UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Inria informatics mathematics

THÈSE DE DOCTORAT

Suivi long terme de personnes pour les systèmes de vidéo monitoring

Long-term people trackers for video monitoring systems

Thi Lan Anh NGUYEN

INRIA Sophia Antipolis, France

Présentée en vue de l'obtention du grade de docteur en Informatiques **d'**Université Côte d'Azur
Dirigée par : Francois Bremond
Soutenue le : 17/07/2018

Devant le jury, composé de :

- Frederic Precioso, Professor, I3S lab – France
- Francois Bremond, Team leader, INRIA Sophia Antipolis – France
- Jean-Marc Odobez, Team leader, IDIAP – Switzerland
- Jordi Gonzalez, Associate Professor, ISE lab, Espanol
- Serge Miguet, Professor, ICOM, Université Lumière Lyon 2, France

Région PACA

Suivi long terme de personnes pour les systèmes de vidéo monitoring

Long-term people trackers for video monitoring systems

Jury:

Président du jury*

Frederic Prescioso, Professor, I3S lab - France

Rapporteurs

Jean-Mard Odobez, Team leader, IDIAP – Swizerland

Jordi Gonzales, Associate Professor, ISE lab, Espagnol

Serge Miguet, Professor, ICOM, Universite Lumiere Lyon 2 – France

Directeur de thèse :

Francois Bremond, Team leader, STARS team, INRIA Sophia Antipolis

Titre : Suivi long terme de personnes pour les systèmes de vidéo monitoring

Résumé
Le suivi d'objets multiples (Multiple Object Tracking (MOT)) est une tâche importante dans le domaine de la vision par ordinateur. Plusieurs facteurs tels que les occlusions, l'éclairage et les densités d'objets restent des problèmes ouverts pour le MOT. Par conséquent, cette thèse propose trois approches MOT qui se distinguent à travers deux propriétés: leur généralité et leur efficacité.
La première approche sélectionne automatiquement les primitives visions les plus fiables pour caractériser chaque tracklet dans une scène vidéo. Aucun processus d'apprentissage n'est nécessaire, ce qui rend cet algorithme générique et déployable pour une grande variété de systèmes de suivi.
La seconde méthode règle les paramètres de suivi en ligne pour chaque tracklet, en fonction de la variation du contexte qui l'entoure. Il n'y a pas de constraintes sur le nombre de paramètres de suivi et sur leur dépendance mutuelle. Cependant, on a besoin de données d'apprentissage suffisamment représentatives pour rendre cet algorithme générique.
La troisième approche tire pleinement avantage des primitives visions (définies manuellement ou apprises), et des métriques définies sur les tracklets, proposées pour la ré-identification et leur adaptation au MOT. L'approche peut fonctionner avec ou sans étape d'apprentissage en fonction de la métrique utilisée.
Les expériences sur trois ensembles de vidéos, MOT2015, MOT2017 et ParkingLot montrent que la troisième approche est la plus efficace. L'algorithme MOT le plus approprié peut être sélectionné, en fonction de l'application choisie et de la disponibilité de l'ensemble des données d'apprentissage.

Mots clés : MOT, suivi de personnes

Title: Long term people trackers for video monitoring systems

Abstract
Multiple Object Tracking (MOT) is an important computer vision task and many MOT issues are still unsolved. Factors such as occlusions, illumination, object densities are big challenges for MOT. Therefore, this thesis proposes three MOT approaches to handle these challenges. The proposed approaches can be distinguished through two properties: their generality and their effectiveness.
The first approach selects automatically the most reliable features to characterize each tracklet in a video scene. No training process is needed which makes this algorithm generic and deployable within a large variety of tracking frameworks. The second method tunes online tracking parameters for each tracklet according to the variation of the tracklet's surrounding context. There is no requirement on the number of tunable tracking parameters as well as their mutual dependence in the learning process. However, there is a need of training data which should be representative enough to make this algorithm generic. The third approach takes full advantage of features (hand-crafted and learned features) and tracklet affinity measurements proposed for the Re-id task and adapting them to MOT. Framework can work with or without training step depending on the tracklet affinity measurement.
The experiments over three datasets, MOT2015, MOT2017 and ParkingLot show that the third approach is the most effective. The first and the third (without training) approaches are the most generic while the third approach (with training) necessitates the most supervision. Therefore, depending on the application as well as the availability of a training dataset, the most appropriate MOT algorithm could be selected.

Keywords : MOT, people tracking

ACKNOWLEDGMENTS

I would like to thank Dr. Jean-Marc ODOBEZ, from IDIAP Research Institute, Switzerland, Prof. Jordi GONZALEZ from ISELab of Barcelona University and Prof. Serge MIGUET from ICOM, Universite Lumiere Lyon 2, France , for accepting to review my PhD manuscript and for their pertinent feedbacks. I also would like to give my thanks to Prof. Precioso FREDERIC - I3S - Nice University, France for accepting to be the president of the committee.

I sincerely thank my thesis supervisors Francois BREMOND for what they have done for me. It is my great chance to work with them. Thanks for teaching me how to communicate with the scientific community, for being very patient to repeat the scientific explanations several times due to my limitations on knowledge and foreign language. His high requirements have helped me to obtain significant progress in my research capacity. He guided me the necessary skills to express and formalize the scientific ideas. Thanks for giving me a lot of new ideas to improve my thesis. I am sorry not to be a good enough student to understand quickly and explore all these ideas in this manuscript. With his availability and kindness, he has taught me the necessary scientific and technical knowledge as well as redaction aspects for my PhD study. He also gave me all necessary supports so that I could complete this thesis. I have also learned from him how to face to the difficult situations and how important the human relationship is. I really appreciate him.

I then would like to acknowledge Jane for helping me to solve a lot of complex administrative and official problems that I never imagine.

Many special thanks are also to all of my colleagues in the STARS team for their kindness as well as their scientific and technical supports during my thesis period, especially Duc-Phu, Etienne,Julien, Farhood, Furqan, Javier, Hung, Carlos, Annie. All of them have given me a very warm and friendly working environment.

Big thanks are to my Vietnamese friends for helping me to overcome my homesickness. I will always keep in mind all good moments we have spent together.

I also appreciate my colleagues from the faculty of Information Technology of ThaiNguyen University of Information and Communication Technology (ThaiNguyen city, Vietnam) who have given me the best conditions so that I could completely focus on my study in France. I sincerely thank Dr. Viet-Binh PHAM, director of the University, for his kindness and supports to my study plan. Thank researchers (Dr Thi-Lan LE, Dr Thi-Thanh-Hai NGUYEN, Dr Hai TRAN) at MICA institute (Hanoi, Vietnam) for instructing me the fundamental knowledge of Computer Vision which support me a lot to start my PhD study.

A big thank to my all family members, especially my mother, Thi-Thuyet HOANG, for their

full encouragements and perfect supports during my studies. It has been more than three years since I lived far from family. It does not count short or quick but still long enough for helping me to recognize how important my family is in my life.

The most special and greatest thanks are for my boyfriend, Ngoc-Huy VU. Thanks for supporting me entirely and perfectly all along my PhD study. Thanks for being always beside me and sharing with me all happy as well as hard moments. This thesis is thanks to him and is for him.

Finally, I would like to thank and to present my excuses to all the persons I have forgotten to mention in this section.

Thi-Lan-Anh NGUYEN
thi-lan-anh.nguyen@sophia.inria.fr
Sophia Antipolis, France

CONTENTS

FIGURES

TABLES

1

INTRODUCTION

A huge amount of data is recorded by video surveillance systems in many different locations such as airports, hospitals, banks, railway stations, stadiums, streets, supermarkets and even at domestic environment (see figure 1.1). These evidences shows a worldwide use of these videos for different applications. The duty of a supervisor of a video surveillance system is to observe these videos and to quickly focus on abnormal activities taking place in the surveillance region (see figure 1.2). However, the simultaneous observation and analysis of these videos is a challenge for the supervisor while ensuring the minimum rate of missing abnormal activities in real time. Moreover, the observation of many screens for a long period of time reduces the supervisor's interest and attention to analyze these videos. Therefore, an automatic video monitoring system can mitigate these barriers.

A video monitoring system is the automatic and logical analysis of information extracted from a surveillance video data. Examples of such monitoring systems can be a counter in each area at supermarkets which could help efficiently managing customer services as well as promote marketing strategies or a follow-on of patient's trajectories and hobbies to detect abnormal activities.

In order to understand the typical building blocks of a video monitoring system, let us consider the work-flow of an activity recognition system described in figure 1.3. The aim of an activity recognition system is to automatically label objects, persons and activities in a given video. As shown in the work-flow, a video monitoring system includes generally different tasks: object detection, object tracking, object recognition and activity recognition. This thesis studies a narrow branch of the object tracking task: multi-object tracking (MOT) in a single camera view.

Figure 1.1: Illustration of some areas monitored by surveillance cameras. (a) stadium, (b) supermarket, (c) airport, (d) railway station, (e) street, (f) zoo, (g) ATM corner, (h) home, (i) highway.

1.1 Multi-object tracking (MOT)

Multiple Object Tracking (MOT) plays a crucial role in computer vision applications. The objective of MOT is to locate multiple objects, maintaining their identities and completing their individual trajectories in an input video. Targeted tracking objects can be pedestrians or vehicles on the street, sport players in the court, or a flock of animals in the zoo, patients in heathcare room, etc. Although different kinds of approaches have been proposed to tackle this problem, many issues are still unsolved and hence it is an open research area. In the following part, we list and discuss five main MOT challenges which directly affect to tracking performance and motivates our researches on this domain.

- **Changes in scene illumination**: Changes in the scene illumination directly affect the appearance of an object. They are not only in lighting intensity but also the lighting direction disturbs can also affect the object's appearance . For example, the light casting

different shadows depending on its direction ca a possible scenario. These challenges due to illumination changes are not only a problem for the detection but also affect the tracking quality. The detector may fail to segment objects from shadows or may detect the shadow instead of the object. Further, the object maybe also mis-detected due to low illumination or low contrast. In these cases, an object trajectory may be segmented into short trajectories (tracklets). Moreover, the object appearance changes prevent trackers to find out the invariant information of objects throughout time.

- **Changes in object shape and appearance**: Objects having linear movement (e.g. cars on highway, people crossing the street ...) are usually easier to track because of their consistent appearance. However when the object rotates around itself or the object disappeared and comes back to the scene can also considerably change the appearance in the 2D image. In addition, deformable objects, like humans, can greatly vary in shape and appearance depending on their movements. Shape can be difficult to model with such variations. In these cases, models based on colour distributions are more reliable and they can help to localize the object.

- **Short-time full or partial occlusions**: Short time full occlusions or partial occlusions occur frequently in real world videos with a high density of moving objects. They can be caused either by the object itself (hand movements in front of a face), by the surrounding obstacles (static occlusions) or by neighbouring objects (dynamic occlusions). It is a difficult task to handle such occlusions because they alter the online learned object model and they prevent from obtaining a continuous trajectory and may cause the tracker to drift.

- **Background**: Complex background, or textured background may have similar patterns or colours to the object. Due to these factors, the tracker can fail or drift.

- **Camera motion**: In real-life videos, the moving camera tends to follow the main target object. However, when the videos are taken by a small consumer camera (like a mobile phone), we can observe a lot of trembling, and jitters causing and motion blur in the images or abrupt zooming. Rapid movements of the object can also have similar effects on the quality of the video.

1.2 Motivations

Tracking approaches from the state-of-the-art have been proposed to improve the tracking quality by handling above challenges. However, these approaches can face either theoretical or

Figure 1.2: A video surveillance system control room.

experimental issues. For example, the trackers may have issues to represent an object appearance adapting to the variation of video scenes, the tracker may require an important training stage which is time-consuming and their setting may depend on many parameters to be tuned.

Furthermore, our researches mainly focus on human tracking because of these three following reasons. Firstly, compared to other conventional objects in computer vision, humans are challenging objects due to their diversity and non-articulated motion. Secondly, the huge number of videos of humans illustrate the huge number of practical applications which have a strong commercial potential. Thirdly, according to our knowledge, humans are objects which at least 70% of current MOT research efforts are devoted to.

Therefore, the objectives of this thesis is to proposed novel methods which improve multi-person tracking performance by addressing the mentioned issues.

1.3 Contributions

This thesis brings three contributions, three algorithms to improve tracking performance by addressing above challenges. All algorithms are categorized as long-term tracking which try to link short person trajectories (tracklets) which have been wrongly segmented due to full occlusion or bad quality detection.

Here are described the three proposed long-term multi-person tracking algorithms:

- **A robust tracker named *Reliable Feature Estimation* (RFE) based on an online estimation of tracklet feature reliability.** The variation of video scenes can induce changes of the person's appearance. These changes often cause the tracking models to drift because

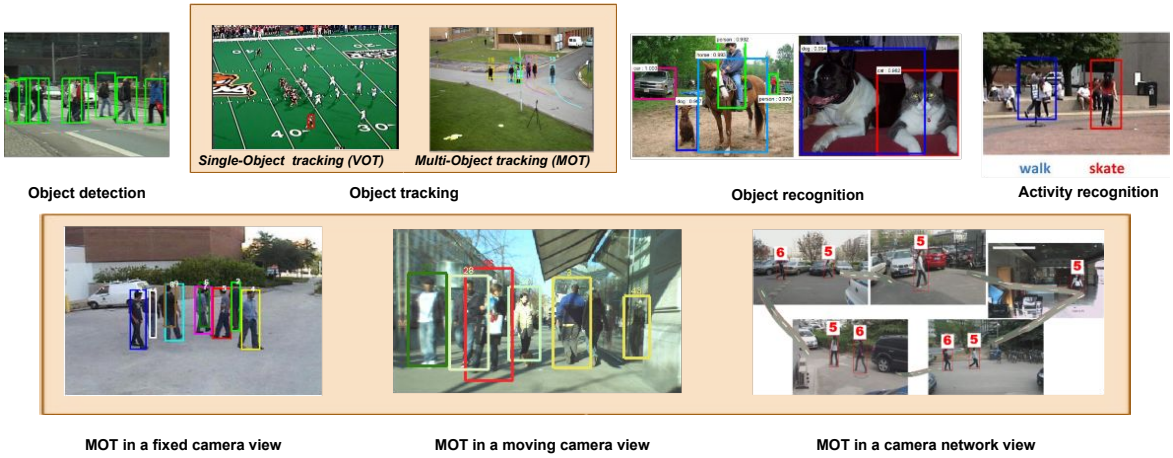


Figure 1.3: Illustration of some tasks of video understanding. The first row shows the workflow of a video monitoring system. The object tracking task is divided into two sub-types: Single-object tracking and multi-object tracking. The second row shows scenes where the multi-object tracking (MOT) is performed, including tracking objects from a fixed camera, from a moving camera and from a camera network, respectively.

their update cannot be able to quickly adapt to these changes. Therefore, we propose a tracking algorithm which selects automatically reliable tracklet features which discriminate trackets from each others. The reliable tracklet feature must discriminate a tracklet with its neighbourhood and pull this tracklet with its corresponding tracklet closer. There are some advantages of our approach over the state-of-the-art: (1) No training process is needed which makes this algorithm generic and employable to a large variety of tracking frameworks. (2) No prior knowledge information is required (e.g. no calibration and no scene models are needed).

- **A new mechanism named *Context-based Parameter Tuning* (CPT) for tuning online tracking parameters to adapt the tracker to the variation of neighborhood of each tracklet.** Two video scenes may have the same person density, occlusion level or illumination, but appearance of persons in the scene may not be the same. Therefore, utilizing the same tracking settings for all persons in the video can be inefficient to discriminate persons. In order to solve this issue, we proposed a new method to tune tracking parameters for each tracklet independently instead of globally share parameters for all tracklets. The offline learning step consists of building a database of tracklet representations together with their best tracking parameter set. In the online phase, the tracking parameters of each tracklet are obtained by retrieving the representation of the current tracklet with its closest learned tracklet representation from the database. In the offline phase, there is no

restriction on the number of tracking parameters as well as their mutual independence within the process of learning the optimal tracking parameters for each tracklet. However, there is a requirement on the training data which should be diverse enough to make this algorithm generic.

- **A tracking algorithm named *Re-id Based Tracker* (RBT) adapting features and methods is proposed for person Re-identification in multi-person tracking**. The algorithm takes full advantages of features (including hand-crafted and learned features) and methods proposed for re-identification and adapt them to online MOT. In order to represent a tracklet with hand-crafted features, each tracklet is represented by a set of multi-modal feature distributions modeled by GMMs to identify the invariant person appearance features across different video scenes. We also learn effective features using Deep learning (CNN) algorithm. Taking advantage of a learned Mahalanobis metric between tracklet representations, occlusions and mis-detections are handled by a tracklet bipartite association method. This algorithm contributes to two scientific points: (1) tracklet features proposed for Re-identification (LOMO, MCSH, CNN) are reliably adapted to MOT, (2) offline Re-identification metric learning methods are extended to online multi-person tracking. The metric learning process can be implemented fully offline or as a batch mode. However, learning the Mahalanobis metric in the offline training step requires the training and testing data should be similar. In order to make this algorithm become generic, instead of using hand-crafted features, we represent a tracklet by CNN feature extracted from a pre-trained CNN model. Then, we associate the CNN feature-person representation with Euclidean distance into a comprehensive framework which works fully online.

1.4 Thesis structure

This manuscript is organized as follows:

- Chapter 2 presents the literature review of Multi-object tracking (MOT). It focuses on categorizing the state-of-the-art MOT algorithms and MOT models as well as MOT trends.

- Chapter 3 presents definitions, pre-post processing functions and MOT evaluation method which are used by the proposed approaches described in upcoming chapters.

- Chapter 4 details a new multi-person tracking approach named *RFE* which keeps person IDs by selecting automatically reliable features to discriminate tracklets (defined as short person trajectories in chapter 3) in a particular video scene. No training process is required in this approach.

- Chapter 5 presents a framework named *CPT* which online tunes tracking parameters to adapt a tracker to the change of video segments. Instead of tuning parameters for all tracklets in a video, the proposed method tunes tracking parameters for each tracklet. The best satisfactory tracking parameters are selected for each tracklet based on a learned offline database.
- Chapter 6 presents a framework named *RBT* which extends the features (han-crafted or CNN features) and tracklet affinity computation methods designed for the people Re-id task (working in an offline mode) to online multi-person tracking.
- Chapter 7 is dedicated to the experimentation which evaluates and compares the proposed approaches to each other as well as to the state-of-the-art trackers. The results not only highlight the robustness of the proposed approaches on several benchmark datasets but also figure out elements affecting the tracking performance.
- Chapter 8 presents the concluding remarks and limitations of the thesis contributions. Thanks to this, future work is given out to address these limitations and to improve the performance of proposed approaches.

2

Multi-Object Tracking, A Literature Overview

Multiple Object Tracking (MOT) is an important task in the pipeline of video monitoring system. Different kinds of approaches have been proposed to tackle the MOT challenges such as abrupt object appearance changes, occlusions or illumination variations, however, these issues have been unsolved yet. With the purpose of deeply understanding this topic as well as clearly presenting our proposed approaches, in this chapter, we endeavor to review challenges, trends and researches related to this topic in the last decades.

A part of this review first focuses on MOT algorithm categorization and MOT models based on the overview in [66]. Then, we discuss in detail about drawbacks of MOT models, trends of the state-of-the-art trackers to address MOT problems. Based on this analysis, we propose methods to enhance tracking performance. The structure of this chapter is organized as follows: Section 2.1 categorizes the MOT algorithms from the state-of-the-art based on their processing modes. Section 2.2 examines a list of MOT models categorized into two parts: observation model and association model where observation models focus on the object representation and their affinity; and association models dynamically investigate the matching mechanisms of objects across frames. Trends of MOT tracking algorithms from the state-of-the-art as well as their limitations is revealed in section 2.3. Finally, section 2.4 briefly describes our proposals beyond the limitations of the state-of-the-art trackers to enhance MOT performance.

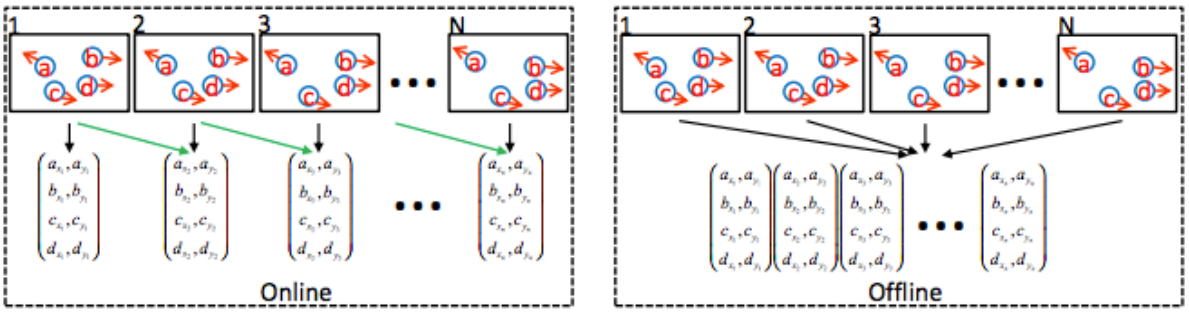


Figure 2.1: Illustration of online and offline tracking. Video is segmented into N video chunks.

2.1 MOT categorization

According to the way of processing data, MOT algorithms could be categorized into online or offline tracking. The difference is how the object detections are utilized when handling the tracking in the current frame. Online tracking utilizes detections up to the current frame or current video chunk to conduct the estimation, while offline tracking employs object detections in the whole video. In this part, we will analyze and compare online and offline tracking in some aspects such as required input, methodology, advantages as well as disadvantages of each method.

2.1.1 Online tracking

Online tracking methods associate object detections in the current frame [84, 95] or between tracklets in a short video chunk [8, 79]. If online tracking utilizes object detection up to the current frame, we categorize it as short-term tracking. Otherwise, we categorize it as long-term tracking. Online tracking algorithms commonly use bipartite matching methods for data association where Hungarian algorithm is the most popular method. These methods are capable of online processing based on frame-to-frame association or with a acceptable latency if detections in a short time-window are achieved in advance. Therefore, they could be applied in online processing applications. Although these methods are less computationally expensive, identifying objects could fail due to inaccurate detections (false positives) and online tracking algorithms can only deal with short-term occlusions.

2.1.2 Offline tracking

Offline tracking consists of algorithms where object observations (detection or tracklet - a short object trajectory) in video or image sequence are obtained in advance. These algorithms

Items	Online Tracking	Offline Tracking
Required input	up-to-current detections	all detections of the whole video
Methodology	- gradually extends existing trajectories with current detections - bipartite graph optimization	- links detections in the whole video into object trajectories - global optimization
Advantages	- suitable for online tasks - less expensive computation cost	- can recover long-term occlusions
Disadvantages	- recovers only short-term occlusions	- delays in outputting final results - huge computation cost - pre-requirement for all object detections in the whole video - huge search space for global optimization

Table 2.1: The comparison of online and offline tracking.

can overcome the shortcomings of online trackers by extension of a bipartite matching into a network flow. The direct acrylic graph in [127] is formed with vertices corresponding to object detection or to tracklets and edges corresponding to the similarity links between vertices. In [90], a track of a person forms a clique and MOT is formulated as a constraint maximum weight clique graph. The data association solutions for these offline tracker are found through minimum-cost flow algorithm. However, offline tracking methods also have their obvious drawbacks, such as: their huge computational cost due to iterative association process to generate globally optimized tracks and their pre-requirement for entire object detection in a given video.

Figure 2.1 illustrates the difference between online and offline tracking algorithms. To be clearer, we compare them in Table 2.1.

2.2 MOT models

MOT is composed of two primary components: observation model and association model. Observation models represent object observations (detection, tracklet) and measures the similarity between two object observations (detection - detection, tracklet - detection, tracklet - tracklet). Association models dynamically investigate the matching mechanisms of object observations across frames. In this section, we present and discuss both models in details.

2.2.1 Observation model

An observation models are categorized into appearance, motion, exclusion and occlusion handling models. Types of observation models are discussed in details in this part but this manuscript focuses more on the appearance model which presents the most important information for object affinity computation in MOT.

2.2.1.1 Appearance model

Almost of the recent trackers pay their attention to represent the object appearance for affinity measurement in MOT. Different from visual object tracking (VOT) which focuses on constructing an object representation to discriminate the target from background, MOT need to discriminate targets from each other. Therefore, beside building representations for objects, the appearance model for MOT measures the affinity or the discrimination power between objects. Appearance model includes two components, *visual representation* and *statistical measurement*. Visual representation describes the visual characteristics of the target based on features while statistic measurement computes the affinity or the discrimination power between two object representations. In the following, we first discuss about features, then describe the appearance model categories.

2.2.1.1.1 Features

Figure 2.2 shows seven types of object features which have been deployed in MOT. In this section, we describe these features as well as the purposes of using these features in MOT as following.

- ***Point-based features*** are features extracted from points-of-interest which bring meaningful object information. Point-based features are not only efficiently utilized for VOT [94] but also are helpful for MOT. For instance, KLT tracker is employed to track feature points and generates a set of trajectories or short tracklets [99, 51]. KLT features [103] are utilized by [12] to estimate object motion. Similarly, point-based features are also employed by [17] for motion clustering.

- ***Color-based features***: These are the most visual and popular features which are utilized for MOT. Based on kinds of color-based features, color intensities of object are extracted and presented under different ways. Color histogram is used by [90, 7, 28, 98, 38]. The simple raw pixel template is employed by [114] to compute the appearance affinity. The color-based features along with a measurement are usually employed to calculate the affinity between two object observations (detection-detection, detection-tracklet, tracklet-tracklet).

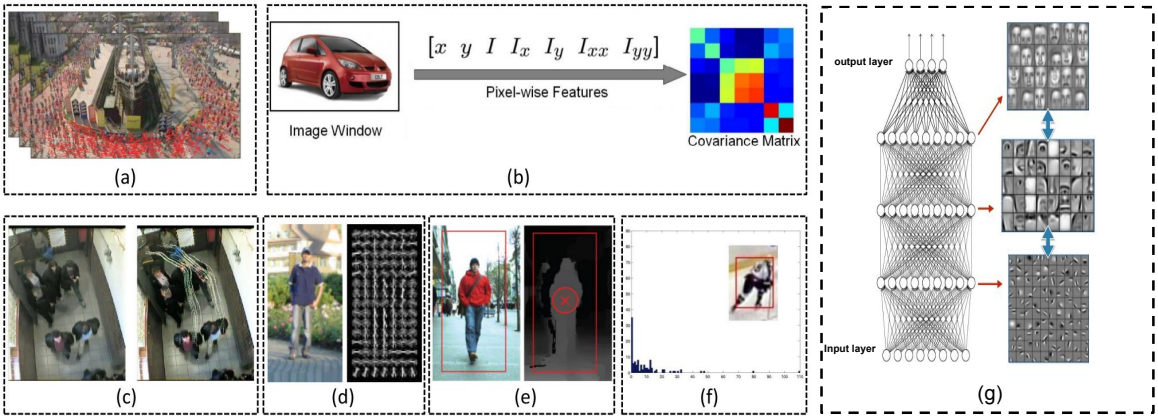


Figure 2.2: Different kinds of features have been designed in MOT. (a) Optical flow, (b) Covariance matrix, (c) Point features, (d) Gradient based features, (e) Depth features, (f) Color histogram, (g) Deep features.

- ***Optical flow*** is composed by trajectories of object's point-of-interest. The optical flow feature can be employed to conduct short-term VOT. Thus many solutions proposed for MOT utilize optical flow to link detections from consecutive frames into tracklets for further data association processing [89] in long-term tracking. Optical flow is also employed to complement HOG for observation model [2]. Additionally, optical flow is popular in extremely crowded scenarios for discovering crowd motion patterns, object movement thanks to flow clustering [67, 88].

- ***Gradient-based features***: An image gradient is a directional change in the intensity or color in an image. There are some features based on gradient proposed to characterize objects in MOT. For example, authors in [76] utilize a variation of the level-set formula, which integrates three terms: penalizing the deviation between foreground and background, an embedding function from a signed distance function and the length of the contour to track objects in frames. Besides the success in object detection, HOG [26] plays a vital role in the multiple object tracking problem as well. For instance, HOG is employed in [38, 53, 24] to detect objects and/or to compute similarity between human detections for data association.

- ***Region covariance matrix features***: Region covariance matrix features [104] are robust to issues such as illumination changes, scale variations, etc. Therefore, it is also employed for the MOT problem. In [5, 40], the region covariance matrix based similarity is used to compare appearance for data association. In different ways, covariance matrices along with other features constitute the feature pool for appearance learning in [53, 42] to

represent object for both single and multiple object tracking.

- ***Depth features***: Depth information is employed for various computer vision tasks. These features are directly extracted from 3D-camera data or indirectly via a projection on different 2D-camera views. With regard to MOT, authors in [76] utilize depth information to correct bounding box of object detection and re-initialize the bounding box for tracking. Authors in [30] employ depth information to obtain more accurate object detections in a mobile vision system and then use the detection result for multiple object tracking. Besides that, method in [35] integrates depth to generate detections and consequently verify their consistency for multiple object tracking from a moving car.

- ***Deep features*** With the success of deep learning in solving classification problems, more and more trackers such as [109, 125, 92] extract deep appearance features to describe objects and obtain significantly higher performance in both online and offline setting. The extracted deep appearance features are feature vectors obtained from convolution layers in deep networks. Different layers encode different types of features. Higher layers capture semantic concepts on object categories, whereas lower layers encode more discriminative features to capture intra class variation.

To sum up, above mentioned features work efficiently in particular cases. However, beside their advantages, there still exist shortcomings. For example, color-based histogram enables to compute effectively the similarity of two object observations, but it ignores the spatial layout of object regions. Gradient-based features like HOG can describe the shape of object and are robust to illumination changes but they are less effective in handling occlusion and deformation. Region covariance matrix features obtain useful information on object, but they bear a high computation cost. Depth features add extra information on objects to get more accurate measures in affinity computation, but they require depth information (captured by 3D cameras) or multiple views of the same scene and additional matching algorithm. Deep features give a diverse information of objects depending on the results of convolution layers. However, choosing effective information from which layers is depended on videos and deep features require high training costs. Therefore, single feature selection and combination for MOT depends on the requirement of the applications and are still a challenge.

2.2.1.1.2 Appearance model categories

We categorize appearance models based on how the state-of-the-art trackers use these features to represent object appearance into two types: *Single feature based appearance model* and *multiple feature based appearance model*.

a. Single feature based appearance model

Utilizing a single feature is a popular option of appearance model in MOT because of its simplicity and efficiency. In the following, we present four ways to build a single feature based appearance model.

- ***Raw pixel template representation***: The raw pixel template representation collects the raw pixel intensity or color of a region. Beside that, it can encode the spatial information. Because of its simplicity and usefulness, some methods use this appearance model when matching two templates. In particular, Yamaguchi *et al.*[114] employ the Normalized Cross Correlation (NCC) to evaluate the predicted position of objects. The method proposed in [1] computes the appearance affinity as the NCC between the object template and a candidate bounding box. Wu *et al.*[112] build a network-flow approach to handle multiple target tracking at each time instant. In this approach, MOT is presented as a network with flows as transitional costs between object observations. These costs are computed by NCC between upper one-fourth bounding-box of object observation pairs. Despite of discussed efficiency, this kind of representation easily suffers from the change of illumination, occlusion or some other issues.

- ***Color histogram representation***: Color histogram is the most popular representation for appearance modeling in MOT approaches. Authors in [51] design a color histogram model [82] to calculate the matching likelihood in terms of appearance, and they use an exponential function to transform the histogram distance into probability. In addition, to capture the similarity, authors in [99] use the Bhattacharyya distance between hue-saturation color histograms when constructing a graph. Appearance model is defined as the RGB color histogram of a trajectory by Leibe *et al.*[60]. It is initialized as the first detection's color histogram and evolves as a weighted mean of all the detections which belong to this trajectory. The likelihood considering appearance is proportional to the Bhattacharyya coefficient of two histograms. Affinity regarding appearance is obtained by calculating the Bhattacharyya distance between the average HSV color histograms of the concerned tracklets [85]. Though color histogram representation is powerful in capturing the statistical information of target region, it has the drawback of losing spatial information.

- ***Covariance matrix representation***: Covariance matrix is robust to illumination change, rotation, etc. The covariance matrix descriptor is employed to represent the appearance of an object by Henriques *et al.*[40]. Then, the likelihood of appearance to link two object regions is modeled as a Gaussian distribution. In [42], an object region is divided into blocks. Within each block, the covariance matrix is extracted as the region descriptor to characterize the block. At the same time, likelihood of each block of this object region is

computed with regard to the corresponding block of the counterpart, and likelihood of the whole region is the product of the likelihood of all blocks.

- ***Bag of words representation***: Clusters of local image features are treated as words. In computer vision, a bag of words is a vector of occurrence vocabularies of clusters of local image features. Fast dense SIFT-like features ([65]) are computed by Yang *et al.*[119] and encoded based on the bag-of-word model. In this model, each image is represented as a collection of vectors of the same dimension and the order of different vectors is of no importance. Therefore, if spatial information is needed, the spatial pyramid matching (SPM) method proposed in [56] is applied. This is used as an observation model for appearance modeling.

b. Multiple feature based appearance model Although a single feature based appearance model is simple and efficient, this model is not effective enough to characterize object in complex videos. Therefore, gathering different kinds of features would make appearance model robust. However, how to combine the information from multiple features could be an issue. We present four types of mechanisms to build multiple feature based appearance models:

- ***Boosting***: The strategy of Boosting usually selects a subset of features from a feature pool sequentially via a Boosting based algorithm (e.g. Adaboost by Kuo *et al.*[50] and RealBoost by Yang and Nevatia [118]). Features are selected according to their discrimination power. A discriminative appearance model proposed by [50] assigns high similarity to tracklets which are of the same target, but low affinity to tracklets of different targets. This model is composed of color histogram in RGB space, HOG and covariance matrix descriptor as features, applied in 15 regions, so that they have 45 cues in total in the feature pool. Collecting positive and negative training pairs according to the spatial-temporal constraints, they employ Adaboost to choose the most representative features to discriminate pairs of tracklets belonging to the same object from those belonging to different objects. A HybridBoost algorithm is proposed by Li et al. [61] to automatically select features with maximum discrimination. This algorithm employs a hybrid loss function composed of a classification term and a ranking term. Correct tracklet associations are set to the higher ranks and wrong associations are dismissed by the classification.

- ***Concatenating***: Brendel *et al.*[16] trains a SVM model classifier to distinguish a specific target from targets in its temporal window. To describe a target, features including color, HOG and optical flow are concatenated and further processed with Principal Component Analysis (PCA) projection for dimension reduction. The similarity S between two object observations is computed by Mahalanobis distance as follow:

$$S = exp(-(f - f')^T M(f - f')) \tag{2.1}$$

where M is the Mahalanobis distance metric matrix which is learned online. f and f' are concatenated features of two object observations.

- ***Summation***: More than one features are gathered to represented the object appearance model. If each single feature is used to compute a matching by a probability, some state-of-the-art trackers [13, 64] present the appearance model of an object O_i as a matching probability which is the weighted summation of single-feature probability P_i^k as follows:

$$P_i = \frac{\sum_k^N w_k \times P_i^k}{\sum_k^N w_k} \tag{2.2}$$

where k is feature index, N is the number of features and w_k is the weight to balance single features.

- ***Product***: In a similar way, the short-term tracker in [119] integrates multiple features including color, shapes and local features to calculate the likelihood linking a new detection with an existing trajectory. The approach in [98] multiples the color histogram likelihood and depth likelihood as the final likelihood to compose the appearance model. These methods share the following similar formula:

$$P(f^1, f^2..f^k|s) = \prod_{k=1}^{N} P(f^k|s) \tag{2.3}$$

where N is the number of features, $P(f^1, f^2..f^k|s)$ is the likelihood linking a detection with a trajectory s and f^k is feature k.

In general, each combination method has its own limitations. The limitations of boosting strategies are time consuming, hardly implementable in real-time platform and increasing the complexity of the classification. The concatenating method requires an important preprocessing step to normalize the dimension of features. Computing the weight in the summation method is a challenge when the video condition changes affect directly to single-feature reliability. The product method treats single-features with equal roles which is limited the discriminative power of single-features. Therefore, selecting which combination method to gather single-features to represent object depends on the requirement of MOT applications.

2.2.1.2 Motion model

The second popular model that the state-of-the-art trackers use to represent objects is the motion model. Object motion model describes the movement of an object. It is important for MOT since it can reduce search space by predicting the potential position of objects in the future

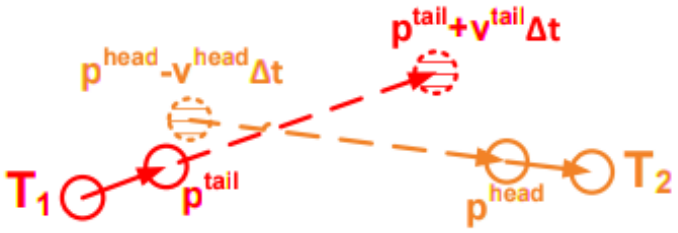


Figure 2.3: Illustration of linear motion model presented in [113] where **T** standing for Target, **p** standing for Position, **v** standing for Velocity of the target.

frames. Motion models employed in MOT are generally divided into the following two classes: Linear and non-linear motion models.

Linear motion models: These models are designed for targets assumed to move with constant velocity. This is the most popular model of pedestrian or vehicle movements which are smooth in video scenes (abrupt motions are a special case). The velocity of object in the next frame is the same as the current velocity and is drawn by some types of distribution.

A constant velocity models, including forward velocity and backward velocity is computed simultaneously by [113] to calculate the motion affinity of two tracklets. The illustration of this linear motion model is shown in figure 2.3. Each velocity model is represented by a Gaussian distribution. Assuming that the last position of target $\mathbf{T}_i$ appears earlier than the first position of target $\mathbf{T}_j$. The forward velocity distribution is centered on p_j^{head}- the head position of target $\mathbf{T}^{\cdot}\mathbf{j}$: $G(\mathbf{p}_j^{head}, \Sigma_j^B)$. It estimates the probability of the position of $\mathbf{p}_i^{tail}$ to reach $\mathbf{p}_j^{head}$ with a forward displacement of tracklet $\mathbf{T}_i$ presented by $v_i^F \Delta t$. The backward velocity distribution is centered on $\mathbf{p}_i^{tail}$ - the tail position of tracklet $\mathbf{T}_i$: $G(\mathbf{p}_i^{tail}, \Sigma_i^F)$. It calculates the probability of position $\mathbf{p}_j^{head}$ backward to $\mathbf{p}_i^{tail}$ with the backward replacement of $\mathbf{T}_j$ presented by $v_j^B \Delta t$.

$$P_m(\mathbf{T}_i, \mathbf{T}_j) = \mathcal{N}(\mathbf{p}_i^{tail} + v_i^F \Delta t; \mathbf{p}_j^{head}, \Sigma_j^B) * \mathcal{N}(\mathbf{p}_j^{head} + v_j^B \Delta t; \mathbf{p}_i^{tail}, \Sigma_i^F) \tag{2.4}$$

the motion model in [3, 73] is also a constant velocity model. In loss function for matching objects, the optimization term which considers differences between the velocities of one object in different time instants, is formulated as follows:

$$O_m = \sum_{t=1}^{N} \sum_{i=1}^{M} \| v_i^t - v_i^{t+1} \|^2 \tag{2.5}$$

where v_i^t is the velocity of object i at time t. It is computed as the displacement between object positions in two consecutive frames. The first summation takes all the N frames into account and the second summation counts all the M trajectories/objects. Intuitively, this term penalizes the difference between velocities and forces object trajectories to be smooth.

Non-linear motion models: Commonly, the movement of objects, especially pedestrian, can be modeled by linear motion models. However, as shown in figure 2.4, there are often non-linear motion patterns in a scene. Therefore, non-linear motion models are proposed to represent more accurately a tracklet motion. The figure 2.5 illustrates the linear as well as non-linear motion models in the same scenario. The red and orange lines represent linear motion estimation while the blue line describes the non-linear motion model proposed by [116]. The authors online learn a non-linear motion map M which is defined as a set of tracklets that include confident non-linear motion patterns. As shown in figure 2.5, the tracklet T_0 is a support tracklet , $T_0 \in M$, to explain the motion link between T_1 and T_2 because there exist elements $\{(p_i, s_i, v_i)\}$ in T_0 which are matched with the last position of T_1 and the first position of T_2. p, s and v are position, size and velocity of each pattern in map M, respectively. Then the real path to link T_1 and T_2 is estimated based on T_0. In order to compute the motion affinity between two tracklets, the authors also use the method formulated by equation 2.4, but based on the non-linear motion positions.

Non-linear motion models can accurately represent non-lear motions of a target. However, targets can share the same motion pattern or a target can fit into more than one motion pattern. These cases confuse MOT algorithms to discriminate targets. Therefore, almost the state-of-the-art trackers [3, 73, 116, 113] use motion models as the additional information to characterize objects in a video scene.

2.2.1.3 Exclusion model

Exclusion is a constraint when solving MOT problem due to physical collisions. There are generally two constraints to be applied on multiple detections and trajectories. The first one is the so-called detection-level exclusion (i.e., two different detections in the same frame cannot be assigned to an identical trajectory). The second one is the so-called trajectory-level exclusion (i.e., two trajectories cannot share an identical detection). The detail of both constraints is presented as follows.

Detection-level exclusion

The detection-level exclusion is modeled as a constraint to penalize physical collisions among detections. The approach in [74] forces that two objects appearing in the same frame have to keep different identities. Similarly, authors in [52] employ label propagation for multiple object tracking. To model exclusion, a special exclusion graph is constructed to capture the constraint that detections with the same time stamp (occurring at the same time) should have different labels.

In different ways, exclusion is modeled as an extra constraint in the objective function of network flow in [18]. Let the detections at frame k be $O_k = \{o_1^k, ..., o_{M_k}^k\}$. Given detections in two consecutive frames as O_k and O_{k+1} , one detection from O_k and another detection from

Figure 2.4: Illustration of non-linear movements

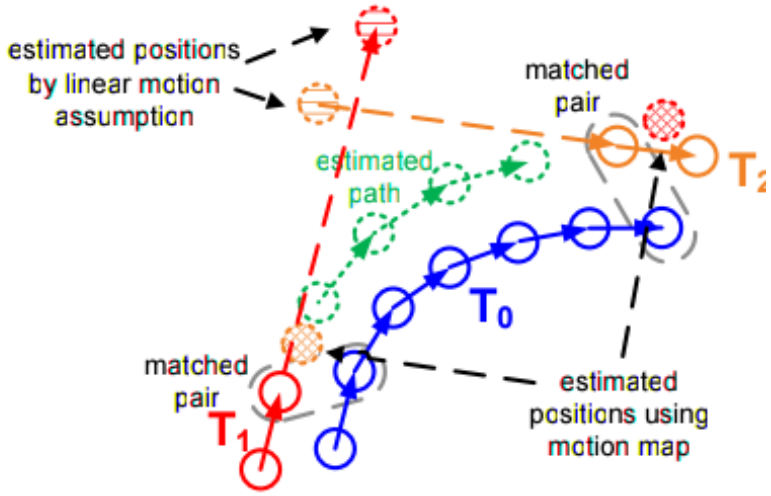


Figure 2.5: Illustration of non-linear motion model in [116]

O_{k+1} can form a match. Based on all matches between these two frames, a graph is constructed as $G = (V, E)$, where each node in G is a pair of detections and each edge belonging to E represents flow in the graph, where flow 1 means linkable and 0 means not. Conflict edges are represented as $E_{conflict}$. Recalling the constraint that one detection should only be occupied by no more than one trajectory, the flow through edge in $E_{conflict}$ is constrained to be at most 1.

Trajectory-level exclusion

Trajectory-level exclusion is defined as a constraint applied on tracklets or trajectories. In approach [7], authors define two constraints named ”must-link” and ”cannot-link” between two tracklets to create exceptions in the clustering algorithm and guarantee the integrity of the proposed algorithm. With ”must-link” constraint, two tracklets that were merged at time $t-1$ stay merged at time t. The cannot-link constraint provides spatio-temporal constraints based on the camera network. For a single camera, two tracklets appearing on the same frame cannot belong to the same object. The object cannot appear on two non-overlaping cameras at the

same time.

In approach [74], the authors also penalize the case when two close trajectories Tr_i and Tr_j have different labels. The penalty is proportional to the spatial-temporal overlap between Tr_i and Tr_j . The closer the two trajectories, the higher penalty it is. Similarly, authors in [3] model the exclusion as an additional cost term to penalize the case when two trajectories are very close to each other. The cost is reversely proportional to the minimum distance between the trajectories in their temporal overlap. By doing so, one of the trajectory would be abandoned to avoid the collision.

2.2.1.4 Occlusion handling model

Occlusion is a big challenge to MOT algorithms. It could lead to ID switch or fragmentation of trajectories. In the literature, various kinds of strategies have been proposed in order to handle occlusion. These strategies are categorized into three following types.

Part-to-whole

This strategy is the most popular one for occlusion handling. It assumes that, part of the object is still visible when occlusion happens, even the complete occlusion still begins with partial occlusion. This assumption allows trackers to utilize the visible part to infer state of the whole object. In [42], an object region is divided into multiple non-overlapped blocks. For each block, an appearance model based on subspace learning is constructed. Likelihood is computed according to reconstruction error in the subspace corresponding to each block. In order to deal with occlusion along with the task of recovering occlusion relationship among objects, the occlusion handing model solves the occlusion problem in tracking in two aspects. Firstly, spatial information is considered as the likelihood of an object region which is the product of likelihood of all its blocks. Secondly, an occlusion map is obtained according to reconstruction errors of all blocks. Then, this occlusion map is utilized to reason on the occlusion relationship among objects.

Part based model is also applied in [38] as a multi-person multi-part tracker. Human body is divided into individual body parts. In the next step, the whole human body and individual body parts are tracked in parallel. The final trajectory estimation is obtained by jointly association between the whole human body and the individual human body parts. Figure 2.6 shows how the part based model handles occlusion. The pedestrian is occluded from frame 47 to frame 134. During this period, the whole-body human detector would be confused. However, thanks to the detected visible parts, trajectories of visible parts are estimated. Furthermore, along with the trajectory of the whole body, the complete trajectory is recovered.

Tracking based on appearance information may fail when occlusion happens. In a different way shown in [99] motion of feature points in visible parts is also applicable to address occlusion.

Figure 2.6: An illustration of occlusion handling by the part based model.

Hypothesize-and-test

This strategy solves occlusion challenges by hypothesizing proposals and testing the proposals according to observations achieved after occlusions.

The long-term tracker proposed in [127] builds a cost-flow framework for each time-window. In order to handle long-term occlusion, increasing the size of time-window is needed. However, it also increases the search space of global optimization. In order to reduce the number of ambiguous objects which are occludable, an Explicit Occlusion Model (EOM) is proposed and integrated into the cost-flow framework. Occlusion hypotheses are generated based on the occlusion constraints that two object observations are occludable if and only if their distance and scale difference are small enough. Assuming o_i is occluded by o_j , a corresponding occlusion hypothesis is $O_{ji} = (p_j, s_i, f_i, t_j)$, where p_j and t_j are the position and the time stamp of o_j, and s_i and f_i are the size and appearance features of o_i. Along with the original observations (tracklets), all the observations are given as input to the cost-flow framework and MAP is conducted to obtain the optimal solution.

Buffer-and-recover

This model allows trackers to overcome full occlusion problem. In this strategy, states of object before occlusion are remembered and buffered. When occlusion ends, object states are recovered based on the buffered information.

The approach proposed in [75] combines a level-set tracker based on image segmentation and a high-level tracker based on detection for MOT. In their approach, the high-level tracker is employed to initialize new tracks from detection and the level-set tracker is used to tackle the frame-to-frame data association. When occlusion occurs, the level-set tracker would fail. To tackle this, the high-level tracker keeps a trajectory alive for up to 15 frames when occlusion happens. In case the object reappears, thanks to buffered object information, the object identity is maintained and object trajectory is recovered by an extrapolation mechanism.

Similarly, in order to handle occlusion, approaches [80, 79] keep tracklet information in a

buffer of two time-windows. Every full occlusions appearing in this two time-windows may be recovered when the distance of buffered tracklets before occlusion and tracklets reappearing after occlusion is close enough.

Occlusion is the biggest challenge of MOT because of two reasons. Firstly, occlusion makes the object appearance changes or invisible to trackers. Secondly, trackers becomes hard to define whether object trajectory is end. These discussed occlusion handling models prove their effectiveness in MOT, however, they still exist some limitations. For example, *part-to-whole* models face to alignment problems, the performance of *hypothesis-and-test* and *buffer-and-recover* models directly depends on the object representation. Therefore, by extending object features proposed for Re-identification such as LOMO, MCSH in [77, 63, 126] to MOT, the MOT algorithms could make a stable object representation against object appearance changes caused by occlusion.

2.2.2 Association model

Association model dynamically investigates state transition of objects across frames. Based on the method to obtain the states of objects, it can be classified into probabilistic inference and deterministic optimization methods.

2.2.2.1 Probabilistic inference

Object tracking can be viewed as the probabilistic estimation or prediction of the future state of an object (size, position and velocity). MOT approaches based on probabilistic inference model typically investigate the states of objects with a probabilistic distribution. Based on the existing observations of objects, this method estimates the probabilistic distribution of objects' states to identity objects in each frame. The two most common probabilistic methods used for tracking: the Kalman filter and the Particle Filter.

Probabilistic inference based methods estimate the new state of objects relying on only existing observations, thus they are especially appropriate for online tracking. However, efficient, probabilistic methods can face to issues such as a high computation cost, especially in models with a large number of parameters, and in selecting a prior to avoid misleading results. In the next section, we mainly focus on presenting *deterministic optimization model* which can overcome those limitations of probabilistic inference models.

2.2.2.2 Deterministic optimization

Different to the probabilistic inference models which estimate or predict the future states of an object, the task of deterministic optimization model in MOT is to define the best matches of obtained object observations via their similarity. The MOT problem is cast as a data association

optimization problem. If object observations are available at the current time instant (detections) or video chunk (detections and tracklets), the data association is processed in every frame or video chunk. We define this type of data association as *local data association* which is mostly employed in online tracking. Inversely. if object observations from all frames are obtained, the data association is applied for all object observations in the video. We categorize this type of data association as *global data association* which is suitable for offline tracking.

2.2.2.2.1 Local data association

Online tracking associates detections at the current frame with the most matching tracked objects [84, 95] or between tracklets in a video chunk [8, 79]. In order to match object observations, a local data association - Bipartite Graph Matching technique is the most popular method - is employed.

Bipartite Graph Matching: By modeling the MOT problem as Bipartite Graph Matching, two disjoint sets of graph nodes are defined, such as existing trajectories and new detections or two sets of tracklets in a video chunk. Weights among nodes are modeled as affinities among object observations. Then, greedy bipartite assignment algorithm [96, 15] or Hungarian algorithm [86, 87, 84, 95, 8, 79] are employed to derive the optimal matches between nodes in both sets.

2.2.2.2.2 Global data association

The global data association compute all matching abilities among obtained object observations in the video. To seek the optimal association, MOT problem is often defined as a flow or a graph where detections or tracklets are vertexes of the graph and the edges illustrate the link ability between two vertexes. The global data association method is popularly applied to the task of offline tracking. Some well-studied global data association approaches are detailed in the following.

Min-cost Max-flow Network Flow. The data association in the MOT problem is represented by a network flow where nodes in the graph for network flow are detections or tracklets. The flow is usually modeled as an indicator to link two nodes (flow is 1) or not (flow is 0). To meet the flow balance requirement, a source node and a sink node corresponding to the start and the end of a trajectory, respectively, are added to the original graph (see Figure 2.7). One trajectory corresponds to one flow path in the graph from the source node to the sink node. The cost to transit the flow from the source node to the sink node is the neg-likelihood of all the associations belonging to this flow. This model is adopted by several tracking approaches [24, 112, 18] to solve the MOT problem.

Conditional Random Field. Approaches including [118, 117, 74, 39] solve MOT problem by using a Conditional Random Field model. In this model, MOT task is represented by a graph

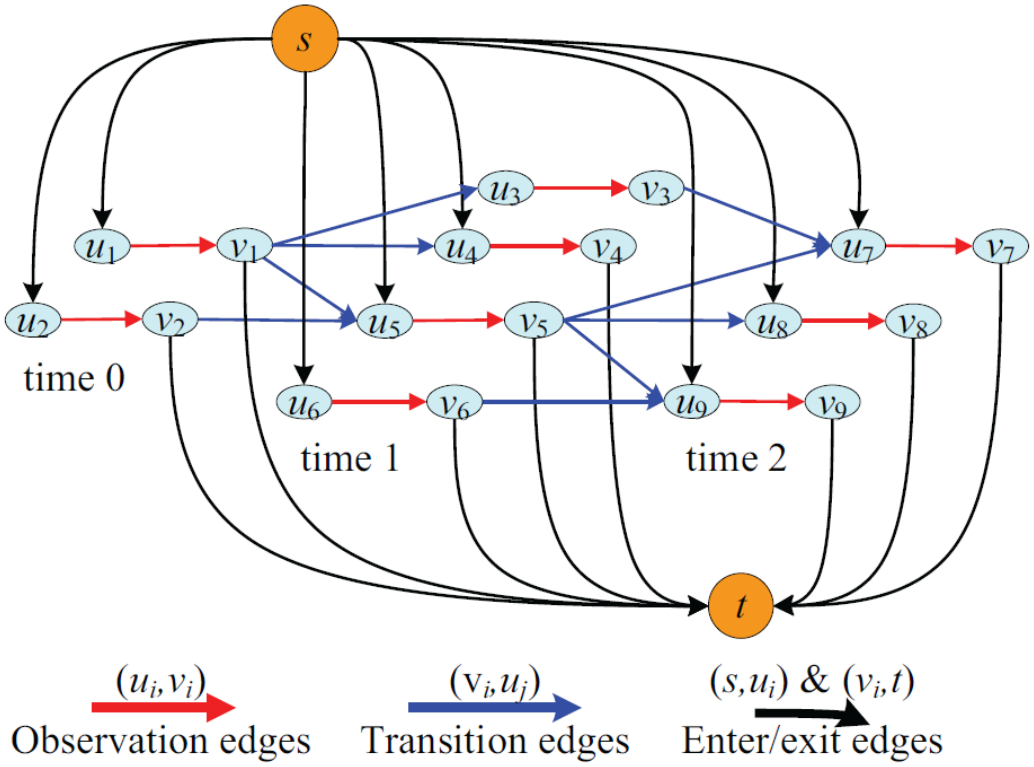


Figure 2.7: A cost-flow network with 3 timesteps and 9 observations [127]

$G = (V, E)$ where V is the set of vertices and E is the set of edges between vertices. The input of the graph is the tracklets. Each node in the graph is defined as a pair of tracklets, and a label is predicted to indicate the link ability of this pair of tracklets (flow is 1) or not (flow is 0). A label map is built up by these labels. Optimizing the label map can achive the optimal association of the tracklets for the MOT problem.

To sum up, with the help of global information, the *deterministic optimization model* could address occlusion and recover mis-detection better than the *probability inference model*. However, approaches based on deterministic optimization face problems such as more time processing and the optimization space. Additionally, with global data association, the requirement of access to all frames in advance prevents online applications.

2.3 Trends in MOT

The previous section discusses advantages and limitations of MOT models. Beyond this analysis, we turn our attention to the main trends in the MOT literature that trackers follow to trade-off the limitations of each MOT model. From the state-of-the-art, we can see that recent

trackers address MOT problems with only one or a combination of three following trends.

2.3.1 Data association

Before 2015, the MOT community mainly paid attention on finding strong, preferably globally optimal methods to handle the data association problem. The MOT problem was often cast as a graphical model and solved with k-shortest paths in [83], as a Linear program solved with the simplex algorithm in [59], as a Conditional Random Field in [71, 39] or as a Bayesian model as in [10]. The pairwise costs for matching two object observations (detection-detection, detection-tracklet, tracklet-tracklet) were based on either simple distances or matching probabilities.

2.3.2 Affinity and appearance

Recently, the attention shifted towards finding strong appearance cues to characterize objects. The impact of this trend is an increase in tracking performance and the ability for trackers to handle more complex scenarios. The top performance methods use sparse appearance models in [32] and integral channel feature appearance models in [46] to enhance object observation affinity. Deep learning base trackers which use deep networks as feature extractors or model data association as CNN classification also have an impact on tracking performance. Because of the power as well as the current strong interest, we spend the next subsection to discuss about some recent deep learning based trackers.

2.3.3 Deep learning

A deep neural network usually works in a standalone mode for most of computer vision tasks, such as image classification, object recognition and detection. The input and output of the deep neural network in this mode are a sample and a predicted label respectively. However, for object tracking, the objective is to estimate the similarity between a target and its candidates (i.e new detections) to decide whether they belong to the same object. The end-to-end training mode (”sample → label”) used by deep neural network is not applicable to object tracking. Therefore, deep-learning based object tracking algorithms switch the traditional deep neural network to work in the another training mechanism, called ”sample pair → similarity”.

Recently, deep neural networks have been widely employed to deal with the Visual Object Tracking (VOT) problem. Authors in [102] proposed a deep architecture containing three networks, a Feature Net, a Temporal Net and a Spatial Net. The Feature Net extracts general feature representation of the target from three convolution layers borrowed from VGGNet. Based on the feature representation, the Temporal Net builds a historical sample tuple by collecting key samples of target trajectory by L1-induced dictionary learning and sparse coding.

This tuple is updated incrementally. The input of the fully connected layer of Temporal Net is the learned tuple and the candidate regions in the current frames. Then, this network outputs the similarity between the current candidate regions and this historical sample tuple. Finally, the Spatial Net learns a spatial response via three convolutional layer combination to refine the estimated position. In another way, the proposed framework in [108] tracks a target via two layer deep network. The top layer encodes more semantic features and serves as a category detector. The lower layer carries more discriminative information and can better separates the target from distracters with similar appearance. The output of each layer is foreground heat map which is initialized in the first frame and updated online via a regression strategy. Finally, the target localization is first performed on the heat map produced by the top layer. If distractor is detected, the heat map of lower layer is utilized.

From state of the art, deep learning methods are also effectively applicable to multi-object tracking (MOT). Authors in [107] propose a novel and efficient way to obtain discriminative appearance-based tracklet affinity models. In this framework, each sample pair is passed to a Siamese CNN including two sub-CNNs to extract the feature vectors. Then, based on the feature vectors obtained from the last layer of both sub-CNNs in each video segment, temporally constrained metrics are learned online to update the appearance-based tracklet affinity model. Finally, MOT problem is formulated as a Generalized Linear Assignment (GLA) problem which is solved by the soft-assignment algorithm. Recently, another robust RNN-based multi-object tracker [92] has been proposed which outperforms previous works on most recent datasets including the challenging MOT benchmark. This method builds multiple-RNN models that learns to encode long-term temporal dependencies across multiple cue (appearance (A), motion (M) and interaction (I)). The output of each RNN model (represents the object in each cue) is a feature vector concatenated by 2 sub-feature vectors (same dimension). One sub-feature vector is extracted from a LSTM network which encodes long-term dependencies of object observations belonging to target trajectory. The other one is the result of RNN fully connected layer when passing directly the detection they wish to compare to the network. Finally, the final RNN is jointly trained end-to-end with the RNNs according to A, M and I cues by concatenating single feature vectors and outputting the score of whether a detection corresponds to a target using Soft-max classifier and cross-entropy loss.

2.4 Proposals

Based on the literature review, in this manuscript, we proposed three MOT approaches which handle discussed problems to improve MOT quality. The object of MOT task in our approaches is human. These approaches are categorized as long-term (tracklet-based) human tracking which processes their inputs with a time latency. We use a diverse feature pool in-

cluding features proposed for MOT (appearance features and motion) and features proposed for Re-identification (LOMO [63], CNN, MCSH [126]) to represent a person. All proposed approaches use *Buffer-and-recover* model as well as build strong person representations to handle partial or full occlusions.

The proposed approaches are presented in details in the upcoming chapters. These approaches can be distinguished through two properties: their generality and their effectiveness. The performances, advantages as well as disadvantages of each approach compared to state-of-the-art methods also also discussed. Depending on the requirement of applications as well as the availability of training data, we can choose which proposed MOT algorithm is the most appropriated.

3

General Definitions, Functions and MOT Evaluation

The proposed algorithms presented in upcoming chapters of the manuscript use some common definitions, object features, pre-or-post processing functions and MOT evaluation methods. Therefore, in order to help the readers easily understanding this manuscript, we spend this chapter on presenting this information.

3.1 Definitions

The content of manuscript focus on the long-term tracking category which tracks objects in a video chunk instead of a frame. Besides that, all approaches use exclusive model (can-match and cannot-match) to add constraints on tracklet during tracking process. Therefore, in this section, the definitions of a tracklet (object's short trajectory), a candidate (can-match tracklet) as well as a neighbour (cannot-match tracklet) of a tracklet are presented.

3.1.1 Tracklet

We define a tracklet Tr_i spanning over consecutive frames $< m, n >$ as a chain of tracked objects called nodes $O_i^t s$ $(m < t < n)$ where i represents the ID of the object and O represents the object bounding-box as follow:

$$Tr_i = \{O_i^m, O_i^{m+1}, ..., O_i^{n-1}, O_i^n\} \tag{3.1}$$

A tracklet is generated by a short-term tracker.

3.1.2 Candidates and Neighbours

For each tracklet Tr_i, we define two sets of related tracklets to Tr_i, including Candidate and Neighbour sets.

$$Rel_i = \{Can_i, Neib_i\} \tag{3.2}$$

Candidate Tracklet Tr_c is determined as a "candidate" of Tr_i if Tr_c satisfies spatial-temporal constraints with Tr_i.

Suppose that Tr_i appears earlier than Tr_c. The *temporal constraint* ensures that the last node of Tr_i must appear earlier than the first node of Tr_c.

Spatial constraint ensures that the last node of Tr_i can reach the first node of Tr_c after a number of frames of potential mis-detection with the current frame rate.

Candidate set Can_i is the set of all candidates of Tr_i:

$$Can_i = \{Tr_c\} \tag{3.3}$$

Neighbour Tracklet Tr_n is a neighbour of Tr_i if Tr_n also satisfies spatial-temporal constraint with Tr_i.

Temporal constraint: Tr_n shares at least one frame with Tr_i.

Spatial constraint: The 2D distance of both tracklets is below a predefined threshold. This threshold is determined by a radius of circle covering the last position of Tr_i and computed based on the width of Tr_i's last node: $width_i$. In upcoming proposed algorithms in the manuscript, the threshold θ is constantly defined by:

$$\theta = 3 \times width_i \tag{3.4}$$

Neighbour set $Neib_i$ is the set of all neighbours of Tr_i:

$$Neib_i = \{Tr_n\} \tag{3.5}$$

3.2 Features

In MOT, object features characterize an object and is extracted from one object region or accumulated from object regions in a period of time. Features can be color, gradient, 2D or 3D information, CNN or a combination of them.

Relying on the way to compute features, we categorize MOT features into two types: node and tracklet features. *Node features* are extracted from a detection bounding-box where *tracklet features* are obtained via accumulated node features within tracklet time-span.

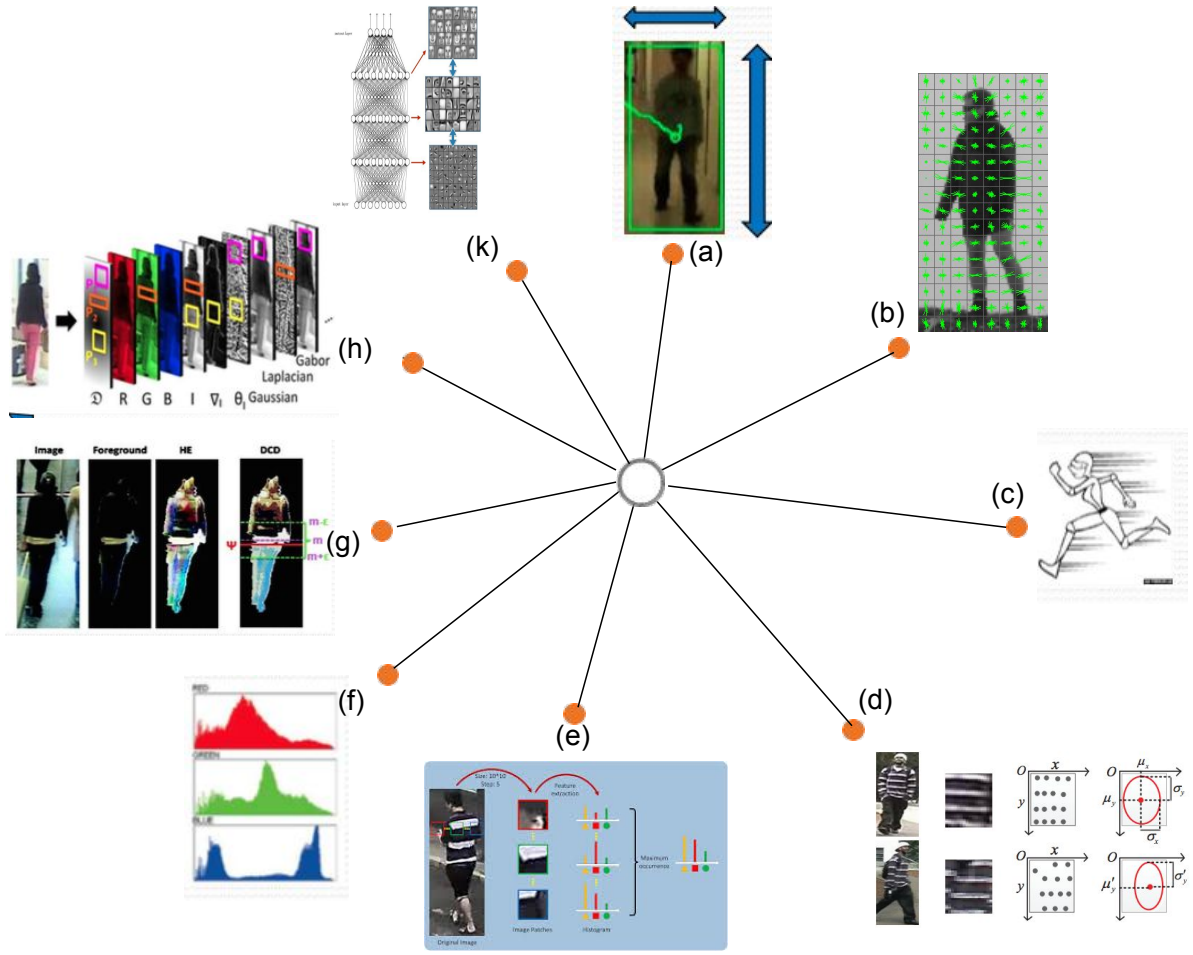


Figure 3.1: Individual feature set (a) 2D information, (b) HOG, (c) Constant velocity, (d) MCSH, (e) LOMO, (f) Color histogram, (g) Dominant Color, (h) Color Covariance, (k) Deep feature.

3.2.1 Node features

A node feature f_i^t is defined as the information characterizing an object at time t: O_i^t. The feature pool gathering features $f_i^t s$ to describe object O_i^t is presented by F_i^t and divided into 2 categories $F_i^t = \{F_i^{O^t}, F_i^{OE^t}\}$ where $F_i^{O^t}$ is the individual features which represent the individual information of an object and $F_i^{OE^t}$ is the surrounding features which characterize the surrounding context around each object. The list of node features selected by the proposed approaches in the manuscript is described in detail. Beside that, their advantages and limitations are discussed as well.

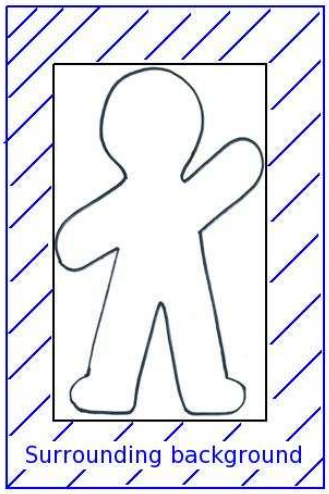


Figure 3.2: Illustration of the object surrounding background.

3.2.1.1 Individual features

Individual feature set $F_i^{O^t}$ consist of features that are computed using only the data extracted from node O_i^t. These features characterize the individual information of a node. Figure 3.1 illustrates the individual features we use in our proposed approaches which are presented in upcoming chapters.

- ***2D information*** (shown in figure 3.1(a)): Let W and H be the width and height of the 2D bounding box of a node. The 2D shape ratio, 2D area of this node are respectively defined as W/H and $W \times H$. The limitation of this feature is that its reliability depends on the detection quality. Once no occlusion occurs and a node is well detected, the shape ratio and area information of a node within a temporal window is independent from the lighting and contrast conditions.

- ***HOG - Histogram of Oriented Gradient*** **[26]** (shown in figure 3.1(b)). The essential thought behind the HOG feature is that local object appearance and shape within an image can be described by the distribution of intensity gradients and edge directions. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The feature is the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

(a)

(b)

(c)

	Occlusion	Density	Contrast
(a)	0.35	1	0.6
(b)	0.6	2	0.5
(c)	0.7	4	0.35

Figure 3.3: Surrounding feature set including occlusion, mobile object density and contrast. The detection of object O_i^t is colored by red, outer bounding-box (OBB) is color by black and neighbours are colored by light-green.

- ***Constant velocity*** (shown in figure 3.1(c)). The objects can move with or without stable velocity and direction. The *constant velocity model* describes a movement with stable velocity and direction while *Brownian motion* describes a movement characterizing with random direction and velocity. Depending on the video context and object movement property, either *constant velocity model* or *Brownian model* is applied to describe an object movement. Our proposed approaches track only people and we suppose that people walk with nearly constant velocity in a small time interval. Therefore, we used the *constant velocity model* in [25] to characterize the object motion. Motion model of an object at time t, O_i^t, is computed based on the position and displacement of node O_i^t and localized object's positions in previous frames $O_i^k s(k < t)$. This feature is useful for discriminating objects that have similar appearances but discriminative motions.

- ***MCSH* - Multi-Channel Spatio-Histogram** [126] (shown in figure 3.1(d)). The spatio histograms are first accumulated on multiple image regions among multiple colour channels (Y, Cb, Cr, H, S, nR, nG, nB) from YCrCb, HSB and normalized RGB color spaces. As revealed in figure 3.1(d), the spatial information of y axis exhibits much better intra-class invariance than x axis due to viewpoint or pose variations. Then the devised feature decomposes the spatio-histogram into three vectorized parts, including multi-channel colour histograms, the first and the second order spatial information (i.e. the mean the standard deviation vector) of y axis.

  Therefore, the spatiogram for an image region R with B color bins is defined as follow:

$$\mathbf{S}_R(b) =< \hat{n}_b, \mu_{by}, \Sigma_{by} >, b = 1, 2, ...B \tag{3.6}$$

$$\mathbf{S}_R = \{\mathbf{h} = \{\hat{n}_b\}; \mu_y = \{\mu_{by}\}; \Sigma_y = \{\Sigma_{by}\}, b = 1, 2, ...B\} \tag{3.7}$$

$$\mu_{by} = \frac{1}{n_b}\sum_{k=1}^{N} y_k \delta_{kb}, \Sigma_{by} = \sqrt{\frac{1}{n_b}\sum_{k=1}^{N}(y_k - \mu_{by})^2 \delta_{kb}} \tag{3.8}$$

where $\hat{n}_b$ is a normalized histogram of bin b, μ_{by} is a spatial mean vector and Σ_{by} is a spatial covariance matrix of y axis, N is the total pixel number of region R, n_b is the count of pixels whose value belonging to $b-th$ bin.

- ***LOMO - Local Maximal Occurrence Representation [63]*** (shown in figure 3.1(e)). The LOMO feature analyzes the horizontal occurrence of local features including features HSV and SILTP (Scale Invariant Local Ternary Pattern) [62] - SILTP is an extension of LBP for handling illumination variations. In particular, sliding windows with a sub-window size 10×10 sliding with an overlapping step of 5 pixel is located in each person region. Within each sub-window, two scales of SILTP histogram and an $8\times8\times8$-bin joint HSV histogram are extracted. Each histogram bin represents the occurrence probability of one pattern in a sub-window. Then, the maximum values of local occurrence of each pattern(i.e the same histogram bin) among these sub-windows at the same horizontal location is achieved to generate a stable representation against viewpoint changes. Besides, this feature applies the Retinex transform [55] which aims at producing a color image that is consistent to human observation of the scene. The restored image usually contains vivid color information, especially enhanced details in shadowed regions. Therefore, LOMO feature also deals with illumination variation issues.

- ***Color histogram*** (shown in figure 3.1(f)) Due to its rapid calculation, efficiency and effectiveness in characterizing objects when the scene lighting condition is good or the image has high resolution, a RGB color histogram of moving pixels is one of the most important appearance features used in object tracking.

  First for each node O_i^t , we compute a normalized histogram of b bins in each channel $C \in \{R, G, B\}$, denoted $H^C_{O_i^t}(k)$(k = 1..b), represents the percentage of occurrence of moving pixels whose color belongs to bin k:

$$\sum_{k=1}^{b} H^C_{O_i^t}(k) = 1 \tag{3.9}$$

- ***Dominant color (DC)*** (shown in figure 3.1(g)): Dominant color is a compact and efficient feature which employs representative colors to characterize the color information in the interesting region of an image. Dominant color feature is suitable for representing local features of images and can be used for quick retrieval in large image databases. This feature has been proposed by MPEG-7 [120]. This feature takes into account only the main colors of the considered node. DC feature of one node is defined as $F = \{\{c_i, p_i\}, i = 1..C\}$ where C is the total number of dominant color bins in the considered node's image region, c_i is a 3-dimensional RGB color vector, p_i is its relative occurrence percentage, with $\sum_{i=1}^{C} p_i = 1$. If dominant color feature which uses a few representative colors to characterize the color information of an image is adopted, the image feature databases size and the time of features matching process will be reduced.

- ***Color covariance*** (shown in figure 3.1(h)) is a very useful feature to characterize the appearance of a node. In particular, the color covariance matrix enables to compare regions of different sizes and is invariant to identical shifting of color values. This becomes an advantageous property when objects are tracked under varying illumination conditions. In [104], for a pixel i in a given image region R, the authors define a vector $\vec{f_i}$ including 11 sub-features :

$$\vec{f_i} = \{x, y, R_{xy}, G_{xy}, B_{xy}, M_{xy}^R, O_{xy}^R, M_{xy}^G, O_{xy}^G, M_{xy}^B, O_{xy}^B\} \tag{3.10}$$

  where x,y are pixel locations, R_{xy} , G_{xy} , and B_{xy} are RGB channel values at position (x, y) ; M and O correspond to gradient magnitude and orientation in each channel at position (x, y). The covariance of region R is characterized by a matrix $C_{R(11x11)} \in \mathbb{R}$:

$$C_R = \frac{1}{n-1} \sum_{i=1}^{n} (\vec{f_i} - \vec{\mu_R})(\vec{f_i} - \vec{\mu_R})^T \tag{3.11}$$

  where n is the number of pixels in region R ; $\vec{\mu_R}$ is a vector of 11 dimensions representing the mean values of the 11 sub-features of all points in the region R; $\vec{f_i}$ is the sub-feature vector of point i, defined in formula 3.10.

- **Deep feature** is extracted from the feature map in convulational layer 4 of modified-VGG16 model. How to extract deep feature is presented in details in chapter 6.

3.2.1.2 Surrounding features

Surrounding feature set $F_i^{OE^t}$ includes features that are computed based on the interaction of a tracklet with its surrounding background. Let $A_i^t = \{C_i^t, W_i^t, H_i^t\}$ be the 2D bounding-box of node O_i^t (tracklet Tr_i at time t) where C_i^t, W_i^t, H_i^t are its 2D center, width and height, respectively. We define an outer bounding-box (OBB) of node O_i^t: $A_i^+ = \{C_i, W_i + \alpha W_i, H_i + \alpha H_i\}$ where α

is a predefined value in interval [0,1]. The surrounding background illustrated in figure 3.2 is defined as $A_i^{sur} = A_i^+ / A_i$. The surrounding features of object O_i^t are described via figure 3.3 where the given object is marked by red bounding-box and the OBB is marked by the black one, three images are extracted from moments t_1, t_2, t_3, respectively.

The ***surrounding object set*** $Surr_i^t$ of a node O_i^t at time t are defined as objects appearing in OBB of O_i^t.

- ***Mobile object density:*** Mobile object density is computed by the number of surrounding objects inside the OBB of the given node O_i^t.

$$Sd_{O_i^t} = |Surr_i^t| \tag{3.12}$$

where $|Surr_i^t|$ corresponds to the size of $Surr_i^t$ set.

- ***Occlusion:*** The occlusion level of a node O_i^t is computed by the mean of the O_i^tś area covered by other surrounding objects.

$$So_{O_i^t} = min(\frac{\sum_{k=1}^{Surr_i^t} o_{O_i^t}^k}{|Surr_i^t|}, 1) \tag{3.13}$$

$o_{O_i^t}^k$ is the occlusion level of O_i^t and its surrounding object O_k^t and is computed as follow:

$$o_{O_i^t}^k = \frac{A^{i,k}}{O_i^t} \tag{3.14}$$

where $A^{i,k}$ is the overlapped area of O_i^t and O_k^t. The value of occlusion level $So_{O_i^t}$ is in the range $< 0, 1 >$, 0 is non-occluded and 1 is full-occluded.

- ***Contrast:***The contrast of a node O_i^t is defined as the color intensity difference between the image region of O_i^t and its surrounding background localized by OBB.

$$Sc_{O_i^t} = 1 - \frac{\sum_c^C simil(H_{O_i^t}^c, H_{OBB}^c)}{3} \tag{3.15}$$

where $simil(H_{O_i^t}^c, H_{OBB}^c)$ is the color intensity similarity between node O_i^t and its OBB in chanel c and defined by:

$$simil(H_{O_i^t}^c, H_{OBB}^c) = \sum_{k=1}^{b} min(H_{O_i^t}^c(k), H_{OBB}^c(k)) \tag{3.16}$$

3.2.2 Tracklet features

A tracklet feature is the accumulation of the node features within the tracklet time-span. Therefore, based on the node feature categories, we also define the tracklet feature pool by F_i which includes $F_i = \{F_i^O, F_i^{OE}\}$ where F_i^O and F_i^{OE} are the individual and the surrounding tracklet feature sets, respectively.

The tracklet features of tracklet Tr_i are extracted and accumulated from features of nodes O_i^t where $t \in < m, n >$ and are presented in detail in each upcoming chapters.

3.3 Tracklet functions

This section lists two functions, which initialize and generate the reliable tracklets as well as interpolate object trajectories, applied in all proposed algorithms in the manuscript. Thank to such pre-or-post processing functions, the tracking performance is improved.

3.3.1 Tracklet filtering

The performance of a short-term tracker is affected by the quality of detection while the performance of a long-term tracker is affected by the quality of input tracklets. Therefore, tracklet filtering is an incremental step for MOT by refining the unreliable tracklets for a long-term tracker's input to improve tracking performance. A tracklet is considered as reliable if it has a smooth trajectory as well as consistent representation, has a size long enough and is not ambiguous with other tracklets. Based on this hypothesis, the proposed tracklet filtering method refines unreliable tracklets based on four following processes.

- **Node anomaly filtering** consists in detecting a node belonging to a tracklet whose features are not consistent compared to other nodes. In all approaches in the manuscript, we use the 2D and color information to determine the node anomalies. In particular, the distance between 2 node positions in two consecutive frames is larger than threshold or the object color changes remarkably in 2 consecutive frames. If any anomaly is detected, this node is removed from the tracklet.

- **Noise filtering:** If a tracklet is too short, it is considered as a noise and is removed. In all our proposed approaches, this value is set to three frames.

- **Node ambiguity filtering:** A tracklet is defined as ambiguous with other tracklets if any node belonging to this tracklet is strongly occluded by other objects. The occlusion level is described by occlusion feature in node's surrounding feature set. If the occlusion level of a node is higher than the threshold, this node is removed from tracklet.

- **Tracklet segmentation:** After two processes including node anomaly filtering and node ambiguity filtering, some nodes belonging to a tracklet are removed. If these nodes are consecutive and the number of these consecutive nodes is higher than a threshold, the tracklet will be segmented at before and after removed nodes. In all our proposed approaches, this threshold is set to five frames.

3.3.2 Interpolation

An object trajectory may miss some nodes. It happens if an object is mis-detected in some frames and the tracking algorithm finds a correct matching when the object reappears. Missed-nodes lead to miss object information to represent tracklets. In order to enhance the tracklet reliability, data interpolation is a necessary step to fill the missed information.

If a tracklet has more than five consecutive missing nodes, the tracklet is considered as unreliable and is segmented by the tracklet filtering step. Otherwise, in order to fill the missing nodes, linear interpolation is performed using the feature pools of the two nodes located just before and just after the missing nodes.

$$\exists t \in < a, b >: f_i^t = f_i^a + (t - a)\frac{f_i^b - f_i^a}{b - a} \tag{3.17}$$

Due to the assumption that a new tracklet is created if more than five consecutive nodes are missing, there is no need to use a more elaborated and time consuming method to fill the missing nodes. Considering this assumption and the fact that the interpolation module is used at every frame, each tracklet contains no empty nodes.

3.4 MOT Evaluation

Metrics and datasets play a significant role to evaluate the performance of any MOT algorithm. In this section, we list metrics and publicly available datasets used to compare our proposed approaches with the state-of-the-art MOT algorithms to verify their robustness. Moreover, some issues which may result in unfair comparison are discussed here.

3.4.1 Metrics

Metrics of MOT approaches provide a standard evaluation for fair quantitative comparison. In this section, we present a brief review on a variety of MOT evaluation metrics including CLEARMOT metrics and completeness metrics which are summarized in Table 3.1.

- **CLEARMOT metrics** consisting of multiple metrics and follow publicly provided toolkit on MOTchallenge website for fair comparison with other approaches. The multiple object tracking precision (MOTP↑) evaluates the intersection area over the union area of

Metric	Description	Note
MOTA	Multiple Object Tracking Accuracy[1]. This measure combines three error sources: false positives, missed targets and identity switches	↑
MOTP	Multiple Object Tracking Precision [1]. The misalignment between the annotated and the predicted bounding boxes	↑
MT	Mostly tracked targets. The ratio of a ground-truth trajectory that is covered by a track hypothesis for at least 80% of their respective life span	↑
ML	Mostly lost targets. The ratio of a ground-truth trajectory that are covered by a track hypothesis for at most 20% of their respective life span	↓
FP	The total number of false positives	↓
FN	The total number of false negatives (missed targets)	↓
ID Sw	The total number of identity switches. Please note that we follow the stricter definition of identity switches as described in [2]	↓
Frag	The total number of times a trajectory is fragmented (i.e. interrupted during tracking)	↓

Table 3.1: The evaluation metrics for MOT algorithm. ↑ represents that higher scores indicate better results, and ↓ denotes that lower scores indicate better results.

detection bounding boxes. The multiple object tracking accuracy (MOTA↑) calculates the accuracy composed of false negatives (FN↓), false positives (FP ↓), and identity switching (IDS↓).

- **Completeness metrics:** Metrics for completeness indicate how well the ground truth trajectories are tracked. These metrics include (MT↑) - the ratio of mostly tracked trajectories (if a ground-truth trajectory is covered by a tracking output for at least 80% of their life-span), (ML ↓) - the ratio of mostly lost trajectories (if a ground-truth trajectory is covered by a tracking output for at most 20% of their life-span) and (FG↓) - the number of track fragments.

3.4.2 Datasets

In MOT evaluation, publicly available datasets are employed to evaluate and compare MOT performances. There are many such datasets experimented by the state-of-the-art trackers. However, we here summarize the most popular and benchmark datasets with which we evaluate and compare our proposed approaches with some state-of-the-art trackers in upcoming chapters.

- **PETs2009-S2_L11** sequence has 794 frames containing 21 people with many occlusions

and people moving with different directions.

- **PETs2015** with W1_ARENA_Tg_TRK_RGB_1 sequence has 240 frames. There are only few people but their size and pose variant throughout time.

- **TUD dataset** includes TUD_Stadtmitte and TUD_Crossing sequences. Both sequences are quite short, with more or less 200 frames, but they contain challenges for trackers such as low light intensity, crowded environment, frequent occlusions and similar object appearances.

- **ParkingLot**: The main challenge of this dataset is occlusion and confusion caused by targets walking together with similar appearance. We choose Parkinglot1 sequence including 14 people in 998 frames for testing because of the availability of detection and Groundtruth bounding-boxes on UCF website [1].

- **MOT2015** consists of 22 sequences, divided into training and testing sets (shown in Figure 3.4 and Figure 3.5). The testing data includes 11 sequences, 5783 frames with 721 people. This dataset shows the diversity of outdoor scenarios with strong and frequent person-person occlusions, people moving with random directions captured by fixed or moving narrow angle cameras, crowded environment (two sequences have 197 and 226 people, respectively). Among the 22 sequences, there are seven new challenging high-resolution videos (ADL-Rundle-6, ADL-Rundle-8, Venice-2, AVG-TownCentre, ADL-Rundle-1, ADL-Rundle-3 and Venice-1), four captured from a static and two from a moving camera held at pedestrian's height. Three of them are particularly difficult: a night sequence from a moving camera (ADL-Rundle-8) and two outdoor sequences with a high density of pedestrians (PETS09-S2L2, ADL-Rundle-1). The moving camera together with the low illumination create a lot of motion blur, making this sequence extremely challenging.

- **MOT17** contains 14 challenging video sequences (7 sequences for training, 7 remaining ones for testing) in unconstrained environments captured with both static and moving cameras (shown in Figure 3.6 and Figure 3.7). This benchmark provides the detections for all sequences produced by three well-known detectors: DPM, SDP and FRCNN. Therefore, in total, the number of training and testing sequences triples: 21 sequences for training and 21 sequences for testing. All sequences have been annotated with high accuracy, strictly following a well-defined protocol. Compared to MOT15, this dataset has higher difficulty and more challenges, e.g. by having scenarios with a 3-folds higher mean density of pedestrians (MOT17-04, MOT17-03, MOT17-07). Aside from pedestrians, the

[1] http://crcv.ucf.edu/data/ParkingLOT/

objects also include other classes like vehicles, bicycles etc. (MOT17-01, MOT17-02, MOT17-03, MOT17-04, MOT17-05, MOT17-06, MOT17-10, MOT17-13 and MOT17-14) in order to provide contextual information for methods to explore.

The video sequences and the public detection of benchmarks MOT15, MOT17 are available on MOTChallenge website. [2]

The MOT evaluation metrics as well as datasets are public to compare new approaches. However, there are some issues which may result in unfairness in case of direct comparison among different approaches on the same dataset. In the last section of this chapter, we list such issues that we face in the experiments. These issues are also discussed in [66].

3.4.3 Some evaluation issues

- Different methodologies. For example, some publications belong to offline methods while others belong to online ones. Due to the difference between online and offline tracking described previous chapter, it is unfair to directly compare them.
- Different detection hypotheses. Some approaches adopt different detectors to obtain detection hypotheses as input. One approach based on different detection hypotheses would output different results, comparing approaches with different inputs is also unfair.
- Some approaches utilize detections from multiple views while some approaches adopt information from a single view. This makes the comparison between them difficult.
- Prior information, such as scenario structure and the number of pedestrians, are employed by some approaches. Direct comparison between these approaches and others is not so convincing.

[2] https://motchallenge.net/

Figure 3.4: Training video sequences of MOT15 dataset.

Figure 3.5: Testing video sequences of MOT15 dataset.

Figure 3.6: Training video sequences of MOT17 dataset.

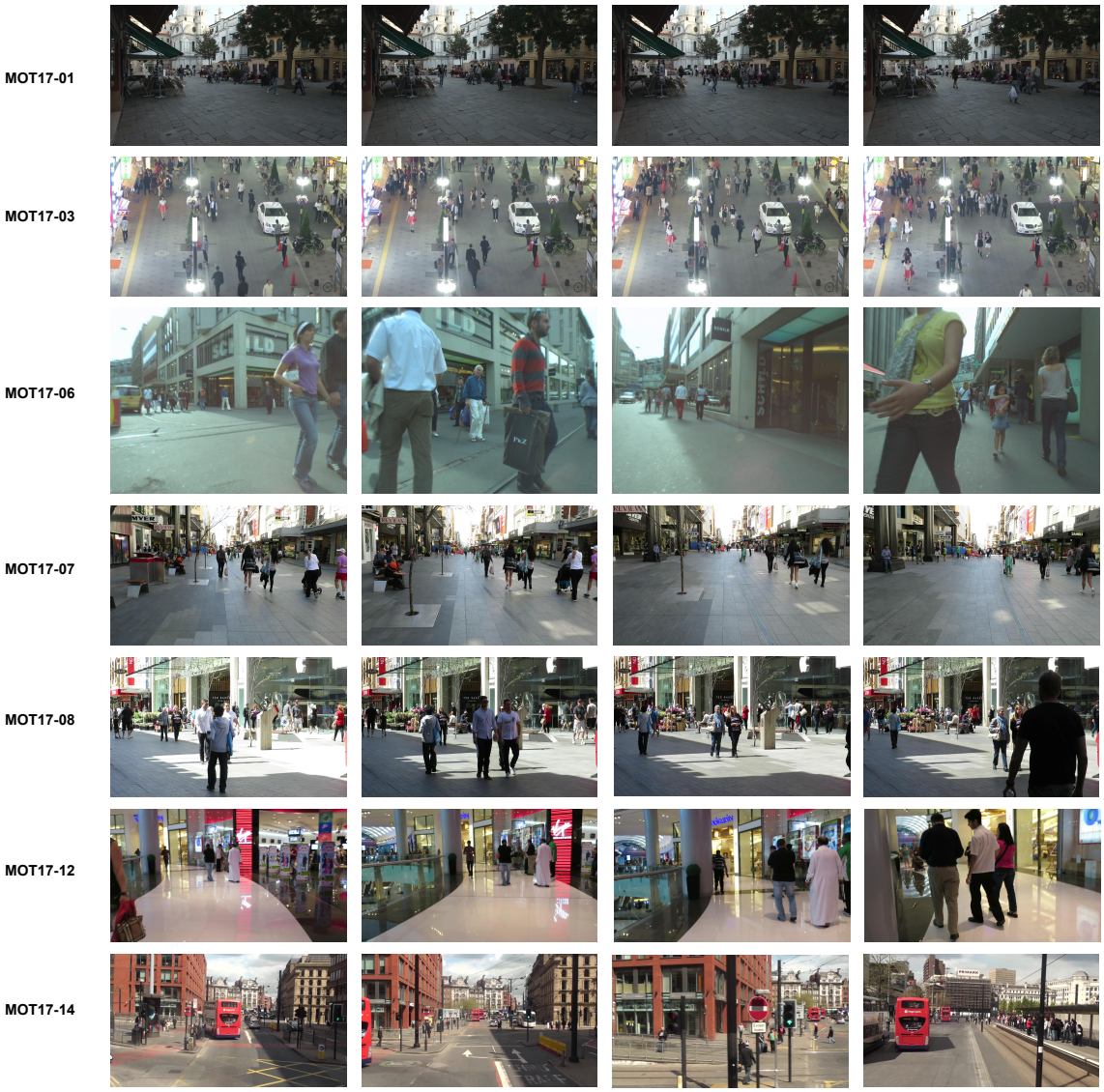


Figure 3.7: Testing video sequences of MOT17 dataset.

4

MULTI-PERSON TRACKING BASED ON AN ONLINE ESTIMATION OF TRACKLET FEATURE RELIABILITY [80]

4.1 Introduction

Multi-object tracking (MOT) has been one of the fundamental problems in computer vision, essential for lots of applications (e.g home-care, house-care, security systems, etc.). The main objective of MOT is to estimate the states of multiple objects while identifying these objects under appearance and motion variations throughout the time. This problem becomes more challenging to multi-person tracking due to frequent occlusion by background or other people, person pose as well as illumination variation, etc. These challenges make person's part or full invisible as well as person appearance change remarkably. Besides that, person mis-detection caused by a detector also remarkably affects to tracking quality. Therefore, finding the discriminative features to characterize a person under scenes (person pose and illumination variations) challenges state-of-the-art trackers.

The first group of approaches [20, 123] proposed to use a pool of powerful features to characterize objects in different video scenes. In order to automatically adapt the tracker to a video scene, these approaches have a controller to select powerful features to discriminate objects overtime. However, these trackers select these features extracted from object detections in every frame (node features) which are sensitive to noise. The second group of approaches [90, 111, 39, 106] formalize the MOT problem as a graph and focus on the optimization prob-

lem in the data association process on this graph to achieve a high tracking performance. These trackers work in the offline mode which needs a huge beforehand detection requirement. Furthermore, the computation cost of global optimization may increase exponentially depending on the number of objects in the scene.

In this chapter, we propose a long-term multi-person tracker named *reliable feature estimation* (RFE) which extracts the features representing a person over its short trajectory called tracklet features. Then, $RFE-Tracker$ links tracklets before and after mis-detection based on the most reliable tracklet features to characterize people in a given video scene. The proposed approach belongs to the first group but using tracklet features instead of node features. In order to select tracklet features, each tracklet feature is set a weight which represents for the reliability of a feature to discriminate the tracklet in a particular video scene. The proposed approach brings two following contributions:

- A simple but effective method which links tracklets based on reliable tracklet features. These features are selected by automatically tuning the corresponding weights to adapt the tracker to the change of video scene. No training process is needed which makes the algorithm generic and deployable to a variety of MOT frameworks.
- A flexible combination of appearance features and motion model to improve tracking quality.

The rest of this chapter is organized as follows: Section 4.2 presents some related works from the state-of-the-art which are proposed to handle the MOT problems such as occlusion, person information variation caused by changes of video conditions, and recover person trajectory from a limited number of mis-detections. Section 4.3 presents the proposed long-term tracking algorithm. Section 4.4 shows brief evaluation results as well as analysis about the performance of the proposed approach. Finally, conclusions are summed up in section 4.5.

4.2 Related work

In this section, we will discuss some online MOT methods from the state-of-the-art proposed to achieve strong object features to discriminate an object in a particular video scene. The discussed methods are categorized into two following groups.

Short-term trackers learn online the discriminative appearance model of an object corresponding to the video scene in every frame. Authors in [20] select discriminative object features via automatically tuning tracker parameters where each parameter controls one features. The more the feature can discriminate objects to each other, the higher value of according parameter is set. Otherwise, the parameter is set to a low value. The approach in [8] tracks

multi-objects by using the tracklet confidence with an automatic discriminative object appearance model which is learned based on an incremental linear discriminant analysis (ILDA). This allows the proposed tracker to distinguish each object to others thanks to the learned object appearance model. Then, the learned object appearance model is also incrementally updated with frame-to-frame tracking results.

However, object features computed by a short-term tracker are unreliable if the detected objects on each frame are noise. In this case, the short-term trackers can fail.

Recently, some researches focus on the second tracking group - *long-term tracking* whose objective is to link short trajectories (tracklets) to create more completed object trajectories. Because tracklet features computed based on tracklet timespan represent objects more completed than node features computed in each frame, the long-term trackers can overcome the above mentioned disadvantages of the short-term tracking. The approach in [6] proposes an algorithm that recovers object trajectory before and after mis-detection by linking segmented tracklets using enhanced covariance-based signatures and an online threshold learning. To gain the object signature of each tracklet, reliable nodes of this tracklet called *key-frames* are extracted, then the signature based on Mean Riemannian Covariance Grid(MRCG) descriptor [91] on these extracted key-frames are generated. The authors in [111] propose a tracking algorithm using the structure of a hierarchical relation hyper-graph. Then the proposed tracker formulates MOT task as a hierarchical dense neighborhoods search problem. In each layer, tracklets are grouped into a dense neighborhoods whose members have high mutual affinity, then these tracklets are linked to form the longer ones. The grouping process finishes when no dense neighborhoods is found out. These long-term tracking algorithms using the object information extracted from a tracklet timespan which are more reliable than extracted from a frame. So, the long-term trackers can gain a better tracking performance than short-term trackers. However, the mentioned long-term trackers more focus on solving MOT task by proposing an optimization algorithm of object information affinity than automatically generating an object representation which can be changed according to the change of video scene. If the video scene is complex and frequently changes, these long-term trackers can fail.

Therefore, we propose a new long-term tracking algorithm which both extracts object features on tracklet timespan and selects the robust features to adapt tracker to the change of video scene. In next section, we will present in detail this proposed approach.

4.3 The proposed approach

Person appearance could change and be mixed with other people after occlusion, mis-detection or people leave and come back in the video scene. Therefore, the objective of the proposed approach is to recover mis-detection and to overcome the occlusion by correctly link-

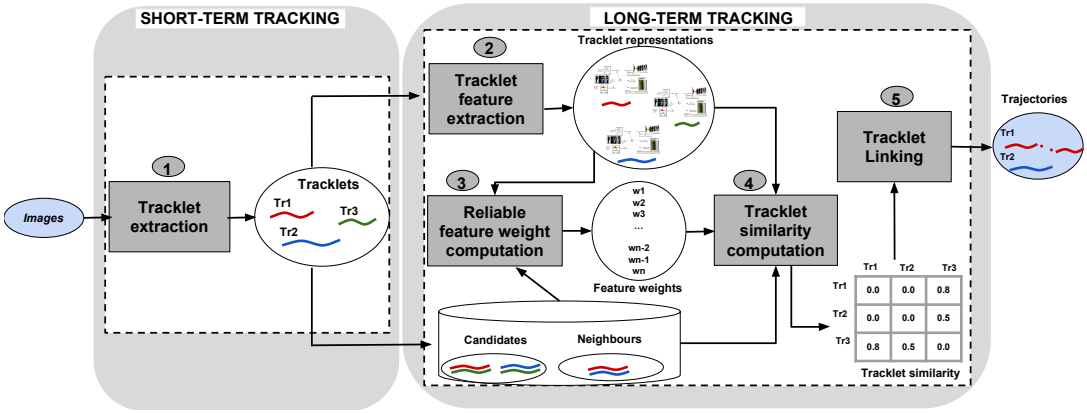


Figure 4.1: The overview of the proposed algorithm.

ing tracklets into a complete person trajectory. To achieve this objective, a robust tracker is proposed to link tracklets based on automatically selecting discriminative features which distinguish ambiguous tracklets to each other.

4.3.1 The framework

The overview of the proposed approach is shown in figure 4.1. It highlights all online steps done in a comprehensive tracking framework. The framework includes two main steps where the short-term tracking is responsible for generating tracklets and the long-term tracking is responsible for extracting the tracklet features to represent a person and selecting discriminative features for correctly tracklet linking. The short-term tracker extracts tracklets in two consecutive time-windows, including the current time-window Δ_t and the previous one Δ_{t-1} (block 1). The purpose of this step is to enable the long-term tracker to later link tracklets appearing in two consecutive time-windows. These tracklets then are smoothed by the filter as well as interpolation methods presented in chapter 3 to achieve reliable tracklets for the input of the long-term tracker.

The proposed long-term tracker processes in each time window Δ_t. We first determine the relationship for each tracklet including "Candidates" and "Neighbours" defined in chapter 3. We extract tracklet representations (block 2) (presented in section 4.3.2) and then select the reliable features to link a given tracklet with its candidate. These features must discriminate this tracklet to its neighbours but still make its distance with its candidate close. The discriminative power of a feature is represented by a weight. This weight is automatically computed (section 4.3.4) based on the feature distance (presented in section 4.3.3) between the given tracklet with its relationship (including candidates and neighbours) (block 3). Based on these weighted features, the tracklet similarities between tracklets are computed (block 4). A similar-

ity matrix is created where each cell is the similarity between a pair of tracklet representations according to two tracklets. Finally, tracklets are linked with their best candidates after optimizing the similarity matrix using Hungarian algorithm (section 4.3.5) (block 5). Each part of the proposed framework is described in detail as follows.

4.3.2 Tracklet representation

Through the whole manuscript, we use the same definition of tracklet presented in Chapter 3:

$$Tr_i = \{O_i^m, O_i^{m+1}, ..., O_i^{n-1}, O_i^n\} \tag{4.1}$$

Each tracklet is described by its representation defined as a set of reliable tracklet features. Tracklet Tr_i is represented by reliable tracklet features $\{(W_i^k, F_i^k)\}$ which is defined as follow:

$$\nabla_{Tr_i} = \{(F_i^k, W_i^k)\} \tag{4.2}$$

where F_i^k is the tracklet feature k used to present tracklet Tr_i and W_i^k is the weight presenting the feature's reliability which is estimated online based on the discriminative power of the tracklet feature. Given a chain of nodes O_i^t where $t \in < m, n >$ belonging to tracklet Tr_i, each tracklet feature $F_i^k \in \nabla_{Tr_i}$ is computed based on the according nodes feature $F_i^k(t)$.

The diversity of object features plays a crucial role in characterizing people in different video scenes. In this approach, we propose to select the following features $F_i^k s$ from the tracklet feature pool F_i discussed in chapter 3 to describe a tracklet Tr_i. These features include 2D information features (2D shape ratio and 2D area), color based features (color histogram, dominant color and color covariance) and motion feature (constant velocity model).

The definition, the advantages as well as disadvantages of each feature are discussed and presented in chapter 3. In the following, we present the tracklet feature similarities which are utilized later to estimate tracklet feature reliability in a particular video scene.

4.3.3 Tracklet feature similarities

The tracklet representation is defined as a set of weighted tracklet features. Therefore, the similarity of two tracklet representation is the combination of weighted tracklet feature similarities. Firstly, these tracklet feature similarities are computed as follows.

2D shape ratio, 2D area and Motion model similarities

Features including 2D shape ratio, 2D area and Constant velocity of each tracklet are represented by Normal Gaussian distribution $F_i^k \simeq G(\mu_i^k, \sigma_i^k)$ whose μ_i^k is the weighted mean and σ_i^k the weighted standard deviation of tracklet feature F_i^k over time t. These values are computed as follows:

$$\mu_i^k = \frac{\sum_{t=m}^{n} w(t) * F_i^k(t)}{\sum_{t=m}^{n} w(t)} \tag{4.3}$$

$$\sigma_i^k = \sqrt{\frac{\sum_{t=m}^{n} w(t) * (F_i^k(t) - \mu_i^k)^2}{\sum_{t=m}^{n} w(t)}} \tag{4.4}$$

where $w(t)$ is the weight function which is used to decrease the impact of "interpolated features" while relying on the directly extracted features from node. "Interpolated feature" is defined as the feature extracted at the interpolated nodes which are estimated by linear interpolation function presented in Chapter 3. The weight function is defined by:

$$w(t) = \begin{cases} w_I & \text{if } F_i^k(t) \text{ is interpolated} \\ w_R & \text{if } F_i^k(t) \text{ is directly extracted} \end{cases}$$

w_R and w_I satisfy:

$$\begin{cases} Nb_R * w_R + Nb_I * w_I = 1 \\ w_I = \alpha * w_R \end{cases}$$

and $F_i^k(t)$ stands for feature F_i^k of node O_i^t, α is a coefficient which determines the reliability of interpolated features compared to directly extracted features. Given that Nb_I and Nb_R are numbers of interpolated nodes and real tracked nodes, correspondingly, α is determined by $\alpha = \frac{Nb_I}{Nb_I + Nb_R}$. It means that the importance of interpolated features is directly proportional to the ratio of interpolated nodes over the tracklet's length.

Upon that, we propose to use Kulback-Leibler divergence [105], a measure of the difference between two probability distributions to compute the distance of these tracklet features:

$$d(F_i^k, F_j^k) = log(\frac{\sigma_j^k}{\sigma_i^k}) + \frac{{\sigma_i^k}^2 + (\mu_i^k - \mu_j^k)^2}{2 \times {\sigma_j^k}^2} - 0.5 \tag{4.5}$$

where (μ_i^k, σ_i^k) and (μ_j^k, σ_j^k) are normal Gaussian distributions for each mentioned feature k of tracklet Tr_i and Tr_j, respectively.

The similarity score between two tracklet features F_i^k and F_j^k are computed by:

$$Simil(F_i^k, F_j^k) = exp(-d(F_i^k, F_j^k)) \tag{4.6}$$

Color Histogram similarity

There are plenty of distance measures between two histograms categorized in [70] including Hellinger distance, Euclidean distance, Chibyshev distance, Histogram intersection, Bhattacharyya distance, Quadratic distance and so on. In this work, we use a metric based on histogram intersection [100] due to its low time consuming computation. The similarity score

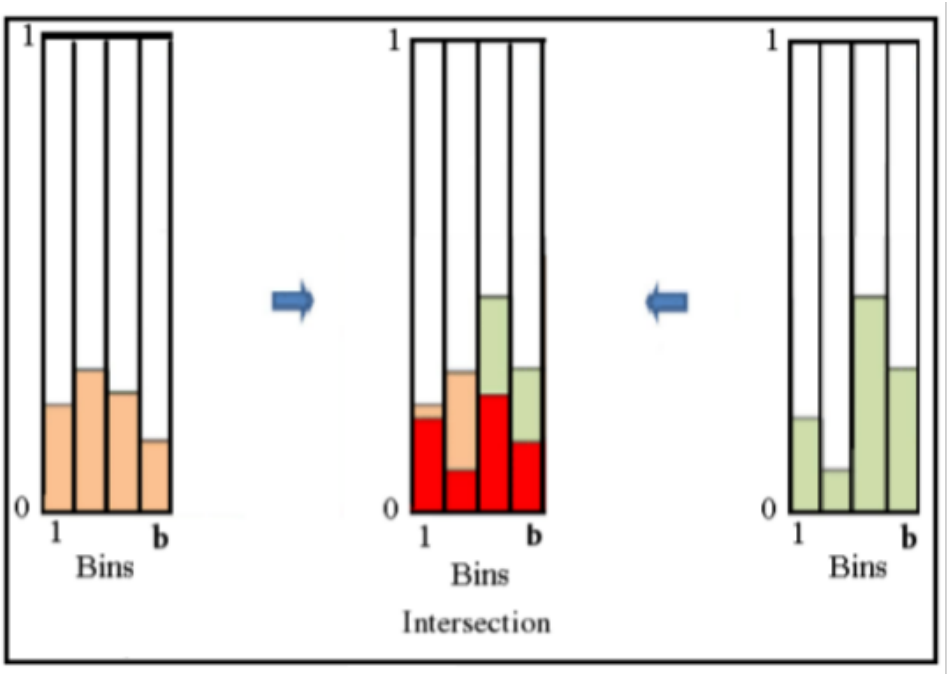


Figure 4.2: Illustration of a histogram intersection. The intersection between left histogram and right histogram is marked by red color in the middle histogram.

$Simil(H_i^c, H_j^c)$ between two histograms H_i^c,H_j^c of tracklet Tr_i and Tr_j for channel c is illustrated in figure 4.2 and defined as follows:

$$Simil(H_i^c, H_j^c) = \frac{\sum_{b=1}^{B} min(H_i^c(b), H_j^c(b))}{max(\sum_{b=1}^{B} H_i^c(b), \sum_{b=1}^{B} H_j^c(b))} \tag{4.7}$$

where B is bins of channel c, $H_i^c(b)$ is the mean of bin b histogram values of nodes $O_i^t s$ $(m < t < n)$ belonging to Tr_i:

$$H_i^c(b) = \frac{\sum_{t=n}^{m} h_i^{c(t)}(b)}{m-n} \tag{4.8}$$

The color histogram similarity score between two tracklets Tr_i and Tr_j is defined as the mean of three histogram similarity scores corresponding to the three channels: red, green and blue:

$$Simil(F_i^k, F_j^k) = \frac{\sum_c^C Simil(H_i^c, H_j^c)}{|C|} \tag{4.9}$$

where C is the color channel set of RGB image, including red, green and blue.

Color covariance similarity

We use the distance defined by [33] to compare two covariance matrices C_i and C_j:

$$\rho(C_i, C_j) = \sqrt{\sum_{f=1}^{\vec{f}} ln^2 \lambda_f(C_i, C_j)} \tag{4.10}$$

$$\vec{f_i} = \{x, y, R_{xy}, G_{xy}, B_{xy}, M^R_{xy}, O^R_{xy}, M^G_{xy}, O^G_{xy}, M^B_{xy}, O^B_{xy}\} \tag{4.11}$$

where $\vec{f}$ is the number of considered point sub-features ($|\vec{f}|= 11$ where x and y are pixel location, R_{xy}, G_{xy}, B_{xy} are RGB channel values and M and O corresponds to gradient magnitude and orientation in each channel, respectively), $\lambda_f(C_i, C_j)$ is the generalized eigenvalue of covariance matrix C_i and C_j, determined by:

$$|\lambda_f C_i - C_j|= 0 \tag{4.12}$$

In order to take into account the person spatial coherence and also to manage occlusion cases, we propose to use the spatial pyramid match kernel defined in [37]. The main idea is to divide the image region of the considered people into a set of sub-regions. At each level l ($l \geq 0$), each of the considered people is divided into a set of $2^l \times 2^l$ sub-regions. Then we compute the color covariance distance for each pair of corresponding sub-regions using equation 4.10 (see figure 4.3). The computation of each sub-region pair helps to evaluate the spatial structure coherence between two considered people. In the case of occlusions, the color covariance distance between two regions corresponding to occluded parts can be very high. Therefore, we take only the lowest color covariance distances (i.e. highest similarities) at each level to compute the final color covariance distance. Let $M^l_z = \{\rho^l_1, \rho^l_2, ..., \rho^l_z\}$ be the set of the z largest distances between corresponding covariance matrices at level l. The covariance distance between two people at level l is defined as follows :

$$D^l = \frac{\sum_{r=1}^{2^l \times 2^l} \rho(C^r_i, C^r_j) - \sum_m^{|M^l_z|} \rho^l_m}{2^l \times 2^l - |M^l_z|} \tag{4.13}$$

where C^r_i and C^r_j are respectively the covariance matrices of Tr_i and Tr_j at sub region r, $\rho(C^i_r, C^j_r)$ is the covariance distance between C^i_r,C^j_r defined in equation 4.10.

Then, the number of distances that are computed at each level are combined using a weighted sum. Distances computed at finer resolutions are weighted more highly than distances computed at coarser resolutions. So the distance of color covariance $d(F^k_i, F^k_j)$ between the two trackets Tr_i, Tr_j is defined as follows :

$$d(F^k_i, F^k_j) = D^L + \sum_{l=0}^{L-1}(\frac{1}{2^L - 1} D^l - D^{l+1}) = \frac{1}{2^L} D^0 + \sum_{l=1}^{L} \frac{1}{2^L - l + 1} D^l \tag{4.14}$$

where L is a parameter representing the maximal considered level ($L \geq 0$). We define the similarity score for color covariance feature between two tracklets Tr^i and Tr^j as follows:

$$Simil(F^k_i, F^k_j) = max(0, 1 - \frac{d(F^k_i, F^k_j)}{D_{cov\_max}}) \tag{4.15}$$

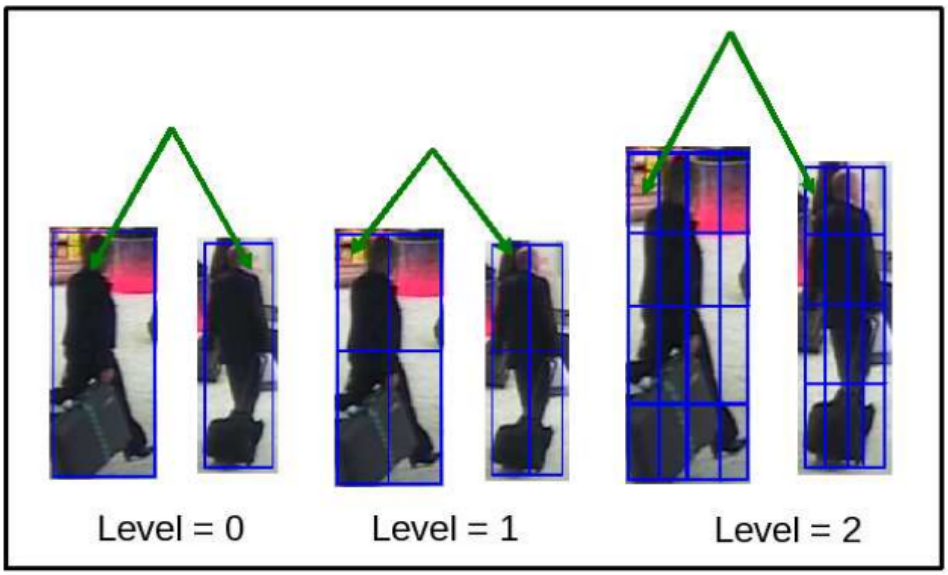


Figure 4.3: Illustration of different levels in the spatial pyramid match kernel.

where D_{cov_max} is the maximal distance for two color covariance matrices to be considered as similar. In experiment, we set D_{cov_max} to 1.5.

Dominant Color similarity

Similar to the color covariance feature, in order to take into account the spatial coherence and also occlusion cases, we propose to use the spatial pyramid match kernel for comparing the Dominant Color Feature(DCF) between two people. We divide the person into sub-regions and compute the dominant color distance between corresponding region pairs. A distance value is computed at each level l thanks to equation 4.13. Finally, the distance $d(F_i^k, F_j^k)$ between two tracklets Tr_i, Tr_j for DCF is computed similarly as equation 4.14. The DCF similarity score is defined as follows :

$$Simil(F_i^k, F_j^k) = 1 - d(F_i^k, F_j^k) \tag{4.16}$$

where $d(F_i^k, F_j^k)$ is the spatial pyramid distance of dominant colors between two considered tracklets.

After achieving tracklet feature similarities, the proposed approach automatically estimate tracklet feature similarity by computing the corresponding reliable feature weights.

4.3.4 Feature weight computation

In this approach, the feature pool F_i of tracklet Tr_i is temporarily divided into 2 types: appearance feature pool F_i^{app} and motion model F_i^{mo}:

$$F_i = \{F_i^{app}, F_i^{mo}\} \tag{4.17}$$

Although people are supposed to move with a constant velocity but they can abruptly change the movement direction. Tracking by estimating the person movement direction can fail in this case. Therefore, the proposed approach firstly tracks people based on person appearance in prior. Then, the motion feature weight is computed based on the reliability of tracket appearance features. If the appearance features are powerful to discriminate people, the appearance weights are set with higher values than the motion weight. Otherwise, the motion weight is set with a higher value (the maximum value is 0.5).

The feature weight of one tracklet must be directly proportional to the feature similarity between this tracklet and its candidate and inversely proportional to the feature similarity of this tracklet with its neighbours. Given a tracklet Tr_i and its relationship including candidate Tr_c and neighbours Tr_ns, we define a feature weight of $F_i^k \in F_i^{app}$ for this pair of tracklets (Tr_i, Tr_c) as follows:

$$\omega_{i,c}^k = \lambda^{Simil(F_i^k, F_c^k) - \tilde{M}(Simil(F_i^k, F_n^k s)) - 1} \tag{4.18}$$

$\tilde{M}()$ is the median of the similarities of feature F^k between tracklet Tr_i and its neighbours. The advantage of the median is that its value is not affected by a few of extremely anomaly values. Therefore, the median is meaningful in coding the similarity of Tr_i with its neighbours even if these similarity values are not distributed uniformly. Furthermore, the function λ^X where $X = DS(F_i^k, F_c^k) - \tilde{M}(DS(F_i^k, F_n^k s)) - 1$ returning value into [0,1] is proposed to normalize the appearance feature weight. We select $\lambda = 10$ in the experiment.

Then, the motion feature weight is computed as follow.

A combination of appearance and motion model features

Depending on how well the appearance feature weights can characterize people in the video scene, the approach proposes a new way to compute the motion model weight based on other appearance features:

$$\omega_{i,c}^m = 0.5 - 0.5 \max(\omega_{i,c}^k) \quad k \in F_i^{app} \tag{4.19}$$

By an inverse transformation in equation (4.19), we can flexibly select appearance features or motion model to track people adapting to a variation of video scenes. If appearance features are reliable enough to discriminate people, the proposed approach takes into account the appearance features more importantly than the motion model. Inversely, when people have similar appearance but different motions, the proposed tracker relies more on the motion

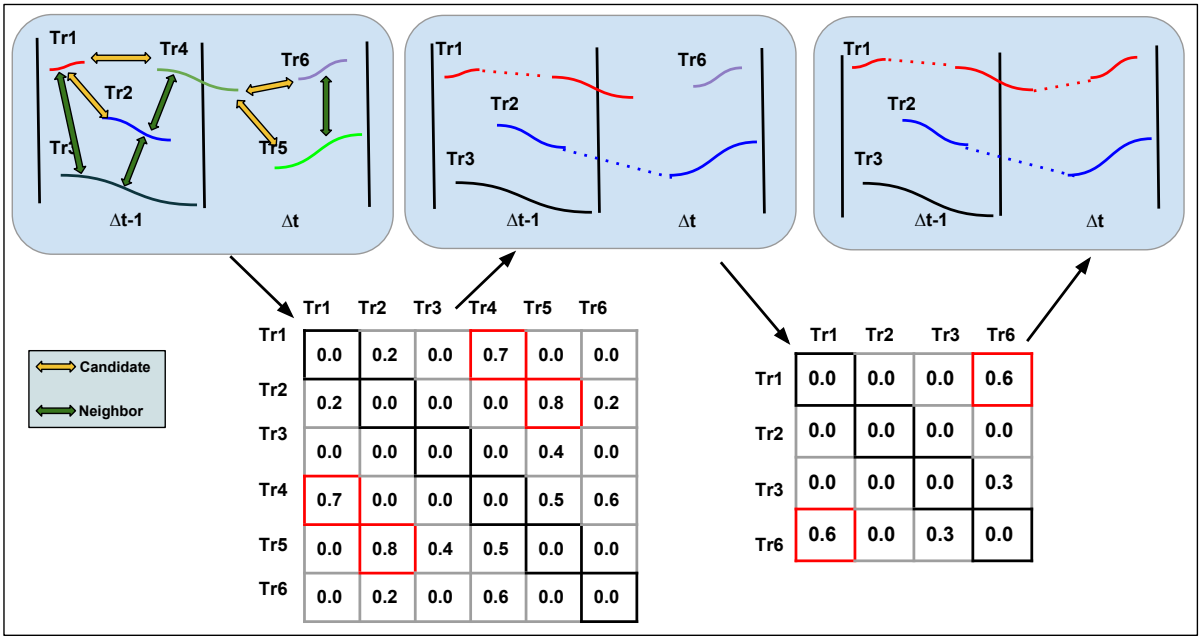


Figure 4.4: Tracklet linking is processed in each time-window Δ_t.

model than appearance features. However, the motion model is not too reliable as the person can change its direction frequently or motion measurement errors can be caused by detection errors or calibration. Therefore, in order to use motion model effectively, in equation (4.19), the value of the motion model weight is fixed with a maximum value of 0.5.

4.3.5 Tracklet linking

Tracklet linking is the last task in the pineline of the proposed approach. Tracklet linking includes two subtasks. In the first subtask, the global similarity between two tracklet representations is computed based on the weighted tracklet feature similarities (shown by block 4 in Figure 4.1). In the second subtask, based on the global similarities of tracklets, bipartite graph optimization such as Hungarian algorithm is applied to optimally link tracklets (shown by block 5).

Figure 4.4 illustrates the tracklet linking process. In each video segment Δ_t, the tracker determines reliable features by computing and updating overtime feature weights. The global representation similarity GS of tracklet Tr_i with each its candidate (represented by Tr_c) is summed up by feature similarities with the corresponding weights as follows:

$$GS(\nabla_{Tr_i}, \nabla_{Tr_c}) = \frac{\sum_{k=1}^{F^{app}} (\omega_{i,c}^{k} + \omega_{c,i}^{k}) \times Simil(F_i^k, F_c^k) + \sum_{m=1}^{F^{mo}} (\omega_{i,c}^{m} + \omega_{c,i}^{m}) \times Simil(F_i^m, F_c^m)}{\sum_{f=1}^{F_i} (\omega_{i,c}^{f} + \omega_{c,i}^{f})} \tag{4.20}$$

After computing these global linking scores, we construct an association matrix M = $\{m_{ij}\}$ with i=1..n, j=1..n, where n is the number of tracklets stacked in two current time consecutive time-windows Δ_{t-1} and Δ_t. $m_{ij} = GS(\nabla_{Tr_i}, \nabla_{Tr_j})$ computed by equation (4.20) if tracklet Tr_j is a candidate of Tr_i; Otherwise, $m_{ij} = 0$. Then, Hungarian algorithm is used to optimize the tracklet linking process. However, the Hungarian algorithm only finds out the best link between 2 tracklets corresponding to one person per time. In order to link all tracklets corresponding a person, the proposed approach applies Hungarian algorithm until there is no more possible linkes. Particularly, as shown in figure 4.4, tracklet Tr_1 is firstly linked with tracklet Tr_4 then is continuously linked with Tr_6 after applying the Hungarian algorithm in the second time.

4.4 Evaluation

We test the proposed approach named $RFE-Tracker$ on four sequences of public datasets: PETS2015, PETS2009 and TUD. The proposed framework can apply any short-term tracker as a first step to extract the tracklets. However, in this chapter, we propose to use the tracker in [20] named PMT because its code is available and it also uses a pool of person appearance features to track people. The performance of $RFE-Tracker$ is compared with the tracker PMT [20], some other short-term tracking and long-term tracking methods from the state-of-the-art.

4.4.1 Performance evaluation

PETS dataset

We choose the sequence PETS2015-W1_ARENA_Tg_TRK_RGB_1 in dataset PETS2015 and sequence PETS2009-S2/L1-View1 in dataset PETS2009 to test our approach because people have pose variation and abrupted movement change in scenes.

Figure 4.5 (six top images belong to PETS2009-S2/L1-View1 while three bottom images belong to PETS2015-W1_ARENA_Tg_TRK_RGB_1) illustrates the tracking performance related to the online computation of feature weights depending on each video scene. With the situation on three top images, tracklet ID_3 (shown by yellow bounding box) and tracklet ID_{14} (shown by red bounding box) are mis-detected because they cross each other at frame 140. The overlapped tracklets are located inside the black eclipses. Almost all appearance features of tracklets are similar but both people move with opposite directions to each other. In this case, the proposed tracker recovers the broken links thanks to the tracklet motion model with a weight value of nearly 0.4.

Sequence	Method	MT(%)↑	PT (%)	ML(%)↓	MOTA(%)↑	MOTP (%)↑	GT	Frag (#)↓
PETS2015-W1_ARENA_Tg_TRK_RGB_1	Chau *et al.* [21]	0.0	100.0	0.0	56.3	60.1	2	2
	Ours ([21] + Proposed approach)	100.0	0.0	0.0	89.4	87.5	2	1
PETS2009-S2/L1-View1	Chau *et al.* [21]	61.9	23.8	14.3	62.3	63.7	21	8
	Bae *et al.with all* [8]	100	0	0.0	83.0	69.6	23	4
	Zamir et al. [90]	–	–	–	90.3	69.0	21	–
	Bae et al.*-global association* [8]	100	0	0.0	77.4	69.0	23	12
	Badie et al. [7]	–	–	–	90.0	74.0	21	–
	Badie et al. [7] + [21]	66.6	23.9	9.5	85.3	70.8	21	6
	Ours ([21] + Proposed approach)	76.2	14.3	9.5	85.7	71.8	21	4
TUD-Stadtmitte	Chau *et al.* [21]	60.0	40.0	0.0	45.3	61.9	10	13
	Milan *et al.* [73]	78.0	22.0	0.0	71.1	65.5	9	–
	Yan *et al.* [115]	70.0	30.0	0.0	–	–	10	–
	Ours ([21] + Proposed approach)	70.0	30.0	0.0	46.8	64.8	10	7
TUD-Crossing	Chau *et al.* [21]	46.2	53.8	0.0	69.1	65.4	11	14
	Tang *et al.* [101]	53.8	38.4	7.8	–	–	11	–
	Ours ([21] + Proposed approach)	53.8	46.2	0.0	72.3	67.1	11	8

Table 4.1: Tracking performance. The best values are printed in red.

The three middle images show a different chunk of the PETS2009-S2/L1-View1sequence. Tracklet ID_{31} (described by yellow bounding box) and tracklet ID_{32} (described by light blue bounding box) move with similar trajectories but their appearance colors are quite discriminative (by the color of hair and coat). The highest weight equals to 0.6 for dominant color and color histogram while the motion model weight is only 0.1. Therefore, the proposed tracker focuses mainly on dominant color and color histogram features and is able to track people correctly (see in frame 565).

Two people in sequence PETS2015-W1_ARENA_Tg_TRK_RGB_1 also have the similar appearance while having the different movement direction. In this case, the proposed approach relies mainly on person motion model to recover the trajectory fragmentation in frame 109.

Moreover, figure 4.6 shows our tracker's performance for the re-acquisition challenge when person (shown by red arrows) leaves and re-enters the scene. Instead of considering the moving people in the frame they have just re-entered, our approach tracks these people after a sufficient number of frames. Thanks to selected features (color histogram with weight value 0.5, dominant color with weight value 0.6) which are updated cumulatively, person IDs are correctly retrieved.

TUD dataset

We also use TUD datasets (including TUD_Stadtmitte and TUD_crossing) sequences to evaluate the performance of our approach compared to other recent trackers. Both of these sequences are quite short, with more or less than 200 frames, but they contain challenges for trackers such as low light intensity, crowded environment, frequent occlusions, similar person appearances.

Figure 4.7 illustrates clearly our approach performance when recovering broken links in scenes that have low light intensity and people moving in different directions. In these scenes, person appearances are not discriminative with each other. Appearance features have similar

reliable weights around of 0.2 while the motion model weight is 0.4. Therefore, based on the motion model of people, our approach tracks person ID_{26} (represented by a purple bounding box) correctly after several mis-detection frames.

4.4.2 Tracking performance comparison

The quantity comparison of tracking performances is shown in table 4.1. The proposed tracker outperforms the tracker PMT [20] over all metrics on sequence PETS2015-W1_ARENA_ Tg_ TRK_RGB_1. With sequence PETS2009-S2/L1-View1, our performance is better than the tracker [8] on both modes: short-term and long-term tracking combination (Bae etal_with all) and long-term tracking (Bae etal_global association) in MOTA and MOTP metrics. However, our results are not compared this tracker on metric MT and ML. This negative point can be explained that the proposed tracker and tracker [8] use different ground-truth and 2 people are not detected by the detector applied in the proposed framework. There is a significant tracking quality improvement when comparing our tracker with our input [21]. In particular, MOTA from 0.62 to 0.86 and 0.63 to 0.72 from MOTP, MT increases from 61.9% to 76.2%, ML reduces from 14.3% to 9.5% and the track fragmentation (Frag) reduces by half. Compared with other tracklet merging algorithms (marked in bold), our approach has a slightly lower results of MOTA and MOTP than trackers [90, 7]. However, the tracker in [90] works offline while our algorithm chooses flexibly object features overtime which is suitable for real-time applications. The tracker in [7] has a better performance than ours when it used its own input. When being tested with the same input (the output of tracker in [21]) with ours, our approach has higher results.

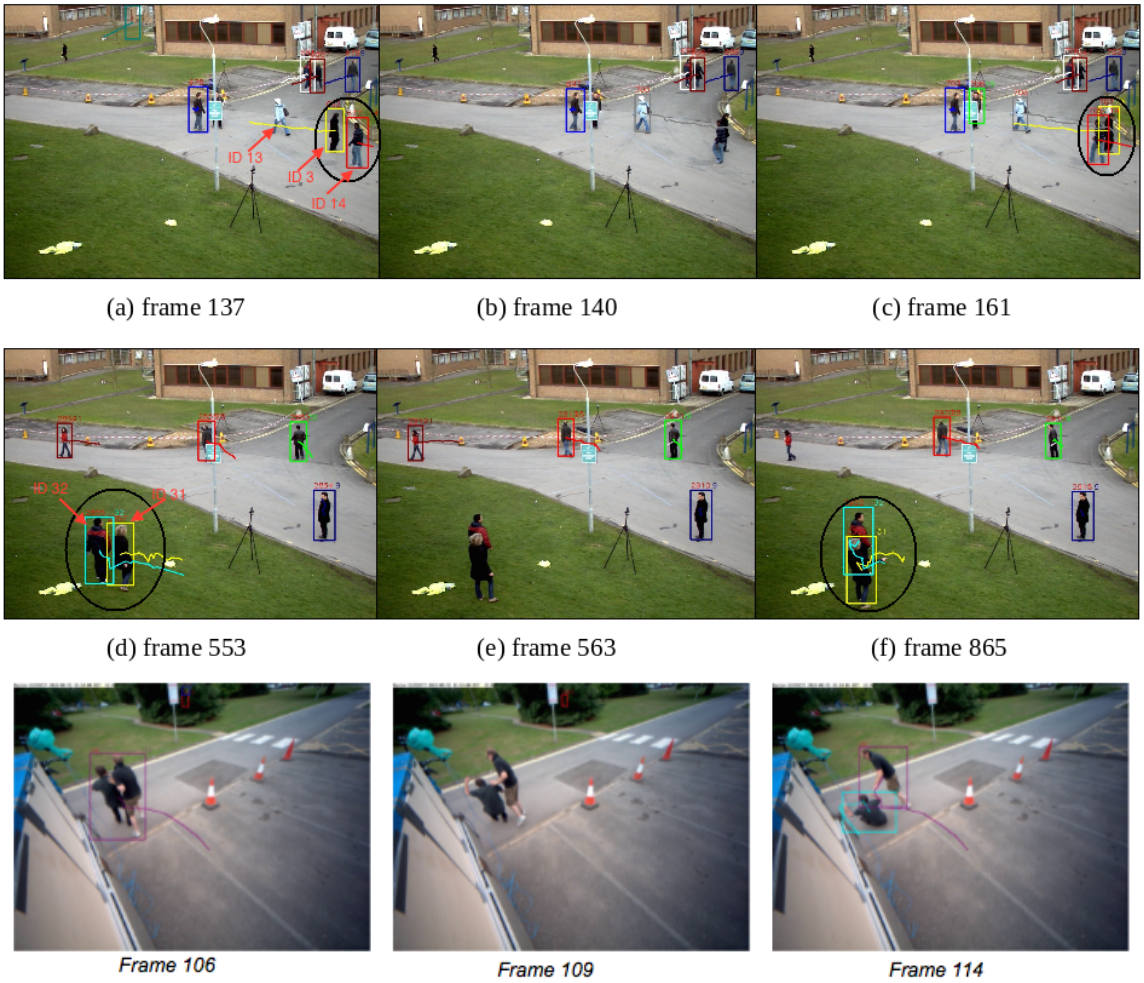
On both sequences TUD-Stadtmitte and TUD-Crossing in the TUD dataset, our approach does not lose any person. The obtained ML values are also the best ones compared to other state-of-the-art trackers in both sequences. Our tracker performance measured by metric MT increases from 60% to 70% with TUD_Stadtmitte and from 46.2% to 53.8% with TUD_Crossing dataset compared to the short-term tracker [21].

The quantitative results on Table 4.1 show that $RFE-Tracker$ improve the tracking performance of the short-term tracker MPT on most of tested datasets by linking tracklets into completed person trajectories. In particular, person trajectories are more completed (Mostly Track (MT), MOTA and MOTP and Frag values increase) while lost person trajectories are reduced (Mostly Lost (MT) values decrease). Tracker $RFE-Tracker$ also have better performance than other state-of-the-art trackers if evaluated trackers use the same ground-truth. Furthermore, the metric Frag plays an important role in evaluating tracklet linking methods. The less the number of fragments are, the better tracklet linking method works.The results on the metric Frag show that the proposed approach always has the least number of track fragmentation compared to other state-of-the-art trackers.

4.5 Conclusions

This chapter presents in detail a new long-term tracker named *reliable feature estimation tracker* (RFE). In order to enhance tracking performance, $RFE-Tracker$ automatically selects the most reliable tracklet features for each tracklet which are specific to a tracked person in a video scene. An adaptive combination of motion model and appearance features is proposed to handle the case that people' appearance information is not discriminative enough but their motions are different. The experimental results over four experimented benchmark sequences show the significant performance improvement of our approach compared to the input as well as the state-of-the-art trackers in the case that all trackers use the same detection and groundtruth. In this approach, no training process is needed which makes this algorithm generic and deployable to a large variety of tracking frameworks.

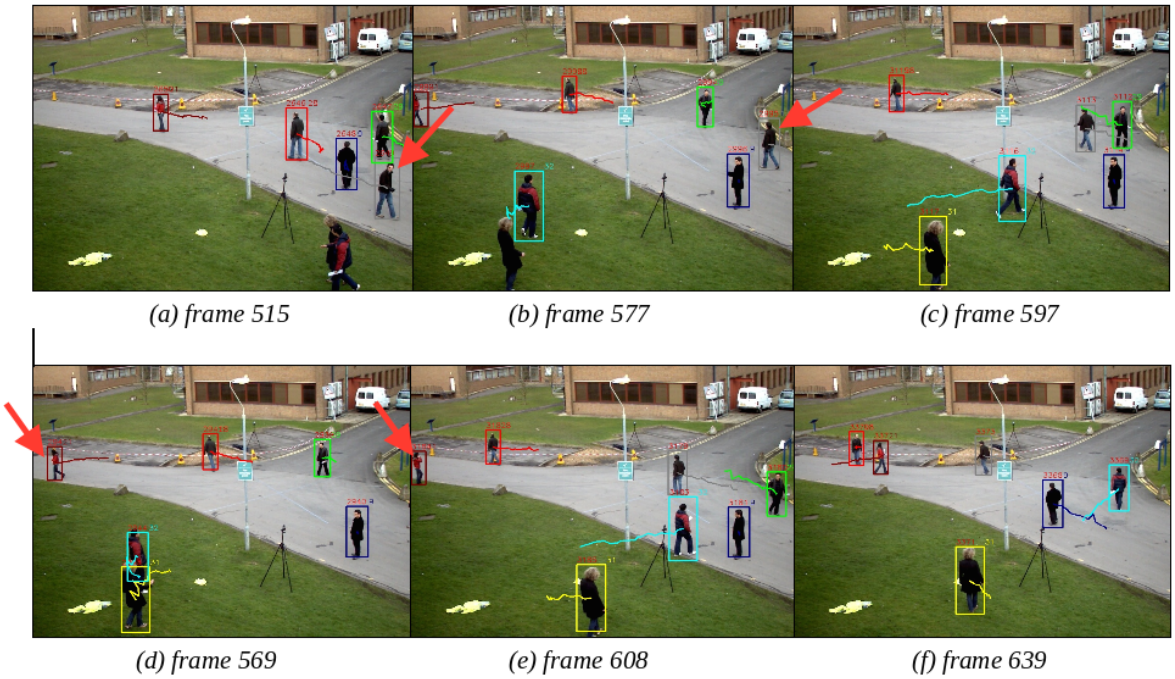
Drawback and future work The tracking performance of $RFE-Tracker$ is affected by the quality of detection as well as input tracklets. If the detector fails to detect a person, multi-person tracking algorithm cannot track this person at all. If there is any ID-switch caused by the short-term tracker, the proposed approach cannot backtrack and correct it. Although the proposed tracklet filter step can make a person trajectory smoother, in the future we still need a backtrack mechanism to correct input tracklets to improve the long-term tracking quality.



(a) frame 137 (b) frame 140 (c) frame 161

(d) frame 553 (e) frame 563 (f) frame 865

Figure 4.5: PETS2009-S2/L1-View1 and PETS2015-W1_ARENA_Tg_TRK_RGB_1 sequences: The online computation of feature weights depending on each video scene.



(a) frame 515 *(b) frame 577* *(c) frame 597*

(d) frame 569 *(e) frame 608* *(f) frame 639*

Figure 4.6: PETS2009-S2/L1-View1 sequence: Tracklet linking with the re-acquisition challenge.

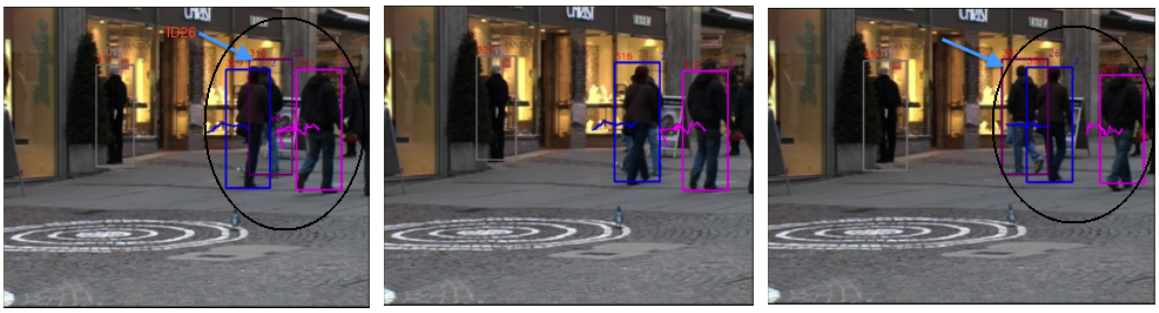


Figure 4.7: TUD-stadtmitte sequence: The proposed approach performance in low light intensity condition, density of occlusion: person ID_{26} (presented by purple bounding box) keeps its ID correctly after 11 frames of mis-detection.

5

MULTI-PERSON TRACKING DRIVEN BY TRACKLET SURROUNDING CONTEXT [79]

5.1 Introduction

Many trackers have been proposed in the past which would expect the multi-person tracking task as solved. It is true for scenes containing a fixed background with a low number of people and few interactions. Besides that, almost of these approaches (including the tracker $RFE\_Tracker$ presented in the previous chapter) track people based on the affinities of people without considering person's surrounding information. Therefore, complex video conditions such as the variations of person occlusion, illumination, high person densities still represent big challenges for these state-of-the-art trackers.

In this chapter, we propose a new long-term tracking framework named *context-based parameter tuning* (CPT) which combines a short-term data association and an online parameter tuning method for each tracklet based on both individual and person surrounding information. This framework has three main contributions as follow:

- We introduce a new long-term tracking framework which combines short-term data association and the online parameter tuning for each tracklet. The proposed framework contrasts to the method [22] that uses the same parameter setting for all tracklets in the video (section 5.3).

- We show that a large number of parameters can be efficiently tuned via the approximate optimization process - multiple simulated annealing. Whereas method [22] could tune only a limited number of parameters and fix the rest to be able to do an exhaustive search to find the best parameter value (section 5.3.4.2).

- We define the surrounding context around each tracklet (section 5.3.3) and the similarity metric among tracklet representations. This metric allows us to match tracklets in an unseen video segment with tracklets in a learned video context (section 5.3.4.3).

The remaining part of this chapter is organized as follows. Section 5.2 discusses some related methods which also try to solve the mentioned MOT problems. The study of video context and the proposed approach are described in detail in section 5.3. Section 5.4 evaluates and compares the tracking performance of the proposed approach with other state-of-the-art trackers. Finally, section 5.5 sums up all the content of this chapter, emphasizes the contributions, tracking performance, the drawbacks of the proposed approach and the future works.

5.2 Related work

Some state-of-the-art trackers [124, 19] track people by automatically tuning the tracking parameters based on the video context information. These methods typically use a pool of object features which are weighted for the new frame based on the most recent context information. The approach in [124] runs multiple trackers at the same time. Each single tracker is responsible for one feature. To fuse these independent trackers, the authors propose two configurations, tracker selection and interaction. The tracker selection extracts one tracking result from among multiple tracker outputs by choosing the tracker that has the highest reliability. The tracker interaction is conducted based on a transition probability matrix (TPM) which is updated by estimating each tracker's reliability. Then, the selected tracker estimates the new state of object. Using only the selected tracker to keep tracking objects, this method has a strong limitation on self-adaptability to the change of video scene characterized by more than one feature (appearance versus motion). Moreover, running multiple trackers also introduces high computational load and restricts the usage of the method in real time. The tracker in [19] firstly learns offline tracking parameters for the video context and saves this information to a database. In the online phase, the tracking parameters of the current video context are retrieved based on a reference to the corresponding learned tracking parameters of the closest context from the database. However, they ignore the individual information of the objects and use the same set of tracking parameters for all objects. This requires a hypothesis of the discrimination of appearances and trajectories among targets, which is not always in the real cases. Moreover, the number of tracking parameters that these trackers can tune is limited to a

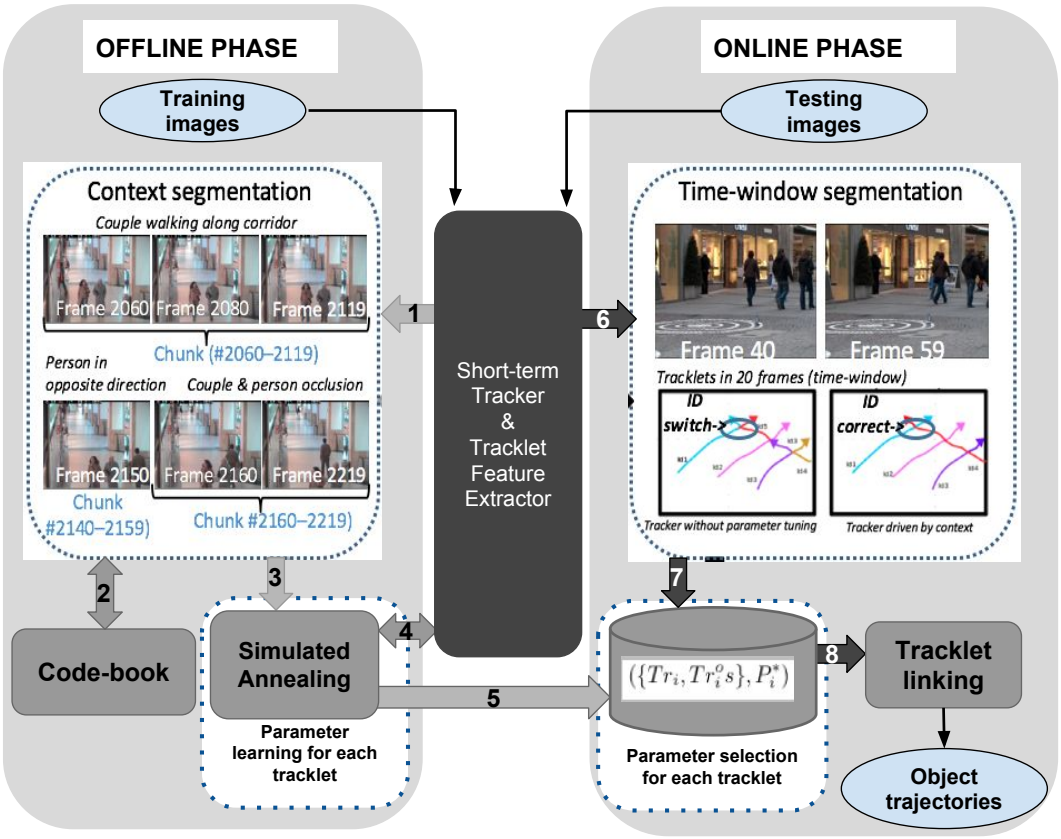


Figure 5.1: Our proposed framework is composed of an offline parameter learning and an online parameter tuning process. Tr_i is the given tracklet, and Tr_i^o is the surrounding tracklet set of tracklet Tr_i.

few. Therefore, in this chapter, we propose a long-term multi-person tracking method to tune tracking parameters based on both person's individual and surrounding information. In this method, tracking parameters are tuned for each tracklet instead of all tracklets appearing in the video context. No requirement on the number as well as the mutual dependence of tunable tracking parameters makes this algorithm generic and be applied to tune tracking parameter sets of different trackers.

5.3 The proposed framework

Figure 5.1 illustrates the proposed MOT framework. It highlights all steps done in the offline and online phases. The objectives of the offline phase are to segment videos based on the "video context", then to learn a database of tracklet representations with according best track-

ing parameters. In the online phase, each tracklet with its "surrounding context" is retrieved to the closest learned tracklet representation in the database to obtain the best tracking parameters. The definitions of "video context" and "surrounding context" are presented in the next subsections.

The framework flow: The framework is composed of 8 steps, including 5 steps in the offline phase and 3 steps in the online phase. Both offline and online phases share a person detector and a short-term tracker to extract the tracklets as well as their features.

In the **offline phase**, the video firstly is segmented into video segments with stable context. In particular, the video is split into video chunks of a fixed size. Each chunk is processed with the short-term tracker [20] to extract the "context features" (*flow_1*). Then, a code-book model of "context features" presents for each chunk (*flow_2*). The distance between two codebook models of two consecutive chunks is computed. If two or more consecutive chunks have same context (the codebook model distance is small), they are merged to form a video segment. Next step is the best tracking parameter learning. The video segment (video chunks with same context) and its tracklets are passed to the simulated annealing optimization process (*flow_3*). In this step, the tracklet representation (including tracket individual and surrounding features) is generated (*flow_4*). The best tracking parameters P_i^* for each tracklet are learned based on the evaluation of tracker performance against the ground truth information (*flow_4*). Finally, a tracklet representation accompany with its best tracking parameter set is stored in database (*flow_5*). The learned data is formalized as follow: (∇_{Tr_i}, P_i^*). More details on the optimization of parameters P_i^* are provided in section 5.3.4.2.

In the **online phase**, the proposed tracking algorithm processes in each fixed-size video chunk defined by a time-window (in our case is 20 frames). The same short-term tracker with the offline phase are applied on each video chunk Δt to extract tracklets as well as tracklet representations (*flow_6*). Then, each extracted tracklet representation is matched against the closest learned tracklet representation in database to retrieve the according best tracking parameters (*flow_7*). The distance of two tracklet representations is provided in section 5.3.4.3. Finally, in order to extend the person trajectory, tracklets with tuned parameters in the current video chunk Δ_t and tracklets are retrieved in previous video chunk Δ_{t-1} are linked to each other by computing their tracklet representation distance and performing a local data association process using Hungarian optimization algorithm (*flow_8*).

5.3.1 Video context

We follow the definition of the video context and how to segment the videos into video segments with stable contexts from the paper [22]. Particularly, a video context is defined by elements in the videos which influence the tracking quality. We called these elements as contextual features. For each training video, we extract contextual features from tracked people

and then use them to segment the training video in a set of consecutive segments. In the following, we present a set of six contextual features which define a video context: density of people, their occlusion level, their contrast with regard to the surrounding background, their contrast variance, their 2D area and their 2D area variance.

- People density: The density of people influences significantly the tracking quality. A high density of people may lead to a decrease of person detection and tracking performance. The person density at time t is defined by the number of all people $|Nb_{det}(t)|$ on the 2D camera view:

$$Den^t = |Nb_{det}(t)| \tag{5.1}$$

- Occlusion level: Occlusion occurs when person is partially or completely hidden by other people (dynamic occlusion) or background (static occlusion). Occlusion level decreases both person detection and tracking performances. In this method, we focus on only the dynamic occlusion. Given two people O_t^i and O_t^j at time t, the occlusion level between both people is computed as follow:

$$o_k^t = \frac{a_{ij}^t}{min(a_i^t, a_j^t)} \tag{5.2}$$

  where k denotes the occlusion index in the set of occlusions occurring at time t, a_{ij}^t is the overlap area, a_i^t, a_j^t are bounding-box areas of two people O_i^t and O_j^t, respectively. Let $\mathcal{N}^t$ be the number of person overlap areas at time t, the occlusion level of the video scene at time t is defined the mean of occlusion levels of all people in the scene:

$$Oc^t = min(\frac{\sum_{k=1}^{\mathcal{N}^t} o_k^t \times 2}{Den^t}, 1) \tag{5.3}$$

- Contrast: The contrast of a person is defined as the color intensity difference between this person and its surrounding background. Let $A_i = \{C_i, W_i, H_i\}$ be the 2D bounding box of person O_i^t where C_i, W_i, H_i are its 2D center, width and height, respectively. We determine an outer bounding box of person O_i^t: $A_i^+ = \{C_i, W_i + \alpha W_i, H_i + \alpha H_i\}$ where α is a predefined value in interval [0,1]. In the experiment, we set α to 0.3. The **surrounding background** is defined as $A_i^{sur} = A_i^+ / A_i$.

  The contrast of a person O_i^t is computed by:

$$Cotr_i^t = 1 - Simil(H^{A_i}, H^{A_i^{sur}}) \tag{5.4}$$

  where $Simil(H^{A_i}, H^{S_i^{sur}})$ is the color histogram similarity of two regions: detection region of O_i^t and its surrounding background. The color histogram similarity is presented in equation 2.8 in Chapter 4.

Figure 5.2: Illustration of the contrast difference among people at a time instant.

a person with low contrast reduces first the person detection quality. So the quality of tracking algorithms indirectly decreases in this case. The contrast feature of a video context at time t, $Cotr^t$, is defined as the mean value of the contrasts of all people at time t as follow:

$$Cotr^t = \frac{\sum_{k=1}^{Den^t} Cotr_i^t}{Den^t} \tag{5.5}$$

- Contrast variance: As shown in figure 5.2, the contrasts of people have different values. Therefore, the contrast feature of a video context at time t computed as the mean as equation 5.5 cannot represent correctly the contrast of all people in the video. We define the variance of person contrasts at time t as their standard deviation value by:

$$\hat{C}^t = \sqrt{\frac{1}{Den^t}\sum_{i=1}^{n}(C_i^t - \bar{C}^t)^2} \tag{5.6}$$

- 2D area: 2D area of a person is defined as the number of pixels within its 2D bounding box. Therefore, this feature also characterizes reliability of the person appearance for the tracking process. The larger person area is, the higher the person appearance reliability

is. The 2D area feature of a video context at time t $Area^t$ is defined as the mean value of the 2D areas of people a_i^ts in the video scene at time t.

$$Area^t = \frac{\sum_{k=1}^{Den^t} a_i^t}{Den^t} \tag{5.7}$$

- 2D Area variance: Similar to the contrast feature, people in the video are able to have different 2D areas. people close to the camera have larger 2D areas tha people far to the camera. Therefore, we also define the 2D area variance feature of a video context at time t as the standard deviation value.

5.3.1.1 Codebook modeling of a video context

During the tracking process, we decide to use a codebook model [47] to represent a compressed form of contextual feature values in a video segment without making parametric assumption. In our approach, a video context is represented by a set of 6 feature codebooks, called context codebook model and denoted CB, CB = $\{cb^k$, k = 1..6$\}$. Each contextual feature is represented by a codebook, called feature codebook and denoted cb^k. A feature codebook includes a set of codewords which describe the values of this feature. The number of codewords presents for the diversity of feature values.

Definition of codeword

A code-word represents the values and their frequencies of a contextual feature. A feature codebook can have many codewords. A codeword i of codebook k (k = 1..6), denoted cw_i^k , is defined as follows :

$$cw_i^k = \{\bar{\mu_i^k}, m_i^k, M_i^k, fr_i^k\} \tag{5.8}$$

where

- $\bar{\mu_i^k}$ is the mean of the feature values belonging to this codeword.
- m_i^k, M_i^k are the minimal and maximal feature values belonging to this word.
- fr_i^k is the number of frames in which the feature values belong to this word.

Algorithm for updating codeword

The training phase for updating a codeword works as follows :

- At the beginning, the codebook cb^k of a contextual feature k is empty.
- For each μ_t^k defined as a contextual feature k computed at time t, whether μ_t^k activates any codeword in cb^k is verified. μ_t^k activates codeword cw_i^k if both conditions are satisfied :

+ μ_t^k is in range $[0.7 \times m_i^k, 1.3 \times M_i^k]$.

+ The distance between μ_i^k and cw_i^k is smaller than a threshold θ_3. This distance is defined as follows :

$$dist(\mu_t^k, cw_i^k) = 1 - \frac{min(\mu_t^k, \bar{\mu_i^k})}{max(\mu_t^k, \bar{\mu_i^k})} \tag{5.9}$$

where $\bar{\mu_i^k}$ is the mean value of codeword cw_i^k (presented in equation 5.14).

- If cb^k is empty or if there is no codeword activated, create a new codeword and insert it into $cb^k$by updating the values of this new codeword as follows :

$$\bar{\mu_i^k} = \mu_t^k \tag{5.10}$$

$$m_i^k = \mu_t^k \tag{5.11}$$

$$M_i^k = \mu_t^k \tag{5.12}$$

$$fr_i^k = 1 \tag{5.13}$$

- If μ_t^k activates cw_i^k, this codeword is updated with the value of μ_t^k:

$$\bar{\mu_i^k} = \frac{\bar{\mu_i^k} \times f_i + \mu_t^k}{f_i + 1} \tag{5.14}$$

$$m_i^k = min(m_i^k, \mu_t^k) \tag{5.15}$$

$$M_i^k = max(M_i^k, \mu_t^k) \tag{5.16}$$

$$fr_i^k = fr_i^k + 1 \tag{5.17}$$

The codewords whose value fr_i^k is lower than a threshold, are eliminated because they are corresponding to very low frequent feature values.

5.3.1.2 Context Distance

The context distance is defined to compute the distance between a context C and a context codebook model $CB = \{cb^k, k = 1..6\}$. The context C of a video segment (Δ frames) is represented by a set of six values : the density, the occlusion level of people, the contrast with regard to the surrounding background, their contrast variance, the 2D areas and the 2D area variance. For each contextual feature k (k = 1..6), the contextual feature value at time t is denoted μ_t^k . For each such value, we consider whether it matches any codeword of the corresponding feature code- book cb^k. The pseudo-code of Algorithm 1 shows how to compute the distance

between a context C and a context codebook model CB. The distance between context C and codebook cb^k is expressed by the number of times of matching a μ_t^k and a codeword cw_i^k are found. The distance $dist(\mu_t^k, cw_i^k)$ is defined as in equation 5.9 and is normalized in the interval [0, 1].

Algorithm 1 Compute_Context_Distance

```
1:  procedure ContextDistance(C, CB, L)
2:      Input: context codebook model CB, context C, L(number of frames of context C)
3:      Output: context distance between CB and C
4:      totalCount = 0;
5:      for each codebook cb^k in CB (k = 1..6) do
6:          count = 0;
7:          for each value μ_t^k of context C do
8:              for each codeword cw_i^k in codebook cb^k do
9:                  if dist(μ_t^k, cw_i^k) < θ1 then
10:                     count ++;
11:                     break;
12:         if count/L < θ2 then return 1;
13:         totalCount+=count;
14:     return 1 - totalCount/(L * 6);
```

5.3.2 Tracklet features

The proposed long-term tracker people to tune tracking parameters for tracklets which are generated by a short-term tracker. In order to characterize tracklet Tr_i, we use the tracklet feature pool F_i which includes features accumulated by node features within the tracklet time-span. The definition as well as how to compute nodes features are presented in detail in chapter 3. In this chapter, the tracklet feature pool F_i is also divided into 2 feature pools $F_i = \{F_i^O, F_i^{OE}\}$:

- F_i^O (individual features) represents the pool of features that are computed using only the data of the tracklet. F_i^O includes 6 features: 2D Shape ratio, 2D Area, Color histogram, Dominant color, Color Covariance and motion model.

- F_i^{OE} (surrounding features) represents the pool of features that are computed based on the interaction of a tracklet to its surrounding background which is defined in section 5.3.1. Any tracklet intersecting in the surrounding background of tracklet Tr_i is considered to interact with tracklet Tr_i. F_i^{OE} consists of 3 features: occlusion, person density and contrast.

A tracklet feature is accumulated by the according feature of nodes belonging to the tracklet. However, the feature reliability is different between nodes. Therefore, each tracklet feature $F_i^k \in F_i$ is represented by (μ_i^k, σ_i^k) where μ_i^k and σ_i^k are the weighted mean and standard deviation of nodes' feature $F_i^k(t)$, respectively. The values of μ_i^k and σ_i^k are computed by:

$$\mu_i^k = \frac{\sum_{t=m}^{n} w(t) * F_i^k(t)}{\sum_{t=m}^{n} w(t)} \tag{5.18}$$

$$\sigma_i^k = \sqrt{\frac{\sum_{t=m}^{n} w(t) * (F_i^k(t) - \mu_i^k)^2}{\sum_{t=m}^{n} w(t)}} \tag{5.19}$$

where $w(t)$ is the weight function which is defined in section 2.3.1.1 in chapter 4.

5.3.3 Tracklet representation

The proposed approach objects to obtain the best tracking parameters for each tracklet in the testing video scenes by retrieving its closest tracklet in the learned database. Because the datasets for training are different with those for testing, in stead of comparing the individual features between two tracklets, this approach compare their surrounding context as well as the discrimination to their neighbourhood. We define the neighbourhood of a tracklet $Tr_i^{Surr}s$ as a set of tracklets Tr_i^{Surr} which intersects inside the surrounding background of tracklet Tr_i.

Therefore, the tracklet representation is defined as follow:

$$\nabla_{Tr_i} = \{F_i^{OE}, \{F_i^O, F_i^{O(Surr)}s\}\} \tag{5.20}$$

where $F_i^{O(Surr)}$ is individual feature pool of each surrounding tracklet $Tr_i^{Surr} \in Tr_i^{Surr}s$ and $F_i = \{F_i^O, F_i^{OE}\}$.

In the following section, the list of tracklet features (consists of surrounding features and individual features) are presented in detail.

5.3.4 Tracking parameter tuning

5.3.4.1 Hypothesis

In order to select the best tracking parameters for each tracklet, the proposed approach relies on a hypothesis that if representations of two tracklets are close enough, the learned best tracking parameter values of one tracklet could be applied effectively for the other one. The hypothesis is formalized as follow:

$$\begin{aligned} &\text{If } (\|\nabla_{Tr_j} - \nabla_{Tr_i}\| < \epsilon_1) \text{ and } (Q(\Im(\nabla_{Tr_i}, P_i^*), GT) > \theta) \\ &\quad \Rightarrow Q(\Im(\nabla_{Tr_j}, P_i^*), GT) > \theta + \epsilon_2 \end{aligned} \tag{5.21}$$

where $\|\nabla_{Tr_j} - \nabla_{Tr_i}\|$ is the ***tracklet representation distance*** (provided in section 5.3.4.3) of two tracklets Tr_i and Tr_j, Q is the tracking performance of tracking algorithm $\Im$, GT stands for tracking ground-truth and P_i^* is the best tracking parameter set of tracklet Tr_i. In this work, we use the Mostly-Track (MT) metric (detailed in the experiment part) and the tracking time metric in [78] to evaluate the tracking performance Q.

The hypothesis is proposed with two main purposes. The first purpose is to justify the tuning online tracking parameters for an extracted tracklet. If the representation ∇_{Tr_j} of the new tracklet Tr_j In the online phase is matched against any record in the database ∇_{Tr_i}, the tracker could gain the optimal performance for the new tracklet when applying the according learned parameter set P_i^*. The second purpose is to avoid redundant records in database. In training phase, if tracklet Tr_j's representation is closed enough to existed tracklet Tr_i in the database, they could use the same best tracking parameters and we store only tracklet Tr_i. The correctness of the hypothesis will be discussed in the experiment part.

5.3.4.2 Offline Tracking Parameter learning

We have a training video segmented by video context and now we want to learn the best tracking parameters for each tracklet in a video context and store it in database. For exploring a large search space to find an optimum, we are using simulated annealing (SA) method which helps in cases where exhaustive search is impossible. SA is meta-heuristic and approximates the global optimum in a large searching space. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing may be preferable to alter such as gradient descent that can get stuck in local optimization.

Simulated annealing based optimization: The tracking parameters are learned to optimize the tracker performance which is evaluated against the ground truth information. Therefore, the objective function of tracking parameter optimization is defined by finding the best tracking parameter set P_i^* to maximize the tracking performance $Q(\Im(\nabla_{Tr_i}, P_i)$. Then, the objective function is determined:

$$P_i^* = \arg\max_{P_i} Q(\Im(\nabla_{Tr_i}, P_i), GT) \tag{5.22}$$

We apply the multiple-SA method to find the best tracking parameter setting. In particular, multiple optimizers run in parallel to increase the searching speed. The starting points SA optimizers are initialized by dividing the searching space into subsets and selecting the middle point of each subset. Therefore, the best performance of optimizers will approximate more accurately the global optimized values. Learned parameter values according to the optimizer getting the highest performance are considered as the best tracking parameter set.

5.3.4.3 Online Tracking Parameter tuning

In the testing phase, the online tracking parameter tuning is applied for each video chunk (Δt). Firstly, the representation of each tracklet in this video chunk is extracted. Then, based on the tracklet representation distance computation, the given tracklet obtains the best tracking parameter set by retrieving its closest one in the learned database.

Tracklet representation distance

To compare two tracklets, we focus on two aspects. The first aspect is the difference between these tracklets' appearance discrimination level with their own surrounding tracklets. The second is the difference between their surrounding context. Therefore, the tracklet representation distance $\|\nabla_{Tr_j} - \nabla_{Tr_i}\|$ shown in Equation 5.21 is formalized as follow:

$$\|\nabla_{Tr_j} - \nabla_{Tr_i}\| \simeq \beta \times \|Disc(F_j^O, F_j^{O(Surr)}s) - Disc(F_i^O, F_i^{O(Surr)}s)\| + (1-\beta) \times \|F_j^{OE} - F_i^{OE}\| \quad (5.23)$$

where $Disc(F_i^O, F_i^{O(Surr)}s)$ and $Disc(F_j^O, F_j^{O(Surr)}s$ are the appearance discrimination levels of tracklets Tr_i and Tr_j with their surrounding tracklets, respectively. $\|F_j^{OE} - F_i^{OE}\|$ is the surrounding context distance of Tr_i and Tr_j. The weight β adapts the importance of appearance discrimination level between two tracklets over the distance of their surrounding context. We set β values to 0.7 in experiment.

We define $p \in \{i, j\}$ and N is the size of F_p^O, $[N+1, N+3]$ are indexes of surrounding features F_p^{OE}. $Disc(F_p^O, F_p^{O(Surr)}s)$ and $\|F_j^{OE} - F_i^{OE}\|$ in equation 5.23 are computed as follows:

$$Disc(F_p^O, F_p^{O(Surr)}s) = \frac{\sum_{k=1}^{N} \omega_p^k \times Disc(F_p^k, F_p^{k(Surr)}s)}{\sum_{k=1}^{N} \omega_p^k} \quad (5.24)$$

$$\|F_j^{OE} - F_i^{OE}\| = 1 - \frac{\sum_{k=N+1}^{N+3} \gamma^k \times Simi(F_j^k, F_i^k)}{\sum_{k=N+1}^{N+3} \gamma_i^k} \quad (5.25)$$

$$Disc(F_p^k, F_p^{k(Surr)}s) = 1 - \tilde{M}(Simi(F_p^k, (F_p^{k(Surr)})) \quad (5.26)$$

With equation 5.24, the appearance discrimination level $Disc(F_p^O, F_p^{O(Surr)}s)$ is computed by the weighted average of all tracklet individual features' discrimination $Disc(F_p^k, F_p^{k(Surr)}s)$ of tracklet Tr_p (k = 1..N) wrt its neighbourhood $Tr_i^{Surr}s$. $Disc^k(Tr_p, Tr_p^c s)$ on tracklet individual feature k, shown in equation 5.3.4.3, is computed based on the median $\tilde{M}$ of this feature similarities between Tr_p and $Tr_p^c s$. The surrounding context distance $\|F_j^{OE} - F_i^{OE}\|$ between

two tracklet Tr_i and Tr_j, shown in equation 5.25 is computed by the weighted average of their surrounding features' similarity. The way to compute tracklet feature similarities is provided in section 5.3.2.

If the tracklet surrounding context changes, the reliability of tracklet features may change and their individual feature weights ω as well as the surrounding feature weights γ need to be set and tuned along the change onf scene. Therefore, the best tracking parameter set P_i^* defined in section 5.3.4.1, which is learned. In the offline phase and tuned In the online phase, is a set of individual feature weights ω and surrounding feature weights γ in equation 5.24 and 5.25, respectively.

5.3.4.4 Tracklet linking

Beside using the ***tracklet representation distance*** to retrieve the closest tracklet in the learned database to a tracklet in online parameter tuning, this distance is also used to compare a tracklet with its candidates in two consecutive video chunks (Δ_{t-1} and Δ_t) in tracklet linking step.

We construct an association matrix M = $\{m_{ij}\}$ with i=1..n, j=1..n, where n is the number of tracklets based on their tracklet representation distance. $m_{ij} = \|\nabla_{Tr_j} - \nabla_{Tr_i}\|$ computed by equation 5.23 if tracklet Tr_j is a candidate of Tr_i; Otherwise, $m_{ij} = 0$. Finally, Hungarian algorithm is used to optimize the tracklet linking process.

5.4 Evaluation

In this section, the performance of the proposed tracker named $CPT-Tracker$ is evaluated. The short-term tracker using different person appearance features [20] is selected to experiment in our framework. We compare the tracking results of $CPT-Tracker$ with methods from the state-of-the-art in three cases: with the short-term tracker (with fixed parameter), with the tracker which tunes tracking parameters for the whole video context and with six other state-of-the-art trackers.

5.4.1 Datasets

Training phase

$CPT-Tracker$ is trained on nine video sequences: four videos from CAVIAR dataset[1] and three from ETISEO dataset[2]. The videos are selected because they represent a variety of tracking contextual information (e.g..low/high density of person in the scene, strong/weak person

[1] homepages.inf.ef.ac.uk/rbf/CAVIAR/

[2] www-sop.inria.fr/orion/ETISEO/

contrast). The offline training phase requires the ground-truth of person tracking as input. From the hypothesis shown in equation 5.21, some tracklets are close to each others then we keep only one tracklet as the representative. Therefore, after training 780 tracklet samples, tracking parameters for only 284 tracklet representations are learned. Tracklet representations together with their own best tracking parameter sets are storaged in the database. Then, this database is used as a reference to automatically retrieve tracking parameters for tracklets in the testing phase.

Testing phase

$CPT - Tracker$ is evaluated on 3 video sequences from 2 public datasets (PETs2009 and TUD). For all these videos, the video scenes are different from the ones of training videos. The proposed tracking algorithm processes on each video chunks of 20 frames. Tracking parameters are tuned for each tracklet in the current video chunk by best tracking parameters which are obtained from its closest tracklet in the learned database. The tuned tracking parameters of each tracklet adapts the tracker $CPT - Tracker$ to the change of this tracklet's surrounding context.

5.4.2 System parameters

All system parameters have been found experimentally, and is kept unchanged for all benchmark datasets. The same threshold $\theta = 0.3$ is used for the data association process. The size of a video chunk in the offline phase is fixed to 50 frames. The size of a video chunk in the online phase is 20 frames. The minimum size of a tracklet is set to 3 frames.

5.4.3 Performance evaluation

5.4.3.1 PETs 2009 dataset

The sequence S2L1_View1, is selected for testing because this sequence is used for the evaluation in several state-of-the-art trackers. It consists of 794 frames with 21 people with different degrees of inter-person occlusion.

The visualization in figure 5.3 shows the surrounding contexts of a testing tracklet from sequence S2L1_View1 and its two closest learned tracklet representations from CAVIAR dataset. The testing tracklet and two learned tracklets have the similarity of surrounding context. However, the discrimination level of these tracklets with their own surrounding tracklets are different. In particular, the learned tracklet in (b) and testing tracklet (represented by "red" bounding-boxes) move in the opposite direction with their own surrounding tracklets (represented by "blue" bounding-boxes) while people move in the same direction in (a). Therefore, based on the tracklet representation distance, the tracklet In the online phase is closer to learned

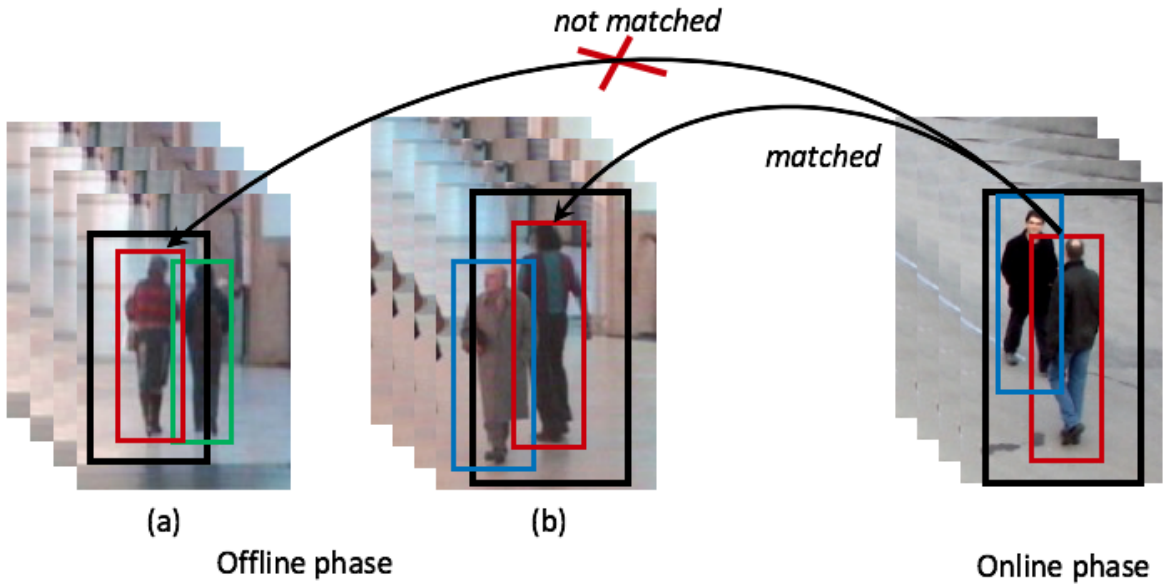


Figure 5.3: Tracklet representation ∇_{Tr_i} and tracklet representation matching. Tracklet Tr_i is identified with "red" bounding-box and fully surrounded by the surrounding background marked by the "black" bounding-box. The other colors (blue, green) identify for the surrounding tracklets.

tracklet in (b) than in (a). The tracker $CPT-Tracker$ uses the best tracking parameters learned for the tracklet in (b) to tune tracklet feature weights for the tracklet in the online phase.

5.4.3.2 TUD dataset

The second test is conducted with the TUD dataset (including TUD-Stadtmitte and TUD-Crossing sequences). Both of these sequences are quite short, with more or less 200 frames, but they contain challenges for trackers due to heavy and frequent person occlusions. Figure 5.4 shows a snapshot of the tracking performance of the proposed algorithm. The testing tracklet (represented by "green" bounding-box) has a low low-contrast and high person density context. The target appearance is not discriminative enough wrt surrounding tracklets but it moves in different direction compared to others. The closest tracklet to the testing tracklet in the learned database is represented by the "red" bounding-box. The best tracking parameters of this learned tracklet (consisting of 0.512 for motion feature, 0.215 for color histogram and 0.193 for color covariance) are tuned for testing tracklet. Thanks to tuned parameters, the tracker $CPT-Tracker$ can correctly link tracklets before and after mis-detections and recover the person trajectory.

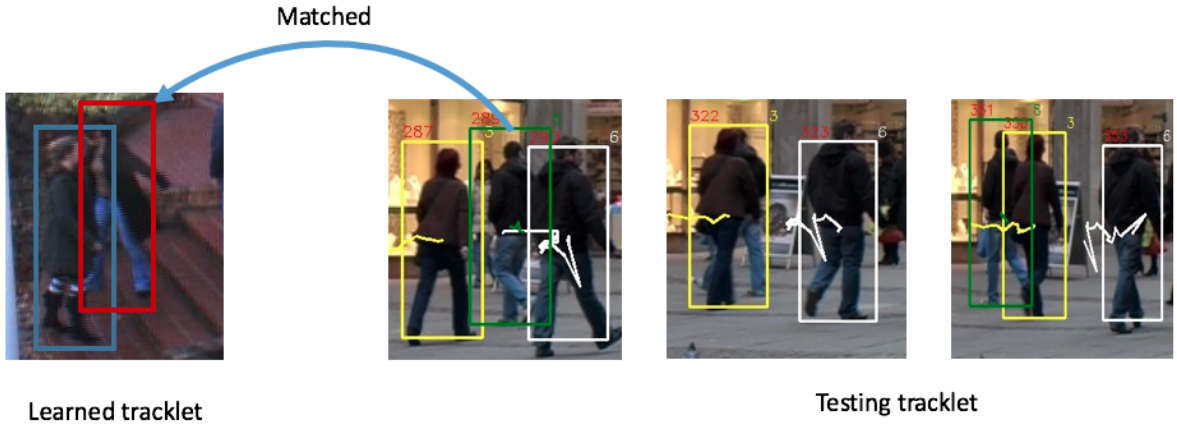


Figure 5.4: TUD-Stadtmitte dataset: The tracklet ID_8 represented by color "green" with the best tracking parameters retrieved by a reference to the closest tracklet in database recovers the person trajectory from misdetection caused by occlusion.

5.4.3.3 Tracking performance comparison

The tracking performance comparison of the proposed tracker $CPT-Tracker$ with state-of-the-art trackers is shown in table 5.1 over three testing video sequences. The detection and evaluation method are shared by trackers ICDP˙Phu,IMAVIS˙Phu and the proposed approach.

MT and ML metrics

Metrics MT and ML evaluate which percentage of ground-truth people are matched by tracking output (at least 80% for MT and less than 20% for ML).

On PETs2009/S2L1/View-1, three trackers [8, 39] and $CPT-Tracker$ are tested. Tracker [8] includes several steps: online local association or global association based on tracklet confidence and person appearance learning. In this evaluation section, we compare $CPT-Tracker$ with tracker [8] in case of global association method because both methods are online long-term tracking. The performance of method [8]- global association can reach 100% mostly-tracked (ML) which are much higher than $CPT-Tracker$'s performance (76.2%). However, the incomparable performance of $CPT-Tracker$ is reasonable when the person detector providing the input for this tracker misses detecting three ground-truth people. Beside that, two methods use different ground-truth. In particular, the tracker [8] is evaluated on 23 ground-truth people while the tracker $CPT-Tracker$ is evaluated on only 21 ones. In this dataset, in the case that people leave the scene and come back, the ground-truth with 21 people labels these people as the same but the another with 23 people considers these people as different. Therefore, even a tracker cannot track people in this case, the evaluation on 23 person ground-truth has higher performance than the evaluation on 21 person ground-truth. The tracker [39] mod-

Dataset	Method	MT(%)↑	PT	ML(%)↓	MOTA(%)↑	MOTP(%)↑	GT
PETS2009 - S2L1_View1	Shitrit *et al.* [11]	–	–	–	81.0	58.0	21
	Bae *et al.-global association* [8]	100	0	0.0	77.4	69.0	23
	Chau *et al.* [20]	–	–	–	62.3	63.7	21
	Chau [22] ([20] + parameter tuning for whole video context)	–	–	–	85.0	71.0	21
	Heili *et al.*[39] (parameter tuning based on detection context)	70.0	25.0	5.0	–	–	20
	Ours ([20] + Parameter tuning based on tracklet context)	76.2	14.3	9.5	86.8	73.2	21
TUD-Stadtmitte	Milan *et al.* [73]	70.0	20.0	0.0	71.1	65.5	9
	Chau *et al.* [20]	60.0	40.0	0.0	45.3	61.9	10
	Chau [22] ([20] + parameter tuning for whole video context)	70.0	10.0	20.0	–	–	10
	Heili *et al.*[39] (parameter tuning based on detection context)	70.0	30.0	0.0	–	–	10
	Ours ([20] + Parameter tuning based on tracklet context)	70.0	30.0	0.0	47.3	65.6	10
TUD-Crossing	Tang *et al.* [101]	53.8	38.4	7.8	–	–	11
	Chau *et al.* [20]	46.2	53.8	0.0	69.1	65.4	11
	Heili *et al.*[39] (parameter tuning based on detection context)	–	–	–	79.0	78.0	13
	Ours ([20] + Parameter tuning based on tracklet context)	53.8	46.2	0.0	72.1	67.3	11

Table 5.1: Tracking performance. The best values are printed in red.

els multi-person tracking task by Conditional Random Field (CRF) which considers long-term connectivity between pairs of detection. Tracking parameters are learned in an unsupervised way from detections and tracklets. Method [39] also uses a different ground-truth with our method, particularly, 20 ground-truth people are annotated. On the ML metric, we outperform the tracker [39], 76.2% comparing to 70.0%, even we use the ground-truth which counts more people in the video. Tracker [39] mostly loses only one person while our approach mostly loses two. However, both trackers uses different ground-truth, the comparison on ML metric is not convinced enough in the case that the proposed approach loses the person who is not counted by the tracker [39] ground-truth.

On TUD dataset including TUD-Stadtmitte and TUD-Crossing, our approach does not lose any person and has highest performance measured by MT metric. Compared to other methods [22], tracker $CPT - Tracker$ can track more people, then reduces the ML value from 20% to 0% on sequence TUD-Stadtmitte. Tracker $CPT - Tracker$ improves the performance of [20] measured by metric MT on both sequences (an increase of 10% on sequence TUD-Stadtmitte and 7% on sequence TUD-Crossing.

MOTA, MOTP metrics

In almost cases, our proposed approach has better MOTA and MOTP values compared to others, the short-term tracker [20] as well as the parameter tuning method for whole video context from [22].

On PETS2009/S2L1/View-1 sequence, the proposed approach performance has higher result than state-of-the-art trackers [11, 8] and parameter tuning method for whole context [22] which uses the same short-term tracker [20]. Especially, thanks to the proposed parameter tuning method, the short-term tracker [20] is improved significantly, from 62.3 to 86.8 for MOTA value and from 63.7 to 73.2 for MOTP value.

On TUD dataset, the tracker $CPT-Tracker$ slightly improves tracking performance of short-term tracker [20] in both metrics. We have lower performance compared to tracker [73] on the sequence TUD-Stadmitte in MOTA metric and tracker [39] on the sequence TUD-Crossing in MOTA, MOTP metrics. However, we evaluate our method using the same ground-truth and detection compared to tracker [20] while using the different ones compared to trackers [73, 39]. Therefore, in order to have more confident comparison, trackers from state-of-the-art need sharing the detection and evaluation method.

5.5 Conclusions and future work

This approach proposes a new framework which online tunes tracking parameters to adapt the tracker to the variation of tracklet surrounding context. It tunes the tracking parameters for each tracklet instead of globally setting up for all tracklets to ensure that tuned parameters can characterize each tracklet in its surrounding context. Moreover, this framework uses the approximate optimization method (SA) which has no restriction on the independence as well as the number of tracking parameters. Therefore, this framework can be also applied to other trackers with different tracking parameter set. A new way to represent a tracklet in its surrounding context is also proposed to highlight its discrimination level of tracklet to other in its context. The experimental results show the remarkable performance improvement of our approach compared to: (1) trackers using static parameter values, (2) a parameter tuner for all people in a video, (3) state-of-the-art trackers over three public benchmark datasets.

However, some limitations exist in the proposed approach. Firstly, the storage of more and more learned tracklets in database makes the database becomes huger and huger. Finding to the best learned tracklet to obtain best tracking parameters for a testing tracklet is time-consuming. Secondly, there is a requirement on the training data which should be diverse enough to make the algorithm generic. Third, the proposed approach cannot refine a tracklet extracted from a short-term tracker's output which is composed of more than one ground-truth person. Forth, the performance of the proposed tracker remarkably affected by the detection and short-term tracking performances.

Therefore, in **future work**, we will propose some methods to overcome above limitations of the proposed tracker: (1) a method to index learned tracklets to reduce the time to find the closest tracklet to retrieve the best tracking parameters in a large learned database. (2) a back-track mechanism to correct the errors of the detector and the short-term tracker.

6

RE-ID BASED MULTI-PERSON TRACKING [81]

6.1 Introduction

Multi-person tracking in a crowded environment faces to many challenging problems, such as long or frequent person occlusions caused by other people or background, pose variation or illumination changing which makes person appearance change overtime.

Multi-person tracking in a single camera can be considered a special case of Multiple shot Re-id applied for one camera view in cases person appearance variation caused by occlusion, illumination changes. Whereas, the recent person Re-id approaches propose effective object features which are invariant to person appearance change as well as metrics to improve their ability in matching people. However, the Re-id works in offline mode which requires person information for the whole video. Therefore, in order to address multi-person tracking problems with Re-id manner, it is necessary to propose a method which performs two tasks: (1) generating reliable person representations which are invariant with person appearance variation and (2) correctly linking person trajectories based on person representation affinities.

In this chapter, we propose a robust multi-person tracking method which takes full advantage of features (including hand-crafted and learned features) and tracklet affinity computation methods proposed for multiple-shot person Re-id and adapts them to MOT. The proposed method not only addresses problems in MOT but also ensures online processing. This method integrates a short-term and a long-term tracker in a comprehensive framework where the short-term tracker generates tracklets and the long-term tracker links generated tracklets together

after a time buffering. In order to represent a tracklet with hand-crafted features, these features are computed for full body and body parts, then, each tracklet is represented by a set of multi-modal feature distribution modeled by Gaussian Mixture Models (GMMs). Thanks to learning a Mahalanobis metric between tracklet representations, the long-term tracker handles occlusion and mis-detection by a tracklet bipartite association method. In order to learn this metric, KISSME [49] algorithm is adopted to learn feature transformations of a person before and after occlusion or mis-detection. The drawback of this metric learning algorithm is the requirement of the similarity between training and testing data. With the objective of making this framework become generic, instead of using hand-crafted features, we represent a tracklet by CNN feature extracted from a pre-trained CNN model. Then, we associate the CNN feature-person representation with Euclidean distance into a comprehensive framework which works fully online.

The rest of the chapter is organized as follow: Section 6.2 discusses about some works from the state-of-the-art which try to solve the same MOT problems as the proposed approach. Section 6.3 presents the details about the structure and flows of the mentioned two-step comprehensive hand-crafted Re-id features based tracking framework. Tracklet representation using learned features (CNN) is presented in section 6.4. The data association method is presented in section 6.5. Section 6.6 evaluates the robustness of the Re-id hand-crafted feature based method by comparing its performance with other state-of-the-art trackers. Finally, section 6.7 concludes the chapter.

6.2 Related work

In order to address problems related to person appearance changing, multiple-shot person Re-id methods [63, 126, 77] have gained high performances in matching people from different camera views. In order to match a query person to the closest person in a gallery, these Re-id methods use efficient features and person representations. These methods are adopted to solve problems that involve pose and camera view setting variation.

From the state-of-the-art, there are some approaches try to apply the Re-id features to tracking. The authors in [9] used Mean Riemannian Covariance Grid (MRCG) descriptor proposed for Re-id for linking tracklets into longer ones to form the final person trajectories. The affinity of two tracklets are computed based on the distance between two tracklet representation in each time-window. Tracket representation based on MRCG descriptor is generated by forming a dense grid structure with spatially overlapping square regions described using mean covariance matrix. Tracklet representation computed by the mean of corresponding cell covariances of all nodes in tracklet can not completely represent for a tracklet if person information changes much in tracklet timespan. To address this problem, authors in [7] select key-frames represent-

ing the most ”reliable nodes” of each tracklet. The ”reliable nodes” are the ones which contain the most significant information concerning the appearance of the person with the least noise coming from interaction with the background or other people (occlusion, pose variance, illumination changing and so on). This method can generate the reliable tracklet appearance signature. However, key-frame selection depends on the ratio between noise and non-noise nodes in tracklet. If noise nodes occupies a large ratio, for example long-term occlusion, selected key-frames are occluded nodes. Therefore, to efficiently link tracklets in the scenario variation, the consistent information of tracklet including noise as well as non-noise nodes needs to be covered and represented.

On the other hand, deep learning methods are also effectively applicable to multi-object tracking (MOT). Authors in [107] propose a novel and efficient way to obtain discriminative appearance-based tracklet affinity models. In this framework, each sample pair is passed to a Siamese CNN including two sub-CNNs to extract the feature vectors. Then, based on the feature vectors obtained from the last layer of both sub-CNNs in each video segment, temporally constrained metrics are learned online to update the appearance-based tracklet affinity model. Finally, MOT problem is formulated as a Generalized Linear Assignment (GLA) problem which is solved by the soft-assignment algorithm. Recently, another robust RNN-based multi-object tracker [92] has been proposed which outperforms previous works on most recent datasets including the challenging MOT benchmark. This method builds multiple-RNN models that learns to encode long-term temporal dependencies across multiple cue (appearance (A), motion (M) and interaction (I)). The output of each RNN model (represents the object in each cue) is a feature vector concatenated by 2 sub-feature vectors (same dimension). One sub-feature vector is extracted from a LSTM network which encodes long-term dependencies of object observations belonging to target trajectory. The other one is the result of RNN fully connected layer when passing directly the detection they wish to compare to the network. Finally, the final RNN is jointly trained end-to-end with the RNNs according to A, M and I cues by concatenating single feature vectors and outputting the score of whether a detection corresponds to a target using Soft-max classifier and cross-entropy loss. Although the effectiveness of these methods are presented, they bear a high computation cost of online tracklet appearance model learning.

In this chapter, we proposed a method which extends the features (hand-crafted and learned features proposed for Re-id) to represent tracklets in MOT. To compute the affinity between tracklets in online MOT, learned Mahalanobis distance are learned to compute the affinity of hand-crafted tracklet representation while we use the Euclidean distance to compare learned tracklet representation. The experimental results show that the performance of both frameworks are comparable with the state-of-the-art trackers.

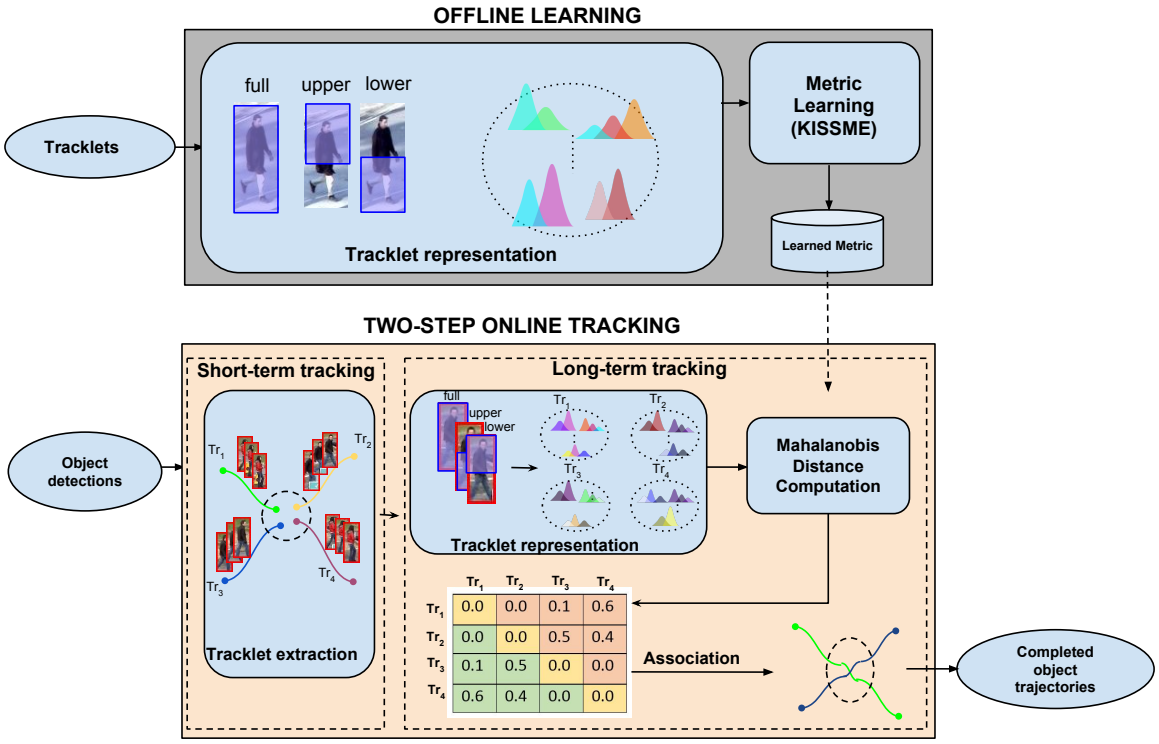


Figure 6.1: The proposed hand-crafed feature based MOT framework.

6.3 Hand-crafted feature based MOT framework

Figure 6.1 illustrates the proposed hand-crafted feature based MOT framework consisting of two blocks: online tracking and offline learning. In the offline block, the framework generates the tracklet representations of input tracklets and learns the similarity metric between the tracklets using data in the training set. Once the similarity metric has been learned, the two-step online block, describes the interaction between short-term (frame-to-frame) and long-term trackers, are processed in every time-window Δt. The short-term tracker's objective is to extract tracklets by linking together potential person detections in consecutive frames. For a reliable tracklet, in the scenario where people are occluded by background or other people, tracklet filtering presented in chapter 3 is applied by splitting spatially disconnected or occluded tracklets, too short tracklets are also filtered out. The long-term tracker generates tracklet representations of extracted tracklets stacked in two consecutive time-windows $[\Delta_{t-1}, \Delta_t]$ instead of the whole video as the Re-id method. The purpose of tracklet stack is to recover all further segmented tracklets from previous time-window in the case of long occlusion. The long-term tracker performs linking generated tracklets (tracklets and their corresponding candidates) based on their Mahalanobis distance and carries out data association using a bipartite graph optimization,

typically Hungarian algorithm.

6.3.1 Tracklet representation

We define tracklet Tr_i spanning over consecutive frames $< m, n >$ as following:

$$Tr_i = \{O_i^m, O_i^{m+1}, ..., O_i^{n-1}, O_i^n\} \tag{6.1}$$

Since person Re-id usually deals with identifying a person from different camera views, it is expected that the appearance model from Re-id representation becomes even more effective in single-view multi-person tracking.

Inspired by person Re-id approach in [77], we represent the tracklet appearance as a multi-modal probability distribution of the selected features. To deal with occlusion, the appearance models are created independently for each part of person (full, upper and lower part of the bounding-box). By this method, each channel in tracklet representation could correspond to a particular object feature for each part.

Appearance models help to overcome occlusion, pose variation and illumination problems. Unlike feature pruning methods that make problem specific, we create models with different features without pruning. Although this can cause a redundancy in feature representation but the features are computed efficiently to be shared between the parts (upper and lower body regions are defined as 60% of the person detection bounding-box). To describe a person, we use appearance features that are locally computed on the person detection bounding-box, including: HOG[26], LOMO[63], MCSH[126] and Color histogram (CH) features where LOMO and MCSH features have never been applied in MOT domain. While the framework exploits HOG feature as a shape-based feature to overcome difficulties of pose variation, it benefits from other features to cope with appearance changes happening in long occlusions.

Given a set of nodes (detection bounding-boxes) belonging to tracklet Tr_i, the tracklet representation ∇_{Tr_i} (illustrated in figure 6.2) is defined as a multi-channel appearance mixture where each channel is a appearance model $M_i^{p,f}$:

$$\nabla_{Tr_i} = \{M_i^{p,f} \mid p \in \{full, upper, lower\}, f \in \{HOG, LOMO, MCSH, CH\}\} \tag{6.2}$$

Each appearance model in the set is a multivariate Gaussian Mixture Model (GMM) distribution of low-level features of part p and feature f.

$$M_i^{p,f}(GMM) = \{(\mu_{i,k}^{p,f}, \sigma_{i,k}^{p,f})\} \tag{6.3}$$

$k = 1..K$ and K is GMM components. The values of each Gaussian distribution $(\mu_{i,k}^{p,f}, \sigma_{i,k}^{p,f})$ is updated in the whole tracklet timespan as follow:

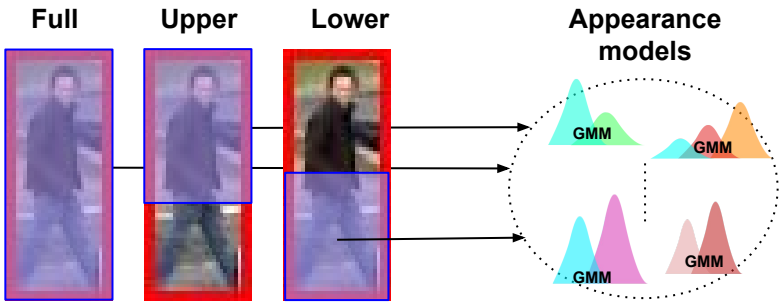


Figure 6.2: Tracklet representation.

$$\mu_{i,k}^{p,f(\Delta_t)} = (1-\alpha)\mu_{i,k}^{p,f(\Delta_{t-1})} + \alpha\frac{\sum_t^{\Delta_t} f_i^t}{\Delta_t} \tag{6.4}$$

$$\sigma_{i,k}^{p,f(\Delta_t)} = (1-\alpha)\sigma_{i,k}^{p,f(\Delta_{t-1})} + \alpha\sqrt{\frac{\sum_t^{\Delta_t}(\mu_{i,k}^{p,f(\Delta_t - f_i^t)})^2}{\Delta_t}} \tag{6.5}$$

where α is a weight to balance the feature reliability achieved in previous time-window Δ_{t-1} and current time-window Δ_t.

6.3.2 Learning mixture parameters

For each body part p and feature f of each person with ID i, the parameters of the appearance model $M_i^{p,f}$ are learned independently. There is no *a priori* knowledge about the number of modes of a person appearance, therefore, both finding the number of modes and description of them using low-level features need to be addressed.

People appearing in a video have different appearance and produce GMMs with variable number of components. Therefore, the number of components are not *a priori* determined and need to be retrieved. In order to infer the number of GMM components (K) for each appearance model automatically, Akaike Information Criterion (AIC) model selection is used. After knowing the fixed number of components, the parameters of a GMM could be learned conveniently using Expectation-Maximization method.

6.3.3 Similarity metric for tracklet representations

6.3.3.1 Metric learning

Recently, metric learning has gained considerable scientific interest in the field of person Re-id, as it provides a very elegant fusion of the descriptive and discriminative techniques typically encountered in the community. The main idea is to build on an existing feature representation,

which is usually designed to generate a descriptive signature of the whole person appearance, and then to learn a suitable metric that reflects the visual camera-to-camera transition. Hence, in contrast to methods that match features directly in the feature space using some standard distance measure, metric learning has the advantage that even less distinctive features, which need not capture the visual invariance between different camera views, are sufficient for achieving high matching performance. Moreover, since the learned metric inherently emphasizes or attenuates directions in the feature space based on their importance for the given task, it can also be seen as a discriminative feature selector. Just like in the case of discriminative methods, to estimate such a metric, a training stage is necessary. However, once learned, metric learning approaches are very efficient during evaluation, since additionally to the feature extraction and the matching, only linear projections have to be computed.

The goal of metric learning is to adapt some pairwise real-valued metric function, say the Mahalanobis distance: $d_M(x, x') = \sqrt{(x - x')^T M (x - x')}$ to the problem of interest using the information brought by training examples. Most methods learn the metric (here, the positive semi-definite matrix M in d˙M) in a weakly-supervised way from pair or triplet based constraints of the following form:

- Must-link / cannot-link constraints (sometimes called positive / negative pairs):

  $X^+ = (x_i, x_j)$: x_i and x_j should be similar

  $X^- = (x_i, x_l)$: x_i and x_l should be dissimilar

- Relative constraints (sometimes called training triplets):

  $R = (x_i, x_j, x_l)$: x_i should be more similar to x_j than to x_l

A metric learning algorithm basically aims at finding the parameters of the metric such that it best agrees with these constraints (see Figure 6.3 for an illustration), in an effort to approximate the underlying semantic metric. This is typically formulated as an optimization problem that has the following general form:

$$\min_M l(M, X^+, X^-, R) + \lambda R(M) \tag{6.6}$$

where $l(M, X^+, X^-, R)$ is a loss function that incurs a penalty when training constraints are violated, $R(M)$ is some regularizer on the parameters M of the learned metric and $\lambda \geq 0$ is the regularization parameter. State-of-the-art metric learning formulations essentially differ by their choice of metric, constraints, loss function and regularizer.

Some of popular algorithms to learn matrix M from a set of vector pair $X = \{x_{ij} | i = 1 : m, j = 1 : n\}$ are LMNN [110],ITML [27] and KISSME [49]. However, for our experiments, we

[2] http://www.vision.caltech.edu/html-files/archive.html

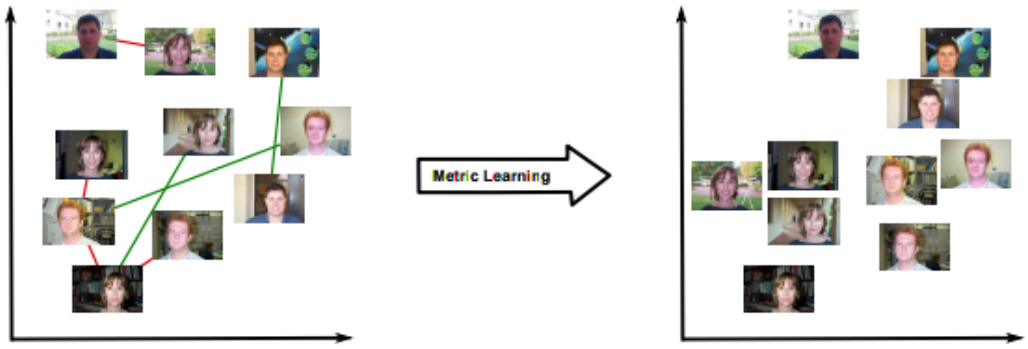


Figure 6.3: Illustration of metric learning applied to a face recognition task. For simplicity, images are represented as points in 2 dimensions. Pair wise constraints, shown in the left pane, are composed of images representing the same person(must-link, shown in green) or different people(cannot-link, shown in red). We wish to adapt the metric so that there are fewer constraint violations (right pane). Images are taken from the Caltech Face dataset. [2]

use KISSME [49] for its simplicity, low computation cost and effectiveness under challenging conditions.

Metric learning sampling method is illustrated in 6.4. In order to learn the metric M, we select positive samples (x_i, x_j) and negative samples (x_i, x_l) as follow:

- Tracklet segments: The training trajectories are divided into fixed size tracklet segments.
- A positive sample is a pair of GMM component means of two segmented tracklets belonging to the same GroundTruth people.
- A negative samples is a pair of GMM component means of two segmented tracklets belonging to the different GroundTruth people.

KISSME algorithm assumes independent Gaussian generation processes with parameters $\theta^+ = (0, \Sigma^+)$ and $\theta^- = (0, \Sigma^-)$ for positive and negative pairs (x_i, x_j) and (x_i, x_l), respectively. We estimate parameter of matrix M using KISSME by:

$$M = (\Sigma^{+^{-1}} - \Sigma^{-^{-1}}) \tag{6.7}$$

where Σ^+ and Σ^- are feature difference covariance metrics of positive and negative classes, respectively. Given pair associations, the covariance matrices Σ^+ and Σ^- are computed as follows:

$$\Sigma^+ = \sum_{(x_i, x_j) \in X^+} (x_i - x_j)(x_i - x_j)^T \tag{6.8}$$

$$\Sigma^- = \sum_{(x_i, x_l) \in X^-} (x_i - x_l)(x_i - x_l)^T \tag{6.9}$$

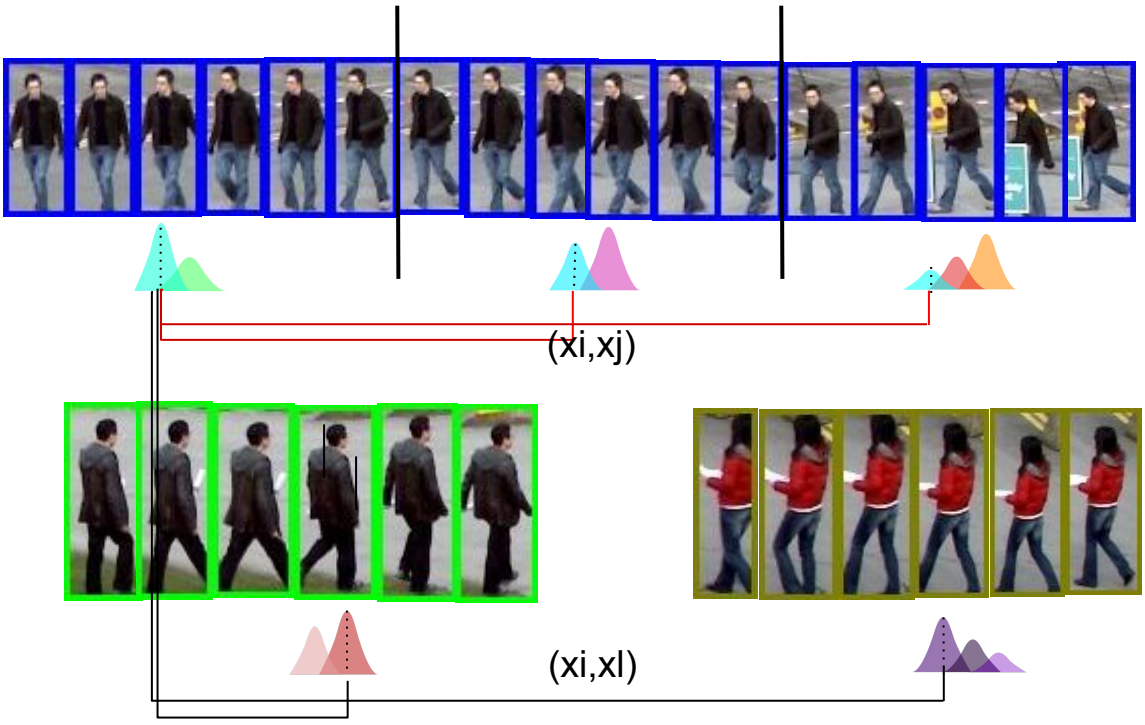


Figure 6.4: Metric learning sampling.

6.3.3.2 Tracklet representation similarity

Similarity metric plays an essential role in comparing two candidate tracklets' representations. Similarity of two tracklet representations is defined as the sum of similarities between the corresponding appearance models. Given the distance between two appearance models $d(M_i^{p,f}, M_j^{p,f})$ of tracklet representations ∇_{Tr_i} and ∇_{Tr_j}, we can convert this distance into similarity using Gaussian similarity kernel as follow:

$$Sim(\nabla_{Tr_i}, \nabla_{Tr_j}) = \sum_{p \in P, f \in F} \exp\left(-\frac{\overline{d(M_i^{p,f}, M_j^{p,f})} - \gamma_j^{p,f}}{(\beta_j^{p,f} - \gamma_j^{p,f})}\right) \tag{6.10}$$

where $P = \{full, upper, lower\}$ and $F = \{HOG, LOMO, MCSH, CH\}$, $\beta_j^{p,f}$ and $\gamma_j^{p,f}$ are the maximum and minimum normalized distance between tracklet representation ∇_{Tr_j} and representations of its candidates Can_j, respectively. The definitions of Can_j is presented in chapter 3.

$\overline{d(M_i^{p,f}, M_j^{p,f})}$ is a maximum normalized distance between two appearance model (part p and feature f) corresponding to tracklet representations ∇_{Tr_i} and ∇_{Tr_j}:

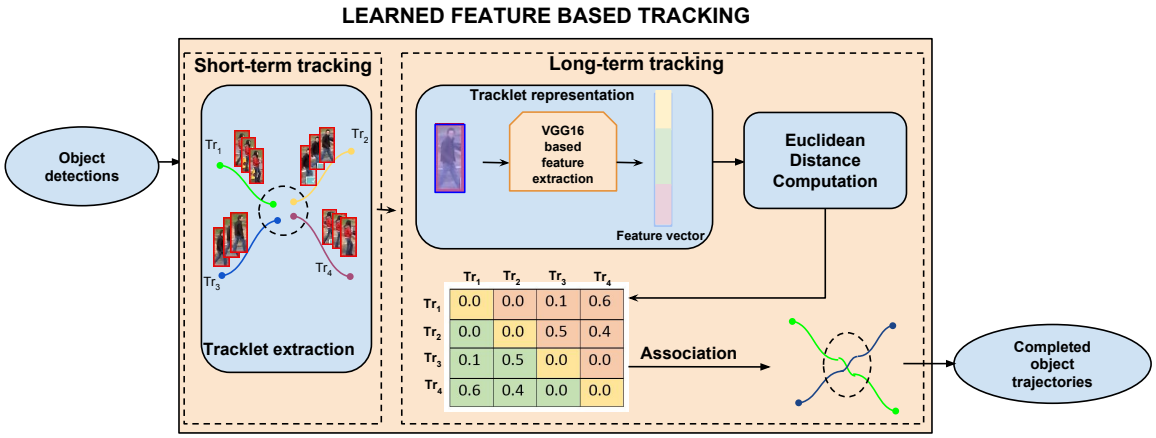


Figure 6.5: The proposed learned feature based MOT framework.

$$\overline{d(M_i^{p,f}, M_j^{p,f})} = \frac{d(M_i^{p,f}, M_j^{p,f})}{max_{\hat{j} \in Can_i} d(M_i^{p,f}, M_{\hat{j}}^{p,f})} \tag{6.11}$$

The distance between two appearance models is defined as sum of distance between GMM components weighted by their prior probabilities:

$$d(M_i^{p,f}, M_j^{p,f}) = \sum_{k_1=1:K_i^{p,f}, k_2=1:K_j^{p,f}} \pi_{k_1} \pi_{k_2} d(G_{i,k_1}^{p,f}, G_{j,k_2}^{p,f}) \tag{6.12}$$

where $G_{i,k}^{p,f}$ is the component k of $M_i^{p,f}$ with corresponding prior π_k and $K_i^{p,f}$ and $K_j^{p,f}$ are numbers of components of $M_i^{p,f}$ and $M_j^{p,f}$, respectively. For a pair of GMM component means (x_i, x_j), squared Mahalanobis distance of two GMM components is defined as:

$$d^2(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j) \tag{6.13}$$

where M is a learned metric from the offline learning phase.

6.4 Learned feature based framework

The learned feature based MOT framework is illustrated in figure 6.5. The framework describes the interaction between short-term and long-term trackers in every time-window Δt. The objectives of both trackers are similar to those in the hand-crafted feature based MOT framework. However, in the long-term tracking algorithm, CNN features extracted by the modified-VGG16 based feature extractor (illustrated in figure 6.6) are used to represent a person. All tracklet representations in two consecutive time-windows $[\Delta_{t-1}, \Delta_t]$ are stacked for the

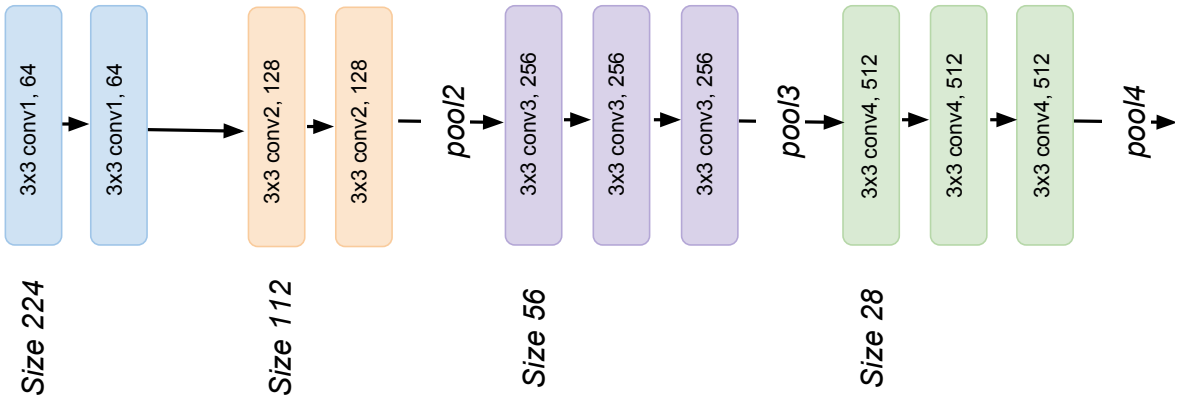


Figure 6.6: The modified-VGG16 feature extractor.

later tracklet association step. In order to compute the tracklet affinity, the Euclidean distance is applied to compare two corresponding tracklet representations. Finally, the tracklet association process is performed by a bipartite graph optimization, typically Hungarian algorithm.

6.4.1 Modified-VGG16 based feature extractor

For MOT task, we retain the structure of VGG16 from the first to the forth convolution layer group except the first max-pooling layer as described in figure 6.6. The size of kernels for all layers is fixed to 3×3. In particular, the first convolution layer group - *conv1* - has two convolutional layers with 64 filters per each. Local response normalization is used for the output of *conv1*, which is then passed to second convolutional layer group - *conv2*. *Conv2* has two convolutional layers (128 filters per each) followed by a max-pooling layer - *pool2*. The third - *conv3* - and the forth - *conv4* - convolutional groups have similar architecture which has three convolutional layers (256 and 512 filters per each), followed by max-pooling layers - *pool3* and *pool4*, respectively. The output of max-pooling layer - *pool2* - of the second convolutional layer group *conv2* is passed to the third convolutional layer group 3 - *conv3*. Then the output of max-pooling layer of the third group - *pool3* - is passed to the forth group - *conv4*. The extracted feature vector FV_i^t from node O_i^t is the output of max-pooling layer *pool4*.

6.4.2 Tracklet representation

The representation of tracklet $Tr_i = \{O_i^m, O_i^{m+1}, ...O_i^{n-1}, O_i^n\}$ using learned features extracted by the modified-VGG16 based feature extractor is defined as follow:

$$\nabla_{Tr_i} = mean(FV_i^t) \quad t \in < min(m, n-\Delta), n > \tag{6.14}$$

where FV_i^t is the CNN feature vector extracted from nodes O_i^t. The tracklet representation is computed by the mean of feature vectors extracted from a recent defined number of nodes belonging to tracklet Tr_i. We set this number to Δ in experiment.

6.5 Data association

In the online phase, we use the learned metric for hand-crafted features or the Euclidean distance for learned features to compute tracklet representation similarity. Then, the framework tries to calculate the global linking scores of a tracklet using candidates from relationship set of the tracklets. Similarity matrix S=$\{m_{ij}\}$ is constructed with similarity scores between all of the candidates, where i=1..n, j=1..n, and n is the number of tracklets in current time interval: $[\Delta_{t-1}, \Delta_t]$. If tracklet Tr_j is in a candidate of tracklet Tr_i, the similarity of the pair (Tr_i, Tr_j) is calculated based on the distance between two corresponding tracklet representations ∇_{Tr_i} and ∇_{Tr_j}. In particular, Mahalanobis distance $m_{ij} = Sim(\nabla_{Tr_i}, \nabla_{Tr_j})$ is applied for hand-crafted features while the Euclidean distance $m_{ij} = 1 - \|\nabla_{Tr_i} - \nabla_{Tr_j}\|_2$ is applied for CNN features. Otherwise, it is set to zero in the similarity matrix. Once the cost matrix is computed, the optimal association pairs, which minimize the global association cost in S, are determined using Hungarian algorithm.

6.6 Experiments

In this section, the performances of both proposed tracking features and tracking algorithm are measured. In the first part, the effects of recent hand-crafted features proposed for Re-id (LOMO and MCSH), the typical tracking features (HOG and CH(RGB)), the hand-crafted feature combination (LOMO + MCSH + HOG + CH) and learned features (CNN) on tracking performance are compared. Then, in the second part, the evaluation of the proposed hand-crafted feature based tracker and some state-of-the-art methods are shown. The performance of CNN feature evaluated in newest MOT dataset MOT17 is discussed in the experiment chapter.

6.6.1 Tracking feature comparison

We evaluate the effects of proposed features on tracking performance by testing the proposed tracking framework using these features on sequence PETS2009-S2L1-View1. The tracking performance is measured by popular metrics MT, ML, MOTA and MOTP and the quantitative comparison is shown in table 6.1. The features MCSH and LOMO are proposed for Re-id issue but are really efficient in tracking. The efficiency is shown by the improvements of tracking performance (an increase of around 25% in MT, a decrease by nearly a half measured by ML, an increase by a double in MOTA and 5% in MOTP) when applying these features compared to the

Feature	MT(%)↑	ML(%)↓	MOTA(%)↑	MOTP(%)↑
HOG - KISSME	47.6	38.0	37.3	67.7
CH(RGB) - KISSME	61.9	19.0	48.1	69.5
MCSH - KISSME	76.2	14.3	79.1	72.2
LOMO - KISSME	76.2	14.3	78.5	74.9
HOG+CH+LOMO+MCSH - KISSME	81.0	9.5	82.2	75.3
CNN - Euclidean distance	81.0	9.5	80.4	72.7

Table 6.1: Quantitative analysis of performance of tracking features on PETS2009-S2/L1-View1. The best values are marked in red.

typical tracking features HOG and CH(RGB). The feature HOG is the least reliable compared to others to characterize people in this sequence because the texture of people are similar. The features MCSH and LOMO show their effectiveness in finding out the invariant information of people over changes of viewpoint and person pose. Therefore, they are useful for the tracker to identify people when they leave and come back to the scene as well as change the motion abruptly. Moreover, features proposed for Re-id (LOMO, MCSH) not only based on color but also consider additionally about the spatio information to distingwuish people. Therefore, these features are more efficient than color based feature (CH(RGB)) (the improvements of 15% in MT, nearly 5% in ML and MOTP and especially 30% in MOTA). Finally, the proposed tracker when combining all features achieve the best performance with 81%, 9.5%, 82.2% and 75.3% measured by metrics MT, ML, MOTA and MOTP, respectively. It slightly improves the tracking performance (nearly 5% metric MT, ML and MOTA)compared to use only features MCSH or LOMO.

Even metric learning methods are powerful than the Euclidean distance in computing the affinity between objects, applying the Euclidean distance is independent to the training data. This advantage of the Euclidean distance makes the tracker be applicable to the real-world applications. In this experiment, we combine CNN features with the Euclidean distance to build a MOT framework. The results in table 6.1 show that the performance of learned feature based framework is equal or better than other referenced frameworks which use hand-crafted features plus metric learning on MT and ML metrics. Its performances measured by MOTA and MOTP metrics are less than the combination of selected hand-crafted features. However, no training step required makes the proposed CNN feature based framework more generic. Therefore, depending on the requirement of applications as well as the availability of training data, we could choose the most appropriate MOT algorithm to each other.

Method	MOTA(%)↑	MOTP(%)↑	GroundTruth	MT(%)↑	PT	ML(%)↓
Shitrit *et al.* [11]	0.81	0.58	21	–	–	–
Bae *et al.-global association* [8]	0.73	0.69	23	100	0	0.0
Chau *et al.* [20]	62.3	63.7	21	76.2	9.5	14.3
Ours ([20] + Proposed approach)	88.4	75.2	21	81.0	9.5	9.5

Table 6.2: Quantitative analysis of our method, the short-term tracker [20] and other trackers on PETS2009-S2/L1-View1. The best values are printed in red.

6.6.2 Tracking performance comparison

In this section, we evaluate our MOT framework with other state-of-the-art tracker on some sequences in public datasets including PETS2009-S2/L1-View1 and ParkingLot1. All compared trackers use hand-crafted features to represent a person. A short-term tracker using different person appearance descriptors [20] is selected to experiment in our framework. We use the public detection and evaluation method to get the fairly comparisons with other state-of-the-art trackers. We spend our discussion of the performance of our CNN-feature-based-MOT-framework on the newest benchmark dataset - MOT17 in the experiment chapter.

On sequence **PETS2009-S2/L1-View1**, the short-term tracker [20] and the proposed tracker share the detection public on website [3] and MOT evaluation toolkit [78] developed by STARS team, INRIA Sophia Antipolis while other trackers use their own detection and MOT evaluation code. From the quantitative results in table 6.2, the proposed tracker does not have as good results as tracker [8] on metric MT and ML. However, these compared trackers use the different ground-truth and detection. The detector applied by tracker [8] localizes completely all people in the video while the detection used by the proposed tracker totally loses two people. Furthermore, when people leave and come back to the scene, Groundtruth used by the proposed trackers set the same identity to these people while the Groundtruth used by tracker [8] sets different identities to them.In this part, in order to have a fair comparison, we focus on comparing the tracking performance of the short-term tracker [20] and the proposed tracker. The proposed method significantly improves the short-term tracker [20] tracking performance measured by almost of metrics. In particular, 26% on metric MOTA, 12% on metric MOTP, 5.8 % on metric MT and 4.8% on metric ML.

On **sequence Parkinglot1**, we use the detection and MOT evaluation toolkit public in website [4] to compare our tracking performance with publicly annotated data. The results of our tracker, the short-term tracker [20] and others trackers are shown in table 6.3. Compared to the short-term tracker [20], the proposed approach improve the tracking performance of

[3] http://www.milanton.de/data/
[4] http://crcv.ucf.edu/data/ParkingLOT/

Trackers	MOTA(%)↑	MOTP(%)↑	MT(%)↑	ML(%)↓	FP(#)↓	FN(#)↓	ID Sw(#)↓	Frag(#)↓
PMPT[95]	79.3	74.1	-	-	-	-	-	-
H2T [111]	88.4	81.9	78.57	0	-	-	21	-
GMCP [90]	90.43	74.1	-	-	-	-	-	-
PMT[20]	78.1	69.3	57.14	28.57	472	1056	10	114
RBT-Tracker(Hand-crafted features) - (Ours)	84.5	74.4	78.57	0	325	925	7	99

Table 6.3: Quantitative analysis of our method, the short-term tracker [20] and other trackers on ParkingLot1. The tracking results of these methods are public on UCF website. The best values are printed in red.

the short-term tracker [20] on most metrics. Dominantly, on metric ML, the proposed tracker keeps track all people and improves 28.57 % while fully tracking more 2 people occupied by 13.29% on metric ML. There are remarkable decreases on other metric including FP, FN, ID Sw and Frag. With other trackers, only tracker [111] and ours are evaluated by MT, ML and IDSw. Both methods have the same performances on MT, ML. While [111] has higher results than ours on metrics MOTA and MOTP, our method reduces two-third of IDSw errors. [90] is evaluated only using MOTA and MOTP metrics. The performance of this method is better than ours on MOTA but it performs worse when using MOTP. With [95], on both metrics MOTA and MOTP, our method has better performances in comparison.

6.7 Conclusions

We have proposed a robust multi-person tracking method which integrates short-term and long-term trackers into a two-step comprehensive framework. The proposed method works in online mode and can track person in unknown videos. It also effectively addresses some of the highly challenging problems in MOT such as mis-detection, person appearance changes by occlusion, pose or illumination variations, etc.. by the extension of person appearance features (hand-crafted and CNN features) and metric learning methods proposed for Re-id domain to MOT. The effectiveness and robustness of our method are verified by extensive experiments compared with state-of-the-art trackers. The evaluation part prove that the features which are powerful in Re-id are obviously effective to MOT.

Future work We are trying to apply the current object features such as fine-tuned deep features, Gaussian of Gaussian (GOG) [68] (proposed for Re-id) to enhance the tracking performance.

7

EXPERIMENT AND COMPARISON

7.1 Introduction

In this chapter, we evaluate the performance of our proposed approaches compared to the state-of-the-art trackers on two most popular benchmark datasets MOT15 and MOT17. However, evaluating all proposed approaches on these datasets implies numerous experiments. To reduce the training cost but still keep a fair comparison with the state-of-the-art trackers, we conduct the experiments in two steps. First, we select the best of the three proposed approaches by comparing their performances on three public sequences: PETS2009-S2/L1-View1 in PETS2009 dataset, TUD-stadmitte in TUD dataset and ParkingLot1 in ParkingLot dataset with a unique system parameter setting, detection, groundtruth and evaluation toolkit. We choose these sequences from three above datasets because these sequences ensure a large diversity of video scenes: Objects have chaotic movements, are occluded by other objects or background, leave and come back to the scene in PETS2009-S2/L1-View1. In TUD-stadtmitte, the video scene has low illumination conditions, is captured with narrow viewing camera angle and has frequent and strong object occlusions. In ParkingLot1, objects have similar appearance and move together in groups. In the second step, as a representative of our three approaches, the best tracker is compared with other trackers from the the state-of-the-art over the most popular benchmark datasets MOT15 and MOT17. These datasets are much more complex with larger diversity of video scenes. These two benchmark datasets also provide a public evaluation method as well as detection and ground-truth to have a fair comparison between trackers. Experiments show that our tracker performs well when compared to state-of-the-art tracking algorithms. For the more convinced evaluation, all compared trackers share the detection,

ground-truth and evaluation method.

7.2 The best tracker selection

The selection is performed based on the performance comparison of the proposed trackers over three public sequences PETS2009-S2/L1-View1, TUD-stadmitte and ParkingLot1 with the same sytem parameter setting, detection, groundtruth and evaluation method. The method having the highest performance is selected as the best tracker.

System Parameter setting Parameters controlling the discriminative feature selection adapting to the variation of video scenarios are automatically tuned. Otherwise, parameters have been found experimentally, and remains unchanged for all proposed trackers over the three selected image sequences. The same threshold $\theta = 0.3$ is used for all of the data association process. The size of a video chunk is fixed to 20 frames. The minimum size of a tracklet is set to 5 frames.

Detection, Groundtruth We use the public detection and ground-truth from the website [1] for the sequences PETS2009-S2/L1-View1, TUD-stadtmitte and from the website [2] for the sequence ParkingLot1.

Evaluation tool We use the ViSEVal toolkit [4] which is developed by STARS team, Sophia Antipolis to evaluate all proposed approaches on PETS2009-S2/L1-View1 and the public toolkit in website [3] for sequence ParkingLot1.

Baseline tracker All proposed approaches are long-term trackers which use tracklets generated by any short-term tracker as the input. In this experiments, we propose to use the short-term tracker PMT [20] as a baseline because this tracker is available, fast and uses a pool of object appearance features to track objects.

7.2.1 Comparison

The ***comparison*** among the proposed trackers and the baseline PMT on three public sequences (RFE-Tracker, CPT-Tracker and RBT-Tracker (hand-crafted features) are presented in hapter 4, chapter 5 and chapter 6, respectively) is shown in table 7.1.

In general, all proposed trackers have better performance than the baseline tracker. With sequence **PETS2009-S2/L1-View 1**, the tracking performance increases around 25% on MOTA metric, around 10% on MOTP metric and reduces 5% on ML metric. **TUD-Stadtmitte** sequence challenges trackers by a low illumination, strong as well as frequent occlusions and narrow captured angle. Even tracking performances of the proposed approaches are better than of

[1] http://www.milanton.de/data/
[2] http://crcv.ucf.edu/data/ParkingLOT/
[3] http://crcv.ucf.edu/data/ParkingLOT/

Sequences	Trackers	MOTA(%)↑	MOTP(%)↑	GT	MT(%)↑	PT(%)	ML(%) ↓
PETS2009-S2/L1-View1	PMT [20]	62.3	63.7	21	76.2	9.5	14.3
	RFE-Tracker [80] + [20]	85.7	71.8	21	76.2	14.3	9.5
	CPT-Tracker [79] + [20]	86.8	73.2	21	76.2	14.3	9.5
	RBT-Tracker (hand-crafted features) [81] + [20]	88.4	75.2	21	81.0	9.5	9.5
TUD-Stadtmitte	PMT [20]	45.3	61.9	10	60.0	40.0	0.0
	RFE-Tracker [80] + [20]	46.8	64.7	10	70.0	30.0	0.0
	CPT-Tracker [79] + [20]	47.3	65.6	10	70.0	30.0	0.0
	RBT-Tracker (hand-crafted features) [81] + [20]	51.4	67.1	10	70.0	30.0	0.0
ParkingLot1	PMT [20]	78.2	69.3	14	57.14	14.29	28.57
	RFE-Tracker [80] + [20]	80.1	70.7	14	71.43	14.29	14.28
	CPT-Tracker [79] + [20]	80.8	71.5	14	64.29	28.57	7.14
	RBT-Tracker (hand-crafted features) [81] + [20]	84.5	74.4	14	78.57	21.43	0.0

Table 7.1: Quantitative analysis of the proposed trackers and the baseline. The best values are marked in red.

the baseline, however, the improvements are still modest. There are increases of tracking performance on MOTA (around 3%), MOTP (around 4%) and MT (10%), particularly. The experiment on **ParkingLot1** sequence also shows the improvement of propose trackers over the baseline on most of metrics. Specially, the RBT-Tracker (hand-crafted features) [81] outperforms the baseline on MOTA from 78% to 85%, on MOTP from 69% to 75%, on MT from 57.14% to 78.57% and reduces the ratio of mostly lost objects (metric ML) from 28.57% to 0%. It proves that the tracking quality is improved when we extend the detection-based-tracking to the tracklet-based-tracking. Object representation accumulated from a tracklet corresponds to more reliable object information by consistent feature cues while object features achieved from an object detection is sensitive to noise. Therefore, object information based on tracklet features is more efficient and reliable than the one based on still object detection features.

Moreover, the experiments on all sequences show that RBT-Tracker (hand-crafted features) [81] is the best over proposed approaches. While RFE-Tracker [80] and CPT-Tracker [79] have the same performance measured by almost all metrics, RBT-Tracker (hand-crafted features) [81] is dominant over these two trackers. Precisely, there are increases of 4% on MOTA, around 3% on MOTP over all sequences, around 5% and 10% measured by MT metric on PETS2009-S2/L1-View1 and ParkingLot1, respectively. With ML metric, RBT-Tracker (hand-crafted features) [81] can keep track all objects on ParkingLot1 and TUD-Stadmitte and increases average 10% over two remaining trackers on ParkingLot1. However, the detector totally loses 2 objects over 21 ground-truth objects in sequence PETS2009-S2/L1-View1, therefore the proposed tracking algorithms cannot improve tracking quality measured by ML metric. The tracking performance remains unchanged over the three proposed approaches on ML metric (9.5%).

In conclusion, when we apply the same configuration in all experiments, the quantitative analysis shows that the proposed trackers improve the baseline's performance on all sequences. Furthermore, the RBT-Tracker (hand-crafted features) [81] is dominant to others. Therefore, in the upcoming part of experiment, we choose RBT-Tracker (hand-crafted features) [81] as the

representative of proposed approaches to compare with other trackers from the state-of-the-art.

7.3 The state-of-the-art tracker comparison

In this section, the performance of selected tracker $RBT-Tracker$ (hand-crafted features) and $RBT-Tracker$ (CNN features) are evaluated and compared with other trackers from the state-of-the-art on two benchmark datasets MOT15 and MOT17, respectively.

7.3.1 MOT15 dataset

The dataset MOT15 includes 22 sequences, half for training and half for testing. The dataset challenges the the state-of-the-art detectors and trackers by its complicated scenes such as low illumination and contrast, strong and frequent occlusion, objects' abrupt motion, crowed environment. First, the performance of our representative tracker $RBT-Tracker$ (hand-crafted features) with each sequence over all metrics is presented. The results show the impact of video scene to tracking performance. Second, we compare the performance of our representative tracker with the others from the state-of-the-art. Both offline and online methods are presented. Based on all experimental results on this dataset, the comparison of our tracker with the best offline tracker on the least and the most challenging sequences are analyzed.

7.3.1.1 System parameter setting

We set the system parameter values based on experiment and keep unchanged for all sequences in MOT15 dataset. In particular, the data association thresholds $\theta_1 = 0.3$ and $\theta_2 = 0.2$ are set for short-term and long-term trackers, respectively. The size of a video chunk is 20 frames while the minimum size of a tracklet is 5 frames.

7.3.1.2 The proposed tracking performance

Table 7.2 shows the performances of $RBT-Tracker$ (hand-crafted features) on 11 sequences belonging to MOT15 dataset. The tracking performances are sorted in the descend order measured by MT metric. The performances are really different among sequences, in particular, 61% objects on TUD-Crossing is mostly tracked while 0% object is tracked on both KITTI-16 and Venice-1.

7.3.1.3 The state-of-the-art comparison

A quantitative comparison between our approach and thirteen state-of-the-art tracking methods on challenging MOT15 dataset is shown in table 7.3. The tracking performances are computed over 11 sequences by the mean of each metric (MT, ML, MOTA, MOTP) and by the sum

Sequences	MT(%)↑	ML(%) ↓	MOTA(%)↑	MOTP(%)↑	FP(#) ↓	FN(#)↓	IDSw(#)↓	Frag(#)↓
TUD-Crossing	61.5	7.7	72.1	73.0	55	230	22	43
ETH-Jelmoli	20.0	33.3	28.9	72.7	648	1,110	45	66
ADL-Rundle-1	18.8	25.0	2.4	71.1	4,200	4,749	138	241
PETS09-S2L2	11.9	7.1	37.1	68.7	1,881	3,744	435	599
ADL-Rundle-3	9.1	20.5	24.7	71.6	2,400	5,062	197	219
AVG-TownCentre	8.8	27.4	21.8	68.8	1,717	3,671	203	396
KITTI-19	6.5	27.4	6.6	65.8	1,856	2,986	147	379
ETH-Linthescher	4.4	66.0	22.0	73.1	424	6,463	80	176
ETH-Crossing	3.8	46.2	24.8	73.4	86	662	6	12
KITTI-16	0.0	23.5	28.4	71.6	272	888	58	103
Venice-1	0.0	23.5	5.2	70.8	1,622	2,647	56	123

Table 7.2: Quantitative analysis of the proposed tracker's performance on dataset MOT15. The performance of the proposed tracker $RBT - Tracker$ (hand-crafted features) on 11 sequences is decreasingly sorted by MT metric.

of each indicator FP, FN, IDSw, Frags. In the evaluation part, we also categorize the state-of-the-art trackers into two groups: Offline and online tracking. Reasonably, offline trackers could have better performance than online trackers because of their beforehand objects' and scenario's information which is invisible to online trackers. In order to emphasis the robustness of the proposed approach which satisfies both requirements of online processing and high tracking performance, we show that our method not only outperforms online methods, but also has comparable performances compared to offline ones.

Looking at the table 7.3, we can see that trackers have best results on some metrics but not on all of the metrics. According to the analysis in [97], trajectory-based metrics, including MT and ML metrics, show the ratios of ground-truth trajectory's life span are covered by a output track (at least 80% for MT and at most 20% for ML, respectively). MT and ML are not influenced by the number of Frag or IDSw. As a result, these metrics give more information about the coverage of the trajectories rather than the ability of the tracker to reproduce them. On the other hand, results on metrics (MOTA, MOTP) and indicators (FP, FN) are too sensitive to detector errors. Particularly, FP and FN indicators are computed based on detector precision and recall, while MOTA and MOTP metrics show how much a tracker is able to find target positions and reject false alarms proposed by the detector. Therefore, in terms of tracking performance evaluation, trajectory-based metrics (MT and ML) are proved to be closer to end-user expectations than the others.

The performance of all selected trackers are sorted in descend order by the MT metric. Our approach outperforms both online and offline methods when ML metric and FN indicator are used. In details, our approach misses the least number of persons shown by ML metric and keeps track of the highest number of persons, shown by the lowest number of false negatives in FN. The results on these two metrics illustrate the remarkable improvement of our method com-

Methods	Trackers	MT↑	ML↓	MOTA↑	MOTP↑	FP↓	FN↓	IDSw↓	Frag↓
Offline	CNNTCM [107]	11.2±13.0	44.0	29.6± 13.9	71.8	7,786	34,733	712	943
	CEM [73]	8.5±20.3	46.5	19.3 ±17.5	70.7	14,180	34,591	813	1,023
	SiameseCNN [58]	8.5 ± 8.08	48.4	29.0 ±15.1	71.2	5,160	37,798	639	1,316
	ELP [69]	7.5 ± 6.3	43.8	25.0 ±10.8	71.2	7,345	37,344	1,369	1,804
	TBD [36]	6.4±13.4	47.9	15.9 ±17.6	70.9	14,943	34,777	1,939	1,963
	MotiCon [57]	4.7±8.6	52.0	23.1 ±16.4	70.9	10,404	35,844	1,018	1,061
Online	RBT-Tracker (hand-crafted features) - (Ours) [81]	9.0±17.4	36.9	20.6 ±18.7	70.3	15,161	32,212	1,387	2,357
	SCEA [121]	8.9 ± 6.6	47.3	29.1 ±12.2	71.1	6,060	36,912	604	1,182
	OMT_DFH [43]	7.1±11.3	46.5	21.2 ±17.2	69.9	13,218	34,657	563	1,255
	RNN_LSTM [72]	5.5±9.9	45.6	19.0 ±15.2	71.0	11,578	36,706	1,490	2,081
	EAMTTpub [93]	5.4 ±7.5	52.7	22.3 ±14.2	70.8	7 ,924	38,982	833	1,485
	RMOT [122]	5.3±9.8	53.3	18.6 ±17.5	69.6	12,473	36,835	684	1,282
	TC_ODAL [8]	3.2±7.9	55.8	15.1 ±15.0	70.5	12,970	38,538	637	1,716
	GSCR [31]	1.8±2.14	61.0	15.8 ±10.5	69.4	7,597	43,633	514	1,010

Table 7.3: Quantitative analysis of our method on MOT15 challenging dataset with state-of-the-art methods. The tracking results of these methods are public on MOTchallenge website. Our proposed method is named "MTS" on the website. The best values in both online and offline methods are marked in red.

pared to others. We reduce nearly one-forth the number of lost persons compared to methods [58, 57, 36, 122, 8] and nearly a half compared to [31] with ML metric. With FN, the number of false negatives in our method is reduced at least by 2,379 compared to [73] and at most by 11,421 compared to [31]. According to MT metric, our tracker performs remarkably better than trackers [57, 36, 93, 43, 72, 122, 31, 8] and in total has the second best performance. However, the best tracker [107] evaluated by this metric works only in offline mode.The proposed method achieves comparable results on MOTP metric but is not impressive for MOTA metric and indicators (FP, IDWs and Frag) compared to the other methods from the state-of-the-art.

On the other hand, the performances shown in table 7.2 shows that Venice-1 is the most challenging sequence while TUD-Crossing is the least one. In order to have a more detailed comparison, two methods having best performances measured by MT metric from table 7.3: $CNNTCM$ (working in offline mode) and the proposed tracker $RBT-Tracker$(hand-crafted features) (working in online mode) are evaluated on these sequences. The results are shown in table 7.4.

With the sequence **TUD-Crossing**, the proposed tracker $RBT-Tracker$ (hand-crafted features) outperforms the tracker $CNNTCM$ on almost all important metrics. Particular, there are increases of nearly 17% (from 46.2% to 61.5%) measured by MT, 12% by MOTA (from 60.5% to 72.1%) while the performance evaluated by MOTP is nearly equal. $RBT-Tracker$ (hand-crafted features) can track more objects than $CNNTCM$ which is shown by 16% decrease of ML (from 23.1% to 7.7%), a remarkable decrease of FN (from 352 to 230). In the opposite side, the tracker $CNNTCM$ has a better results on IDSw and Frag, including a reduction of 7 ID-switches (IDSw) and nearly 30 trajectory fragments (Frag).

The tracking performance of both compared trackers is illustrated on Figure 7.1. The left

Sequences	Trackers	Methods	MT↑	ML↓	MOTA↑	MOTP↑	FP↓	FN↓	IDSw↓	Frag↓
TUD-Crossing	CNNTCM(CVPR-2016) [107]	Offline	46.2	23.1	60.5	73.7	66	352	17	14
	RBT-Tracker (Hand-crafted features) - (Ours) [81]	Online	61.5	7.7	72.1	73.0	55	230	22	43
Venice-1	CNNTCM(CVPR-2016) [107]	Offline	0.0	41.2	19.2	74.1	582	3,091	12	13
	RBT-Tracker (Hand-crafted features) - (Ours) [81]	Online	0.0	23.5	5.2	70.8	1,622	2,647	56	123

Table 7.4: Comparison of the performance of proposed tracker [81] with the best offline method $CNNTCM$ [107]. The best values are marked in red.

column is the public detection used by both trackers. The middle and the right column are the performance of $CNNTCM$ and $RBT-Tracker$ (hand-crafted features), respectively. This sequence challenges trackers due to strong and frequent occlusions. As illustrated in Figure 7.1, where frames 33 and 55, frames 46 and 58, 86 and 92 show the scenes before and after of occlusions, tracking performance of both selected trackers are different. In particular, in order to solve the same occlusion case, the tracker $CNNTCM$ filters out the input detected objects (pointed by white arrows) and track only selected objects (pointed by red arrows). Thus, this is the pre-processing step (and not the tracking process) which manages to reduce the people detection errors. Meanwhile, $RBT-Tracker$ (hand-crafted features) still tries to track all occluded objects detected by the detector. The illustration completely explains why the $CNNTCM$ has worse performance than $RBT-Tracker$ (hand-crafted features) measured by MT, ML and FN.

Venice-1 is a difficult videos for trackers because of the low illumination and contrast and objects move in group, so the detection is not so good. From the quantitative results evaluated on this sequence shown in table 7.4, both trackers completely fail to track any object (0% of MT). On almost all remaining metrics, tracker $CNNTCM$ outperforms tracker $RBT-Tracker$ (hand-crafted features) except FN. These results are explained by illustrations on the Figure 7.2 and Figure 7.3. The Figure 7.2 shows tracking performance of these trackers for the occlusion case. The first, second and the last rows are the scene before, during, and after occlusion. The tracker $RBT-Tracker$ (hand-crafted features) tracks correctly the occluded objects (pointed by red arrows, marked by cyan and pink bounding-boxes). However, instead of tracking all occluded objects, tracker $CNNTCM$ filters the occluded object (pointed by the white arrow) and track only the object (marked by the yellow bounding-box). The Figure 7.3 shows how many detection are filtered and tracked by both trackers. The left column is the detection performance, the middle and right columns show tracking performance of $CNNTCM$ and $RBT-Tracker$ (hand-crafted features), respectively. $RBT-Tracker$ (hand-crafted features) tries to track almost all detected objects in the scene while $CNNTCM$ filters much more objects than $RBT-Tracker$ (hand-crafted features) and manages to track these filtered objects in order to achieve better tracking performance. In particular, $CNNTCM$ can reduce more than 18% lost objects (measured by ML metric), increases 14% of MOTA and modestly 3% of MOTP. The

more detections are filtered, the more false negatives (FN) increase. Therefore, $CNNTCM$ has more false negatives than $RBT-Tracker$ (hand-crafted features) (3,091 compared to 2,647, respectively). On the other side, the illustration shows that the people detection results include a huge number of noise. Because of keeping more fake detected objects to track, tracking performance of $RBT-Tracker$ (hand-crafted features) has more false positives than $CNNTCM$, 1,622 compared to 582 (measured by FP).

7.3.2 MOT17 dataset

The dataset MOT17 has 14 sequences including 7 sequences for training and 7 sequences for testing. On each sequence, trackers are provided 3 detections run by detectors DPM, F-RCNN and SDP. This dataset is a combination of challenges for both detectors and trackers, including high people density, strong and frequent occlusion, low illumination and contrast, abrupt person motion change caused by fast camera moving. As the experiment on MOT15, we first show the performance of our approach $RBT-Tracker$ (CNN features) on each sequence over all metrics. The results present the impact of video scene as well as quality of detection to tracking performance. Then, we compare the proposed approach with both offline and online state-of-the-art trackers and figure out factors which challenge trackers to address multi-person tracking problem.

7.3.2.1 System parameter setting

System parameter values have been found experimentally, remain unchanged for all 21 sequences. The thresholds $\theta_1 = 0.55$ and $\theta_2 = 0.2$ are set for all data associations in the short-term and long term trackers, respectively. The size of video chunk is set to 16 frames. The miminum tracklet size is set to 5 frames.

7.3.2.2 The proposed tracking performance

The performances of our proposed tracker named $RBT-Tracker$(CNN features) on dataset MOT17 are shown in table 7.5. The tracking performances vary on sequences and detection qualities. Comparisons based on the detection quality depict that the tracking performances on MT, ML metrics are correlated. In particular, the tracker using SDP detector has the best performances while the performance of tracker using DPM detector is the least. On the other hand, experimental results show that the tracking performances using the same detector are different among sequences. While the performances on sequences MOT17-03 is the best, the performance on sequences MOT17-08 and MOT17-14 are the worst.

For the explanation, two factors, including the detection quality and the video condition, are analyzed. Figure 7.4 illustrates the detection quality on MOT17 sequences. Mis-detection

Sequences	MT(%)↑	ML(%) ↓	MOTA(%)↑	MOTP(%)↑	FP(#) ↓	FN(#)↓	IDSw(#)↓	Frag(#)↓
MOT17-01-DPM	12.5	45.8	27.0	70.7	522	4144	41	129
MOT17-03-DPM	14.2	20.3	42.4	74.5	8933	50234	1076	1174
MOT17-06-DPM	14.0	44.6	42.6	72.3	865	5771	128	264
MOT17-07-DPM	10.0	40.0	35.1	73.1	1327	9457	177	331
MOT17-08-DPM	5.3	47.4	22.8	77.3	960	15205	136	233
MOT17-12-DPM	13.2	46.2	35.8	75.9	780	4749	31	87
MOT17-14-DPM	3.7	57.3	19.8	73.7	1423	13261	146	279
MOT17-01-FRCNN	12.5	37.5	26.2	76.1	1282	3436	42	71
MOT17-03-FRCNN	23.6	18.2	56.7	77.8	1858	43201	295	486
MOT17-06-FRCNN	26.1	28.8	49.8	78.2	995	4818	106	205
MOT17-07-FRCNN	6.7	26.7	33.3	74.9	1294	9807	166	332
MOT17-08-FRCNN	9.2	52.6	22.4	80.3	680	15649	70	111
MOT17-12-FRCNN	12.1	51.6	35.4	79.1	416	5159	24	41
MOT17-14-FRCNN	5.5	47.0	20.3	71.8	2446	12029	255	508
MOT17-01-SDP	33.3	25.0	39.6	73.2	972	2837	87	147
MOT17-03-SDP	43.9	12.8	69.5	76.2	3484	27934	520	1104
MOT17-06-SDP	34.2	31.5	53.9	75.6	1162	4160	116	184
MOT17-07-SDP	21.7	25.8	44.9	75.0	1119	8058	137	263
MOT17-08-SDP	13.2	44.7	28.8	77.5	863	14021	161	239
MOT17-12-SDP	18.7	45.1	39.8	78.0	607	4587	26	55
MOT17-14-SDP	4.9	40.9	29.4	73.0	1786	10976	293	400

Table 7.5: Quantitative analysis of the performance of the proposed tracker $RBT - Tracker$ (CNN features) on MOT17 dataset.

zones are marked by the red circles. While almost people on MOT17-03 sequence are well localized, the detector fails to detect people in some video conditions on MOT17-08 and MOT17-14 sequences. In particular, the red circle on MOT17-14 sequence figures out that the detector cannot detect a large group of people on the bus stop. This video condition challenges the detector because the people have small-size, stand stably and are strongly occluded. Meanwhile, MOT17-08 sequence contains some scenarios that challenge the detector, for examples, people concretely go together, are full occluded by others or are strongly partially occluded by background. Once the people are not detected, the tracker fails to identify them in the video scene.

Video conditions also affect to tracking performance. The statistic information in table 7.5 shows that the tracking performances on sequences are still modest where the highest MT value is 43.9% on MOT17-03 sequences and more than a half of sequences have performance measured by ML metric lower than 22% (MOT17-07, MOT17-08, MOT17-12 and MOT17-14 sequences). Figures 7.5, 7.6 and 7.7 show the failures (shown by big colored arrows) of the proposed tracker $RBT - Tracker$(CNN features) on MOT17-01, MOT17-08 and MOT17-14 sequences, respectively. Even all people are well detected in these cases, but the tracker cannot identify people throughout time. The information extracted from small bounding-boxes are not discriminative enough to characterize people shown in figures 7.5 and 7.7. The video condition

Methods	Trackers	MT↑	ML↓	MOTA↑	MOTP↑	FP↓	FN↓	IDSw↓	Frag↓
Offline	EDMT17 [23]	21.6	36.3	50.0 ± 13.9	77.3	32,279	247,297	2,264	3.260
	FWT [41]	21.4	35.2	51.3 ± 13.1	77.0	24,101	247,921	2,648	4,279
	JCC [44]	20.9	37.0	51.2 ± 14.5	75.9	25,937	247,822	1,802	2,984
	MHT_DAM [48]	20.8	36.9	50.7 ± 13.7	77.5	22,875	252,889	2,314	2,865
	IOU17 [14]	15.7	40.5	45.5 ±13.6	76.9	19,993	281,643	5,988	7,404
Online	RBT-Tracker (CNN features) - (Ours)	17.2	37.0	45.5 ± 12.7	75.9	33,774	269,493	4,033	6,643
	PHD_DCM [34]	16.9	37.2	46.5 ±13.8	77.2	23,859	272,430	5,649	9,298
	EAMTT [93]	12.7	42.7	42.6 ±13.3	76.0	30,711	288,474	4,488	5,720
	GMPHD_KCF [54]	8.8	43.3	39.6 ± 13.6	74.5	50,903	284,228	5,811	7,414
	GM_PHD [29]	4.1	57.3	36.4 ± 14.1	76.2	23.723	330,767	4,607	11,317

Table 7.6: Quantitative analysis of our MOT framework $RBT-Tracker$ (CNN features) on MOT17 challenging dataset with state-of-the-art methods. The tracking results of these methods are public on MOTchallenge website. Our proposed method is named "MTS˙CNN" on the website. The best values in both online and offline methods are marked in red.

on MOT17-08 sequence illustrated in figure 7.6 challenges the tracker by frequent and full occlusions and the illumination changes.

To sum up, the quantitative results in table 7.5 show the challenges for the proposed tracker are not only the video conditions but also the detection quality.

7.3.2.3 The state-of-the-art comparison

The table 7.6 show the quantitative comparison between our approach $RBT-Tracker$ (CNN features) and nine state-of-the-art trackers on benchmark dataset MOT17. These compared methods are categorized into offline and online tracking. The tracking performances are computed over 7 sequences with 3 different detectors: DPM, F-RCNN and SDP. The values shown in table 7.6 are computed by the mean of each metric (ML, ML, MOTA, MOTP) and by the sum of each indicator (FP, NP, IDSw, Frags). As the discussion in the experiments on MOT15 dataset, metrics MT and ML are proved to be closer to user expectations than the others. Therefore, the performance of all compared trackers are sorted in descent order by the MT metric. Generally, offline trackers with their beforehand information of objects and scenarios have better performance than online trackers. Comparing two trackers including $EDMT17$ - the best offline tracker and our approach $RBT-Tracker$ (CNN-features) - the best online tracker, we can see the modest increases of around 4.5% of metrics MT and MOTA, 1.7% of metric MOTP and a slight decrease of 0.7% of metric ML.

Our $RBT-Tracker$ (CNN features) is also compared with four other online tracking methods. The results show that $RBT-Tracker$ has the best performances on the metrics (MT, ML, FN and IDSw) and the second best performances on the metrics (MOTA, Frags). On MOTP metric, there is a slight decrease of 1.3% (from 77.2 % to 75.9%) when comparing our $RBT-Tracker$ and the best performance belonging to tracker PHD_DCM.

It is shown in the table 7.6 that the performance of state-of-the-art trackers are modest on this challenging benchmark dataset. The best results (only 21.6% and 50% measured by ML and MOTA metrics, respectively) belong to tracker $EDMT17$ which works in the offline mode. In order to analyze factors affecting to tracking, we illustrate the performances of $EDMT17$ - the best offline tracker, our proposed approach $RBT-Tracker$(CNN features) and PHD_DCM are the best and the second best online trackers, respectively on some challenging video conditions in figures 7.5, 7.6, 7.7.

Figure 7.5 illustrates some cases on **MOT17-01** sequence where all selected trackers fail to keep the identity of people even they are well detected. The visualization shows that people closing to the camera are correctly tracked while people being far from the camera are lost. The yellow arrows point lost people at the time instants that before and after occlusions, including frame pairs (69,165), (181,247), (209,311), respectively. It is proved that if a person is far from the camera(the detection bounding-box is small), the information extracted on this bounding-box is not discriminative enough to characterize this person to neighbourhood. Therefore, *tracking small people over occlusion* becomes a hard MOT task.

As the visualization on figure 7.6, the selected trackers also fail to recover the person ID after *strong partial or full occlusions* (pointed by red arrows) on **MOT17-08** sequence. Different to the challenge shown in figure 7.5, people appearance extracted from detection bounding-boxes are discriminative. However, people are strongly and frequently occluded by others (illustrated in frame pairs (126,219) and (219,274)) or background (shown in frame pairs (10,82) and (226,322)). This challenge prevents trackers from building a reliable representation to keep invariant person information over time.

The illustration on figure 7.7 focuses on the challenges of *fast camera moving and the high people density*, for examples, in frame 409 or 623 on **MOT17-14** sequence. The fast camera moving can cause the abrupt change of person motion. High people density obstacles not only detection but also tracking task. In this sequence, trackers are required to clarify the ambiguity of people standing (shown in frames 409 and 421) or walking (visualized in frames 161, 588 and 623) in a concrete group. The failures of selected trackers in identifying people are marked by orange arrows. The illustrations in frame pair (161, 199) show the affect of fast camera moving challenge to the performance of trackers. Meanwhile, frame pairs (409,421) and (588,623) figure out the tracking drifts caused by both of camera moving and high people density.

7.4 Conclusions

This chapter shows quantitative analyses of experiments of proposed trackers and the state-of the-art methods on two most common benchmark datasets: MOT15 and MOT17. These

analyses focus on two main issues: Evaluating the tracking performances of the proposed approaches with the state-of-the-art trackers, proving experimentally factors impacting to MOT quality.

Firstly, the proposed approaches are compared to each other to select the representative to compare with the state-of-the-art trackers. The representative is evaluated with both online and offline trackers. Reasonably, the offline trackers have better performances than the online trackers thanks to their beforehand information of objects as well as scenarios which are invisible to online trackers. However, on both datasets, the proposed tracker has the best performance compared to online methods on metrics ML and ML which are proved to be closer to end-user expectations than the others.

Secondly, the experimental results show that the proposed trackers as well as trackers from the state-of-the-art trend to have good or bad performances in the same sequences. In additions, the performances of the proposed tracker on sequences are correlated to the detection quality. Therefore, based on the experimental results, we can conclude that video conditions as well as detection quality are the factors which impact to MOT performance.

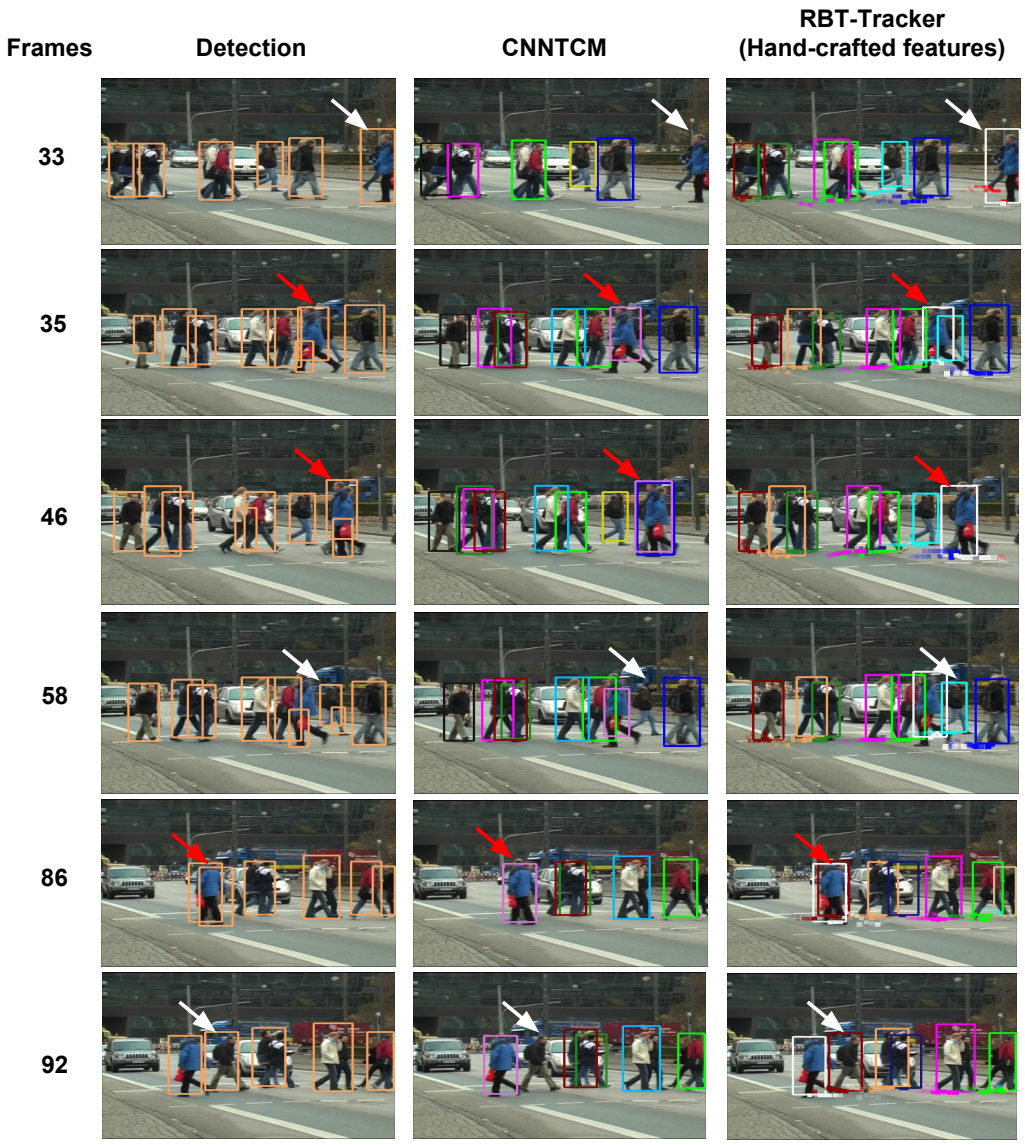
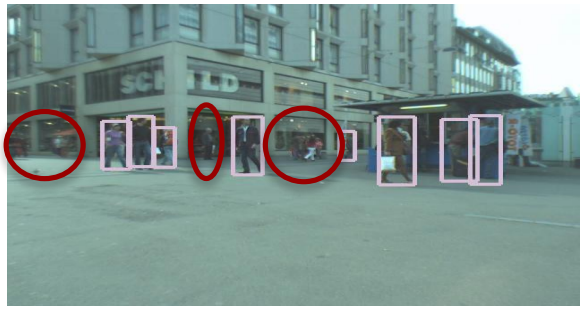


Figure 7.1: The tracking performance of $CNNTCM$ and $RBT-Tracker$ (hand-crafted features) with occlusion challenge on sequence TUD-Crossing. The left to right columns are the detection, the tracking performance of $CNNTCM$ and $RBT-Tracker$ (hand-crafted features), respectively. The top to bottom rows are the scenes at frame 33, 55, 46, 58, 86 and 92. In particular, in order to solve the same occlusion case, the tracker $CNNTCM$ filters out the input detected objects (pointed by white arrows) and track only selected objects (pointed by red arrows). Thus, this is the pre-processing step (and not the tracking process) which manages to reduce the people detection errors. Meanwhile, $RBT-Tracker$ (hand-crafted features) still tries to track all occluded objects detected by the detector. The illustration completely explains why the $CNNTCM$ has worse performance than $RBT-Tracker$ (hand-crafted features) measured by MT, ML and FN.

Figure 7.2: The illustration of the tracking performance of $CNNTCM$ and $RBT-Tracker$ (hand-crafted features) on sequence Venice-1 for the occlusion case. The left to right columns are the detection, the tracking performance of $CNNTCM$ and $RBT-Tracker$ (hand-crafted features) in order. The top to bottom rows are the scenes at frame 68, 81 and 85 which illustrate the scene before, during, and after occlusion, respectively. The tracker $RBT-Tracker$ (hand-crafted features) tracks correctly the occluded objects (pointed by red arrows, marked by cyan and pink bounding-boxes). However, instead of tracking all occluded objects, tracker $CNNTCM$ filters the occluded object (pointed by the white arrow) and track only the object (marked by the yellow bounding-box).

Figure 7.3: The noise filtering step of $CNNTCM$ and $RBT-Tracker$ (hand-crafted features) on Venice-1 sequence. The left to right columns are the detection, the tracking performance of $CNNTCM$ and $RBT-Tracker$ (hand-crafted features), respectively. The top to bottom rows are the scenes at frame 67, 166, 173, 209 and 239. $RBT-Tracker$ (hand-crafted features) tries to track almost all detected objects in the scene while $CNNTCM$ filters much more objects than $RBT-Tracker$ (hand-crafted features) and manages to track these filtered objects in order to achieve better tracking performance. The more detections are filtered, the more false negatives (FN) increase. Therefore, $CNNTCM$ has more false negatives than $RBT-Tracker$ (hand-crafted features). On the other side, the illustration shows that the people detection results include a huge number of noise. Because of keeping more fake detected objects to track, tracking performance of $RBT-Tracker$ (hand-crafted features) has more false positives than $CNNTCM$.

MOT17-01-SDP-det



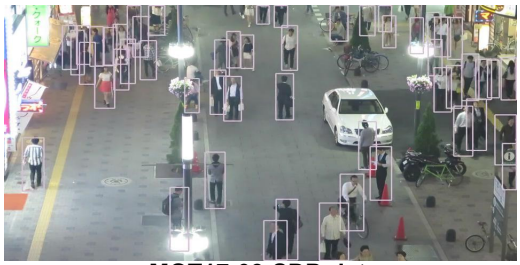
MOT17-06-SDP-det

MOT17-07-SDP-det

MOT17-08-SDP-det

MOT17-12-SDP-det

MOT17-14-SDP-det



MOT17-03-SDP-det

Figure 7.4: The illustration of the detection of sequences on MOT17 dataset. We use the results of the best detector *SDP* to visualize the detection performance. The red circles point out groups of people are not detected. Therefore, the tracking performance is remarkably reduced.

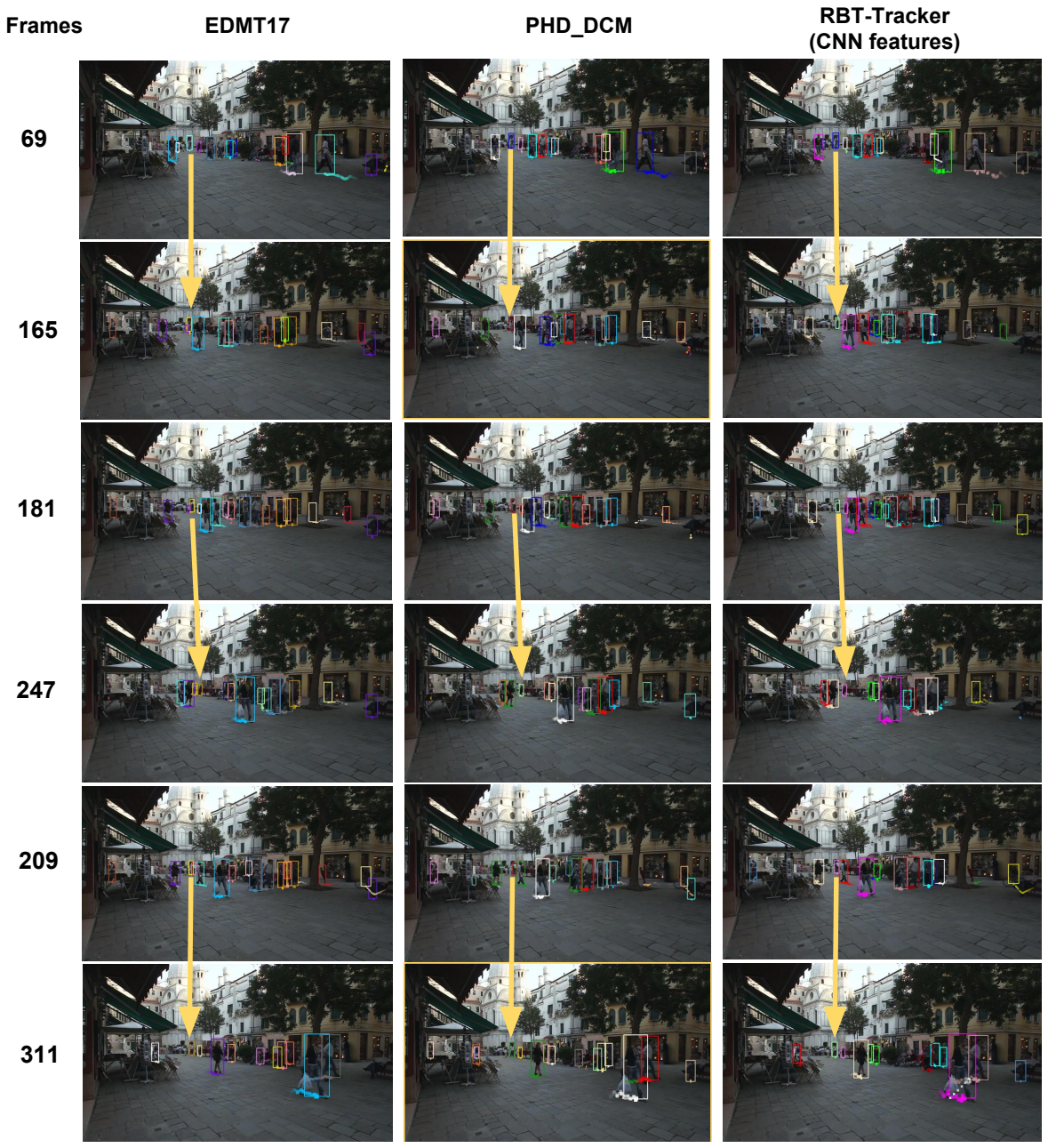


Figure 7.5: The illustration of the failures of state-of-the-art trackers on MOT17-01-SDP sequence. Frame pairs (69,165), (181,247) and (209,311) are the time instants at before and after occlusion, respectively. The yellow arrows show that selected trackers lose people after occlusion in the case that people are far from the camera and the information extracted from their detection bounding-boxes are not discriminative enough to characterize them with the neighbourhood.

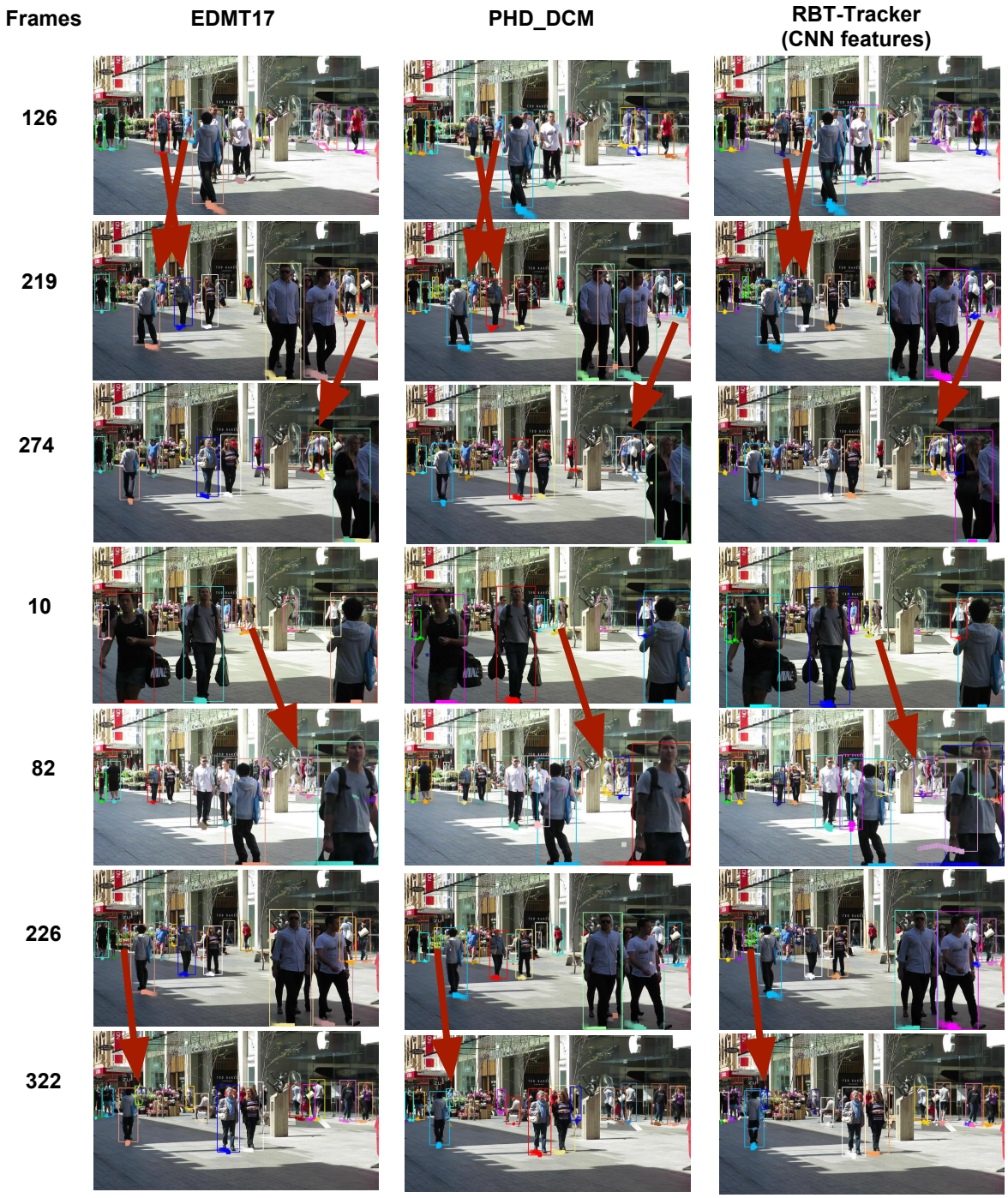


Figure 7.6: The illustration of the failures of state-of-the-art trackers on MOT17-08 sequence. All selected trackers fail to keep person ID over strongly and frequent occlusions. These occlusions are caused by other people (shown in frame pairs (126,219) and (219,274)) or background (shown in frame pairs (10,82) and (266,322)).

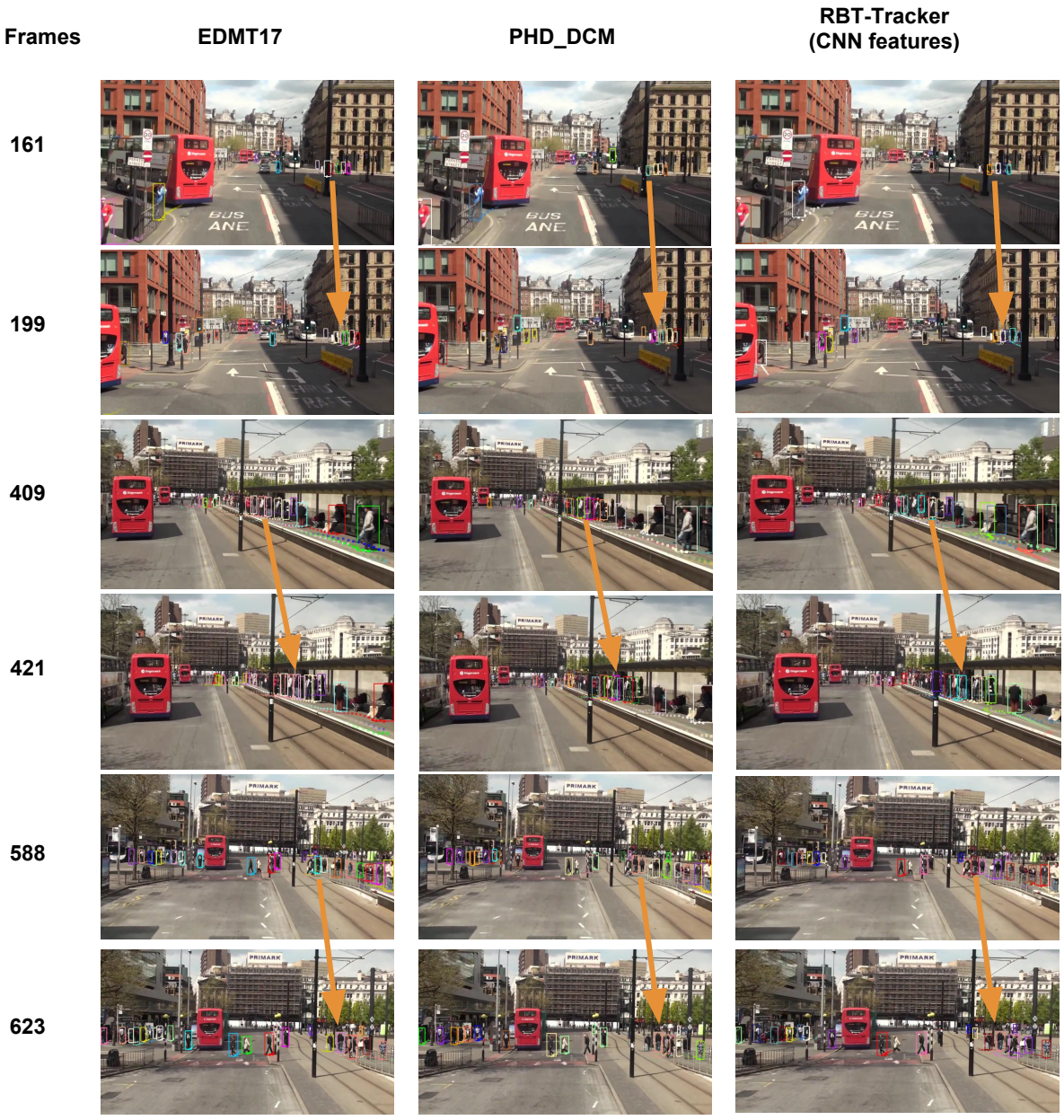


Figure 7.7: The illustration of the failures of state-of-the-art trackers on MOT17-14 sequence. The challenges of fast camera moving or high people density affect directly to the performance of selected trackers. Tracking drifts marked by orange arrows are caused by fast camera moving (shown in frame pair (161,199)) or by both high people density and camera moving (shown in frame pairs (409,421),(588,623)).

8

CONCLUSIONS

This chapter summarizes the content of the thesis, the chapter organized into 2 sections. The first section concludes about the contributions, including advantages as well as limitations of the proposed approaches. In the second section, the future work scope of research is highlighted.

8.1 Conclusion

In this thesis, different online tracking algorithms to tackle the task of multi-person tracking using a single camera view is researched. These approaches categorized into long-term (tracklet-based) tracking because they obtain the outputs from a short-term (detection-based) tracker which act as their inputs and they process these inputs with a time latency. These approaches link tracklets by building robust pairwise similarity between them, mostly based on their strong appearance. The experimental result proves these approaches by showing an increase in tracking performance and the ability of the trackers to handle more complex video scenarios. Each approach is summarized as follows.

- The first approach is termed as *Reliable Feature Estimation tracker* (RFE), which proposes a mechanism to automatically select reliable features that can discriminate tracklets in the current video scene. In particular, a tracklet is represented by a pool of six features, including their 2D area, 2D shape ratio, color histogram, dominant color, color covariance and constant velocity. Two sets are defined for each given tracklet: the candidate set is composed of ”can-match” tracklets while the neighbour set consists of ”cannot-match” tracklets to the given tracklet. In order to compute the affinity between a tracklet and its candidate, the proposed approach selects reliable features which cannot only discrim-

inate this tracklet from its neighbours but also enhance its similarity to the candidate. In particular, the reliability of each feature is established by a feature weight. The feature weight value is computed proportionally to the similarity between one tracklet with its candidate and inversely proportional to its distance from its neighbours. The reliable feature weights are computed overtime depending on the variation of video conditions. This method do not need any training step and works online with a short time buffering.

- The second approach is termed as *Context-based Parameter Tuning* (CPT). This method proposes a technique that tunes tracking parameters to adapt the tracker to the variety of video conditions. Instead of using only tracklet's individual features as in the first approach to characterize a tracklet, surrounding features including occlusion level, person density level and the contrast defining the surrounding context of this tracklet are added in this technique. The approach consists of two phases: offline parameter learning and online parameter tuning. In the offline phase, the optimal tracking parameters are learned for each tracklet by the simulated-annealing optimization method. In the online phase, each tracklet is used to retrieve the closest tracklet in the learned database to obtain the optimal tracking parameters. This approach outperforms the first approach because of the following reasons: it uses additionally tracklet surrounding information which has been experimentally proved to improve the tracking performance; it tunes tracking parameters such as tracking thresholds or feature parameters which are fixed by the first approach. This approach can be run online once the database of tracklets and their corresponding tracking parameters are learned. However, this approach requires an offline training step which necessitates the diverse collect of annotated videos to produce this algorithm generic.

- The third approach named *Re-id based Tracker* (RBT) takes full advantage of features (including hand-crafted and learned features) and matching methods proposed for Re-identification and adapting them to MOT. In order to represent a tracklet with hand-crafted features, a tracklet is represented by a multi-channel appearance model where each channel includes both spatial and appearance information. The offline metric learning method proposed for Re-id is applied for computing the tracklet affinity (Mahalanobis distance). In order to extend the features and methods proposed for Re-identification (working in an offline mode) to online tracking, the proposed approach processes the detection with a latency by using a sliding time-window. However, the similarity of training and testing data is required which limits the generality of the algorithm. In order to make this framework become generic, instead of using hand-crafted features, we represent a tracklet by CNN feature extracted from a pre-trained CNN model. Then, we associate the CNN feature-person representation with Euclidean distance into a comprehensive frame-

work which works fully online.

The comparison in chapter 7 shows that the third approach is the most robust among our proposed trackers on three benchmark datasets (MOT15, MOT17 and ParkingLot) and outperforms almost of selected state-of-the-art trackers according to standard MOT metrics.

8.1.1 Contributions

This thesis brings three following contributions to the state-of-the-art.

- The first approach contributes a simple and effective method which can automatically select reliable features to represent a person which helps the tracker discriminating tracklets to the variation of video conditions with no prior training step. Therefore, this method is generic and can be embedded into other tracking frameworks and does not incur any training cost.
- The second approach contributes a new method to tune tracking parameters for each tracklet independently instead of setting up globally for all tracklets in the video. This method ensures that tracking parameters are tuned to adapt the tracker to the variation of each tracklet's surrounding context which can be also different even if detection scenario is in the same video condition. This method can also tune a large number of tracking parameters by using an approximate optimization algorithm which does neither require the parameter independence nor a limitation on the number of variables. Therefore, this method can be applied to tune different tracking parameter sets of other trackers.
- The third approach is an extension of the features and methods proposed for the person Re-id process (working on offline mode) to online multi-person tracking. The experimental results prove that powerful features and methods proposed for Re-id task are also efficient in MOT.

8.1.2 Limitations

The limitations in the proposed approaches are summarized in this section. We divide these limitations into two groups: Theoretical and experimental ones.

8.1.2.1 Theoretical limitations

- The features used to characterize the tracklets can be redundant. For example, there are several color-based features in the tracklet feature pool utilized to characterize a tracklet. Tuning tracking parameters shared by redundant features may end-up in a non-converging loop.

- The surrounding features are not accurate enough to retrieve a learned tracklet from the database given by a testing tracklet as query.
- The performance of the proposed approaches is dependent on the quality of person detection and the short-term tracker. The low quality of person detection reduces the performance of short-term tracker. Therefore, the input of proposed long-term trackers generated by short-term tracker becomes less reliable.
- There is no back-track mechanism to correct the output of the short-term tracker which provides the input to proposed long-term trackers. The long-term approaches have a preprocessing step to filter unreliable tracklets with a heuristic manner. However, tracking errors, for example, a tracklet covering two groundtruth-persons or a tracklet generated by a tracking drift cannot be refined. A back-track mechanism is necessary to improve such limitations.
- The proposed tracker *CPT* may require a high processing time when looking for the best matched tracklet to retrieve the optimal tracking parameters in the huge learned database. Therefore, besides the requirement of the diversity of the learned database, the size of the learned database is also important.

8.1.2.2 Experimental limitations

- Experiments and comparisons between the proposed approaches and the state of the art are not performed entirely. Firstly, we only evaluate each approach on few video sequences and compare the performances of this tracker with state-of-the-art trackers at a given time of publication. Secondly, in order to reduce the training cost, instead of evaluate all proposed approaches, we select the best one to compare with current state of the art trackers. The selection is done by comparing the tracking performances of the proposed approaches together. However, this comparison is not completely validated due to the small number of video sequences.
- The detection and ground-truth of evaluated video sequences (PETs2009-S2/L1-View1, TUD-stadtmitte and TUD-crossing) are not shared between trackers from state-of-the-art. Therefore, the comparisons on these sequences are not fair enough, especially for the comparisons of the proposed approaches with other trackers.

8.2 Proposed tracker comparison

The presented approaches as well as their experimental performances show that these approaches can be distinguished through two properties: their generality and their effectiveness.

Trackers	The generality	The effectiveness
RFE	✓✓✓	✓
CPT	✓✓	✓✓
RBT (handcrated features - Mahalanobis distance)	✓	✓✓✓
RBT (CNN features - Euclidean distance)	✓✓✓	✓✓

Table 8.1: The proposed trackers can be distinguished through two properties: their generality and their effectiveness. The number of symbol ✓ stands for the generality or effectiveness levels of proposed trackers. The more number of symbols ✓ in a property is shown, the higher level of this property a tracker has.

Their comparison is shown in table 8.1. Depending on the application as well as the availability of training data, the most appropriate multi-person tracking algorithm is selected.

8.3 Future work

- A first interesting work to conduct is to learn recent powerful features such as deep features, which can better characterize person appearance. Some features can be added to better describe the tracklet surrounding context such as the color variance of tracklets or the complexity of person trajectories.
- A second work is to limit the redundancy of object features. This work can reduce the complexity of proposed approaches and enhance the effectiveness of feature based parameter tuning. We can limit the redundancy by proposing a mechanism to evaluate the contribution of each features to the tracking or by using dimension reduction algorithm to compact the person representation.
- A third task is to propose a method to index learned tracklets in a large learned database. The indexation can help to reduce the processing-time consumption for retrieving the closest learned tracklet to obtain the optimal tracking parameters. In order to index tracklets in the database, we can use the PH-tree indexing technique whose effectiveness is presented in [45].
- A back-track mechanism is needed to correct the errors of detector and short-term tracker which we use to generate the input for our proposed trackers. For each tracklet generated by a short-term tracker, we firstly can set the matching confidence between detections. The lower confidence matches could be temporary defined as ambiguities. In a second step, an evaluation mechanism could be provided to select the best matches between ambiguous nodes. Thank to this method, the short-term tracker errors could be corrected.

9

PUBLICATIONS

[1] Thi-Lan-Anh Nguyen, Duc-Phu Chau, and Francois Bremond. *Robust global tracker base on an online estimation of tracklet descriptor reliability*. In 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1-6, Aug 2015.

[2] Thi-Lan-Anh Nguyen, Francois Bremond, and Jana Trojanova. *Multi-object tracking of pedestrian driven by context*. In 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 23-29, Aug 2016.

[3] Thi-Lan-Anh Nguyen, Furqan-M. Khan, Farhood Negin, and Francois Bremond. *Multi-object tracking using multi-channel part appearance representation*. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1-6, Aug 2017.

BIBLIOGRAPHY

[1] Saad Ali and Mubarak Shah. Floor fields for tracking in high density crowd scenes. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 1–14, Berlin, Heidelberg, 2008. Springer-Verlag.

[2] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *CVPR 2011*, pages 1265–1272, June 2011.

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

[4] Nghiem Anh-Tuan, Francois Bremond, Monique Thonnat, and Ruihua Ma. A new evaluation approach for video processing algorithm. In *Proceedings of WMVC 2007*, February.

[5] Ivar Austvoll and Bogdan Kwolek. *Region Covariance Matrix-Based Object Tracking with Occlusions Handling*, pages 201–208. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[6] J. Badie, S. Bak, S.T. Serban, and F. Bremond. Recovering people tracking errors using enhanced covariance-based signature, 2012. AVSS.

[7] Julien Badie and Francois Bremond. Global tracker: an online evaluation framework to improve tracking quality, 2014. AVSS.

[8] S. H. Bae and K. J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1218–1225, June 2014.

[9] Slawomir Bak, Duc Phu Chau, Julien Badie, Etienne Corvee, Francois Bremond, and Monique Thonnat. MULTI-TARGET TRACKING BY DISCRIMINATIVE ANALYSIS ON RIEMANNIAN MANIFOLD. In *ICIP - International Conference on Image Processing - 2012*, volume 1 of *People re-identification and tracking from multiple cameras*, pages 1–4, Orlando, United States, September 2012. IEEE Computer Society.

[10] Yutong Ban, Sileye Ba, Xavier Alameda-Pineda, and Radu Horaud. Tracking Multiple Persons Based on a Variational Bayesian Model. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, volume 9914 of *Lecture Notes in Computer Science*, pages 52–67, Amsterdam, Netherlands, October 2016. Springer.

[11] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints, 2011. ICCV.

[12] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*, pages 3457–3464, June 2011.

[13] Charles Bibby and Ian Reid. Robust real-time visual tracking using pixel-wise posteriors. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 831–844, Berlin, Heidelberg, 2008. Springer-Verlag.

[14] E. Bochinski, V. Eiselein, and T. Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Aug 2017.

[15] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1515–1522, Sept 2009.

[16] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR 2011*, pages 1273–1280, June 2011.

[17] Gabriel J. Brostow and Roberto Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 594–601, Washington, DC, USA, 2006. IEEE Computer Society.

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

[19] Duc Phu Chau, Julien Badie, and Francois Bremond. Online tracking parameter adaptation based on evaluation. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 189–194, Aug 2013.

[20] Duc Phu Chau, Francois Bremond, and Monique Thonnat. A multi-feature tracking algorithm enabling adaptation to context variations. *CoRR*, abs/1112.1200, 2011.

[21] Duc Phu Chau, Francois Bremond, and Monique Thonnat. A multi-feature tracking algorithm enabling adaptation to context variations, 2011. ICDP.

[22] Duc Phu Chau, Monique Thonnat, Francois Bremond, and Etienne Corvee. Online parameter tuning for object tracking algorithms. *Image and Vision Computing*, 32(4):287 – 302, 2014.

[23] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong. Enhancing detection model for multiple hypothesis tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2143–2152, July 2017.

[24] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ECCV'12, pages 215–230, Berlin, Heidelberg, 2012. Springer-Verlag.

[25] Cristina Garcia Cifuentes, Marc Sturzel, Frederic Jurie, and Gabriel J.Brostow. Motion models that only work sometimes, 2012. BMCV.

[26] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[27] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 209–216, New York, NY, USA, 2007. ACM.

[28] A. Dehghan, S. M. Assari, and M. Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4091–4099, June 2015.

[29] V. Eiselein, D. Arp, M. PÃtzold, and T. Sikora. Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors. In *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, pages 325–330, Sept 2012.

[30] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[31] L. Fagot-Bouquet, R. Audigier, Y. Dhome, and F. Lerasle. Online multi-person tracking based on global sparse collaborative representations. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 2414–2418, Sept 2015.

[32] Loïc Fagot-Bouquet, Romaric Audigier, Yoann Dhome, and Frédéric Lerasle. Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 774–790, 2016.

[33] Wolfgang Forstner and Boudewijn Moonen. A metric for covariance matrices, 1999.

[34] Z. Fu, F. Angelini, S. Naqvi, and J. Chambers. Gm-phd filter based online multiple human tracking using deep discriminative correlation matching. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[35] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. J. Comput. Vision*, 73(1):41–59, June 2007.

[36] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):1012–1025, May 2014.

[37] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *IN ICCV*, pages 1458–1465, 2005.

[38] Izadinia H, Sallmi I, Li W, and Shah M. $(mp)^2t$: Multiple people multiple parts tracker, 2012. ECCV.

[39] Alexandre Heili, Adolfo Lopex-Mendez, and Jean-Marc Odobez. Exploiting long-term connectivity and visual motion in crf-based multi-person tracking, 2014. Idiap-RR.

[40] J.F. Henriques, R. Caseiro, and j. Batista. Globally optimal solution to multi-object tracking with merged measurements, 2011. ICCV.

[41] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. Improvements to frank-wolfe optimization for multi-detector multi-object tracking. *CoRR*, abs/1705.08314, 2017.

[42] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2420–2440, Dec 2012.

[43] Jaeyong Ju, Daehun Kim, Bonhwa Ku, David K. Han, and Hanseok Ko. Online multi-object tracking with efficient track drift and fragmentation handling. *J. Opt. Soc. Am. A*, 34(2):280–293, Feb 2017.

[44] Margret Keuper, Siyu Tang, Zhongjie Yu, Bjoern Andres, Thomas Brox, and Bernt Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. *CoRR*, abs/1607.06317, 2016.

[45] Amir Khatibi, Fabio Porto, Joao Guilherme Rittmeyer, Eduardo Ogasawara, Patrick Valduriez, and Dennis Shasha. Pre-processing and Indexing techniques for Constellation Queries in Big Data. In *DaWaK 2017: 19th International Conference on Big Data Analytics and Knowledge Discovery*, number 10253 in Big Data Analytics and Knowledge Discovery, pages 74–87, Lyon, France, August 2017. Springer.

[46] H. Kieritz, S. Becker, W. HÃ$\frac{1}{4}$bner, and M. Arens. Online multi-person tracking using integral channel features. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 122–130, Aug 2016.

[47] Chalidabhongse T. Harwood D. Davis L. (2004) Kim, K. Background modeling and subtraction by codebook construction. In *The International Conference on Image Processing (ICIP)*, Singapore.

[48] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4696–4704, Washington, DC, USA, 2015. IEEE Computer Society.

[49] M. Kostinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2288–2295, June 2012.

[50] Huang.C Kou C.H and Nevatia.R. Multi-target tracking by online learned discriminative appearance models, 2010. CVPR.

[51] L. Kratz and K. Nishino. Tracking with local spatio-temporal motion patterns in extremely crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 693–700, June 2010.

[52] K. C. A. Kumar and C. D. Vleeschouwer. Discriminative label propagation for multi-object tracking with sporadic appearance features. In *2013 IEEE International Conference on Computer Vision*, pages 2000–2007, Dec 2013.

[53] C.H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by online learned dis- criminative appearance models, 2010. CVPR.

[54] T. Kutschbach, E. Bochinski, V. Eiselein, and T. Sikora. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–5, Aug 2017.

[55] Edwin H. Land and John J. McCann. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61(1):1–11, Jan 1971.

[56] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178, 2006.

[57] L. Leal-Taixe, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3542–3549, June 2014.

[58] Laura Leal-Taixe, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese CNN for robust target association. *CoRR*, abs/1604.07866, 2016.

[59] L. Leal-TaixÃ©, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 120–127, Nov 2011.

[60] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1683–1698, Oct 2008.

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

[62] S. Liao, G. Zhao, V. Kellokumpu, M. PietikÃinen, and S. Z. Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1301–1306, June 2010.

[63] Shengcai Liao, Yang Hu, and Stan Z. Li. Joint dimension reduction and metric learning for person re-identification. *CoRR*, abs/1406.4216, 2014.

[64] Ye Liu, Hui Li, and Yan Qiu Chen. Automatic tracking of a large number of moving targets in 3d. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ECCV'12, pages 730–742, Berlin, Heidelberg, 2012. Springer-Verlag.

[65] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[66] Wenhan Luo, Junliang Xing, Xiaoqin Zhang, Xiaowei Zhao, and Tae-Kyun Kim. Multiple object tracking: A literature review. 2014.

[67] A. El Maadi and M. S. Djouadi. Suspicious motion patterns detection and tracking in crowded scenes. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6, Oct 2013.

[68] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato. Hierarchical gaussian descriptor for person re-identification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1363–1372, June 2016.

[69] N. McLaughlin, J. M. D. Rincon, and P. Miller. Enhancing linear programming with motion modeling for multi-target tracking. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 71–77, Jan 2015.

[70] K. Meshgi and S. Ishii. Expanding histogram of colors with gridding to improve tracking accuracy. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 475–479, May 2015.

[71] A. Milan, L. Leal-TaixÃ©, K. Schindler, and I. Reid. Joint tracking and segmentation of multiple targets. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5397–5406, June 2015.

[72] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, February 2017.

[73] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):58–72, Jan 2014.

[74] A. Milan, K. Schindler, and S. Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3682–3689, June 2013.

[75] D. Mitzel and B. Leibe. Real-time multi-person tracking with detector assisted structure propagation. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 974–981, Nov 2011.

[76] Dennis Mitzel, Esther Horbert, Andreas Ess, and Bastian Leibe. *Multi-person Tracking with Sparse Detection and Continuous Segmentation*, pages 397–410. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[77] Furqan M.Khan and Francois Bremond. Multi-shot person re-identification using part appearance mixture. WACV.

[78] Anh-Tuan Nghiem, Francois Bremond, and Monique Thonnat adn V Valentin. Etiseo, performance evaluation for video surveillance system, 2007. AVSS.

[79] T. L. A. Nguyen, F. Bremond, and J. Trojanova. Multi-object tracking of pedestrian driven by context. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 23–29, Aug 2016.

[80] T. L. A. Nguyen, D. P. Chau, and F. Bremond. Robust global tracker based on an online estimation of tracklet descriptor reliability. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Aug 2015.

[81] T. L. A. Nguyen, F. M. Khan, F. Negin, and F. Bremond. Multi-object tracking using multi-channel part appearance representation. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Aug 2017.

[82] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 661–675, London, UK, UK, 2002. Springer-Verlag.

[83] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*, pages 1201–1208, June 2011.

[84] Fabio Poiesi, Riccardo Mazzon, and Andrea Cavallaro. Multi-target tracking on confidence maps: An application to people tracking. *Computer Vision and Image Understanding*, 117(10):1257 – 1272, 2013.

[85] Zhen Qin. Improving multi-target tracking via social grouping. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 1972–1978, Washington, DC, USA, 2012. IEEE Computer Society.

[86] Zhen Qin and Christian R. Shelton. Improving multi-target tracking via social grouping. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1972–1978, 2012.

[87] Vladimir Reilly, Haroon Idrees, and Mubarak Shah. Detection and tracking of large number of targets in wide area surveillance. In *Proceedings of the 11th European Conference*

on Computer Vision Conference on Computer Vision: Part III, ECCV'10, pages 186–199, Berlin, Heidelberg, 2010. Springer-Verlag.

[88] M. Rodriguez, J. Sivic, I. Laptev, and J. Y. Audibert. Data-driven crowd analysis in videos. In *2011 International Conference on Computer Vision*, pages 1235–1242, Nov 2011.

[89] Mikel Rodriguez, Saad Ali, and Takeo Kanade. Tracking in unstructured crowded scenes. In *ICCV*, pages 1389–1396. IEEE Computer Society, 2009.

[90] A. Roshan Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[91] F. Bremond S. Bak, E. Corvee and M. Thonnat. Multiple-shot human re-identification by mean riemannian covariance grid, 2011. IEEE International Conference on Advanced Video and Signal-Based Surveillance.

[92] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *CoRR*, abs/1701.01909, 2017.

[93] Ricardo Sanchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. *Online Multi-target Tracking with Strong and Weak Detections*, pages 84–99. Springer International Publishing, Cham, 2016.

[94] Jianbo Shi and C. Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Jun 1994.

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

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

[97] F. Solera, S. Calderara, and R. Cucchiara. Towards the evaluation of reproducible robustness in tracking-by-detection. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Aug 2015.

[98] Bi Song, Ting-Yueh Jeng, Elliot Staudt, and Amit K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 605–619, Berlin, Heidelberg, 2010. Springer-Verlag.

[99] Daisuke Sugimura, K. M. Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1467–1474, Sept 2009.

[100] Michael J. Swain and Dana H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, November 1991.

[101] Siyu Tang, Mykhaylo Andriluka, Anton Milan, Konrad Schindler, Stefan Roth, and Bernt Schiele. Learning people detectors for tracking in crowded scenes, 2013. ICCV.

[102] Zhu Teng, Junliang Xing, Qiang Wang, Congyan Lang, Songhe Feng, and Yi Jin. Robust object tracking based on temporal and spatial deep networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[103] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.

[104] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and clas- sification, 2006. ECCV.

[105] Tim van Erven and Peter Harremoe. Re Ìnyi divergence and kullback-leibler divergenc, 2014. IEEE transations on information theory.

[106] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang. Tracklet association with online target-specific metric learning. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1234–1241, Washington, DC, USA, 2014. IEEE Computer Society.

[107] Bing Wang, Li Wang, Bing Shuai, Zhen Zuo, Ting Liu, Kap Luk Chan, and Gang Wang. Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[108] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3119–3127, Dec 2015.

[109] L. Wang, N. T. Pham, T. T. Ng, G. Wang, K. L. Chan, and K. Leman. Learning deep features for multiple object tracking by using a multi-task learning strategy. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 838–842, Oct 2014.

[110] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, June 2009.

[111] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[112] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1948–1955, June 2012.

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

[114] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *CVPR 2011*, pages 1345–1352, June 2011.

[115] Xu Yan, Ioannis A. Kakadiaris, and Shishir K. Shah. What do i see? modeling human visual perception for multi-person tracking, 2014. ECCV.

[116] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2034–2041, June 2012.

[117] Bo Yang, Chang Huang, and Ram Nevatia. Learning affinities and dependencies for multi-target tracking using a crf model, 2011. CVPR.

[118] Bo Yang and Ram Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part I*, ECCV'12, pages 484–498, Berlin, Heidelberg, 2012. Springer-Verlag.

[119] M. Yang, Fengjun Lv, Wei Xu, and Yihong Gong. Detection driven adaptive multi-cue integration for multiple human tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1554–1561, Sept 2009.

[120] Nai Chung Yang, Wei Han Chang, Chung Ming Kuo, and Tsia Hsing Li. A fast mpeg-7dominant color extraction with new similarity measure for image retrieval, 2008. Visual Communication and Image Representation.

[121] J. H. Yoon, C. R. Lee, M. H. Yang, and K. J. Yoon. Online multi-object tracking via structural constraint event aggregation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1392–1400, June 2016.

[122] J. H. Yoon, M. H. Yang, J. Lim, and K. J. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 33–40, Jan 2015.

[123] Ju Hong Yoon, Du Yong Kim, and Kuk-Jin Yoon. Visual tracking via adaptive tracker selection with multiple features. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ECCV'12, pages 28–41, Berlin, Heidelberg, 2012. Springer-Verlag.

[124] Kim D.Y Yoon J.H and Yoon K.J. Visual tracking via adaptive tracker selection with multiple features, 2012. ECCV.

[125] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. *POI: Multiple Object Tracking with High Performance Detection and Appearance Feature*, pages 36–42. Springer International Publishing, Cham, 2016.

[126] Mingyong Zeng, Z. Wu, C. Tian, Lei Zhang, and Lei Hu. Efficient person re-identification by hybrid spatiogram and covariance descriptor. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 48–56, June 2015.

[127] Li Zhang, Yuan Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.