintaining group consistency.

Let $v_i = (T_i^1, T_i^2)$ and $v_j = (T_j^1, T_j^2)$ be two possible track associations, without knowing the edge configuration of the graph, the probability of maintaining group consistency for a specific label assignment of (l_i, l_j) is $\frac{1}{C}$, where C is the number of all possible values for (l_i, l_j). Assuming we know v_i is connected to v_j in the graph, which indicates that $Cam(T_i^1) = Cam(T_j^1)$, $Cam(T_i^2) = Cam(T_j^2)$. If both (T_i^1, T_j^1) and (T_i^2, T_j^2) are elementary group, then assigning (l_i, l_j) to $(1, 1)$ should produce $P_{group} = 1$ as it maintains the group consistency. For instance, in the example shown in Fig. 4.2, as (T_1, T_2) and (T_3, T_4) are both elementary groups in Cam_1 and Cam_2, then assigning $(1, 1)$ to veritices (T_1, T_3) and (T_2, T_4) keeps the group consistency compared to the other alternatives (i.e., $(1, 0)$, $(0, 1)$, and $(0, 0)$).

Based on the above analysis, we define P_{group} as

$$P_{group}(l_i, l_j|T) = \begin{cases} 1 & \text{if } l_i = l_j = 1, \\ & (T_i^1, T_j^1) \in EG,\ (T_i^2, T_j^2) \in EG, \\ \frac{1}{C} & \text{otherwise.} \end{cases} \tag{4.10}$$

Note that if (v_i, v_j) is a non-conflicting edge, $C = 4$, as there are four possibilities, i.e., $(1, 1)$, $(1, 0)$, $(0, 1)$, $(0, 0)$, for the label assignment of (l_i, l_j). But if (v_i, v_j) is a conflicting edge, indicating l_i and l_j cannot have label 1 at the same time, thus $C = 3$ for such cases.

Local Appearance Consistency

It is obvious that from group consistency alone we cannot obtain sufficient information to make confident track association decisions. Therefore, we integrate local appearance consistency into the pairwise energy functions. An edge possesses local appearance consistency if the label given to each related vertice in accordance with appearance similarity/dissimilarity of the corresponding track pair.

Mathematically, given an edge (v_i, v_j), where v_i contains track pair (T_i^1, T_i^2) and v_j includes (T_j^1, T_j^2). Let App_{ik} be a discriminative appearance model learned for track T_i^k, which produces high similarity for track that contain similar target as T_i^k, and gives low similarity otherwise. Then we define P_{app_2} as

$$\begin{aligned} P_{app_2}(l_i = 1, l_j = 1|T) &= P(l_i = 1)P(l_j = 1), \\ P_{app_2}(l_i = 1, l_j = 0|T) &= P(l_i = 1)(1 - P(l_j = 1)), \\ P_{app_2}(l_i = 0, l_j = 1|T) &= (1 - P(l_i = 1))P(l_j = 1), \\ P_{app_2}(l_i = 0, l_j = 0|T) &= (1 - P(l_i = 1))(1 - P(l_j = 1)), \end{aligned} \tag{4.11}$$

where P is the probability of two tracks contain the same person based on the discriminative appearance model App, it is defined as $P(l_i = 1) = 0.5 \times (App_{i1}(T_i^2) + App_{i2}(T_i^1))$.

The discriminative appearance model for each track is online learned using AdaBoost. First, we capture the appearance information of each target using various features: HSV color histograms [116], Local Binary Pattern (LBP) [81], Histogram of Gradient

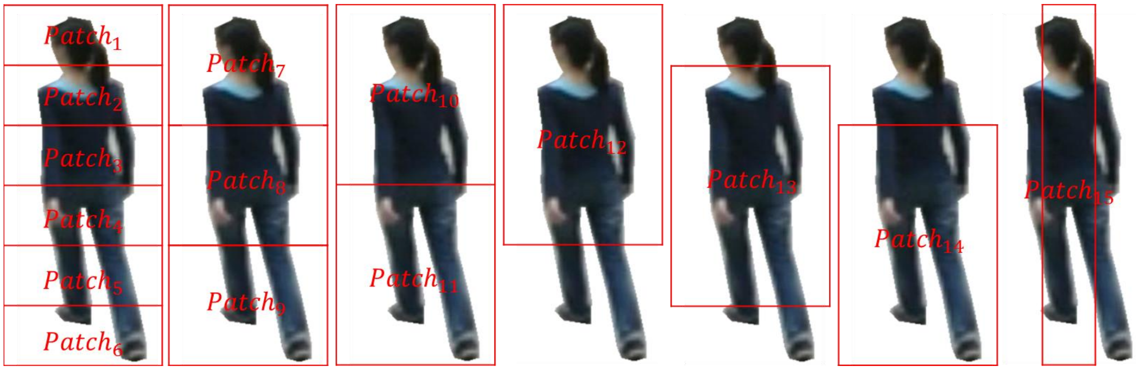


Figure 4.4: Local patches with various scales are defined at different locations of a detection. Patches 1 to 6 have the same size and are served as basic patches, patches 7 to 14 are different combinations of basic patches, and patch 15 captures the middle third region of a detection.

(HOG) [82], and Color Names [114]. Each feature descriptor is computed at different local patches defined on a detection, as shown in Fig. 4.4. We resize each detection to 63 × 27, and extract the containing target using background subtraction. Local patches are defined at different locations with various scales to increase the descriptive ability, and features of the same type in one subtrack are averaged to construct a concise representation for the contained target. In general, one track may contain several subtracks, and there are in total $15 \times 4 = 60$ features for each subtrack.

Given two subtracks t_a and t_b, comparing each of the 60 appearance feature descriptors produces one appearance similarity. A concatenation of the 60 similarities scores forms a feature vector $f(t_a, t_b)$. In our experiments, different methods are used to measure the similarity between different types of features. Bhattacharyya coefficient [33] is used for color histograms and HOG features, χ^2 distance is used for LBP features, and cosine similarity is used for Color Names.

AdaBoost adaptively learns a strong classifier using a number of weak classifiers that minimizes the overall classification error. The generated strong classifier is a linear combination of weak classifiers, and the weight for each selected weak classifier indicates its importance. In our target-specific appearance model, the similarity computed from each feature is used in a weak classifier, and the learned appearance model is formulated as:

$$H(f(t_a, t_b)) = \sum_{t=1}^{T} \alpha_t h_t(f(t_a, t_b)) \tag{4.12}$$

where T is the number of total iterations, α_t is the weighting parameter assigned during the learning process, and $h_t(f(t_a, t_b))$ is a weak classifier based on one of the features extracted from subtracks t_a and t_b.

In order to online learn the discriminative appearance model for each target, we collect training samples during track association. Given a track T_x, a pair of subtracks can form a positive training sample if they are two different subtracks in T_x. A negative sample can be generated by two subtracks if one of them is from T_x, and the other is from another track that has time overlap with T_x. Therefore, a positive sample consists of feature similarities of the same target, while in a negative sample the feature similarities are calculated from two different targets.

Once the discriminative appearance model is learned for a target, we can compute the appearance similarity between this target and other targets using the following equation:

$$App_{i1}(T_i^2) = \sum_{r=1}^{R} H_{i1}(f(t_{i1}^r, t_{i2}^r))) \tag{4.13}$$

where App_{i1} is the target-specific appearance model learned for track T_i^1, and it is used to compute the similarity between T_i^1 and T_i^2. We randomly select R subtrack pairs from both tracks, and use the average their similarity for the similarity of the track pair.

4.2.5 Energy Minimization Algorithm

We formulated the across camera multi-target tracking task as a energy minimization problem using CRF model, as shown in Eq. 4.2. Since the proposed CRF model does not follow the submodularity principle (see APPENDIX), we cannot obtain exact inference using global graph cut optimization techniques [70]. Moreover, traditional approximation approaches for CRF, such as Loopy Belief Prorogation (LBP) and Alpha Expansion, cannot be directly applied for our problem, as solutions produced by these methods may not satisfy the constraint for a valid label set, see Eq. 4.3. Therefore, we developed an iterative approximation algorithm to find a good labeling solution.

More precisely, we first obtain an initial labeling of all vertices using only Hungarian algorithm with unary costs, similar to [26]. As Hungarian algorithm allows only one assignment for each participant, this ensures the initial label set to be a valid one. Then vertices assigned with label 1, i.e., the selected track associations, are sorted in ascending order according to their unary costs. Next, for each label 1 vertice, we find all edges that are connected to current vertice. For each of these edges, all other label configurations are considered, and the one with the minimal graph energy cost is selected. Note that, for a conflicting edge, there are only three labeling possibilities: $(1, 0)$, $(0, 1)$, and $(0, 0)$. If the chosen label configuration generates a energy cost smaller than the current one, we update the label set with the change. In order to maintain the constraints for a valid label set, each time when the label of a vertice changes from 0 to 1, we check if the constraint in Eq. 4.3 is violated. When violations exist, the new update is preferred.

A summary of the energy minimization algorithm is provided in Algorithm 3.

Algorithm 3 Algorithm for finding labels with low energy cost.

```
Input: Tracklet set $T = \{T_1, .., T_n\}$; CRF graph $G = \{V, E\}$
Output: A label set $L$
Use Hungarian algorithm to find an initial label set $L$ with the lowest unary energy
  cost, and evaluate current graph energy cost $\Psi$ in Eq. 4.2.
Sort label 1 vertices according to their unary costs as $\{v_1, ..., v_m\}$
for $i = 1, ..., m$ do
    Find a set $E_i$ including all edges connecting to $v_i$
    Set updated graph energy cost $\Psi^{'} = +\infty$
    for all $e = (v_i, v_x) \in E_i$ do
        Change labels of $(v_i, v_x)$ to a untested possibility,
        maintain constraints for a valid label set,
        evaluate the new graph energy cost $\Psi_{new}$
        if $\Psi_{new} < \Psi^{'}$ then
            $\Psi^{'} = \Psi_{new}$
    if $\Psi^{'} < \Psi$ then
        $\Psi = \Psi^{'}$
        Update $L$ with the change
```

Our proposed energy minimization algorithm finds the label set in a greedy fashion, thus may lead to a local optimal solution. However, a better solution, i.e., a label set with lower energy cost, is achieved after each iteration. Therefore, it ensures us to obtain a better tracking results than using unary costs only.

4.3 Experiments

To validate the effectiveness of the proposed tracking approach, it is compared with several baseline methods as well as the state-of-the-art. We carried out experiments on four different sets of data sequences that are publicly available.

4.3.1 Datasets

Although multi-target tracking in surveillance cameras has been studied for several years, there are fewer publicly available datasets designed for real-world multi-camera tracking as compared to single camera tracking. In this work, we use the NLPR_MCT dataset [3] to evaluate the performance of our proposed method. The NLPR_MCT dataset has both outdoor and indoor scenarios. In addition, there exist obvious illumination variation across cameras, which makes it a very challenging dataset for multi-target tracking.

There are in total four different sub-datasets contained in the NLPR_MCT dataset, each corresponds to a non-overlapping multi-camera networks. Dataset1 and Dataset2 have the same camera setting, including three cameras (two outdoor and one indoor), as shown in Fig. 4.5. Dataset3 contains four videos that are capture by four indoor cameras, the topology of these cameras are presented in Fig. 4.6. The corresponding camera network of Dataset4 consists of five outdoor non-overlapping cameras, the topology of cameras is shown in Fig. 4.7. More specifics for each sub-dataset are listed in Table 4.1.

It is obvious that the quality of input tracks, i.e., within camera tracking results, will greatly affect the performance of multi-target tracking across cameras. In order to have a fair comparison on the cross camera tracking ability, we use the the same input tracks for all the tested methods in our experiments. The input tracks are the single camera tracking ground truth provided in the NLPR_MCT dataset.

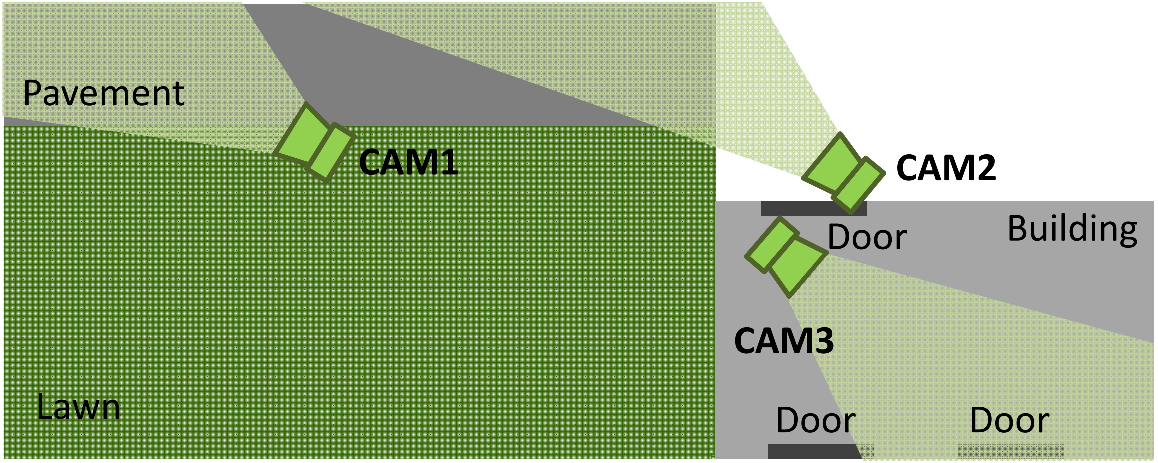


Figure 4.5: Camera topology for Dataset1 and Dataset2. Cam1 and Cam2 are outdoor cameras, and Cam3 is a indoor camera.

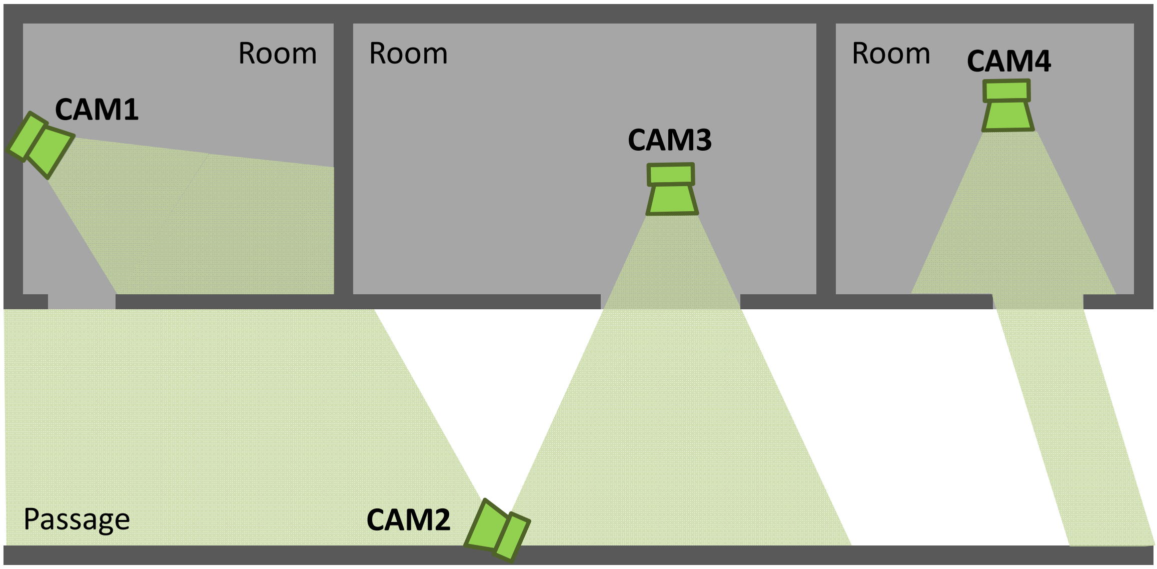


Figure 4.6: Camera topology for Dataset3. Cam1 to Cam4 are all indoor cameras.

Table 4.1: Specifics for each sub-dataset in the NLPR_MCT dataset.

	Dataset1	**Dataset2**	**Dataset3**	**Dataset4**
# of Cameras	3	3	4	5
Resolution	320 × 240	320 × 240	320 × 240	320 × 240
Duration	20min	20min	3.5min	24min
# of Targets	235	255	14	49
Frame Rate	20fps	20fps	25fps	25fps

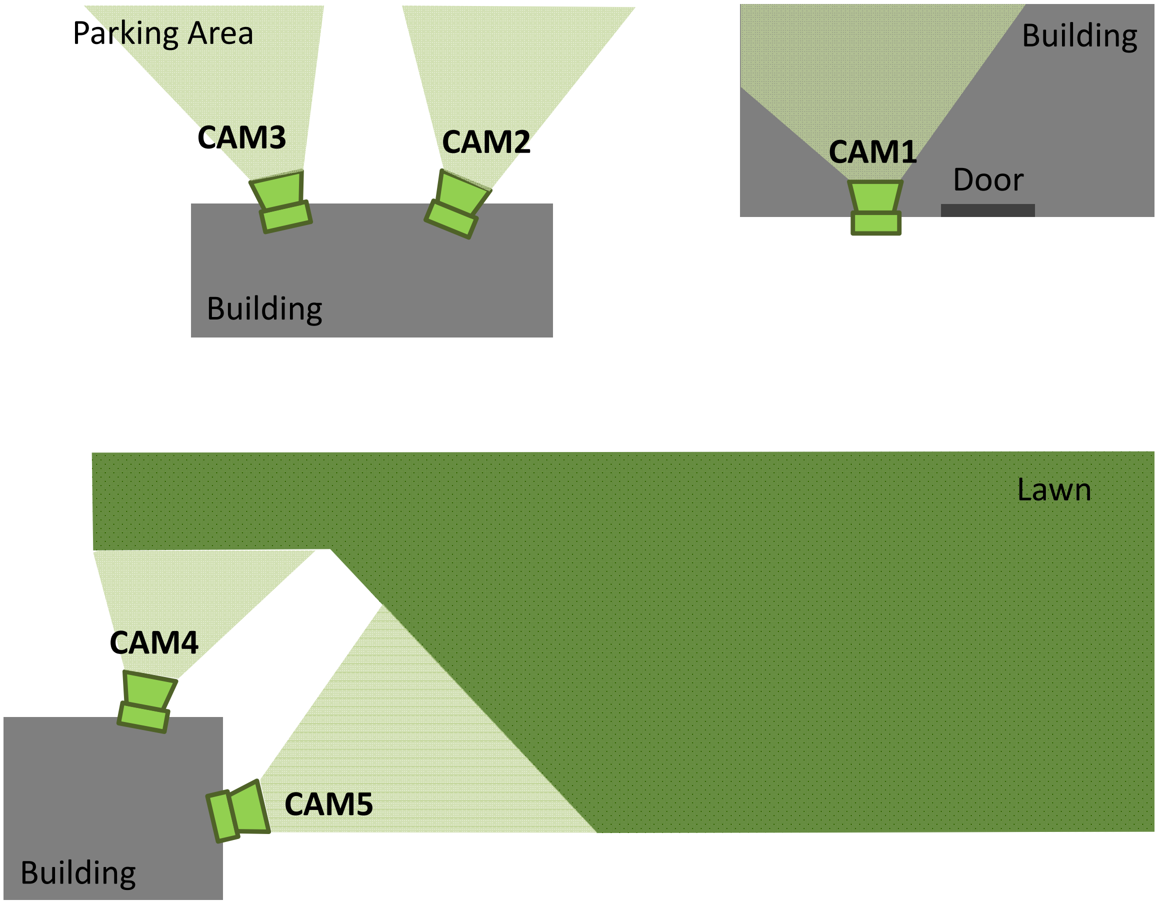


Figure 4.7: Camera topology for Dataset4. Cam1 to Cam5 are all outdoor cameras.

Table 4.2: The number of cross camera true positive in each sub-dataset.

	Dataset1	Dataset2	Dataset3	Dataset4
True Positive	334	408	152	256

4.3.2 Evaluation Metrics

As has been noticed in multi-target tracking in a single camera that it is very difficult to have a direct quantitative comparison of different tracking approaches due to the lack of a standardized benchmark [87]. The same issue persists in multi-target tracking across cameras. Inspired by the widely used CLEAR MOT metrics [17] for single camera multi-object tracking, the NLPR_MCT dataset provides a evaluation metric, Multi-Camera Tracking Accuracy (MCTA), which is a single number metric that combines detection accuracy, single camera tracking accuracy and cross camera tracking accuracy. The definition of MCTA is given in Eq. 4.14 .

$$\begin{aligned} &MCTA \\ &= Detection \times Tracking^{SCT} \times Tracking^{ICT} \\ &= (\frac{2 \times precision \times recall}{precision + recall})(1 - \frac{\sum_t mme_t^s}{\sum_t tp_t^s})(1 - \frac{\sum_t mme_t^c}{\sum_t tp_t^c}), \end{aligned} \tag{4.14}$$

where *precision* and *recall* reflect the performance of the object detector, mme_t^s is the number of mismatches (i.e., ID-switches) for time t in a single camera, and mme_t^c is the number of mismatches for time t across different cameras, tp_t^s and tp_t^c are the number of true positive for time t within camera and cross cameras, respectively. Note that, according to the defined criteria, when a new target first enters the scene, it produces a new cross camera true positive instead of a within camera true positive.

The MCTA metric ranges from 0 to 1, a higher value indicates a better tracking performance. In order to focus on the ability of cross camera multi-target tracking, single camera tracking ground truth is used as input tracks. Therefore, the first two terms in Eq. 4.14, i.e., *Detection* and $Tracking^{SCT}$, will be 1. The cross camera tracking performance is only affected by mme^c, the number of mismatches across cameras.

Table 4.3: Comparison of cross camera tracking results on the NLPR_MCT dataset.

Method	Dataset1		Dataset2		Dataset3		Dataset4	
	mme^c	MCTA	mme^c	MCTA	mme^c	MCTA	mme^c	MCTA
Baseline1	156	0.5329	197	0.5172	89	0.4145	150	0.4141
Baseline2	91	0.7275	102	0.7500	62	0.5921	118	0.5391
Ours	54	0.8383	81	0.8015	51	0.6645	70	0.7266
USC-Vision [23]	27	0.9152	34	0.9132	70	0.5163	72	0.7052
Hfutdspmct	86	0.7425	141	0.6544	40	0.7368	155	0.3945
CRIPAC-MCT	113	0.6617	167	0.5907	44	0.7105	110	0.5703

4.3.3 Experimental Results

In this evaluation, our goal is to link tracks in different camera views that contain the same target under certain spatial temporal constraints. The number of cross camera true positive in each sub-dataset is shown in Table 4.2. We introduce three baseline models for comparison:

- Baseline1: use only Hungarian algorithm with global appearance model, no grouping information.
- Baseline2: our proposed CRF model without the local appearance consistency in Eq. 4.9.

A quantitative comparison of our proposed model and the baseline models are presented in Table 4.3. It is observed that our proposed model significantly improves the tracking performance on all sub-datasets compared to Baseline1. For Dataset1 and Dataset2, our model increases MCTA by almost 0.3. For Dataset3, the improvement with respect to MCTA is 0.25. The largest improvement is achieved in Dataset4, where the MCTA improves by 0.46 when our proposed model is used. Therefore, it is validated that by integrating social grouping information we can achieve better tracking performance, as high-level context provides us other useful information that are not included in low-level features. A visual comparison of our model and Baseline1 on Dataset1 is shown in Fig. 4.8. In Baseline2, only group consistency is taken into account for edge cost calculation in the CRF graph. The tracking performances of Baseline2 on all sub-datasets are better than that of Baseline1, which further validate the effectiveness of grouping information for track association. Comparison between our proposed model and Baseline2 indicates that local appearance consistency plays an important role in eliminating incorrect track association, as it requires the linked track pair should not only have high appearance similarity in the global appearance model but also be visually similar according to the local appearance model. A visual comparison of our model and Baseline2 on Dataset2 in presented in Fig. 4.9. In Fig. 4.10, we provide some tracking results of our model on Dataset3. More tracking results on Dataset4 using the proposed method are shown in Fig. 4.1 and Fig. 4.11.

In addition, the proposed CRF model is compared with other methods for tracking in multiple non-overlapping cameras. These methods are reported in the Multi-Camera Object Tracking (MCT) Challenge [2] in ECCV 2014 visual surveillance and re-identification workshop. We select the top 3 methods for comparison, their corresponding tracking performances on each sub-dataset are shown in Table 4.3, with USC-Vision [23] being rank 1, Hfutdspmct being rank 2, and CRIPAC-MCT being rank 3. According to the results shown in Table 4.3, our proposed model takes advantage of the adequate grouping information contained in the videos in Dataset4 and achieves the highest MCTA on this sub-dataset. For Dataset1 and Dataset2, where there are less grouping information, our proposed model has the second highest MCTA compared to the state-of-the-art. Due to the narrow view point for cameras in Dataset3 (see Fig. 4.10), each target enters and exits the scene in a short time. It is difficult to detect elementary groups, as two targets can form an elementary group if they co-exist for at least 2 seconds in our experiments. In Dataset3, the median length of all tracks is 3.9 seconds, while in other sub-datasets the median length is at at least 5.5 seconds. Therefore, the proposed method has the lowest MCTA on this dataset. However, the tracking performance is still comparable to other methods.

4.4 Conclusions

In this chapter we present a novel CRF model based framework for multi-target tracking across cameras. The proposed model is able to systematically integrate social grouping behavior as high-level context information for reducing ambiguities in track asso-

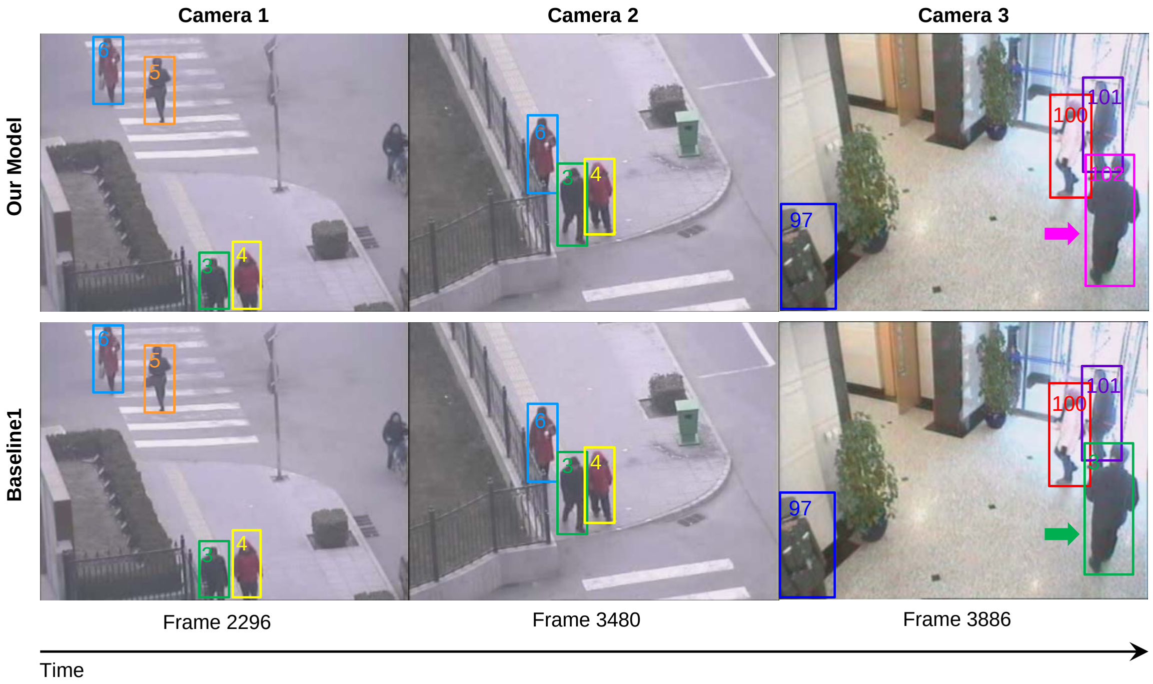


Figure 4.8: A visual comparison of our model (the first row) and Baseline1 (the second row) on Dataset1. It is observed that Baseline1 mistakenly identifies a new target in Camera 3 (the one pointed by arrow) as target 3, while our model avoid this error by maintaining the group consistency between target 3 and 4. Bounding box with the same color indicates the same target. Best viewed in color.

ciation. Experiments on four challenging real-world data sequences validate the effectiveness of our model. When there is rich grouping information in the scene, the tracking performance is significantly improved with the learned high-level context information. Possible future work would be learning more discriminative representations for the targets and evaluate our method on more datasets.

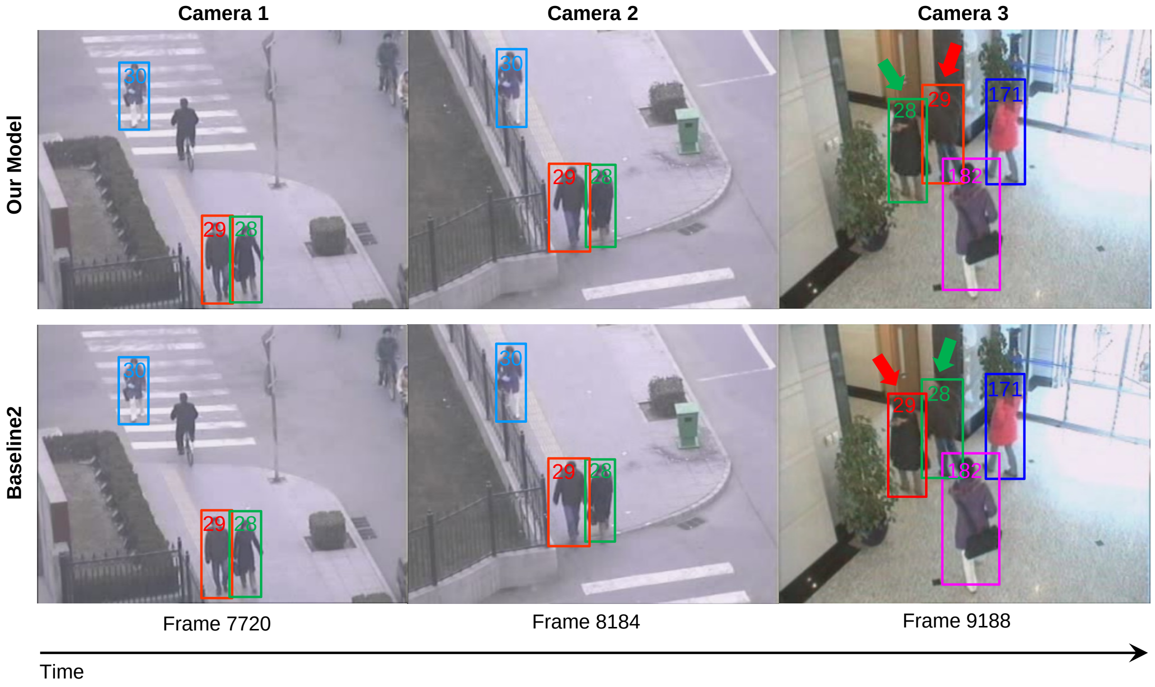


Figure 4.9: A visual comparison of our model (the first row) and Baseline2 (the second row) on Dataset2. In the result of our model, target 28 and 29 are correctly tracked in all cameras. But their IDs are switched in Camera 3 in the result of Baseline2, due to the lack of local appearance consistency. Bounding box with the same color indicates the same target. Best viewed in color.

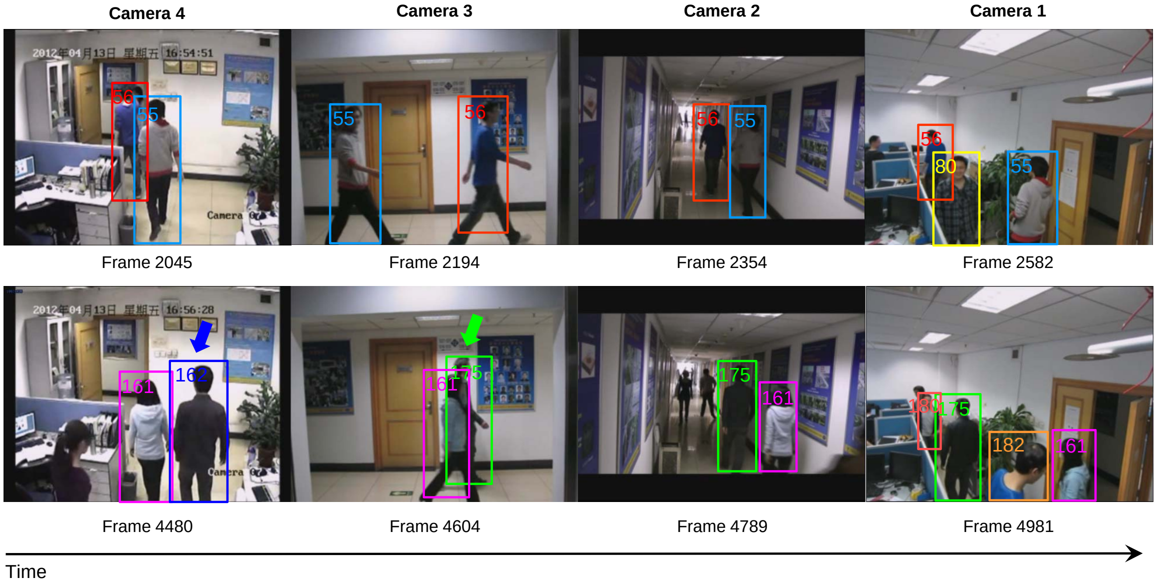


Figure 4.10: Sample tracking results of our proposed method on Dataset3. In the first row, by taking advantage of the grouping information, target 55 and 56 are successfully tracked in all cameras, even under significant within and across camera illumination changes. In the second row, target 162 in Camera 4 is not correctly link to the same target (target 175) in Camera 3. This target is severely occluded by target 161 in Camera 3, even with group information we are unable to link them, as such association does not maintain appearance consistency. Bounding box with the same color indicates the same target. Best viewed in color.

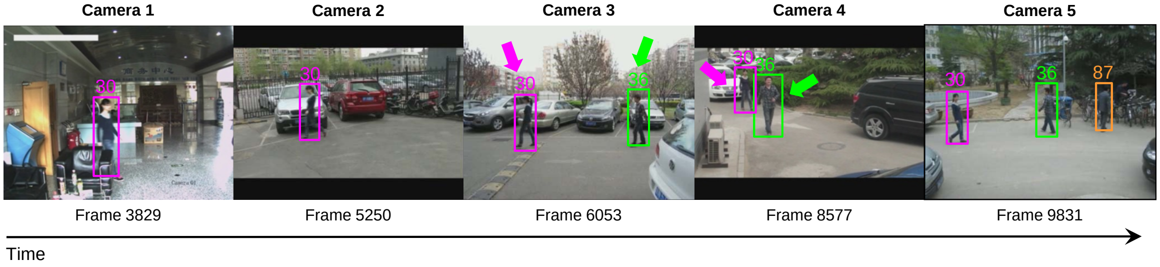


Figure 4.11: Sample tracking results of our proposed method on Dataset4. Target 30 is correctly tracked in all cameras. However, target 36 in Camera 3 is mistakenly linked to another target in Camera 4 (pointed by green arrow). Since both of them form an elementary group with target 30, and are visually very similar. Bounding box with the same color indicates the same target. Best viewed in color.

Chapter 5

Conclusions

In order to facilitate multi-target tracking in surveillance cameras in real-world scenarios, we proposed several tracking methods in this dissertation that cover both within camera and across cameras tracking tasks.

In Chapter 2, an online learned elementary grouping model with non-linear motion context is introduced for improving multi-target tracking performance in a single camera. In this method high-level contextual information, social grouping behavior, is integrated via elementary groups into a basic association-based tracking framework to mitigate visual ambiguities that are too challenging for low-level information. An elementary group is a group that contains only two targets. Therefore, a group of any size can be represented by a set of elementary groups. This property gives the proposed method flexibility to handle group merge and split. During tracklet association, two targets are not only matched according to their appearance and motion affinities, but also the probability of maintaining elementary group consistency. In addition, we use a non-linear motion map to explain non-linear motion pattern between elementary groups. Experimental comparisons between the proposed methods and other alternatives are carried out on four real-world datasets. Both quantitative and visual results validate the effectiveness and efficiency of the proposed method, and further prove the importance of using contextual information in within camera multi-target tracking.

In Chapter 3, we looked into the problem of multi-target tracking in non-overlapping cameras, and proposed a reference set based appearance model for more robust appearance match. Since observations of the same target in a camera network are often separated by time and space, the appearance of the same target might be significantly different in two neighboring cameras due to changes in illumination conditions, poses, and camera imaging characteristics. Instead of comparing tracks from two different cameras directly, a reference set is constructed for each pair of cameras and appearance comparisons are carried out indirectly via the corresponding reference set. A reference set contains subjects that appear in both cameras, and each subject has several appearance instances that are generated by track division. Given two tracks that are from two different cameras, each track is first compared to the appearance instances of all subjects in the reference set that are from the same camera. In other words, each reference subject is an indirect feature that describes some characteristics of the target's appearance. Having two tracks compared with the same set of reference subjects enables us to generate two reference set based descriptors with the same length. These descriptors are later used to compute the appearance similarity of the two tracks. We performed in-depth experiments and analysis on two challenging real-world datasets. Experimental results on both datasets demonstrate the superiority of the proposed reference set based appearance model over baseline methods and state-of-the-art Brightness Transfer Functions based method.

In Chapter 4, we explored social grouping information for inter-camera multi-target tracking. The multi-target tracking problem is formulated using an online learned Conditional Random Field (CRF) model that minimizes a global energy cost. Each node in the CRF graph represents a pair of linkable tracks, and two nodes are connected by an edge if the corresponding tracks can form at least one elementary group. The proposed CRF model prefers track associations that not only have high affinities in appearance and motion but also maintain within camera grouping consistencies. Extensive experiments on three different camera networks showed that the proposed tracking method is effective on associating difficult track pairs with additional high-level contextual information. When there are rich grouping information in the scene, the tracking performance can be significantly improved.

Bibliography

[1] Caviar dataset.

[2] Multi-Camera Object Tracking (MCT) Challenge.

[3] NLPR_MCT Dataset.

[4] Herve Abdi. *Kendall Rank Correlation*, pages 509–511. SAGE Publications, Inc., 2007.

[5] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. Socially-aware large-scale crowd forecasting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[6] Saad Ali and Mubarak Shah. Floor fields for tracking in high density crowd scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–14, 2008.

[7] L. An, M. Kafai, S. Yang, and B. Bhanu. Person re-identification with reference descriptor. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–1, 2015.

[8] Le An, M. Kafai, and B. Bhanu. Dynamic bayesian network for unconstrained face recognition in surveillance camera networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(2):155–164, June 2013.

[9] Le An, M. Kafai, Songfan Yang, and B. Bhanu. Reference-based person re-identification. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 244–249, 2013.

[10] Le An, Songfan Yang, and B. Bhanu. Person re-identification by robust canonical correlation analysis. *IEEE Signal Processing Letters*, 22(8):1103–1107, Aug 2015.

[11] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1272, June 2011.

[12] L. Bazzani, M. Zanotto, M. Cristani, and V. Murino. Joint individual-group modeling for tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[13] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 137–144, Nov 2011.

[14] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3457–3464, June 2011.

[15] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR*, 2006.

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

[17] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Journal on Image and Video Process*, 2008:1–10, January 2008.

[18] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Proceedings of Workshop on Motion and Video Computing*, pages 169–174, 2002.

[19] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1515–1522, Sept 2009.

[20] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[21] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1273–1280, June 2011.

[22] A.A. Butt and R.T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1846–1853, June 2013.

[23] Yinghao Cai and G. Medioni. Exploring context information for inter-camera multiple target tracking. In *2014 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 761–768, March 2014.

[24] J. Candamo, M. Shreve, D.B. Goldgof, D.B. Sapper, and R. Kasturi. Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):206–224, 2010.

[25] T.-H. Chang and Shaogang Gong. Tracking multiple people with a multi-camera system. In *Proceedings of IEEE Workshop on Multi-Object Tracking*, pages 19–26, 2001.

[26] Xiaojing Chen, Le An, and B. Bhanu. Multitarget tracking in nonoverlapping cameras using a reference set. *IEEE Sensors Journal*, 15(5):2692–2704, May 2015.

[27] Xiaojing Chen, Le An, and Bir Bhanu. Reference set based appearance model for tracking across non-overlapping cameras. In *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2013.

[28] Xiaojing Chen, Zhen Qin, Le An, and Bir Bhanu. An online learned elementary grouping model for multi-target tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1242–1249, June 2014.

[29] Xiaotang Chen, Kaiqi Huang, and Tieniu Tan. Object tracking across non-overlapping views by learning inter-camera transfer models. *Pattern Recognition*, 47(3):1126 – 1137, 2014.

[30] Chun-Te Chu, Jenq-Neng Hwang, Kung-Ming Lan, and Shen-Zheng Wang. Tracking across multiple cameras with overlapping views based on brightness and tangent transfer functions. In *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, 2011.

[31] Chun-Te Chu, Jenq-Neng Hwang, Jen-Yu Yu, and Kual-Zheng Lee. Tracking across nonoverlapping cameras based on the unsupervised learning of camera link models. In *Proceedings of IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, 2012.

[32] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[33] T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, USA, 1991.

[34] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893 vol. 1, June 2005.

[35] A. Dantcheva, C. Velardo, A. D'angelo, and J-L. Dugelay. Bag of soft biometrics for person identification : New trends and challenges. *Mutimedia Tools and Applications*, 2010.

[36] M. Demirkus, K. Garg, and S. Guler. Automated person categorization for video surveillance using soft biometrics. In *Proceedings of Biometric Technology for Human Identification VII*, 2010.

[37] Giovanni Denina, Bir Bhanu, HoangThanh Nguyen, Chong Ding, Ahmed Kamal, Chinya Ravishankar, Amit Roy-Chowdhury, Allen Ivers, and Brenda Varda. Videoweb dataset for multi-camera activities and non-verbal communication. In *Distributed Video Sensor Networks*, pages 335–347. Springer London, 2011.

[38] S. L. Dockstader and A. M. Tekalp. Multiple camera fusion for multi-object tracking. In *Proceedings of IEEE Workshop on Multi-Object Tracking*, pages 95–102, 2001.

[39] T. D'Orazio and G. Cicirelli. People re-identification and tracking from multiple cameras: A review. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 1601–1604, 2012.

[40] T. D'Orazio, P.L. Mazzeo, and P. Spagnolo. Color brightness transfer function evaluation for non overlapping multi camera tracking. In *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, 2009.

[41] M.-P. Dubuisson and A.K. Jain. A modified hausdorff distance for object matching. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 1, pages 566–568 vol.1, Oct 1994.

[42] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sept 2010.

[43] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sept 2010.

[44] L. Feng and B. Bhanu. Understanding dynamic social grouping behaviors of pedestrians. *IEEE Journal of Selected Topics in Signal Processing*, 9(2):317–329, March 2015.

[45] J. Ferryman and A. Shahrokni. PETS2009: Dataset and challenge. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–6, Dec 2009.

[46] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 1997.

[47] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, April 2000.

[48] Weina Ge, R.T. Collins, and R.B. Ruback. Vision-based analysis of small groups in pedestrian crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1003–1016, May 2012.

[49] N. Ghosh and B. Bhanu. Evolving bayesian graph for three-dimensional vehicle model building from video. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):563–578, April 2014.

[50] Andrew Gilbert and Richard Bowden. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 125–136, 2006.

[51] A. Gyaourova and A. Ross. Index codes for multibiometric pattern retrieval. *IEEE Transactions on Information Forensics and Security*, 7(2):518–529, April 2012.

[52] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282–4286, May 1995.

[53] J.F. Henriques, R. Caseiro, and J. Batista. Globally optimal solution to multi-object tracking with merged measurements. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2470–2477, Nov 2011.

[54] M. Hofmann, D. Wolf, and G. Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3650–3657, 2013.

[55] Semislav Dimitrov Hristov. *Multi-target tracking in unevenly illuminated scenes.* PhD thesis, University of Trento, 2015.

[56] Weiming Hu, Min Hu, Xue Zhou, Tieniu Tan, Jianguang Lou, and S. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):663–671, April 2006.

[57] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 788–801, 2008.

[58] C. Hue, J.-P. Le Cadre, and P. Perez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *Signal Processing, IEEE Trans.*, 2002.

[59] A. K. Jain and U. Park. Facial marks: Soft biometric for face recognition. In *ICIP*, 2009.

[60] R. Jain and K. Wakimoto. Multiple perspective interactive video. In *Proceedings of the International Conference on Multimedia Computing and Systems*, pages 202–211, 1995.

[61] Anil K. Jaina, Sarat C. Dassb, and Karthik Nandakumara. Can soft biometric traits assist user recognition. In *Proceedings of SPIE*, 2004.

[62] Omar Javed, Zeeshan Rasheed, Khurram Shafique, and Mubarak Shah. Tracking across multiple cameras with disjoint views. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2003.

[63] Omar Javed, Khurram Shafique, Zeeshan Rasheed, and Mubarak Shah. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109:146 – 162, 2008.

[64] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual tracking via adaptive structural local sparse appearance model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1822–1829, 2012.

[65] Hao Jiang, S. Fels, and J.J. Little. A linear programming approach for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.

[66] Z. Jin and B. Bhanu. Pedestrian tracking with crowd simulation models in a multi-camera system. *Computer Vision and Image Understanding*, 2014.

[67] Zhixing Jin and B. Bhanu. Integrating crowd simulation for pedestrian tracking in a multi-camera system. In *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, 2012.

[68] M. Kafai, L. An, and B. Bhanu. Reference face graph for face recognition. *IEEE Transactions on Information Forensics and Security*, 9(12):2132–2143, Dec 2014.

[69] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[70] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, Feb 2004.

[71] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, 2011.

[72] Cheng-Hao Kuo, Chang Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 685–692, 2010.

[73] Cheng-Hao Kuo, Chang Huang, and Ram Nevatia. Inter-camera association of multi-target tracks by on-line learned appearance affinity models. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, pages 383–396, 2010.

[74] I. Kviatkovsky, A. Adam, and E. Rivlin. Color invariants for person reidentification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1622–1634, July 2013.

[75] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *IEEE International Conference on Computer Vision*, pages 1–8, Oct 2007.

[76] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. Robust visual tracking based on simplified biologically inspired features. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 4113–4116, 2009.

[77] Yuan Li, Chang Huang, and R. Nevatia. Learning to associate: HybridBoosted multi-target tracker for crowded scene. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2953–2960, June 2009.

[78] Chengjun Liu. Discriminant analysis and similarity measure. *Pattern Recognition*, 47(1):359 – 367, 2014.

[79] Chunxiao Liu, Shaogang Gong, and Chen Change Loy. On-the-fly feature importance mining for person re-identification. *Pattern Recognition*, 47(4):1602 – 1615, 2014.

[80] Jingen Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3337–3344, 2011.

[81] Xiuwen Liu and DeLiang Wang. Texture classification using spectral histograms. *IEEE Transactions on Image Processing*, 12(6):661–670, June 2003.

[82] DavidG. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[83] Wenhan Luo, Xiaowei Zhao, and Tae-Kyun Kim. Multiple object tracking: A review. *Computing Research Repository*, abs/1409.7618, 2014.

[84] F. Madrigal and J.-B. Hayet. Multiple view, multiple target tracking with principal axis-based data association. In *Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 185–190, 2011.

[85] L. Marcenaro, P. Morerio, and C.S. Regazzoni. Performance evaluation of multi-camera visual tracking. In *Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 464–469, 2012.

[86] Riccardo Mazzon and Andrea Cavallaro. Multi-camera tracking using a multi-goal social force model. *Neurocomputing*, 100(0):41 – 50, 2013. Special issue: Behaviours in video.

[87] A. Milan, K. Schindler, and S. Roth. Challenges of ground truth evaluation of multi-target tracking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 735–742, June 2013.

[88] Mehdi Moussaid, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5(4):e10047, 04 2010.

[89] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1), 1957.

[90] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J. Little, and David G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

[91] Unsang Park and A.K. Jain. Face matching and retrieval using soft biometrics. *IEEE Transactions on Information Forensics and Security*, 5(3):406–415, 2010.

[92] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 261–268, Sept 2009.

[93] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–465, 2010.

[94] A.G.A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and Wensheng Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 666–673, June 2006.

[95] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1201–1208, June 2011.

[96] B. Prosser, S. Gong, and T. Xiang. Multi-camera matching using bi-directional cumulative brightness transfer functions. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 64.1–64.10, September 2008.

[97] Zhen Qin and Christian Shelton. Improving multi-target tracking via social grouping. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1972–1978, 2012.

[98] Zhen Qin, Christian Shelton, and Lunshao Chai. Social grouping for target handover in multi-view video. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2013.

[99] V. Rabaud and S. Belongie. Counting crowded moving objects. In *CVPR*, 2006.

[100] Daniel A. Reid and M.S. Nixon. Using comparative human descriptions for soft biometrics. In *Proceedings of International Joint Conference on Biometrics (IJCB)*, pages 1–6, 2011.

[101] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, Sep 2001.

[102] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1472–1485, Aug 2009.

[103] F. Schroff, T. Treibitz, D. Kriegman, and S. Belongie. Pose, illumination and expression invariant pairwise face-similarity measure via doppelgänger list comparison. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2494–2501, 2011.

[104] C. Seiffert, T.M. Khoshgoftaar, J. Van Hulse, and A Napolitano. Rusboost: Improving classification performance when training data is skewed. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[105] Guang Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1815–1821, June 2012.

[106] Nils T Siebel and Stephen J Maybank. The advisor visual surveillance system. In *Proceedings of the European Conference on Computer Vision Workshop (ECCVW)*, 2004.

[107] J. Sochman and D.C. Hogg. Who knows who - inverting the social force model for finding groups. In *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 830–837, Nov 2011.

[108] B. Song, T. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[109] Bi Song, A.T. Kamal, C. Soto, Chong Ding, J.A. Farrell, and A.K. Roy-Chowdhury. Tracking and activity recognition through consensus in distributed camera networks. *IEEE Transactions on Image Processing*, 19(10):2564–2579, 2010.

[110] S. Srivastava, Ka Ki Ng, and E.J. Delp. Color correction for object tracking across multiple cameras. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1821–1824, May 2011.

[111] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2011.

[112] M. Taj and A. Cavallaro. Multi-camera track-before-detect. In *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, 2009.

[113] D. Tao, L. Jin, Y. Wang, and X. Li. Person reidentification by minimum classification error-based KISS metric learning. *IEEE Transactions on Cybernetics*, 45(2):242–252, Feb 2015.

[114] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, July 2009.

[115] Manik Varma and Bodla Rakesh Babu. More generality in efficient multiple kernel learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1065–1072, 2009.

[116] Xiang-Yang Wang, Jun-Feng Wu, and Hong-Ying Yang. Robust image retrieval based on color histogram of local feature regions. *Multimedia Tools and Applications*, 49(2):323–345, 2010.

[117] Xiaogang Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3 – 19, 2013.

[118] Zheng Wu, T.H. Kunz, and M. Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1185–1192, June 2011.

[119] X. Shao X. Song, H. Zhao, J. Cui, R. Shibasaki, and H. Zha. An online approach: Learning-semantic-scene-by-tracking and tracking-by-learning-semantic-scene. In *CVPR*, 2010.

[120] J. Xing, H. Ai, L. Liu, and S. Lao. Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling. *IEEE TIP*, 2011.

[121] Junliang Xing, Haizhou Ai, and Shihong Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1200–1207, June 2009.

[122] Fei Xiong, Mengran Gou, Octavia Camps, and Mario Sznaier. Person re-identification using kernel-based metric learning methods. In *European Conference on Computer Vision*, pages 1–16. Springer, 2014.

[123] K. Yamaguchi, A.C. Berg, L.E. Ortiz, and T.L. Berg. Who are you with and where are you going? In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1345–1352, June 2011.

[124] Xu Yan, Anil Cheriyadat, and Shishir Shah. Hierarchical group structures in multi-person tracking. In *Proceedings of IEEE Conference on Pattern Recognition (ICPR)*, pages 1242–1249, June 2014.

[125] B. Yang and R. Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *ECCV*, 2012.

[126] Bo Yang, Chang Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a CRF model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1233–1240, June 2011.

[127] Bo Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1918–1925, June 2012.

[128] Bo Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[129] Bo Yang and Ramakant Nevatia. Multi-target tracking by online learning a crf model of appearance and motion patterns. *International Journal of Computer Vision*, 107:203–217, 2014.

[130] Ming Yang, Fengjun Lv, Wei Xu, and Yihong Gong. Detection driven adaptive multi-cue integration for multiple human tracking. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2009.

[131] Yang Yang, Jimei Yang, Junjie Yan, Shengcai Liao, Dong Yi, and StanZ. Li. Salient color names for person re-identification. In *European Conference on Computer Vision (ECCV)*, pages 536–551, 2014.

[132] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), December 2006.

[133] Qi Yin, Xiaoou Tang, and Jian Sun. An associate-predict model for face recognition. In *Proceedings of IEEE Conferenceon Computer Vision and Pattern Recognition (CVPR)*, pages 497–504, 2011.

[134] Qian Yu, Gerard Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. In *CVPR*, 2007.

[135] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, 2012.

[136] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.

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

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

[139] Shu Zhang, Yingying Zhu, and Amit Roy-Chowdhury. Tracking multiple interacting targets in a camera network. *Computer Vision and Image Understanding*, 134:64 – 73, 2015.

[140] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3586–3593, June 2013.

[141] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Learning mid-level filters for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 144–151, June 2014.

[142] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Transfer re-identification: From person to set-based verification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2650–2657, June 2012.

[143] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):653–668, 2013.

[144] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[145] Yingying Zhu, N.M. Nayak, and A.K. Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):91–101, 2013.