Human detection and tracking in surveillance videos

Wang, Bing

2016

Wang, B. (2016). Human detection and tracking in surveillance videos. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65919>

<https://doi.org/10.32657/10356/65919>

Human Detection and Tracking in Surveillance Videos

Bing WANG

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2015

*To my parents and my wife,
who always give me love and support.*

Acknowledgments

First of all, I would like to express my deepest appreciation and sincerest gratitude to my supervisor Professor Kap Luk Chan and my co-supervisor Professor Gang Wang, who have gave me strong support and sincere encouragement throughout my PhD study in the Nanyang Technological University (NTU). During these years of study, I have learned not only the professional knowledge but also the spirit of exploring science from them. This work would not have been possible without their insightful guidance and encouragement.

I would like to acknowledge the School of Electrical and Electronics Engineering, NTU, for awarding me the research scholarship and providing me with excellent environment and computing facilities for my research.

I would like to thank Li Wang, Junyan Wang, Xiaoyan Zhang for their useful advices and discussions on my work. I am also indebted to Ye Luo, Haijian Zhang, Zhen Zuo, Bing Shuai, Lifan Zhao, Ting Liu and many other friends in NTU for their help in my research and life in Singapore.

Finally, I would like to dedicate this thesis to my parents and my wife, who always encourage me and support me.

Abstract

The thesis addresses the following challenging problems of detecting and tracking humans in the presence of occlusions in typical surveillance videos: (1) adaptation of semantic-part-based human detectors to new surveillance video sequence when trained detectors using other video data not performing well on the new video data; (2) tracking of humans with person identification minimizing identification errors over longer tracking periods; and (3) hierarchical spatial and temporal analysis for discriminative tracking of human targets. The thesis aims to improve the state-of-the-art performance in human detection and tracking by studying the human detectors, extended tracking of track segments (tracklets) generated from short term tracking of detection responses.

For the adaptation of semantic-part-based human detectors to new surveillance video sequence, a unified deep CNN model for joint learning of features, semantic pedestrian part detectors and a transfer learning model is developed. The components within this deep CNN model interact with each other in the learning process, which facilitates the optimization of the learned components during the co-operative learning. In particular, an adaptation layer is proposed to embed the capability of knowledge transfer into the CNN model. As a result, the proposed transferred CNN (T-CNN) model is able to transfer the visual knowledge of the semantic pedestrian parts from the source data to target data. Extensive experimental evaluations show that the proposed method is better than other deep learning based methods in terms of detection performance. Moreover, the adaptive deep features can be complementary to the pre-defined features used by other state-of-the-art methods.

For tracking of humans with person identification minimizing identification errors over longer tracking periods, a novel method, based on online target-specific

metric learning and coherent dynamics estimation, for tracklet association by network flow optimization is developed. The proposed framework aims to exploit appearance and motion cues to prevent identity switches during tracking and also to recover missed detections. The target-specific metrics (appearance cue) and motion dynamics (motion cue) are proposed to be learned and estimated online, i.e. during the tracking process. Furthermore, a learning algorithm to learn the weights of motion and appearance tracking cues for tracklet affinity models is proposed to handle some difficult situations. Extensive evaluations following state-of-the-art practices have been conducted and the results from these evaluations show the improvements by the proposed method over some existing state-of-the-art methods.

In hierarchical spatial and temporal analysis for discriminative tracking of human targets, inspired by recent advances in convolutional neural network (CNN) architectures, a novel unified deep model for tracklet association, which can jointly learn the CNNs and temporally constrained metrics, is developed. Furthermore, a novel loss function incorporating temporally constrained multi-task learning mechanism is developed to make the deep model more effective in solving the tracklet association problem. Extensive experimental results comparing with the state-of-the-art methods demonstrate the effectiveness and superiority of the proposed unified deep model.

Contents

Acknowledgments	i
Abstract	ii
List of Figures	viii
List of Tables	xii
List of Abbreviations	xiv
1 Introduction	1
1.1 Challenges	2
1.1.1 Viewpoint and Illumination Variations	2
1.1.2 Target Occlusions	3
1.1.3 Appearance Modeling in Human Tracking	5
1.2 Motivation and Objectives	5
1.2.1 Human Detection	5
1.2.2 Human Tracking	6
1.3 Major Contributions	7
1.4 Organization of the Thesis	9

2	Literature Review	11
2.1	Object Detection	11
2.2	Object Tracking	15
2.2.1	Single Object Tracking	16
2.2.2	Multi-Object Tracking	17
2.3	Concluding Remarks	24
3	Human Detection with Occlusion Handling	26
3.1	Pedestrian Detection in Highly Crowded Scenes Using “Online” Dic- tionary Learning for Occlusion Handling	27
3.1.1	Introduction	27
3.1.2	Initial Detection with Enhanced Head Detection	30
3.1.3	“Online” Dictionary Learning	31
3.1.4	Experiments	33
3.1.5	Concluding Remarks	34
3.2	Learning Semantic Pedestrian Part Detectors with a Transferred CNN Model	35
3.2.1	Introduction	35
3.2.2	Related Work	38
3.2.3	The Overview of the Proposed Transferred CNN	39
3.2.4	Training the Transferred CNN (T-CNN)	41
3.2.5	The Transferred CNN at the Test Stage	44
3.2.6	Experiments	45
3.2.7	Concluding Remarks	56

4	Tracklet Association by Online Target-Specific Metric Learning and Coherent Dynamics Estimation	58
4.1	Introduction	58
4.2	Cost-flow Network Formulation for Trajectory Recovery by Tracklet Association	64
4.3	Online Learning of Tracklet Affinity Models	67
4.3.1	Online Target-Specific Metric Learning	68
4.3.2	Tracklet Refinement	70
4.3.3	Online Tracklet Dynamics Estimation	72
4.3.4	Tracklet Affinity Measurement	74
4.4	Learning of Affinity Weights	76
4.4.1	Assessment of Difficult Situations for Motion Dynamics . . .	77
4.4.2	Learning of the Weighting Parameter	78
4.5	Experiments	80
4.5.1	Datasets	80
4.5.2	Experimental Settings	82
4.5.3	Evaluation Metrics	82
4.5.4	Quantitative Evaluation	84
4.5.5	Tracking Results by Using the proposed Human Detector . .	90
4.5.6	Computational Speed	92
4.6	Concluding Remarks	92
5	Joint Learning of Convolutional Neural Networks and Temporally Constrained Metrics for Tracklet Association	95

5.1	Introduction	95
5.2	The Unified Deep Model	98
5.2.1	Architecture	98
5.2.2	Loss Function and Temporally Constrained Metric Learning	99
5.3	Tracklet Association Framework	105
5.3.1	Tracklet Association with Generalized Linear Assignment . .	105
5.3.2	Tracklet Affinity Measurement	107
5.4	Experiments	109
5.4.1	Datasets	109
5.4.2	Experimental Settings	110
5.4.3	Performance Evaluation	111
5.4.4	Tracking Results by Using the proposed Human Detector . .	115
5.5	Concluding Remarks	116
6	Conclusions and Future Research	117
6.1	Conclusions	117
6.2	Recommendations for Future Research	120
	Author's Publications	123
	References	124

List of Figures

1.1	Sample frames from the PETS 2009 dataset [1].	3
1.2	Examples of occlusion from the TUD Stadtmitte dataset [2]. Missed detections with partial occlusions are labeled in yellow; and missed detections with full occlusions are labeled in red.	4
2.1	The deformable part-based model from [3].	12
2.2	An example of the cost-flow network with 3 consecutive frames and 9 observations from [4].	18
3.1	Examples of crowded scenes and their corresponding detection re- sults with the proposed approach are presented in row 1 and row 2, respectively.	28
3.2	The framework of the proposed approach.	29
3.3	The deformable part-based model from [3].	30
3.4	Performance comparison on PETS 2009 dataset.	34
3.5	The framework of the transferred CNN (T-CNN).	36
3.6	The flowchart of pedestrian detection system at the testing stage. .	44
3.7	Performance comparison on PETS 2009 dataset. Since the two se- quences in this dataset are densely crowded, Baseline1 is not effec- tive. Only the results of Baseline2 are shown here.	48

3.8	Results of the baseline detectors and ours on the Caltech dataset under standard evaluation settings.	49
3.9	Results of the state-of-the-art detectors and ours on the Caltech dataset under standard evaluation settings.	51
3.10	Results of the deep network based detectors and ours on the Caltech dataset under standard evaluation settings.	52
3.11	Detection results under different occlusion conditions on the Caltech dataset.	53
3.12	Results of the state-of-the-art detectors and ours on the ETH dataset under standard evaluation settings.	54
3.13	Results of the baseline detectors and ours on the ETH dataset under standard evaluation settings.	55
3.14	Detection results of the deep network based detectors under different settings on the ETH dataset. Reasonable setting: based on the pedestrians at a height of more than 50 pixels and a visibility of at least 65%. Overall setting: based on all the annotated pedestrians. .	56
4.1	A difficult scenario of high appearance similarity among targets. (Frames from PETS dataset with pedestrian identities labeled by our method): Despite individuals 1 and 8 dressed in similarly colored clothes and severe occlusions and interactions between individuals 1, 8, and 9, their identities should remain unchanged.	61

4.2	The proposed framework. In the cost-flow network, each node denotes a reliable tracklet; The flow costs of edges are defined by negative log of the affinity scores, which are obtained through the online learning of target-specific metrics and motion dynamics with the off-line learned weights on segments of short-time sequences known as local segments.	63
4.3	An example of the difficult situations.	78
4.4	Columns 1 and 2 compare the tracking results of LRMCNF [5] (left) and our approach (right) respectively on the ETH BAHNHOF sequence (frames 104 - first row, 130 -second row, 187- third row). Note the detection windows pointed by red arrows. We can see that our approach maintains ID labels more reliably.	86
4.5	Tracking results of our method (TSML) on (top to bottom) <i>TUD Crossing</i> , <i>TUD Stadtmitte</i> , <i>ETH BAHNHOF</i> and <i>ETH SUNNY DAY</i>	93
4.6	Tracking results of our method (TSML+TD+WP) on (top to bottom) <i>PETS 2009</i> and <i>Town Centre</i>	93
5.1	Tracking framework of our method. In the Generalized Linear Assignment (GLA) graph, each node denotes a reliable tracklet; each edge denotes a possible link between two tracklets. This work jointly learns the siamese CNN and the temporally constrained metrics for tracklet affinity model, as shown in the red-dashed box, which is exploited to estimate the linking probability between two tracklets in the GLA graph. The tracking results are obtained by combinatorial optimization via the softassign algorithm.	97
5.2	The structure of 5-layer sub-CNN used in the unified deep model.	100

5.3	An illustration of the temporally constrained multi-task learning mechanism. n is the total number of the segments and the segments are shown in the temporal space.	103
-----	--	-----

List of Tables

4.1	Comparison of tracking results with other network flow based methods on TUD Crossing and ETH BAHNHOF (first 350 frames) sequences. The entries in the table are (IDS)/(correctly matched detections). Columns 1 and 2 use the pre-trained pedestrian detector of [3]. Column 3 shows the results when ground truth detections are used to generate the initial tracklets. The ground truth detections are from [6].	85
4.2	Comparison of tracking results between state-of-the-art methods and ours on PETS 2009 dataset.	87
4.3	Comparison of tracking results between state-of-the-art methods and ours on Town Centre dataset.	89
4.4	Comparison of tracking results between state-of-the-art methods and ours on TUD Stadtmitte dataset. Note that very good performance has been achieved by TSML, resulting in limited room of improvement. We thus do not employ TD and WP in the experiment.	89
4.5	Comparison of tracking results between state-of-the-art methods and ours on ETH dataset. Note that the forward and panning motions of the cameras lead to the unreliable motion affinities between tracklets of this dataset. We thus do not employ TD and WP in the experiment.	90

4.6	Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on PETS 2009 dataset. .	91
4.7	Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on Town Centre dataset.	91
4.8	Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on TUD Stadtmitte dataset.	91
4.9	Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on ETH dataset.	92
5.1	Comparison of tracking results between state-of-the-art methods and ours on PETS 2009 dataset.	112
5.2	Comparison of tracking results between state-of-the-art methods and ours on Town Centre dataset.	113
5.3	Comparison of tracking results between state-of-the-art methods and ours on ParkingLot dataset.	113
5.4	Comparison of tracking results between state-of-the-art methods and ours on ETH dataset.	114
5.5	Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on PETS 2009 dataset. .	115
5.6	Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on Town Centre dataset.	115
5.7	Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on ParkingLot dataset. .	115
5.8	Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on ETH dataset.	116

List of Abbreviations

BoW	Bag-of-Words
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CSS	Color-Self-Similarity
DPM	Deformable Part Model
FC	Feature Context
GLA	Generalized Linear Assignment
GMCP	Generalized Minimum Clique Problem
GMMCP	Generalized Maximum Multi Clique Problem
HOG	Histogram of Oriented Gradients
LBP	Local Binary Pattern
MAP	Maximum-a-Posteriori
MMDL	Max-Margin Multiple-Instance Dictionary Learning
MWIS	Maximum-Weight Independent Set
RBC	Radial Basis Coding
SIFT	Scale-Invariant Feature Transform
SVM	Support Vector Machine
TAT	Tracklet Association-Based Tracking
T-CNN	Transferred Convolutional Neural Network

Chapter 1

Introduction

Millions and millions of surveillance videos are recorded daily by surveillance systems around the globe. These massive amount of video data has to be manually processed and analyzed by human operators laboriously in many situations to extract useful information of interest. Consequently, efficient video analysis tools are in great demand to process these surveillance videos. Among these video analysis tools, object detection and tracking are two essential ones, which are usually used as the basis for many higher level tasks of video content understanding, such as behavior reasoning, event analysis. Hence, they attract an enormous amount of research efforts from researchers all over the world.

Object detection and tracking in real scenes are two important problems in computer vision too, which are generally understood as follows:

- *Object detection.* The goal is to localize the objects of interest in individual video frame, which may be highlighted with bounding boxes, and each frame of the surveillance video is often considered independently. Object detection is often adversely affected by many uncontrollable causes including scene clutters, illumination changes and object occlusions.
- *Object tracking.* The tracking task often takes temporal information and more

recently together with appearance information into consideration to follow the objects of interest, tracing out the complete trajectories of the objects of interest. However, when many target interactions and serious occlusions occur, long-term tracking of individual objects is a very challenging problem.

As human behaviors and activities are often the key objectives in surveillance video analysis, this thesis focuses on the human as the object of interest.

1.1 Challenges

Human detection and tracking can be easy tasks, if all humans are separate from each other without any occlusions and the scene background is plain. Such a thinking is unrealistic in real surveillance scenes where detecting and tracking multiple humans are quite challenging due to viewpoint, illumination, articulation, target interactions and occlusions, and so on. Some of the main challenges are further discussed in this subsection.

1.1.1 Viewpoint and Illumination Variations

The shape and appearance of a human will vary substantially when the viewpoint of the camera is changed or the person turns towards other directions. Changes of the viewpoint often leads to distortions of human appearance, as shown in Figure 1.1. Larger changes of the viewpoint usually reveal shapes that were previously not observed. In the extreme case, no shape similarity is shared by two different views of the same person.

The same person in the same scene, imaged under different illumination conditions, can result in very different appearances. As we can see in Figure 1.1, the appearances of a group of pedestrians vary a lot. To deal with this problem, a pre-defined features: histogram of oriented gradients (HOG) [7], which are invariant to

some illumination changes, are proposed. The effect of illumination changes can be reduced by working with these oriented gradients obtained from pixel intensities.

Figure 1.1: Sample frames from the PETS 2009 dataset [1].

Instead of directly handling viewpoint and illumination variations, the research reported in this thesis focuses on other sources of variations: inter-class and intra-class variations to differentiate different targets and backgrounds.

1.1.2 Target Occlusions

Occlusion for a target object is defined as some or all parts of the target are not visible in a video frame. Based on the extent of visibility, occlusions can be classified as partial occlusion and full occlusion as shown in Figure 1.2. For partial occlusions, the appearance for the same target may change greatly, which makes the object detection and tracking tasks more challenging. For full occlusions, the trajectories of the targets can only be inferred from previous and future motion patterns, and failing which the tracked trajectories will be broken. Figure 1.2 shows some examples of partial and full occlusions. In real surveillance scenes, humans are frequently occluded by each other.

There has been plenty of research reported in literature that addresses occlusion

Figure 1.2: Examples of occlusion from the TUD Stadtmitte dataset [2]. Missed detections with partial occlusions are labeled in yellow; and missed detections with full occlusions are labeled in red.

handling in object detection [8–12]. To deal with occlusions, modeling occlusions as regions inconsistent with the target statistics is a common approach. Girshick *et al.* [8] propose to introduce an occluder part in their grammar model when some of the parts are occluded. Wang *et al.* [9] develop a pedestrian detection approach capable of handling partial occlusions by combining Histograms of Oriented Gradients (HOG) and Local Binary Pattern (LBP) as the feature set. Meger *et al.* [10] propose to use depth inconsistency from 3D data to handle occlusions. Hsiao *et al.* [11] propose to explicitly model occlusions by reasoning 3D interactions of objects. Shu *et al.* [12] propose to select the subset of parts, which maximizes the probability of detection, in pedestrian detection for occlusion handling. However, these methods still cannot handle severe occlusions in crowded scenes when 3D information is unavailable.

For human tracking, there exists additional challenges. The appearance of a human varies greatly as the visible parts of the human body change. Hence, maintaining the identities of humans in the tracking is very challenging when there

are many occlusions and interactions with each other.

1.1.3 Appearance Modeling in Human Tracking

To correctly track multiple humans in surveillance videos, appropriate appearance-based affinity models should be developed so that the same humans would have high affinity scores while different humans would have low affinity scores. In real surveillance scenes, appearances of the same human usually change over time due to viewpoint variation, illumination change, occlusions or interactions as shown in Figure 1.1 and 1.2. Moreover, different humans can be similar in appearance due to wearing similar dresses. All of these factors make maintaining identities of humans a very challenging problem too.

1.2 Motivation and Objectives

1.2.1 Human Detection

The performance of a human detector is often hindered by difficulties such as partial or severe occlusions. Even part-based detectors such as DPM may fail due to the lack of effective representations of pedestrian parts in such difficult situations. Intuitively, a large number of annotated training images of pedestrian parts are needed for learning their visual patterns. However, most pedestrian detection datasets do not have annotated training data for parts. To solve this problem, a transfer learning model to learn these semantic pedestrian part detectors is proposed in this thesis.

On the other hand, feature representation is an important pre-requisite in human detection. In the past, pre-defined features such as SIFT [13], HOG [7] and LBP [9] have been used. However, they are not able to represent data-driven visual patterns in human detection. Recently, deep learning models such as the convo-

lutional neural networks (CNN) [14] have achieved very promising performance in many computer vision applications such as image classification [15] for example. Inspired by this, many deep learning based methods [16–20] have been proposed for human detection. For occlusion handling, Ouyang et al. [16, 18] utilize deep models to learn the relationships between different parts. However, their method does not explicitly learn visual patterns of semantic pedestrian parts such as head, upper body, and so on, which are critical for occlusion handling. Previous work on learning semantic pedestrian part detectors via a deep model is yet to be seen. In this thesis, it is shown to be beneficial to use visual patterns learned from semantic pedestrian parts via a deep model for occlusion handling in human detection.

1.2.2 Human Tracking

Recently, significant progress has been reported in human detection [3, 7, 9, 21–24], and this promotes the popular tracking paradigm: detect-then-track [4–6, 25–30]. The main idea is that a human detector is run on each frame to detect targets of interest, and then detection responses are linked across multi-frames to obtain target trajectories. In [4–6, 28], the authors formulate the multi-target data association as a network flow optimization problem. Zhang et al. [4] use a push-relabel method [31] to solve the network flow optimization problem as a min-cost flow problem. Berclaz et al. [28] and Pirsiavash et al. [6] propose to use successive shortest path algorithms, which can provide roughly the same globally optimal tracking results. In a more recent paper, Butt et al. [5] incorporate higher-order track smoothness constraints, such as constant velocity, for multi-target tracking. However, due to the limitation of the appearance cues used for tracking, the methods mentioned above usually cannot deal with longer term tracking to obtain a complete trajectory of a target. This is because prolonged occlusions and target-to-target interactions will result in fragmentation of a trajectory. If we make full

use of the information from the whole sequence (previous, current, and subsequent frames), trajectory can be recovered from the fragments and tracking errors such as missed tracks or identity switches can be corrected. Then, similarity measurement between two track fragments (tracklets) to determine whether they belong to the same person becomes very critical in tracklet association problem. Some of the state of art methods, such as in [32] and [25], fuse several features based on motion, time, position, size and appearance to improve the similarity measurement. However, their appearance models are still inadequate to handle large appearance variations, thus adversely affecting tracking performance. In this thesis, a discriminative target-specific appearance-based affinity model is advocated to reinforce the appearance cues for multi-person tracking.

Another issue in state-of-the-art human tracking methods is that they make use of image representations often not good enough for constructing robust appearance-based tracklet affinity models. Current methods usually utilize pre-selected features, such as HOG features [7] and local binary patterns [9], or color histograms, which are not “tailor-made” for the tracked objects in question. Recently, deep convolutional neural network architectures have been successfully applied to many challenging tasks, such as image classification [14] and object detection [33], and provide highly promising results. The core to its success is to take advantage of deep architectures to learn richer hierarchical features through multiple nonlinear transformations. Hence, deep convolutional neural networks for multi-person tracking is explored in this thesis.

1.3 Major Contributions

The major contributions of this thesis can be summarized as follows:

- For human detection, a unified deep CNN model for jointly learning fea-

tures, semantic pedestrian part detectors, and a transfer learning model is proposed. Components within this deep CNN model interact with each other in the learning process, enabling the components to maximize their potentials during the cooperative learning. In particular, a margin-based loss function for the adaptation layer of the deep CNN model to transfer knowledge from the source dataset to the target one is developed. As a result, the proposed transferred CNN (T-CNN) model can extract more effective pedestrian part detectors for handling pedestrian occlusions on the target dataset. Experimental results demonstrate that the human detection system proposed in this thesis outperforms other state-of-the-art detectors based on deep networks.

- For human tracking, the target-specific metrics with enhanced discriminative ability are proposed to be online learned through a two-step target-specific metric learning and metric refinement processes. Moreover, the novel tracklet affinity models consisting of both appearance-based model and motion dynamics-based model are developed to prevent identity switches and recover missed detections during tracking. These tracklet affinity models are updated within each local segment for reduced computation as well as obtaining locally adaptive affinity models. To obtain a further improvement in performance, a learning algorithm to learn the weights of motion and appearance tracking cues for tracklet affinity models is proposed. The proposed method is found to be effective even when the appearance or motion cues fail to identify or follow the target due to occlusions or target-to-target interactions. The experimental results on several public datasets verify the effectiveness and superiority of the proposed method.
- A unified deep model for joint learning of “tailor-made” hierarchical features for tracked targets and temporally constrained segment-wise metrics

for tracklet affinity models is proposed. With this deep model, the feature learning and the discriminative tracklet affinity model learning can efficiently interact with each other, maximizing their performance by cooperating with each other. In particular, a novel temporally constrained multi-task learning mechanism is proposed to be embedded into the last layer of the unified deep neural network, which makes the deep neural network more effective in tackling the tracklet association problem. By employing the proposed method, correct tracklet association can be attained even in challenging situations. Experimental results on several challenging public datasets show that the proposed method outperforms the state-of-the-art approaches in multi-human tracking.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows:

- A literature review of related works in the fields of object detection and tracking is presented in Chapter 2.
- In Chapter 3, an extension of the deformable part model (DPM) [3] for occlusion handling in highly crowded scenes is first introduced. The score computation of body parts in the original DPM detector is reformulated. Then the detection responses are refined by online learned multiple-instance dictionaries. Moreover, more general semantic pedestrian part detectors for dealing with pedestrian occlusions are also presented in this chapter. These semantic pedestrian part detectors are learned through a novel transferred convolutional neural network (T-CNN). Extensive experimental results and analysis are presented therein.

- Chapter 4 describes a novel method based on target-specific metric learning and coherent dynamics estimation for tracklet (track fragment) association by network flow optimization in long-term multi-person tracking. Target-specific metrics (appearance cue) and motion dynamics (motion cue) are proposed to be learned and estimated online, i.e. during the tracking process. Furthermore, the weights of these two tracking cues are proposed to be learned to handle the difficult situations, such as severe occlusions and target-to-target interactions. Comparisons and quantitative analysis are provided in this chapter.
- In Chapter 5, a novel unified deep model for tracklet association is proposed. This deep model can jointly learn the convolutional neural networks (CNNs) and temporally constrained metrics for tracklet affinity models. In the proposed method, a siamese convolutional neural network (CNN) is first pre-trained on the auxiliary data. Then the siamese CNN and temporally constrained metrics are jointly learned online to construct the appearance-based tracklet affinity models. Comparisons with other state-of-the-art tracking methods are presented in the experimental evaluations. Extensive experimental results and discussions are also provided in this chapter.
- In Chapter 6, concluding remarks of this thesis and discussions on future works are given.

Chapter 2

Literature Review

In this chapter, prominent object detection and tracking methods published in literature are introduced and reviewed. The key elements in object detection and tracking are discussed and analyzed in a systematic manner.

2.1 Object Detection

Object detection is an indispensable step for most of vision tasks, such as surveillance video analysis. It has been quite successful for some specific objects such as faces [34, 35] and humans [7, 9, 23, 24]. A complete survey of object detection is not the focus of this thesis. Instead, related object detection methods within the last decade are reviewed and the development trend of object detection is briefly analyzed.

Feature representation is known to be one of the key elements in object detection. Many features have been utilized in object detection. Haar-like features [36], HOG [7], and dense SIFT [37] have been proposed to capture the shape of objects for detection. First-order color features like color histograms [38], second-order color features like color-self-similarity (CSS) [39] and co-occurrence features [40] have also been proposed for object detection. Texture feature such as local binary

Figure 2.1: The deformable part-based model from [3].

patterns (LBP) has been used in [9]. There are still some other types of features, such as covariance descriptor [23], depth [41], segmentation outcomes [42], 3D geometry [43], and combinations of different features [9, 38, 39, 42]. All the features above are pre-defined. Recently, some researchers [19, 44] have tried to learn features from training data. In their works, the local max pooling or average pooling has been exploited, which makes the learning features more robust to small local misalignment. However, the deformation properties of body parts cannot be learned by these methods.

Since humans and some other kinds of objects can undergo non-rigid deformation, the deformable part model (DPM) [3] has been proposed to improve detection performance. The DPM detector is widely used in tracking-by-detection based methods. To make a fair comparison with other state-of-the-art methods, this DPM detector is also utilized in the proposed multi-human tracking methods in this thesis. As shown in Figure 2.1, the model defined in [3] consists of a coarse root filter, eight part filters and a spatial model for modeling locations of the parts relative to the root. The proposed detection framework in [3] comprises three key parts: low-level features: histograms of oriented gradients (HOG), matching deformable part-based models, and discriminative learning of latent variables using latent SVM. These key parts enable the DPM detector to deal with some pose variations and partial occlusions of targets.

Some extensions of the original DPM detector have been proposed in recent years [45–48]. Zhu *et al.* [45] propose a latent hierarchical structural learning method for object detection. In their work, they represent objects by mixture of hierarchical models with two or three layers, and extend the number of parts into the deeper layers. Park *et al.* [46] introduce a multi-resolution model that acts as a deformable part-based model when scoring large instances but as a rigid template when scoring small instances. Azizpour *et al.* [47] propose to adapt the original DPM with strong supervision setting in which part annotations are used in training to improve model initialization and handling of partial occlusions. In a more recent paper, Ouyang *et al.* [48] propose a probabilistic framework to model the relationship between the configurations estimated by single- and multi-pedestrian detectors trained with the DPM in [3], and to refine the single-pedestrian detection results with help of multi-pedestrian detections. These methods assume that the distribution of source samples is similar to that of target samples. Doing away with this assumption, the contributions of the proposed human detection method in this thesis aim at tackling the domain adaptation problem, where the data distributions in two domains can be significantly different and no part annotations of the training data are available. Furthermore, instead of using the HOG features [7], more discriminative features are proposed to be extracted from the learned transferred CNN model in this thesis.

Recently, a new object detection strategy [49, 50] is proposed to use multi-scale image segmentation to provide a couple thousands of candidate bounding boxes for each image and the categories of the bounding boxes are determined by strong classifiers. In [49], Sande *et al.* propose to use bag-of-words (BoW) model with spatial pyramid matching for object detection. Since the BoW models ignore spatial relations among local features, this method is able to tolerate large deformations.

Context information from local or global appearance has also been explored to improve the performance of object detection [51–55]. Song *et al.* [51] propose an iterative contextualization scheme to mutually boost performance of object detection. Wang *et al.* [52] introduce a novel Feature Context (FC) with multiple reference points, which is efficient to incorporate spatial arrangement, and a new coding method called Radial Basis Coding (RBC) for object detection and image classification. Ding *et al.* [53] present a framework that propagates contextual cues to wider areas through iterations to improve the detection accuracy. Cinbis *et al.* [54] propose a new method for object detection that is based on set representations of the contextual elements. The proposed set-based model is able to learn the contextual relationships of objects without predefining the relationship types. Li *et al.* [55] propose to exploit the unlabeled regions to extract adaptive contextual cues for enhanced object detection.

More recently, Wang *et al.* [56] propose to model an object class by a cascaded boosting classifier with various types of features from local regions, known as regionlets. Different from existing BoW approaches and DPM approaches, regionlets provide a different way to model object deformation, which can handle rigid objects, objects with small local deformations as well as long-range deformations. In the work of [18], Ouyang *et al.* propose a unified deep model that jointly learns feature extraction, deformation handling, occlusion handling and classification for pedestrian detection. Through interaction among these four independent task objectives, the joint learning algorithm boosts the detection performance. Girshick *et al.* [33] propose a novel object detection framework, called R-CNN, that combines bottom-up region proposal candidates with rich hierarchical features obtained from a convolutional neural network (CNN). This work shows that it is very effective to pre-train the deep network with supervision on the auxiliary data and then fine-tune the deep network on the target data in which limited labeled

training data is available.

From the above literature review on object detection, some potential research directions in this area are identified: (1) Novel features. The state-of-the-art detectors usually utilize several feature types in combination with HOG, such as [9]. (2) Contextual information. Especially at low resolutions, the context is quite important for object detection. (3) Temporal integration. Inspired by tracking, object detection also can exploit temporal information to improve performance. (4) Grouping cues. In crowded scenes, object detection is a very challenging problem. However, grouping cues are relevant in crowded scenes. A recent work [48] has tried to use multi-pedestrian detection to refine single-pedestrian detection. (5) Data. Most detectors were trained on datasets of limited sizes, such as Caltech dataset [57], INRIA dataset [7]. ImageNet [15], which is a quite huge dataset, has been created for image classification. The work of [33] has demonstrated that the deep neural network pre-trained on ImageNet [15] can boost the detection performance substantially.

2.2 Object Tracking

Object tracking is one of the most significant components in computer vision, which can be applied to many important domains such as surveillance systems, robotics, human computer interaction, etc. In terms of the number of tracked targets, it can be divided into two sub-problems: single object tracking and multiple object tracking. Given the initialized state of a target object in a frame of a video, the goal of single object tracking is to follow the single target in the subsequent frames. Single object tracking has achieved great progress in last decade and some typical approaches are online AdaBoost (OAB) tracker [58], incremental visual tracking (IVT) [59], multiple instance learning (MIL) tracker [60], l_1 tracker (L1T) [61],

Tracking-Learning-Detection (TLD) tracker [62] and multitask tracker (MTT) [63]. Recently, inspired by the advances in deep learning architectures, some researchers propose to exploit these deep architectures in single object tracking. However, multiple object tracking is a more challenging problem. It aims to simultaneously track multiple targets over a sequence of images. This thesis will focus more on multiple target tracking. In the following, the deep learning-based single object tracking methods are first briefly reviewed. Then the literature review of multi-object tracking is presented. Moreover, the important issues involved in multi-object tracking are discussed and analyzed.

2.2.1 Single Object Tracking

Instead of a complete survey of single object tracking, the up-to-date deep learning-based visual trackers are reviewed in this subsection. As deep learning architectures have achieved very promising performance in many computer vision applications such as image classification [14], nowadays many researchers [64–68] propose to utilize deep architectures in single object tracking.

[64] is the first work to apply deep neural networks on single object tracking. In this work, a stacked denoising autoencoder is first utilized to offline learn generic deep features from a large auxiliary image dataset. Then the generic deep features are online tuned to adapt to the specific target during the tracking process. In [65], Li *et al.* propose to exploit a CNN architecture and a structural loss function for single object tracking. Wang *et al.* [66] propose a two-layer CNN with temporal slowness constraint to learn generic features from auxiliary data for visual tracking. Moreover, an adaptation module is developed to adapt these generic features in accordance with the specific target. The work of [67] demonstrates that a simple two-layer deep neural network even without learning is still powerful enough to yield an effective representation for single object tracking. In [68], Hong

et al. propose to online learn discriminative target-specific saliency map via a CNN architecture for single object tracking.

2.2.2 Multi-Object Tracking

Many previous multi-object tracking methods [69–72] rely on background subtraction from static scenes. However, recently significant progress in object detection has promoted interests in combining tracking and detection for more complex and dynamic scenes. From the literature review, it is found that there are four important elements in multi-object tracking: data association framework, motion-based model, appearance-based model and grouping-based model. A literature review according to these important elements of multi-object tracking is presented in the following subsections.

2.2.2.1 Data Association Framework

Data association framework is a strategy to solve the multi-object tracking problem through global optimization. Many kinds of data association frameworks have been proposed for multi-object tracking in literature. This thesis only focuses on some typical and popular data association frameworks.

Network Flow. Network flow (also known as the transportation network flow) is a directed graph in which each edge has a capacity and each node usually represents a low-level observation. In the multi-object tracking problem, the low-level observations can be detection responses or tracklets (track fragments). The flow of an edge is usually modeled as an indicator which indicates whether the two adjacent nodes are linkable or not. To satisfy the flow restriction in the network flow graph, a source node representing the starting point of a trajectory and a sink node representing the ending point of a trajectory are added. An example of such a network flow graph is shown in Figure 2.2.

Figure 2.2: An example of the cost-flow network with 3 consecutive frames and 9 observations from [4].

The previous works [4, 6, 28] formulated the multi-frame, multi-target data association as a network flow problem. Zhang *et al.* [4] propose a method that uses a cost-flow network with a non-overlap constraint on trajectories to solve the maximum-a-posteriori (MAP) data association problem. It also employs a push-relabel method [31] to solve the min-cost flow problem. Berclaz *et al.* [28] and Pirsiavash *et al.* [6] propose to use a more efficient successive shortest path algorithm, which can provide roughly the same globally optimal tracking results with less running time.

Conditional Random Field. The Conditional Random Field (CRF) model is exploited to solve the multi-object tracking problem in [73–75]. The CRF is an undirected graphical model, in which each node represents a possible link between a pair of tracklet and each edge represents a correlation between two adjacent nodes.

In [73], Yang *et al.* propose a CRF model to transform the multi-person tracking problem into an energy minimization problem. They use RankBoost algorithm to select features. Recently, Yang *et al.* [74] propose an online learned CRF model

for solving the multi-target tracking problem. To avoid identity switch errors, this CRF framework also incorporates pairwise models to differentiate difficult pairs of targets. In a more recent work [75], Milan *et al.* propose a discrete-continuous CRF for multi-target tracking which can handle the inter-object exclusions at two levels: (1) at the data association level based on non-submodular constraints, such that each detection can only explain one target and vice versa; (2) at the trajectory level, where a novel co-occurrence label cost penalizes solutions with colliding trajectories.

GMCP. In [76], Zamir *et al.* propose to utilize Generalized Minimum Clique Problem (GMCP) to solve the global optimization problem for multi-object tracking. The Generalized Minimum Clique Graph is defined as $G = (V, E, w)$, where G is undirected and weighted, V , E and w represent the nodes, edges and the corresponding weights, respectively. Each node in the graph G denotes a detection response. In a more recent work [77], Dehghan *et al.* propose to extend the Generalized Maximum Clique Problem (GMCP) by formulating multi-object tracking as a Generalized Maximum Multi Clique Problem (GMMCP). The authors propose to utilize Binary Integer Programming to solve this GMMCP. Moreover, a significant speed-up is achieved by reformulating GMMCP via a Mixed-Binary-Integer Program.

Others. Brendel *et al.* [78] propose to formulate data association problem in multi-object tracking as finding the **Maximum-Weight Independent Set (MWIS)** of a graph of tracklets. In [12], Shu *et al.* propose to utilize a **Greedy Bipartite** algorithm to associate the detection responses into long trajectories. Jiang *et al.* [79] propose a novel **Linear Programming** relaxation scheme for handling multi-object tracking problem. **Hungarian** algorithm is adopted to solve the data association problem for multi-object tracking in [25, 27, 80, 81]. In [82], Dicle *et al.* propose to formulate the multi-target tracking problem as a **Gener-**

alized Linear Assignment (GLA) of tracklets which are associated into long trajectories progressively.

2.2.2.2 Motion-Based Model

A constant velocity motion model considering both the forward and backward velocity is presented in [27]. In this work, the link probability of two tracklets is estimated based on the proposed constant velocity motion model. Recent approaches [5, 83, 84] have proved that higher-order motion constraint is effective to enhance motion-based models for improving multi-target tracking performance. As for representing higher order motion constraint, Ochs *et al.* [85] propose a method to project a hyper-graph onto its initial graph, allowing higher-order motion-based models to be represented by the edges of a standard graph. Their method is used for motion segmentation of images, but can be adapted to multi-target tracking by projecting higher order motion constraints onto pairwise constraints in a flow network. Collins [84] introduces an approximate algorithm similar to the iterated conditional modes (ICM) algorithm to improve an initial feasible multi-frame solution. The proposed method is capable of handling arbitrary higher-order cost functions defined over entire trajectories. However, this method does not exploit shape and appearance information. Butt *et al.* [83] try to incorporate higher-order motion constraints by solving a series of independent multi-dimensional assignment problems over frame triplets. The three-frame tracklets are associated into longer trajectories gradually. Nevertheless, this approach is unable to revisit and correct a trajectory.

More recently, Butt *et al.* [5] propose an algorithm which can incorporate higher-order motion constraints such as piecewise constant-velocity path smoothness constraint for multi-target tracking. Their proposed method can still maintain a manageable computational complexity. Different from [83], which builds up a

solution from short tracks determined independently over frame triplets, the work of [5] optimizes the tracks globally over the entire sequence. Hence, the proposed method in [5] is capable of finding better solutions.

In another recent work [86], Yang *et al.* propose an online approach to learn non-linear motion patterns for multi-target tracking in a tracklet association framework. Unlike most previous methods, which use linear motion patterns only, they build a non-linear motion map in an online learning manner to better represent direction changes and produce more robust motion affinities.

2.2.2.3 Appearance-Based Model

Appearance-based model is another important element in multi-target tracking. Some recent approaches [12, 26, 32, 87, 88] attempt to enhance the appearance-based models to improve tracking performance.

Shitrit *et al.* [32] propose a framework which can exploit image appearance cues to prevent identity switches. The appearance model in their framework for pedestrian datasets, which only utilizes color similarity, is very simple. So the performance of their methods on pedestrian datasets is not so good. Moreover, they also use background subtraction method to segment the detection responses, which makes their method only capable of handling static camera datasets. Li *et al.* [26] propose to progressively associate detection responses to form longer tracklets and finally obtain the desired target trajectories. This work uses HybridBoost algorithm to learn the affinity model integrated in a hierarchical framework to track multiple targets as a joint problem of ranking and classification. Kuo *et al.* [87] propose a system named PIRMPT which uses the merits of person identity recognition to improve the multi-person tracking performance. However, this method needs off-line learned local descriptors which may be infeasible for tracking people whose identities are yet unknown. Yang *et al.* [88] introduce an online learning

approach to produce discriminative part-based appearance models (DPAMs) for multi-human tracking in real scenes. The DPAMs can exclude occluded regions in appearance modeling in order to improve tracking performance.

Shu *et al.* [12] propose an effective multi-person tracking method using part-based model and occlusion handling. Their appearance-based model can capture rich information about individuals, which makes it be robust against appearance changes and occlusions. Since the part-based model requires detailed body information, high quality imagery datasets are more suitable for the proposed method. In their experiments, the smallest frame resolution of the three datasets is 1920×1080 .

2.2.2.4 Grouping-Based Model

Recently, some approaches [89–91] successfully incorporate grouping cues to help improve multi-person tracking performance. Pellegrini *et al.* [89] propose an effective dynamic model based on social grouping behaviors to improve tracking in crowded scenarios. In [90], Pellegrini *et al.* present a third-order graphical model that is able to jointly estimate individual trajectories and group memberships over a short time interval for multi-person tracking. In a recent paper, Zhen *et al.* [91] propose a principled way of incorporating grouping behavior information, as a high-level reasoning tool, to improve multi-person tracking performance.

Multi-target tracking in crowded scenes is a very challenging task. Due to the prominent grouping behaviors in crowded scenes, grouping-based models are very important in this difficult situation. A general problem formulation, which can incorporate grouping behaviors into the tracklet association problem, is introduced in [91]. The objective is to recover the trajectories of each target based on motion, appearance and grouping cues within a time interval $[0, T]$. Given a set of n tracklets (track fragments) $X = \{F_1, F_2, \dots, F_n\}$ within $[0, T]$. The task is to

determine which tracklets are from the same person. This can be represented by a correspondence function ϕ :

$$\phi_{ij} = \begin{cases} 1, & \text{if tracklet } j \text{ immediately follows tracklet } i, \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

with the additional constraints that $\sum_j \phi_{ij} = 1$ and $\sum_i \phi_{ij} = 1$, indicating each tracklet should follow and be followed by one other tracklet (except for the first and last tracklets of each track). Let Φ be the set of valid correspondences.

A pairwise transition score, M_{ij} , which denotes negative log-likelihood that tracklet j should be linked after tracklet i , is defined. This is the basic affinity model based on motion and appearance cues in traditional works. The traditional data association based tracking can be formulated as follows:

$$\arg \min_{\phi \in \Phi} \sum_{ij} \phi_{ij} M_{ij} \quad (2.2)$$

This data association problem can be solved optimally by the Hungarian algorithm [25].

Here, the authors propose to take the grouping behaviors into consideration to help eliminate visual ambiguities within the video sequence to improve multi-person tracking performance. The objective is to maximize the consistency of motion, appearance and grouping cues. For grouping behavior evaluation, an optimal number of K groups are assumed to be formed by people within a time interval of a video sequence. For each group, there is a group mean trajectory G_k . Then a new group measurement term is added to the original objective function.

Hence, the data association based tracking problem is formulated as:

$$\begin{aligned} \arg \min_{\phi \in \Phi, \psi \in \Psi} & \sum_{ij} \phi_{ij} M_{ij} + \alpha \sum_{ik} \psi_{ik} D(F_i, G_k) \\ \text{s.t. } & \forall i, j, k \quad \phi_{ij}(\psi_{ik} - \psi_{jk}) = 0 \end{aligned} \quad (2.3)$$

where ψ is a grouping correspondence function:

$$\psi_{ik} = \begin{cases} 1, & \text{if tracklet } i \text{ is assigned to group } k, \\ 0, & \text{otherwise} \end{cases} \quad (2.4)$$

with an additional constraint that $\sum_k \psi_{ik} = 1$, which denotes that one tracklet should only belong to one group. Let Ψ be the set of valid group correspondences.

The constraint of Eqn. 2.3 denotes that if two tracklets are linked, they should also be assigned to the same group. $D(F_i, G_k)$ is the relative distance measure between tracklet i and group trajectory G_k . α is selected as a weighting parameter to balance the contributions of traditional visual cues and grouping behaviors.

Nevertheless, the proposed grouping-based model of [91] is performed at a pedestrian level and the K-means clustering is utilized to generate the initial groups, which means the number of groups is a fixed value. More recently, Chen *et al.* [92] propose an online learned elementary grouping-based model for multi-target tracking. Compared with [91], the proposed grouping-based model in [92] is more flexible by exploiting elementary groups. Moreover, the elementary grouping approach in [92] is more effective and efficient for modeling grouping behaviors.

2.3 Concluding Remarks

In this chapter, a brief literature review of prominent object detection and tracking methods, which motivates the works in this thesis, is presented. Although the

above two topics are different, they are implicitly related to each other. Recently, tracking-by-detection based tracking methods has been significantly benefited from the progress of object detection.

For human detection, a transferred convolutional neural network (T-CNN) is proposed to learn semantic pedestrian part detectors in later chapter of the thesis. Different from the previous methods described in Section 2.1, the proposed method explicitly learn visual patterns of semantic pedestrian parts, which are critical for occlusion handling.

In the first work on multi-target tracking reported in later chapter of this thesis, the problem is formulated in a network flow framework as introduced in Section 2.2.2.1. Unlike previous network flow based methods [4–6, 28], each node in the network represents a tracklet, and each edge represents the likelihood of neighboring tracklets belonging to the same trajectory as measured by the proposed tracklet affinity score.

In the second multi-target tracking work reported in later chapter of this thesis, a unified deep model for tracklet association is proposed. Although deep architectures have been employed in single object tracking [64–68] as presented in Section 2.2.1, to the best knowledge of the author, it is the first piece of work to exploit deep convolutional neural networks in multi-object tracking.

Chapter 3

Human Detection with Occlusion Handling

Human detection plays an important role in surveillance video analytics. Although many promising methods have been proposed for human detection, it is still very challenging due to cluttered background, view variations, illumination changes and severe occlusions.

In this chapter, an effective detection framework to improve the performance of generic detectors in highly crowded scenes is first presented. It can be considered as an extension of the deformable part model (DPM) [3] for severe occlusion handling. However, this detection framework only focus on enhancing the head part of human with the assumption that the heads of humans is less likely to be occluded in highly crowded surveillance videos. In the second part of this chapter, more general semantic pedestrian part detectors, which are learned through a novel transferred CNN model, are introduced. This work can be viewed as a more comprehensive and principled way of dealing with human detection with occlusion handling. Hence, more emphasis is put on the latter work.

3.1 Pedestrian Detection in Highly Crowded Scenes

Using “Online” Dictionary Learning for Occlusion Handling

3.1.1 Introduction

The state-of-the-art pedestrian detectors [3, 7, 21, 46] have achieved very good performance when detecting pedestrians in relatively less crowded scenes, where the occlusions are not severe. However, their performance on crowded scenes is poor, due to the severe mutual occlusions of the pedestrians, and sometime the problem of small target size.

There has been plenty of literature that addresses occlusion handling in object detection [8–12], which can further promote the tracking-by-detection based multi-target tracking methods [29, 93] within surveillance applications. Modeling occlusions as regions which are inconsistent with target statistics is one common approach. A review of related work has been presented in Chapter 2. Nevertheless, these methods cannot handle severe occlusions in crowded scenes without any 3D information.

This work addresses such difficulties by proposing the use of “online” multiple-instance dictionary learning to refine a generic detector for enhancing head detection. The intuition is that, in a real crowded scene captured by a surveillance camera, individual target is usually mutually occluded by others, while the head of each individual is less likely to be occluded or at most slightly occluded, as shown in Fig. 3.1. Based on this assumption, this work increases the weight of the head detection in the deformable part-based model (DPM) [3] and learns a discriminative dictionary for heads in an “online” manner to improve detection performance.

Figure 3.1: Examples of crowded scenes and their corresponding detection results with the proposed approach are presented in row 1 and row 2, respectively.

Learning is an important part in computer vision tasks, such as stereo image representation [94], image classification [95, 96], image retrieval [97], object categorization [98], object tracking [27, 99], face recognition [100] and human gender recognition [101]. Dictionary learning is chosen in this work. The dictionary (code-book) enables the proposed method to enforce the explicit representations rather than individual features or simple combinations of the features. The dictionary can also facilitate hierarchical representations. Moreover, dimensionality reduction can be achieved through the quantization process.

Figure 3.2: The framework of the proposed approach.

The proposed detection framework consists of the steps illustrated in Fig. 3.2. First, the weight of the head part in the deformable part-based model [3] is increased and this modified detector with a relatively low detection threshold is applied on each frame of a video sequence. The reliable detection responses are selected by their high detection confidences. The head parts of the reliable detection responses are used as the positive samples and the other body parts of the reliable detection responses are used as one of the two types of the negative samples. The other type of negative samples are collected automatically from background, where there is no detection responses. Second, the samples of above three different types are used to learn a discriminative dictionary. Third, the online learned dictionary is used to refine the head parts of the initial detection responses. Finally, the refined pedestrian detection results are obtained.

The rest of this section is organized as follows. In Section 3.1.2, the initial pedestrian detection with enhanced head detection is introduced. Section 3.1.3 presents the “online” dictionary learning. Section 3.1.4 shows the experimental evaluations on several challenging crowded video sequences. Finally, Section 3.1.5 concludes the work in this part.

Figure 3.3: The deformable part-based model from [3].

3.1.2 Initial Detection with Enhanced Head Detection

The deformable part-based model for pedestrian detection similar to [3] often fail when the pedestrians are severely occluded. This is because the detection score in [3] is computed from all the body parts, without considering that most of the parts may be occluded in crowded scenes. The detection score at location (x, y) is computed in [3] as:

$$S_1(x, y) = b + \sum_{i=1}^{i=n} s(p_i) \quad (3.1)$$

where b is a bias term, n is the number of the body parts, and $s(p_i)$ is the score of body part i .

In this formulation, it is obvious that even if a body part was occluded, its corresponding score still makes equal contribution to the final detection score. This can drastically influence the detection performance especially when dealing with highly crowded video sequences. To address this problem, the weight of the head part score in equation (3.1) is proposed to be increased. As shown in Fig. 3.3, the second part represents the head in the DPM model. Hence, equation (3.1)

can be reformulated as:

$$S_2(x, y) = b + s(p_1) + \alpha \cdot s(p_2) + \sum_{i=3}^{i=n} s(p_i) \quad (3.2)$$

where α is a weighting coefficient of the head part ($\alpha = 2.5$ in our implementation).

Here, a simple sweep search scheme was used to find the value of α . A parameter value of α is uniformly sampled in the range $[1, 6]$. The step value is 0.1. 50 independent runs were conducted on the first half of the PETS S1.L2 sequence with the above varying parameter values. Finally, the weight $\alpha = 2.5$ was found to achieve the best performance in terms of log-average miss rate. Hence, $\alpha = 2.5$ was selected.

Nevertheless, relying on DPM detector with head part enhanced is not sufficient to obtain satisfactory detection results. Hence, the “online” dictionary learning is utilized to further refine the initial detections.

3.1.3 “Online” Dictionary Learning

The objective here is to “online” learn a discriminative dictionary while keeping the computational complexity low. The learning involves “online” training sample collection, feature representation and “online” learning.

The initial detections with scores higher than a certain threshold ω ($\omega = 3$ in our implementation) are selected as the reliable detections. Then perform learning as outlined in Fig. 3.2 and the positive samples are collected from the reliable detections while the negatives samples are collected from other body parts and background.

For feature representation, image patches are sampled at an interval of 4 pixels in horizontal and vertical directions. At each sampling location, patches of sizes 8×8 and 16×16 pixels are taken. Each image patch is resized to 8×8 and five

types of features are computed to represent it. The patch size for feature extraction is chosen according to the actual sizes of the heads of the pedestrians in PETS dataset [1]. The computed features are HOG [7], LBP [9], GIST [102], encoded SIFT [13] and LAB color histograms.

The discriminative dictionary is “online” learned by the max-margin multiple-instance dictionary learning (MMDL) algorithm in [103]. This discriminative dictionary comprises a set of linear classifiers (G-code classifiers) for different patch clusters from the three sample classes. For each sample class, the proposed method uses the training images in this class as positive samples, and the remaining training images from other classes as negative samples. Suppose $K+1$ G-code classifiers are learned through MMDL. Given a test image, patch-level features are densely extracted. x is defined as a patch feature vector, whose response is given by the i th G-code: $w_i^T x, i \in \{0, 1, \dots, K\}$. Hence, a response map for each G-code classifier can be obtained. A three-level spatial pyramid representation [104] is utilized for each response map, resulting in $(1^2 + 2^2 + 4^2)$ grids. In each grid, the maximal response for each G-code classifier is calculated. Thus, $3 \times (K+1)$ length feature vector is obtained for each grid. Therefore, the test image is compactly represented by the concatenation of features in all grids.

Note that the feature encoding by using G-code involves no more than a dot product operation. Hence, its computation complexity is very low. This can significantly accelerate the speed of the classification process of our pedestrian detection framework.

Subsequently, the “online” learned G-code classifiers are used for feature encoding of each head image. Then, the refined head parts of the initial detection responses are obtained. Finally, the proposed method projects the refined head part images back into the video sequence and generates the final pedestrian detection output.

3.1.4 Experiments

The proposed pedestrian detection framework has been evaluated on two publicly available sequences with highly crowded scenes: PETS S1.L2 and PETS S2.L3. The S1 sequences of PETS dataset are used for person count and density estimation, which means that the density level is higher than the other sequences of PETS dataset. The high density crowd sequence S1.L2 and S2.L3 are chosen for the evaluation. In all the experiments, only the first camera view sequences in PETS dataset are used. Moreover, only the visual information of the sequences are used and no other prior knowledge such as the camera calibration or the statistic obstacles are used.

The proposed method is compared with the original DPM detector and the modified DPM detector with head part enhanced. The criterion of [105] for evaluations is utilized in the experiments. A detection, which has more than 0.5 overlap with the groundtruth, is determined as true positive in the evaluations. The performance is evaluated by the measurements of miss rate and false positives per image. Moreover, as in [105], the log-average miss rate is employed to summarize the detection performance, which is calculated by averaging miss rates evenly spaced in log-space within the range 10^{-2} to 10^0 .

In the implementation, a detection threshold is set as $t_d = 0.3$ for the modified DPM detector to achieve a high recall. The sizes of the images are doubled for better resolution. For the online dictionary learning, the numbers of randomly selected positive training samples and negative training samples are set as $N_{pos} = 200$ and $N_{neg} = 400$ respectively.

As shown in Fig. 3.4, “DPM” denotes the results of the original DPM detector [3]. “DPM+head” denotes the results of the DPM detector with head part enhanced. “Ours” denotes the results of the proposed method. For the ROC evaluation curves shown in Fig. 3.4, the closer to the bottom left corner, the bet-

Figure 3.4: Performance comparison on PETS 2009 dataset.

ter performance it is. It is found that the proposed approach achieves the best performance in both highly crowded sequences.

In addition, the computation speed of the proposed method is relatively fast. Although the initial modified DPM detector takes about 15 seconds for each frame, the extra steps take only 10 seconds on average for each frame on a 3GHz PC with 8 GB memory. Moreover, the proposed approach is implemented in Matlab, which can be further optimized.

3.1.5 Concluding Remarks

In this work, a detection framework to improve the detection performance in highly crowded scenes is proposed. This work first reformulates the score computation of body parts in the original DPM detector to enhance the head part of the deformable part-based model and then uses the “online” learned dictionary to refine the detection responses. The experimental results on two benchmark sequences demonstrate the effectiveness of the proposed method in detecting pedestrians with serve occlusions. Nevertheless, the limitation is that the proposed method

only focuses on utilizing the visual information of the head part of a human, which is assumed to be less likely occluded in crowded surveillance videos. As a result, the proposed method achieves little improvements in performance on Caltech dataset [57] and ETH dataset [106] compared with the DPM detector. Hence, the results of these two datasets are not provided in this work. In the following part of this chapter, a more generic and comprehensive detection framework, which is based on a deep architecture, is introduced.

3.2 Learning Semantic Pedestrian Part Detectors with a Transferred CNN Model

3.2.1 Introduction

Feature representation is an important component in pedestrian detection. Traditionally, pre-defined features such as SIFT [13], HOG [7] and LBP [9] are used. However, they are not able to extract data-driven visual patterns in pedestrian detection. Recently, deep learning models such as the convolutional neural networks (CNN) [14] have achieved very promising performance in many computer vision applications such as image classification [15] as an example. Inspired by this, many deep learning based methods [16–20] have been proposed for pedestrian detection. For occlusion handling, Ouyang et al. [16, 18] utilize deep models to learn the relationships between different parts. However, their method does not explicitly learn visual patterns of semantic pedestrian parts such as head, upper body, and so on, which are critical for occlusion handling. To the best knowledge of the author, no previous work has been done to learn semantic pedestrian part detectors. This section shows that it is beneficial to use visual patterns learned from semantic pedestrian parts for occlusion handling in pedestrian detection.

Figure 3.5: The framework of the transferred CNN (T-CNN).

As mentioned before, learning semantic pedestrian part detectors requires a lot of annotations on body parts. However, most pedestrian detection datasets do not contain pedestrian part annotations. To solve this problem, the proposed method transfers learned pedestrian part detectors from a source dataset with annotations to the target one without annotations by using a transferred CNN (T-CNN) model. The proposed pedestrian detection framework is presented in Figure 3.5. First, the CNN model [14] is pre-trained on the ILSVRC 2012 dataset [15]. Then, the pre-trained CNN model is adapted based on the source data (Oxford Town Center dataset [107]) by using semantic pedestrian part annotations. To make semantic pedestrian parts generic and effective, this work designs a semantic part pool based on part annotations, which includes 9 categories: head, less upper body, upper body, left upper body, right upper body, lower body, left whole body, right whole body and whole body. The learned parameters of this source CNN model is used to initialize the T-CNN model. Finally, the T-CNN model is trained in an iterative way with the samples of semantic pedestrian parts discovered in each iteration. Furthermore, an activation rule of semantic pedestrian parts from

the semantic part pool is designed to only activate effective semantic part detectors on the target data. The activation rule is that only the semantic part categories discovered in the iterative learning process are kept.

As shown in Figure 3.5, the proposed framework consists of two deep CNNs for the source domain and target domain, respectively. First, the CNN [14] of the source domain is pre-trained on the ILSVRC 2012 dataset. Then the last two layers of this CNN is replaced by one classification layer and the whole network is fine-tuned on the auxiliary data. The learned parameters of the internal layers (C1-FC6) is used to initialize the T-CNN (bottom row) of target domain. The learned weight vector w^s of FC7 is passed to the adaptation layer FCb of the T-CNN.

In this work, a margin-based loss function for the adaptation layer is proposed to transfer knowledge from the source dataset to the target one. As a result, the proposed transferred CNN (T-CNN) model can extract more effective pedestrian part detectors for handling pedestrian occlusions on the target dataset.

The main contributions of this work are two-fold:

- Semantic pedestrian part detectors are explicitly learned for handling pedestrian occlusions.
- A unified deep CNN model for jointly learning features, semantic pedestrian part detectors, and a transferring model is proposed. Components within this deep CNN model interact with each other in the learning process, enabling the components to maximize their potentials during the cooperative learning.

Experimental results demonstrate that the proposed pedestrian detection system outperforms other state-of-the-art detectors based on deep networks.

3.2.2 Related Work

In the literature, many generic pedestrian detection methods learn part based models for deformation and occlusion handling. A typical one is the deformable part model (DPM) introduced earlier, which models parts with additional learned-filters in positions anchored with respect to the whole object bounding box. It allows parts to be displaced from the anchor through the learned deformation costs. The extended strong DPM adapts the original DPM with strong supervision setting in which part annotations are used in training. These methods assume that the distribution of source samples is similar to that of target samples. Our contributions aim at tackling the domain adaptation problem, where the data distributions in two domains are significantly different and no part annotations of the training data are available. Moreover, another limitation of these methods is the use of weak discriminative features (usually HOG [7]). Hence, more discriminative features extracted from the learned transferred CNN model are proposed to be used in this work.

Recently, inspired by the success of convolutional neural network, deep network based pedestrian detection methods [16–20, 108] have been proposed. [16] and [17] propose to learn the visibility relationship among overlapping parts of pedestrians by deep networks. Zeng et al. [108] propose a new multi-stage contextual deep model for pedestrian detection. A deep model that jointly learns feature extraction, deformation handling, occlusion handling and classification for pedestrian detection is proposed in [18]. Luo et al. [20] propose a switchable deep network to model background clutter and complex appearance variations for pedestrian detection. An unsupervised multi-stage feature learning method based on deep network is proposed in [19]. However, these methods model part locations without explicit part training data, which may result in errors in predicted the part locations. Different from these deep network based methods, our method utilizes explicit part

annotations to learn semantic pedestrian part detectors, which are effective for classifying pedestrian candidate regions with partial or serve occlusions.

Transfer learning aims to transfer knowledge from the source domain to the target one. Saenko et al. [109] propose to adapt object models from a specific visual domain to another one with different imaging conditions. Aytar and Zisserman [110] propose to utilize an object detector learned for the source category to regularize the training of a new category. Gopalan et al. [111] propose to utilize the incremental learning to produce intermediate representations of data between two domains for object recognition. This work proposes to transfer the learned semantic pedestrian part detectors from the source dataset to the target one through deep models.

In summary, previous deep network based pedestrian detection methods usually utilize the deep models to model contextual information, background clutters, visibility relationship, deformations, and occlusions for pedestrian detection. Nevertheless, these methods cannot explicitly learn visual patterns of the pedestrian parts, which are very important for pedestrian detection. This work aims to learn semantic pedestrian part detectors through the transferred CNN model in the absence of annotated training data. Moreover, the proposed unified deep CNN jointly learns features, semantic pedestrian part detectors and the transferring model.

3.2.3 The Overview of the Proposed Transferred CNN

A novel transferred convolutional neural network (T-CNN), which is a deep model for learning adaptive semantic pedestrian part detectors, is proposed. With this deep model, the proposed method is able to detect pedestrians occluded at different levels (partially or severely), even though the part annotations of the training data are not available. Different from previous part-based models, such as DPM [3], JointDeep [18] and MT-DPM [112], which use latent or hidden variables to rep-

resent part locations, the proposed deep model can explicitly learn representative visual patterns for semantic pedestrian parts. By leveraging the part annotations of the auxiliary data and the strong transfer learning power of the proposed transferred CNN model, our pedestrian detection system can help to deal with the occlusion problem without part annotations in the training data.

As shown in Figure 3.5, the proposed framework consists of two deep networks. For the deep network in the source domain, the architecture of [14], which achieved good performance on the challenging ILSVRC 2012 dataset [15], is adopted. This deep network consists of five consecutive convolutional layers followed by three fully connected layers. Since this architecture contains more than 60 million parameters, the proposed framework starts by pre-training the convolutional neural network (CNN) on the ILSVRC 2012 dataset. Then, the last two fully connected layers are replaced by one classification layer with K linear SVM classifiers, corresponding to the semantic part categories plus background as shown in Figure 3.5. This CNN model is fine-tuned on the auxiliary data.

The source CNN model is learned on the auxiliary data with detailed part annotations. However, on the target dataset, there is no such annotations. Thus, it is not possible to directly train a similar model from scratch. And, directly applying the source CNN to the target test data is not wise, because the data statistics (typical viewpoints, background, imaging conditions, etc) are different. Hence, a new method is proposed to transfer knowledge from source domain to target domain. To achieve this, a novel adaptation layer FCb (as seen in Figure 3.5) is introduced to replace the weight layer FC7 in the source CNN. The details of this adaptation layer will be described in Section 3.2.4. The parameters of layers C1,..., C5, and FC6, FC7 from the CNN of the source domain are used to initialize the parameters of the transferred CNN of the target domain. The dimensionality of the last hidden layer for the two deep networks (FC6 and FCa in Figure 3.5) is

4096. This design choice is consistent with the state-of-the-art deep architectures [14, 113–115].

3.2.4 Training the Transferred CNN (T-CNN)

In this subsection, the details of training the Transferred CNN (T-CNN) will be described.

3.2.4.1 Training the Source CNN Model

Nine possible subsets of parts are utilized to construct the semantic pedestrian part pool, which includes head, less upper body, upper body, left upper body, right upper body, lower body, left whole body, right whole body and whole body. We find that such subsets are representative enough for most of the realistic scenarios. First, part annotations of the auxiliary data are used to generate these nine semantic pedestrian parts. Then the CNN model pre-trained on the ILSVRC 2012 dataset is fine-tuned on the source data of semantic pedestrian parts. Moreover, a category for background is added. As a result, the classification layer of this source CNN model consists of ten linear SVM classifiers. Background samples are collected by using the region proposal method of [116]. All the training samples are warped to a fixed 224×224 image patch size.

3.2.4.2 Online Sample Collection for Semantic Pedestrian Parts

For the target domain, only the whole body annotations are available. It is needed to online collect training samples. Hence, the region proposal method [116] is employed to generate candidate training samples.

At first, the learned source CNN model is used to classify region proposals of the training images in the target domain. By setting a relatively high classification score threshold, a small number of positive training samples are obtained for the

semantic part categories. Then, the transferred CNN model with the adaptation layer is used to learn semantic pedestrian part detectors in an iterative way. The number of the positive training samples is gradually increased during this mining process. This process usually takes 3-5 rounds.

Nevertheless, the number of online collected training samples is still too small to learn good discriminative and representative models for the target data. Hence, the transferred CNN model with strong transfer learning power is proposed to adaptively learn semantic pedestrian part detectors at each round.

3.2.4.3 The Proposed Transferred CNN Model

To learn semantic pedestrian part detectors on the target dataset, a novel fully-connected adaptation layer is proposed to be added in the deep convolutional neural network. The adaptation layer (FCb) is comprised of K adaptive SVM classifiers for the activated semantic part categories and background respectively. For each adaptive classifier, it aims to learn the target model w by regularizing the distance between the target model w and the source model w^s .

Given training data $x_i \in R^D$ and its corresponding labels $t_i \in \{-1, 1\}$, $i = 1, \dots, N$, the adaptive SVM learning consists of the following constrained optimization:

$$\begin{aligned} \min_{w, b} \quad & \|w - \lambda w^s\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & \xi_i = \max(0, 1 - t_i(w^T x_i + b)) \end{aligned} \quad (3.3)$$

where λ controls the amount of transfer regularization, w^s is the model parameter learned from the source data, ξ_i is the hinge loss function, C controls the weight of the hinge loss, and N is the number of samples.

The objective function of Equ. (3.3) is the primal form of L1-norm SVM with standard hinge loss. As L1-norm SVM is not differentiable, a popular variation L2-norm SVM is introduced, which minimizes the squared hinge loss:

$$\begin{aligned} \min_{w,b} \quad & \|w - \lambda w^s\|^2 + C \sum_{i=1}^N \xi_i^2 \\ \text{s.t.} \quad & \xi_i = \max(0, 1 - t_i(w^T x_i + b)) \end{aligned} \quad (3.4)$$

The above objective function of L2-SVM form is differentiable and imposes a bigger loss (quadratic vs linear) for data points that violate the margin.

The output of the k -th adaptive classifier in the adaptation layer is:

$$a_k(x) = w_k^T x + b_k \quad (3.5)$$

Then the predicted category of a region proposal is:

$$\arg \max_k a_k(x) \quad (3.6)$$

3.2.4.4 Back Propagation in the Transferred CNN Model

The objective function in Equ. (3.4) is utilized to train convolutional neural networks for pedestrian detection. The lower layer weights are learned by back propagating the gradients from the top adaptation layer. It is necessary to differentiate the objective with respect to the activation of the penultimate layer. Assume that the objective function in Equ. (3.4) is $L(w, b)$, and the activation of the penultimate layer is h , which is used to replace the input x . Hence, the objective function for back propagation (BP) learning is as follows:

Figure 3.6: The flowchart of pedestrian detection system at the testing stage.

$$\frac{\partial L(w, b)}{\partial h_i} = 2w - 2\lambda w^s - 2Ct_i w (\max(0, 1 - t_i(w^T h_i + b))) \quad (3.7)$$

Based on the objective function in Equ. (3.7), the backpropagation algorithm utilized in training the proposed transferred CNN model is the same as the standard softmax-based deep learning networks.

3.2.5 The Transferred CNN at the Test Stage

Training the transferred CNN model uses both auxiliary and target training samples. At the test stage, the transferred CNN model is applied to detect pedestrians in test images. An overview of the pedestrian detection at the test stage is shown in Figure 3.6.

Main steps of the proposed pedestrian detection system at the test stage are as follows:

1. Proposal generation: The region proposal method of EdgeBoxes [116] is employed in the proposed detection system.

2. Feature extraction: The learned transferred CNN model is used to extract hierarchical features from each region. The dimension of the features extracted from the last hidden layer (FCa) is 4096.

3. Region classification: The multiclass adaptive SVMs, which are learned together with the deep networks at the training stage, are used to assign a score

to each pedestrian candidate region.

4. Region refinement: A class-specific bounding-box regressor, similar to the one introduced in [8], is used to do bounding-box regression for pedestrians with different parts. The goal is to learn a transformation that maps a proposed pedestrian box P to a ground-truth whole body box G . Since nine different semantic part categories are proposed in the pedestrian detection framework, nine such transformations are needed to learn to map the semantic pedestrian part boxes to the final whole body boxes. Subsequently, non-maximum suppression (NMS) is utilized to further refine the detection results.

The complete step-by-step pseudo-code algorithm description of the proposed detection method is summarized in Algorithm 1.

3.2.6 Experiments

3.2.6.1 Experimental Settings

All the experiments are conducted on the PETS 2009 dataset [1], the Caltech dataset [57] and the ETH dataset [106]. The Oxford Town Center dataset [107] with part annotations is used as the auxiliary data to train the source CNN model. The resolution in this Oxford Town Center dataset is 1920×1080 . However, the resolutions of PETS 2009 dataset, Caltech dataset and ETH dataset are 768×576 , 640×480 and 640×480 , respectively. To better obtain pedestrian candidates with relative small resolution, the sizes of images are doubled for the input of the EdgeBoxes [116] region proposal method.

The number of positive training samples of the auxiliary data for each part detector is 6000. The number of negative training samples of the auxiliary data is 18000. When training the T-CNN, the ratio of positive and negative samples in each learning batch is 1:3. The negative samples are randomly sampled from the background patches of the auxiliary data. For PETS 2009 dataset, the number of

Algorithm 1 Learning Semantic Pedestrian Part Detectors with a Transferred CNN Model

Input:

- Auxiliary data with part annotations;
- Target data without part annotations;

Output:

- The learned transferred CNN model; semantic pedestrian part detectors; detection results.
- 1: The training samples of each semantic part categories from the auxiliary data are warped to a fixed 224×224 image patch size for fine-tuning the source CNN model.
 - 2: The source CNN model with linear SVM classification layer is off-line learned on the auxiliary data.
 - 3: For target data, the region proposal method [116] is utilized to generate candidate training samples.
 - 4: The learned source CNN model is used to classify the generated candidate training samples. By setting a relatively high classification score threshold, a small number of positive samples of each semantic part categories are online collected for learning the transferred CNN model.
 - 5: The transferred CNN model is learned through Eq. 3.4 and 3.7 via BackPropagation algorithm with the online collected samples and the auxiliary data. With the current transferred CNN model, the positive samples of each semantic part categories are online collected following the mining rules of previous step.
 - 6: The previous step usually takes 2-4 rounds. The adaptive classifiers of semantic pedestrian parts from the last layer of the transferred CNN model are actually the semantic pedestrian part detectors.
 - 7: At the test stage, given the testing images of the target data, the region proposals are first generated by the method of [116]. Then all the candidate regions are warped to a fixed 224×224 image patch size.
 - 8: The learned transferred CNN model is used to extract hierarchical features for each candidate region.
 - 9: The learned adaptive classifiers (semantic pedestrian part detectors) are utilized to assign a score to each pedestrian candidate region through Eq. 3.6.
 - 10: After the post processings: bounding-box regression and non-maximum suppression, the final detection results are obtained.
-

positive training samples collected online for each part detector is about 300; the number of negative samples collected online is about 900. For Caltech dataset, the number of positive training samples collected online for each part detector is about 800; the number of negative samples collected online is about 2400. For

ETH dataset, the number of positive training samples collected online for each part detector is about 1000; the number of negative samples collected online is about 3000.

The training of the deep model is based on batch-mode and the outputs of the classifiers in the last layer of the deep model are normalized. The regularization parameter λ in the objective function (3.4) is obtained by cross-validation.

For PETS 2009 dataset, the online training performed 100 frames; For Caltech dataset, the online training performed 4309 frames; For ETH dataset, the online learning performed 1804 frames. The online training performed 3 rounds for all datasets.

In fact, the semantic pedestrian part detectors are proposed for occlusion handling. In practical situations, when the pedestrians are occluded, only parts of the pedestrians are visible. The region proposal method can generate the region proposals of the parts of the pedestrians which are not occluded. Hence, the proposed part detectors can detect the parts of the pedestrians from the region proposals. Finally, the detection bounding boxes of the occluded pedestrians are recovered by transforming the semantic part boxes into the whole body boxes. Therefore, the training data does not contain occlusions as the only some non-occluded parts are needed for successful detection.

For the purpose of evaluation, the same experimental protocol as in [105] is utilized and the algorithm performance is evaluated by using the ROC curves. Measurements of *miss rate and false positives per image* are used to evaluate the overall performance of different pedestrian detection methods, which are the same with the first work of this chapter.

To show the effectiveness of the semantic pedestrian part detectors and the transferred CNN model, two baselines for our method are designed:

- **Baseline1:** no semantic part detectors (only whole body detector is used).

- **Baseline2:** with semantic part detectors, but the semantic part detectors are not adapted using the target data.

Figure 3.7: Performance comparison on PETS 2009 dataset. Since the two sequences in this dataset are densely crowded, Baseline1 is not effective. Only the results of Baseline2 are shown here.

3.2.6.2 Results on PETS 2009 Dataset

The dense crowd sequences PETS S1.L2 and PETS S2.L3 are chosen for the experiments. In these two sequences, most pedestrians are occluded severely by each other, which makes it challenging. Moreover, no other prior knowledge such as camera parameters are employed in the experiments.

The first half of the PETS S1.L2 sequence is used to train the transferred CNN model. Since these two sequences are obtained under the same conditions, the learned transferred CNN model is used to do testing for both sequences. Our detector is compared with the DPM detector [3] and Baseline2. Because these two sequences are densely crowded, in which most pedestrians are severely occluded, the detector trained only using the whole body samples is not effective. Hence, the results of Baseline1 are not presented here.

Figure 3.8: Results of the baseline detectors and ours on the Caltech dataset under standard evaluation settings.

The results in Figure 3.7 show that our detector outperforms the DPM and Baseline2 in both sequences, which substantiates the effectiveness and superiority of the learned semantic pedestrian part detectors of the transferred CNN model in handling severe pedestrian occlusions. Moreover, Baseline2 has better performance than the DPM detector on the two sequences. This indicates that the proposed semantic pedestrian part detectors are more effective than the DPM detector even in the lack of transferring power.

3.2.6.3 Results on Caltech Dataset

As known, the Caltech dataset is the most challenging dataset for pedestrian detection. The subsets set00 - set05 are used for training and the subsets set06 - set10 are used for testing as in [105].

For this dataset, our detector is compared with the state-of-the-art detectors whose detection results are publicly available [57] and two baseline detectors. 49

detectors are compared. Among them, the detection results of the 46 detectors are obtained from the Caltech Pedestrian Detection Benchmark website [57]. The results are produced based on the reasonable setting [105], which only uses pedestrians at a height of more than 50 pixels and a visibility of at least 65%.

The target training data is used to train the CNN model for Baseline1. For Baseline2, the Oxford Town Center dataset with part annotations is used to train the CNN model. Moreover, the target training data is also used to train the whole body detector in Baseline2. As shown in Figure 3.8, Baseline2 achieves 5% improvement in performance compared with Baseline1, which proves the effectiveness of the semantic part detectors. Furthermore, our method achieves 9% improvement in performance compared with Baseline 2, which verifies the superiority of the transferred CNN model.

The overall performance of the state-of-the-art detectors is shown in Figure 3.9. This figure only shows the performance of the detectors rank in the first 15 places excluding VJ [117] and HOG [7]. Our detector ranks the fourth place among all the 49 detectors. The detectors rank in the first 3 places (SpatialPooling+ [118], Katamari [119], and LDCF [120]) utilize pre-defined features, which are different from ours. However, these 3 methods requires annotated learning samples in the target domain. They do not have the ability to transfer learned models from source domain. Hence, our method becomes useful when there is insufficient annotated samples to properly train the detectors. The proposed adaptive deep features can be complementary to these features. Note that our detector outperforms the recent InformedHaar detector [121] by 6%. Compared with MT-DPM [112] and MT-DPM+Context [112], which are two typical part-based detectors, our performance is better for 12% and 9%, respectively.

To better present the performance of all the deep network based detectors, the results are shown in a different figure. As shown in Figure 3.10, our detec-

Figure 3.9: Results of the state-of-the-art detectors and ours on the Caltech dataset under standard evaluation settings.

tor achieves the best performance using the standard evaluation settings. This demonstrates the superiority and effectiveness of the proposed transferred CNN model. Notice that our detector outperforms JointDeep detector [18], which uses deep neural networks to jointly model parts and occlusions, by 10%.

The ground truth annotations of the Caltech dataset contains the information about the occlusion percentage for each human box. The experimental results of human detection under different occlusions on Caltech dataset are generated by using the above ground truth annotations with occlusion information. Figure 3.11 shows the evaluation results under different occlusion conditions on the Caltech dataset. Three occlusion levels: no occlusion (0% occluded), partial occlusion (1-35% occluded), and heavy occlusion (35%-80% occluded) are used for the experiments. Our detector achieves the best performance in all the three oc-

Figure 3.10: Results of the deep network based detectors and ours on the Caltech dataset under standard evaluation settings.

clusion settings among all the deep network based detectors, which demonstrates the superiority of our detector in handling occlusions.

Figure 3.11 shows the performance of the proposed methods against other methods on varying degree of occlusions. The performance on heavily occluded situations is not significantly improved, but better improvement is shown in partial occlusion and no occlusion cases. This is because the parts are often occluded in heavy occlusion case. This is an indication of the limit on methods based on a single camera view images. Hence, it is recommended to explore other imaging methods such as 3D imaging to handle heavy occlusions (see chapter 6, section 6.2).

Figure 3.11: Detection results under different occlusion conditions on the Caltech dataset.

3.2.6.4 Results on ETH Dataset

To evaluate the generalization capacity of our method, only the Oxford Town Center dataset (auxiliary data) is used as the training data, which is similar with the state-of-the-art methods [18, 20, 122]. In this setting, the proposed T-CNN model is online learned on the ETH dataset by using the online collected training samples. Our detector is compared with the state-of-the-art detectors whose detection results are available in [57]. 32 state-of-the-art detectors are included in

Figure 3.12: Results of the state-of-the-art detectors and ours on the ETH dataset under standard evaluation settings.

the comparison. Figure 3.12 plots the results on the ETH dataset, which shows the performance of the detectors rank in the first 15 places excluding VJ [117] and HOG [7]. Our detector ranks the second place among all the 33 detectors. Furthermore, the proposed adaptive deep features can be complementary to the handcrafted features used in [123], which ranks in the first place.

Only the whole body annotations of the Oxford Town Center dataset are used to train the CNN model for Baseline1. For Baseline2, the Oxford Town Center dataset with part annotations is used to train the CNN model. As shown in Figure 3.13, Baseline2 outperforms Baseline1 by 3%, which verifies the effectiveness of the semantic part detectors. And our method outperforms Baseline2 by 6%, which verifies the superiority of the transferred CNN model.

Figure 3.14 shows the evaluation results of the deep network based detectors under different settings on the ETH dataset. Two different settings: reasonable

Figure 3.13: Results of the baseline detectors and ours on the ETH dataset under standard evaluation settings.

setting [105], which is based on the pedestrians at a height of more than 50 pixels and a visibility of at least 65%; overall setting [105], which is based on all the annotated pedestrians. As shown in Figure 3.14a, our detector achieves the best performance under the reasonable setting. Due to the lack of occlusion labels of the ETH dataset, all the annotated pedestrians, which include many occluded pedestrians, are used to do further evaluation. As shown in Figure 3.14b, our detector achieves 44% log-average miss rate, which is 7% better than the second best deep network based detector [20].

3.2.6.5 Run-Time Analysis

The region proposal generation and feature extraction are two main time-consuming modules of our detector. Our detector is implemented using the MatConvNet toolbox [37] on a 1.80GHz PC with a Tesla K40c GPU.

Moreover, the time of the off-line training of the source CNN model on the

Figure 3.14: Detection results of the deep network based detectors under different settings on the ETH dataset. Reasonable setting: based on the pedestrians at a height of more than 50 pixels and a visibility of at least 65%. Overall setting: based on all the annotated pedestrians.

auxiliary data is about 1 hour. The average speed of the online training is about 13 seconds per frame for all the datasets. The average detection time of our detector is around 12 seconds per frame. Currently, the proposed detection framework cannot be used to implement real-time applications.

3.2.7 Concluding Remarks

In this section, a novel pedestrian detection system, which pre-learns the semantic pedestrian part detectors on the source dataset with annotated pedestrian parts and then adapts the pre-learned semantic pedestrian part detectors based on the target dataset without the annotations, is proposed. In particular, an adaptation layer is proposed to embed the capability of knowledge transferring into the CNN model. As a result, the proposed transferred CNN (T-CNN) model is able to transfer the visual knowledge of the semantic pedestrian parts from the source data to target data. Experimental results demonstrate that the proposed pedestrian detection method achieves very promising performance for handling pedestrian

occlusions.

The proposed detection algorithms still have some limitations. They are far from meeting the requirements of real time applications using conventional computing platform due to the use of deep learning for feature extraction from thousands of region proposals. Parallel implementations on powerful GPUs can help to speed up but may not achieve real time performance. Hence, the proposed detection algorithms are more suitable for human detection in archived surveillance videos in which real time performance is not critical. Hence, better detection is achieved at higher cost of computation.

The human detector in section 3.1 is not combined or used in the later T-CNN based human detector. The human detector presented in section 3.1 was developed to improve the detection performance for highly crowded scene, its performance in other types of scene is not significantly better than other state-of-art methods. The T-CNN based human detector is for more different situations and achieve better performance than several state-of-the-art detectors in all these situations. However, it is much more computationally demanding than the one in section 3.1.

Chapter 4

Tracklet Association by Online Target-Specific Metric Learning and Coherent Dynamics Estimation

4.1 Introduction

Automatic tracking of multiple humans in a real world video sequence is a significant but challenging topic in computer vision, and it is needed in many applications such as surveillance, robotics, traffic safety, sports analysis and entertainments. In these applications, it is often crucial to detect and track people precisely in complex and crowded real scenes. However, most existing multi-human tracking approaches still have some limitations, and one of them is in preserving human identities.

In this chapter, this work addresses the challenges in long-term tracking of multiple humans in a complex scene captured by a single, uncalibrated camera with an aim of achieving consistent human identity tracking (i.e. no identity

switches). This is a challenging problem due to many sources of uncertainty, such as clutter, serious occlusions, targets interactions, and camera motion.

Recently, significant progress has been reported in human detection [3, 7, 9, 21–24], and this promotes the popular tracking paradigm: detect-then-track [4–6, 25–30]. The main idea is that a human detector is run on each frame to detect targets of interest, and then detection responses are linked across multi-frames to obtain target trajectories. In [4–6, 28], the authors formulate the multi-target data association as a network flow optimization problem. However, due to the limitation of the appearance cues used for tracking, the methods mentioned above usually cannot deal with longer term tracking to obtain a complete trajectory of a target. This is because prolonged occlusions and target-to-target interactions will result in fragmentation of a trajectory. If we make full use of the information from the whole sequence (previous, current, and subsequent frames), trajectory can be recovered from the fragments and tracking errors such as missed tracks or identity switches can be corrected. Then, similarity measurement between two track fragments (tracklets) to determine whether they belong to the same person becomes very critical in this tracklet association problem. Some of the state of art methods, such as in [32] and [25], fuse several features based on motion, time, position, size and appearance to improve the similarity measurement. However, their appearance models are still inadequate to handle large appearance variations, thus adversely affecting tracking performance.

In the earlier work [93], a discriminative target-specific appearance-based affinity model is advocated to reinforce the appearance cues for multi-human tracking. Unlike the PIRMPT system proposed by [87], which needs off-line learned local descriptors, the proposed target-specific metrics are online learned during the tracking. In [93], a motion constraint based on heuristics is utilized. In this chapter, motion dynamics are exploited to further improve tracking of target’s identity.

Furthermore, the significance of the appearance and motion cues on tracking performance are studied separately. Different from previous works [86–88], which simply multiply the motion and appearance affinities to obtain the linking probabilities of two tracklets, a learning algorithm to automatically learn the weights of the two terms from labeled training data is developed. The learned weights can enhance the tracking cues with strong discriminative power and suppress the tracking cues with weak discriminative power. As a result, the weighted tracking cues can disambiguate the targets’ respective identities better even in situations such as the one depicted in Figure 4.1.

Among the early multi-target tracking methods, particle filtering has been a popular approach. Okuma et al. [124] combines Adaboost for object detection with mixture of particle filters for multiple-object tracking. Cai et al. [125] extend this boosted particle filter by using a rectification technique and embedding mean-shift algorithm to increase the robustness for multiple objects tracking. Breitenstein et al. [29] uses the continuous confidence of human detectors and online-trained, instance-specific classifiers as a graded observation model. These methods rely on the past and current frames to achieve the final tracking results. So they are suitable for online applications. Nevertheless, they may fail when prolonged occlusions and frequent interactions of targets occur and they are sensitive to false detection responses.

Recently, significant progress has been reported in human detection and tracking, based on a new popular tracking paradigm: detect-then-track. A typical way of doing this is to track multiple targets frame by frame, which often encounters irrecoverable errors if a target is undetected in one or more successive frames or if two detections are erroneously linked. To overcome this weakness, the global trajectory optimization methods over batches of frames have been proposed in recent years, such as Linear Programming [79, 126] and Dynamic Programming [6, 127].

Figure 4.1: A difficult scenario of high appearance similarity among targets. (Frames from PETS dataset with pedestrian identities labeled by our method): Despite individuals 1 and 8 dressed in similarly colored clothes and severe occlusions and interactions between individuals 1, 8, and 9, their identities should remain unchanged.

These methods are often based on graphical network optimization in which the nodes are represented by detection responses. Such methods often fail to handle the problems of long-term tracking in crowded scenes well. To alleviate this, some researchers [74, 87, 128] try to use the track fragments (tracklets) as graph nodes aiming at linking tracklets into long trajectories. This kind of Tracklet Association-based Tracking (TAT) methods can increase robustness and reduce the computation complexity of the graph optimization.

There are two key components of a TAT approach: (1) The tracklet affinity

model that estimates the likelihood of two tracklets belonging to the same target;

(2) The global optimization framework for tracklet association that determines the links of the tracklets based on their affinity scores. Yang et al. [86] propose to learn non-linear motion patterns and robust appearance models with a multiple instance learning (MIL) algorithm for tracklet affinity measurements. In [129], Yang et al. introduce online learned discriminative part-based appearance models for tracklet affinity measurements in a tracklet association framework. The Generalized Linear Assignment (GLA) framework is proposed for tracklet association by Dicle et al. [82]. Yang et al. [74] present an online conditional random field (CRF) framework, which incorporates pairwise models to differentiate difficult pairs of targets, for tracklet association. However, due to the long duration gaps between tracklets or less effective appearance-based models, many existing TAT methods are not capable of handling long-term occlusions and interactions between targets. The proposed framework utilizes online learned target-specific metrics with strong discriminative power and allows longer gaps between tracklets to be linked.

The proposed multi-target tracking approach in this chapter is a TAT method based on network flow optimization. This chapter reports that our method is applied to tracking pedestrians in real scenes, but it can be generalized to tracking any other objects in diverse situations. The framework of this method is shown in Figure 4.2.

Given a video sequence, an existing detector, such as the Deformable Part Model (DPM) detector [21], is first utilized to detect pedestrians in each frame. The initial tracklets are generated based on motion trajectory using the successive shortest path algorithm [6]. The initial tracklets may be unreliable because the detection responses in one tracklet may come from more than one person. The proposed target-specific metric learning using the online collected samples from these initial tracklets is introduced. Then the online learned target-specific metrics are

Figure 4.2: The proposed framework. In the cost-flow network, each node denotes a reliable tracklet; The flow costs of edges are defined by negative log of the affinity scores, which are obtained through the online learning of target-specific metrics and motion dynamics with the off-line learned weights on segments of short-time sequences known as local segments.

used to refine these initial tracklets for reliable tracklets. The cost-flow network is based on the reliable tracklets and its optimization yields the long-term trajectories of multiple persons. Estimating the transition costs is the key factor in the min-cost network flow optimization. This work proposes to learn tracklet affinity models, which include weighted discriminative appearance and motion cues, in an online manner for estimating the transition costs.

The main contributions of the proposed work in this chapter are as follows:

- Online learning of target-specific metrics with strong discriminative power through a two-step target-specific metric learning and metric refinement processes.
- Utilizing both appearance and motion dynamics in the tracklet affinity models, which are updated within each local segment for reduced computation and locally adaptive affinity models.

- A learning algorithm to learn the weights of motion and appearance tracking cues for tracklet affinity models.

The rest of this chapter is organized as follows: Section 4.2 describes the cost-flow network formulation. Section 4.3 presents the online learning of the tracklet affinity models. The learning of weights is presented in section 4.4. Experimental results and comparisons are shown in section 4.5. Section 4.6 concludes the whole chapter.

4.2 Cost-flow Network Formulation for Trajectory Recovery by Tracklet Association

The cost-flow network has been shown to be effective for estimating the trajectories in the previous studies [4, 6, 28]. However, in these works, the graph nodes are defined by the detection responses. In recent works [27, 74, 86, 87, 130, 131], Tracklet Association-based Tracking (TAT) methods were proposed for multi-target tracking. In these methods, the tracklets were generated based on association of detection responses. In this chapter, the initial tracklets are generated based on motion trajectory using the successive shortest path algorithm. The initial tracklets are then refined by the proposed online learned target-specific metrics for reliable tracklets. This work can construct a smaller graph based on such tracklets which are of a higher order of abstraction than those based on detection responses. The problems in long-term multi-human tracking can be solved by directly linking tracklets instead of detection responses.

An objective function, which takes a similar form as detection association in [4], is defined for tracklet association. Let $X = \{F_i\}$ be the collection of all the tracklets. A single trajectory hypothesis is defined as an ordered list of N tracklets: $T_k = \{F_{k_1}, F_{k_2}, \dots, F_{k_l}\}$, where $F_{k_i} \in X$, and $i = 1, \dots, l; 1 \leq l < N$. A

tracklet association hypothesis $\mathcal{T}$ is defined as a set of single trajectory hypotheses:

$$\mathcal{T} = \{T_k\}.$$

The objective of tracklet association is to maximize the posteriori probability of $\mathcal{T}$ given X :

$$\begin{aligned} \mathcal{T}^* &= \arg \max_{\mathcal{T}} P(\mathcal{T}|X) \\ &= \arg \max_{\mathcal{T}} P(X|\mathcal{T})P(\mathcal{T}) \\ &= \arg \max_{\mathcal{T}} \prod_i P(F_i|\mathcal{T})P(\mathcal{T}) \end{aligned} \quad (4.1)$$

assuming that the likelihood probabilities of F_i are conditionally independent.

It is assumed that the motion of each tracklet is independent and one tracklet can only belong to one trajectory. Then the above equation can be further decomposed into:

$$\mathcal{T}^* = \arg \max_{\mathcal{T}} \prod_i P(F_i|\mathcal{T}) \prod_{T_k \in \mathcal{T}} P(T_k) \quad (4.2)$$

$$s.t. \quad T_k \cap T_l = \Phi, \forall k \neq l \quad (4.3)$$

The second term in Equ. (4.2) is defined as follows:

$$\begin{aligned} P(T_k) &= P(\{F_{k_1}, F_{k_2}, \dots, F_{k_l}\}) \\ &= P_s(F_{k_1}) \left(\prod_{n=1}^{l-1} P(F_{n+1}|F_n) \right) P_t(F_{k_l}) \end{aligned} \quad (4.4)$$

$P(F_i|\mathcal{T})$ is the likelihood function of tracklet F_i . It is assumed that there are no false alarms from the reliable tracklets, so $P(F_i|\mathcal{T}) = 1$. Then Equ. (4.2) can

be further simplified as follows:

$$\begin{aligned}
 \mathcal{T}^* &= \arg \max_{\mathcal{T}} \prod_i P(F_i | \mathcal{T}) \prod_{T_k \in \mathcal{T}} P(T_k) \\
 &= \arg \max_{\mathcal{T}} \prod_{T_k \in \mathcal{T}} P(T_k)
 \end{aligned} \tag{4.5}$$

$P(T_k)$ is modeled as a Markov chain, which includes a starting probability $P_s(F_{k_1})$, a termination probability $P_t(F_{k_l})$, and transition probability $P(F_{n+1} | F_n)$ between temporarily adjacent tracklets. Finding the optimal association hypothesis $\mathcal{T}^*$ is equivalent to minimizing the cost of flow from source s to sink t in a network flow graph. A network graph can be constructed as follows:

Given an observation set X : for every tracklet $F_i \in X$, we create a node v_i , an edge from source s to a node, (s, v_i) , with cost $c(s, v_i) = c_i^s$ and flow $f(s, v_i) = f_i^s$, and an edge from a node to sink t , (v_i, t) with cost $c(v_i, t) = c_i^t$ and flow $f(v_i, t) = f_i^t$. For every transition $P(F_j | F_i) \neq 0$, create an edge (v_i, v_j) , $i \neq j$, with cost $c(v_i, v_j) = c_{ij}$ and flow $f(v_i, v_j) = f_{ij}$. The logarithm of the objective function is taken to simplify the expression while preserving the maximum a posteriori probability (MAP) solution. Then, Equ. (4.5) can be re-written as follows:

$$\mathcal{T} = \arg \min_{\mathcal{T}} \left(\sum_i c_i^s f_i^s + \sum_{ij} c_{ij} f_{ij} + \sum_i c_i^t f_i^t \right) \tag{4.6}$$

$$s.t. \quad f_{ij}, f_i^s, f_i^t \in \{0, 1\},$$

$$and \quad f_i^s + \sum_j f_{ji} = f_i^t + \sum_j f_{ij} \tag{4.7}$$

subject to Equ. (4.6), where

$$c_i^s = -\log P_s(F_i), \quad c_i^t = -\log P_t(F_i),$$

$$c_{ij} = -\log P(F_j | F_i).$$

Equ. (4.7) ensures that the tracklet association hypothesis $\mathcal{T}$ is non-overlapping. The above formulation can be mapped into a cost-flow network with a source s and a sink t . Estimating the transition costs c_{ij} is very critical in solving this min-cost network flow problem. Previous network flow approaches [4–6, 28] only utilize motion cues across consecutive frames and simple appearance features such as color histograms to calculate c_{ij} . Nevertheless, these cues are not very reliable when long-term occlusions and interactions between targets occur. In this chapter, the segment-wise tracklet affinity models, consisting of weighted tracking cues, are proposed to be online learned for estimating c_{ij} .

4.3 Online Learning of Tracklet Affinity Models

In this section, the online learning of tracklet affinity models, consisting of online target-specific metric learning and online motion dynamics estimating, are introduced. The affinity scores of adjacent tracklets, which are used as the transition probabilities between two corresponding nodes in the cost-flow network, can be obtained through tracklet affinity measurements. The local transition probabilities estimation is performed within a local segment of S frames ($S = 50$ in the implementation).

In order to obtain effective appearance cues for reliable transition probability estimation, a novel target-specific appearance-based model is proposed. The appearance-based model learning problem is formulated as a metric learning problem, which can enhance the features with strong discriminative power and suppress the features with weak discriminative power. Here, target-specific metrics are learned so that target-specific properties can be efficiently explored for more discriminative models. In contrast to the previous work of [87] in which local descriptors are learned offline, our learning is online throughout and our target-

specific metrics are adaptive to local segments. Moreover, to create a more discriminative tracklet affinity model, this work also explores the motion dynamics cue and embeds it into the proposed tracklet affinity model. The motion dynamics are online estimated without any assumed priors. As a result, the learned tracklet affinity models can better represent the appearance and motion cues adaptively and provide reliable transition probability estimation.

4.3.1 Online Target-Specific Metric Learning

It is aimed to online learn discriminative target-specific metrics while keeping the computational complexity low. For each tracklet F_i , we learn a distance metric function.

The learning involves feature representation, online training sample collection and online training. To create a strong appearance-based model, this work starts from a rich set of basic features, which includes color, shape and texture, to describe a pedestrian's appearance.

Given a training dataset $Z = \{(z^t, l_i)\}_{i=1}^n$, where z^t is a feature vector representing the appearance of a detection response at frame t , and l_i is the tracklet label which the detection response belongs to. This work defines a positive difference vector x_i^p computed between a pair of relevant samples (detection responses belonging to the same person) and a negative difference vector x_i^n computed from a pair of irrelevant samples (detection responses belonging to different persons). Here, it is assumed that the first M frames of each initial tracklet are reliable and the detection responses are from the same person. Training samples are therefore collected from these frames.

The difference vectors x_i^p and x_i^n are defined as follows:

$$\begin{aligned} x_i^p &= d(z_i, z'_i) = |z_i - z'_i| \\ x_i^n &= d(z_i, z'_j) = |z_i - z'_j|, \quad i \neq j \end{aligned} \quad (4.8)$$

where d is an absolute difference function, z_i and z'_i are two samples from the same tracklet F_i , z'_j is a sample from a different tracklet F_j .

Given the difference vectors x_i^p and x_i^n , a distance function D_i for tracklet F_i can be learned based on relative distance comparison so that $D_i(x_i^p) < D_i(x_i^n)$. This distance function D_i is parameterized as a Mahalanobis distance function:

$$D_i(x) = x^T M_i x, \quad M_i \succeq 0 \quad (4.9)$$

The logistic function as in [132] is adopted to learn D_i to force $D_i(x_i^p)$ to be small, and $D_i(x_i^n)$ to be big:

$$\min_{D_i} r(D_i) = -\log \left((1 + \exp(D_i(x_i^p) - D_i(x_i^n)))^{-1} \right) \quad (4.10)$$

Furthermore, the term M_i in the distance function D_i can be decomposed by eigendecomposition:

$$M_i = A_i \Lambda_i A_i^T = W_i W_i^T, \quad W_i = A_i \Lambda_i^{\frac{1}{2}} \quad (4.11)$$

where A_i is the orthonormal eigenvector matrix of M_i and the diagonal of Λ_i are the corresponding eigenvalues.

Therefore, learning a distance function D_i is equivalent to learning the matrix

W_i as follows:

$$\begin{aligned} \min_{W_i} r(W_i), s.t. \quad & w_i^T w_j = 0, \forall i \neq j, w_i, w_j \in W_i \\ r(W_i) = & \log(1 + \exp\{\|W_i^T x_i^p\|^2 - \|W_i^T x_i^n\|^2\}) \end{aligned} \quad (4.12)$$

Online training sample collection is another important issue in online learning. The q strongest ($q = 4$ in this work) detection responses in each tracklet are used as training samples. For x_i^p , we collect relevant sample pairs from the same tracklet. However, for x_i^n , we collect irrelevant sample pairs from different persons. To determine the relevance of sample pairs, two constraints are employed: spatio-temporal and exit constraints. The first constraint is based on the fact that one person cannot appear at two or more different locations at the same time. The second constraint is based on the observation that the person who has already exited the view cannot be the person who is still within the view. The irrelevant samples, which satisfy the above two constraints, are online collected from F_i and F_j respectively to form irrelevant pairs.

Once online training sample collection is finished, the gradient descent method is utilized to learn W_i for each tracklet F_i . Finally, the target-specific transform matrices for all the tracklets are obtained:

$$W = \{W_i\}, \quad i = 1, \dots, N \quad (4.13)$$

4.3.2 Tracklet Refinement

To solve the objective function in Equ. (4.6), it is needed to identify reliable tracklets for the nodes in the network graph. The initial tracklets are created by the successive shortest path algorithm as in [6]. This method uses spatio-temporal information such as distance between corresponding observations in adjacent frames

to link the detections into tracklets. Without effective use of appearance cues, the initial tracklets may be not consistent in appearance and hence unreliable when there are many interactions or occlusions between targets. A typical error is that there are some detection responses belonging to different persons in one tracklet. Hence, tracklet refinement is needed to separate tracklets into multiple short but reliable ones.

The online learned target-specific metrics are employed to refine the initial tracklets. To construct the probe set, the detection with the strongest detection response, g_i , is selected from the first M frames of an initial tracklet, F_i , which are assumed to be reliable. It is defined as $G = \{g_i\}$, $i = 1, \dots, N_s$, where N_s is the number of tracklets in a local segment. Each tracklet F_i has only one selected g_i in G .

The target-specific transform matrix W_i for each initial tracklet is learned after collecting training samples as described in previous sub-section. The identity test is carried out within a local segment frame by frame to obtain the relative distance between the detection response at frame t of F_i and the corresponding g_i in the probe set:

$$\begin{aligned} x_i^t &= d(z_i^t - g_i) = |z_i^t - g_i|; \quad i = 1, \dots, N_s \\ d_i^t &= \|W_i^T x_i^t\|^2 \end{aligned} \tag{4.14}$$

where z_i^t is an instance from tracklet F_i at frame t , g_i is the corresponding detection response of F_i in G , and d_i^t is the relative distance between z_i^t and g_i .

For a reliable tracklet, the relative distance between the current detection response z_i^t and the probe g_i should be small; otherwise, it is an unreliable tracklet. A distance threshold ω is used to identify reliable tracklets. In a tracklet F_i , if K ($K = 5$ in our implementation) consecutive detection responses having relative

distance values (from g_i) above ω , we split F_i into two parts from the first consecutive detection response. The above process is repeated multiple times until there are no unreliable tracklets.

4.3.3 Online Tracklet Dynamics Estimation

To disambiguate targets with similar appearance, motion dynamics are proposed to be exploited together with appearance cues as described above to keep track of target's identity. The main idea of tracklet dynamics estimation is to model the evolution of target motions as a sequence of piecewise linear regressors whose orders can be estimated from the available data.

Similar to [133, 134], this work collects the position information of all the detection responses within one tracklet in a vector y and assumes that its value at current time q is related to its past values y_{q-i} by an m^{th} order autoregressive model of the form:

$$\begin{aligned} y_q &= a_1 y_{q-1} + a_2 y_{q-2} + \dots + a_m y_{q-m} \\ &= \sum_{i=1}^m a_i y_{q-i}, \quad m \leq N_f, q \geq s + m \end{aligned} \tag{4.15}$$

where $\mathbf{a} = [a_m \ a_{m-1} \ \dots \ a_1]^T$ is the regressor vector, N_f is the total number of frames of the tracklet, m is the number of frames of the trajectory y_q and s is the starting frame of the tracklet.

Hence, a tracklet F_i can be interpreted as an ordered sequence of dynamic measurements $\{y_q\}$, $s \leq q \leq t$, where s and t are the starting and terminating frames, respectively.

The order of the autoregressive model m measures the complexity of the underlying tracklet dynamics. Specifically, a well known result from the realization

theory [135, 136] is that, under mild conditions, given an ordered sequence of measurements $\{y_q\}$ generated by Equ. (4.15), the order m of the autoregressive model equals to the rank of the corresponding Hankel matrix, *i.e.*, $m = \text{rank}(H_{F_i}^{(n)})$ where $H_{F_i}^{(n)}$ is the Hankel matrix with $n \geq m$ columns:

$$H_{F_i}^{(n)} \doteq \begin{bmatrix} y_s & y_{s+1} & \cdots & y_{s+n-1} \\ y_{s+1} & y_{s+2} & \cdots & y_{s+n} \\ \vdots & \vdots & \vdots & \vdots \\ y_{t-n+1} & y_{t-n} & \cdots & y_t \end{bmatrix} \quad (4.16)$$

The motion dynamics similarity $P_m(F_i, F_j)$ between two tracklets F_i and F_j , which takes a similar form as in [134], is defined as follows:

$$P_m(F_i, F_j) = \begin{cases} -\infty, & \text{if temporal conflict exists;} \\ \frac{\text{rank}(H_{F_i}) + \text{rank}(H_{F_j})}{\text{rank}(H_{F_{ij}})} - 1 & \end{cases} \quad (4.17)$$

where F_{ij} is the joint tracklet with the gap between F_i and F_j interpolated.

Here, it is assumed that the targets do not significantly change their dynamics between tracklets. The intuition of the above motion dynamics similarity is that if two tracklets are from the same trajectory then they can be approximated by one relatively low order regressor. Otherwise, if two tracklets are from different trajectories, the joined trajectory needs a higher order regressor than the regressors of each single tracklet. Hence, if $\text{rank}(H_{F_i}) = r(F_i)$ and $\text{rank}(H_{F_j}) = r(F_j)$, then $\text{rank}(H_{F_{ij}}) = r(F_{ij}) \leq (r(F_i) + r(F_j))$. Consequently, if F_i and F_j are of the same trajectory, then $r(F_i) = r(F_j) = r(F_{ij})$ and $P_m(F_i, F_j) = 1$, but if not, $P_m(F_i, F_j) \approx 0$.

Tracklet dynamics are online estimated without any prior knowledge based on the reliable tracklets. The IHTLS (iterative Hankel Total Least Squares) method in

[82] is employed to estimate the rank of the Hankel matrices for tracklet dynamics estimation.

4.3.4 Tracklet Affinity Measurement

In this subsection, the measurement of the affinity between F_i and F_j , or equivalently, the transition probability, P_{ij} , in the network graph between node i and node j is presented. The tracklet affinity score, $\mathcal{S}_{ij}$, which is equivalent to P_{ij} , is defined as follows:

$$\mathcal{S}_{ij} = P_m(F_i, F_j)P_a(F_i, F_j)\mathcal{C}_{ij} \quad (4.18)$$

where $P_m(F_i, F_j)$ is the motion-based affinity model, which is defined by Equ. (4.17), $P_a(F_i, F_j)$ is the appearance-based affinity model and $\mathcal{C}_{ij}$ is a limiting function.

To obtain the appearance-based affinity model $P_a(F_i, F_j)$, this work first computes the relative distances d_{ij}^t between each detection response in F_i and the probe g_j , and $d_{ji}^{t'}$ between each detection response in F_j and the probe g_i ,

$$\begin{aligned} x_{ij}^t &= |z_i^t - g_j|, \quad x_{ji}^{t'} = |z_j^{t'} - g_i|; \quad i, j = 1, \dots, N_s \\ d_{ij}^t &= \|W_i^T x_{ij}^t\|^2, \quad d_{ji}^{t'} = \|W_j^T x_{ji}^{t'}\|^2 \end{aligned} \quad (4.19)$$

where z_i^t denotes the feature vector of a detection response in tracklet F_i at frame t , $z_j^{t'}$ denotes the feature vector of a detection response in tracklet F_j at frame t' , and $g_i, g_j \in G$.

Subsequently, this work calculates the mean values of the relative distances and

uses them to define the appearance-based affinity model $P_a(F_i, F_j)$:

$$d_{ij} = (\sum_t d_{ij}^t)/m, \quad d_{ji} = (\sum_{t'} d_{ji}^{t'})/n \quad (4.20)$$

$$P_a(F_i, F_j) = (d_{ij}d_{ji})^{-1}\gamma \quad (4.21)$$

where γ is a normalization term and m, n are the number of frames of F_i and F_j respectively.

It is no need to apply Equ. (4.18) to every pair, since there are a lot of obviously non-related tracklet pairs which do not belong to the same trajectory. Because a limiting function $\mathcal{C}_{ij}$ is included in Equ. (4.18), this work actually applies it for every tracklet pair. This limiting function $\mathcal{C}_{ij}$ is proposed based on spatio-temporal, and exit constraints:

$$\mathcal{C}_{ij} = C_t(F_i, F_j)C_e(F_i, F_j) \quad (4.22)$$

The spatio-temporal constraint is defined as follows:

$$C_t(F_i, F_j) = \begin{cases} 1, & \text{if } F_i \cap F_j = \phi \\ 0, & \text{otherwise} \end{cases} \quad (4.23)$$

where $\cap$ is an intersection operator that is used to find the overlap between two tracklets.

The exit constraint is defined as:

$$C_e(F_i, F_j) = \begin{cases} 1, & \text{if } t_i^s > t_j^e \text{ \& } p_j^{t_j^e} \notin E \\ 0, & \text{otherwise} \end{cases} \quad (4.24)$$

where t_i^s is the starting frame of tracklet F_i , t_j^e is the ending frame of tracklet F_j , $p_j^{t_j^e}$ is the position of the detection response of tracklet F_j at time t_j^e and E is the

exit area which is near image borders. For static cameras, the incremental learning algorithm for exit map as in [86] is adopted to obtain E .

$C_t(F_i, F_j)$ and $C_e(F_i, F_j)$ associate F_i and F_j if they have no overlap and F_i does not exit the screen when F_j appears.

The transition costs of the adjacent nodes in the cost-flow network is obtained by taking negative logarithm of the affinity scores between corresponding tracklets:

$$c_{ij} = -\log \mathcal{S}_{ij} \quad (4.25)$$

Finally, the optimal tracklet association hypothesis $\mathcal{T}^*$ in Equ. (4.6) is estimated based on c_{ij} .

However, in the proposed tracklet affinity model as depicted in Equ. (4.18), the motion-based affinity model, $P_m(F_i, F_j)$, and appearance-based affinity model, $P_a(F_i, F_j)$, are treated equally without any voting weights. This may result in inaccurate affinity scores if one of the tracking cues is dominant and the other one is confusing, having an effect as a noise factor. Therefore, the influences of the two tracking cues on tracking performance are investigated in next section.

4.4 Learning of Affinity Weights

In difficult situations, where severe occlusions and interactions occur, the motion-based affinity model and appearance-based affinity model may not be consistent. Hence, it is needed to weight them properly for stable performance.

A weighting parameter λ , which controls the weight of the motion-based affinity score, is proposed to be added in Equ. (4.18).

$$\mathcal{S}_{ij} = [P_m(F_i, F_j)]^\lambda P_a(F_i, F_j) \mathcal{C}_{ij}, \quad 0 \leq \lambda \leq 1 \quad (4.26)$$

where λ is learned from labeled data. If the value of λ is larger, the motion-based affinity score, $P_m(F_i, F_j)$, contributes more to $\mathcal{S}_{ij}$.

4.4.1 Assessment of Difficult Situations for Motion Dynamics

To investigate the difficult situations where the motion affinities are not reliable, only the motion-based affinity model is used to estimate c_{ij} for Equ. (4.6) in the experiments. After analyzing the inconsistencies between the tracking results and the labeled ground truth data, these typical situations, where the motion affinities are not reliable, are obtained. Based on the analysis, a rule is designed to automatically assess these difficult situations.

There are two constraints in this rule. The first one is as follows:

$$\begin{aligned} S(z_i^{t^e}) \cap S(z_k^{t^e}) &\geq \eta * \min(S(z_i^{t^e}), S(z_k^{t^e})) \text{ or} \\ S(z_i^{t^s}) \cap S(z_k^{t^s}) &\geq \eta * \min(S(z_i^{t^s}), S(z_k^{t^s})) \end{aligned} \quad (4.27)$$

where $z_i^{t^e}$ and $z_i^{t^s}$ are the detection responses of F_i at the ending frame t^e and the starting frame t^s , respectively. $z_k^{t^e}$ and $z_k^{t^s}$ are the detection responses of F_k at the ending frame t^e and the starting frame t^s , respectively. $S(\cdot)$ is the operator to capture the area of the detections in pixels. η is a sensitivity threshold ($\eta = 0.3$ in our implementation).

The second constraint is that at least one linked candidate of each tracklet has gaps in the temporal space.

If the tracklet pair $\{F_i, F_k\}$ match the above two conditions, then the weighting parameter λ is added to the motion-based affinity models related to F_i and F_k as $[P_m(F_i, F_j)]^\lambda$ and $[P_m(F_k, F_l)]^\lambda$. F_j, F_l are the linked candidate tracklets of F_i and F_k respectively. As shown in Figure 4.3, the tracklet pairs with identities 269, 289

Figure 4.3: An example of the difficult situations.

and 305,306 match these two constraints. Hence, the weighting parameter λ for these 4 tracklets in terms of the motion-based affinity models is added.

4.4.2 Learning of the Weighting Parameter

The weighting parameter λ in Equ. (4.26) defines the weight of the motion-based affinity model for tracklet association. Based on the frames of the gaps between

tracklets, the weighting parameter λ is divided into 2 levels:

$$\lambda = \begin{cases} \lambda_1, & 1 \leq u \leq B_1 \\ \lambda_2, & u > B_1 \end{cases} \quad (4.28)$$

where u is the number of frames in the gap between corresponding tracklets ($B_1 = 20$ in the implementation).

The intuition is that if the gaps are longer, it is more difficult to accurately estimate the joint tracklet dynamics. Therefore, 2 difficulty levels are defined as in Equ. (4.28), making the weighting parameter λ more adaptive to the difficult situations. The number of the difficulty levels and the upper bound value of level 1 (B_1) are empirically determined. Furthermore, tracking performance evaluation and network flow optimization are employed jointly to optimize the weighting parameters for tracklet association.

Given the reliable tracklets and the labeled ground truth data of a video sequence, it is aimed to learn the weighting parameters in a supervised manner so that the tracking performance can be optimized. The different difficult levels of the weighting parameters are optimized independently in a greedy fashion. After some repeatedly iterations with a fixed step value, the weighting parameters $\{\lambda_1, \lambda_2\}$ are obtained. The learning algorithm is summarized in Algorithm 2.

The weighting parameters λ_1 and λ_2 are learned from the ground truth data of PETS 2009 [137] in this chapter. The proposed tracking algorithm is then run with the learned weighting parameters on all the datasets for evaluation. Based on the analysis of the experimental results with different B_1 , it is found that the tracking performance is slightly affected by the changes of B_1 . A upper bound value B_1 of (20-30) frame gap works well for all sequences.

The complete step-by-step pseudo-code algorithm description of the proposed tracking method in this chapter is summarized in Algorithm 3.

Algorithm 2 Weighting parameter learning for tracklet association

Input:

Reliable tracklets;
 Labeled ground truth data;

Output:

The learned weighting parameters: $\{\lambda_1, \lambda_2\}$;

- 1: Initialize the weighting parameters: $\lambda_1 = \lambda_1^l = 0, \lambda_2 = \lambda_2^l = 0$ and the step value $\Delta\lambda = 0.1$;
- 2: Online estimate transition costs for all graph node (tracklet) pairs based on:
 $c_{ij} = -\log(P_m(F_i, F_j)P_a(F_i, F_j)\mathcal{C}_{ij})$;
- 3: **for** $i=1$ to 2 **do**
- 4: **while** $\lambda_i^l \leq 1$ **do**
- 5: **for all** graph node (tracklet) pair $\{F_i, F_k\}$ matches the rule of the assessment of difficult situations **do**
- 6: **if** $u \geq 1$ and $u \leq B_1$ **then**
- 7: $\lambda = \lambda_1$;
- 8: **else if** $u > B_1$ **then**
- 9: $\lambda = \lambda_2$;
- 10: **end if**
- 11: **for all** tracklet pairs related to F_i, F_k **do**
- 12: $c_{ij} = -\log([P_m(F_i, F_j)]^\lambda P_a(F_i, F_j)\mathcal{C}_{ij})$;
- 13: $c_{kl} = -\log([P_m(F_k, F_l)]^\lambda P_a(F_k, F_l)\mathcal{C}_{kl})$;
- 14: **end for**
- 15: **end for**
- 16: Obtain tracking results through network flow optimization;
- 17: **if** Current tracking results better converge to ground truth data **then**
- 18: $\lambda_i = \lambda_i^l$;
- 19: **end if**
- 20: $\lambda_i^l = \lambda_i^l + \Delta\lambda$;
- 21: **end while**
- 22: **end for**
- 23: **return** $\{\lambda_1, \lambda_2\}$.

4.5 Experiments

4.5.1 Datasets

To evaluate the performance of the proposed approach, four challenging publicly available pedestrian datasets are used for the experiments.

TUD. The TUD Crossing sequence [138] and TUD-Stadtmitte sequence [2] are real-world videos filmed in busy pedestrian streets. The cameras are positioned at a

Algorithm 3 Tracklet Association by Online Target-Specific Metric Learning and Coherent Dynamics Estimation

Input:

Input video sequence;

Output:

Tracking results.

- 1: A state-of-the-art detector such as DPM detector is used to generate human detections.
 - 2: After tracklet initialization step, it is the online training sample collection step within a local segment. Based on the spatio-temporal and exit constraints, the negative and positive training pairs are online collected.
 - 3: All the online collected samples are warped to a fixed 128×64 pixel size.
 - 4: The target-specific metrics are learned via Eq. 4.12 by using the online collected samples from initial tracklets.
 - 5: The learned target-specific metrics are used to refine the initial tracklets for reliable tracklets.
 - 6: The target-specific metrics are relearned via Eq. 4.12 by using the online collected samples from reliable tracklets. Meanwhile, the motion dynamics are online estimated via Eq. 4.17 based on reliable tracklets.
 - 7: The tracklet affinities within this local segment is obtained via Eq. 4.26 with an off-line learned weighting parameter λ .
 - 8: The above steps 2-7 are repeated until all the local segments are processed.
 - 9: The reliable tracklets (graph nodes) and tracklet affinities (transition probabilities) are passed to the cost-flow network. This global network flow optimization yields the long-term trajectories of multiple humans. After the post processing step – trajectory completion, the final tracking results are obtained.
-

quite low angle, resulting in more complex occlusion patterns and rather inaccurate ground plane locations. Furthermore, for the TUD-Stadtmitte sequence, the size of the pedestrians on the image plane vary drastically.

PETS 2009. The PETS 2009 benchmark dataset [137] presents an outdoor scene with large number of pedestrians captured from multiple cameras at 7 fps. Note that the pedestrians vary significantly in appearance due to shadows and lighting changes. Moreover, there are frequent occlusions, which are caused by pedestrian occluding each other, or static occlusions such as the traffic sign. In the experiments, the sequence S2L1 in the first view, which is widely used in multi-target tracking literature, is selected.

Town Centre. The Town Centre dataset [139] is captured by a single elevated camera in a busy town centre street. There are 16 pedestrians visible at any time on average, leading to frequent dynamic occlusions and interactions. Furthermore, due to the severe occlusions caused by static obstacles such as benches, many pedestrians are not detected by the state-of-the-art detectors.

ETH. The ETH BAHNHOF and SUNNY DAY sequences [41] show busy street scenes from a pair of cameras on a moving stroller. The stroller is moving forward at most of the time, however there are still some panning motions, which leads to the unreliable motion affinities between tracklets. Moreover, frequent full or partial occlusions occur due to the low view angles of cameras. And the size of the pedestrians also varies significantly on the image plane.

4.5.2 Experimental Settings

The online collected training samples from video frames are normalized to 128×64 pixels for target-specific metric learning. For the color feature, RGB, YCbCr and HSV color histograms are extracted with 16 bins for each channel respectively and concatenated into a 144-element vector. To capture shape information, this work adopts the Histogram of Gradients (HOG) feature [7] by setting the cell size to be 8 to form a 3968-element vector. Two types of texture features are extracted by Schmid and Gabor filters. In total, 13 Schmid channel features and 8 Gabor channel features are obtained to form a 336-element vector by using a 16-bin histogram vector to represent each channel. Each person image is thus represented by a feature vector in a 4448-dimensional feature space.

4.5.3 Evaluation Metrics

Since it is difficult to use one single score to evaluate multi-target tracking performance, the evaluation metrics defined in [5, 26], as well as the standard CLEAR

MOT metrics [140], are utilized:

- $Recall(\uparrow)$: correctly matched detections / total detections in ground truth.
- $Precision(\uparrow)$: correctly matched detections / total detections in the tracking result.
- $FAF(\downarrow)$: number of false alarms per frame.
- $FP(\downarrow)$: number of false positives.
- $FN(\downarrow)$: number of false negatives.
- GT : number of trajectories in ground truth.
- $MT(\uparrow)$: number of mostly tracked trajectories.
- PT : number of partially tracked trajectories.
- $ML(\downarrow)$: number of mostly lost trajectories.
- $Frag(\downarrow)$: number of fragmentations.
- $IDS(\downarrow)$: number of id switches.
- $MOTA(\uparrow)$: Multi-object tracking accuracy.
- $MOTP(\uparrow)$: Multi-object tracking precision.
- $IDS/correctly\ matched\ detections(\downarrow)$.

For evaluation items with $(\uparrow)$, higher scores denote better performance; for evaluation items with $(\downarrow)$, lower scores denote better performance. The evaluation codes are downloaded from [15].

4.5.4 Quantitative Evaluation

The quantitative evaluations are presented in three sub-sections: abbreviations of the proposed methods in the experiments, comparison with network flow based methods, and comparison with other state-of-the-art methods on benchmark datasets.

4.5.4.1 Abbreviations of Different Methods

- **CML**: The proposed method with only an online learned common class metric for all tracklets.
- **TSML**: The proposed method with online target-specific metric learning.
- **TD**: The proposed method with only tracklet dynamics.
- **TSML+TD**: The proposed method with online target-specific metric learning and tracklet dynamics.
- **TSML+TD+WP**: The proposed method with full tracklet affinity model including target-specific metric learning (**TSML**), tracklet dynamics (**TD**) and weighting parameters (**WP**).

The baselines **CML** and **TSML** are from our previous work [93].

4.5.4.2 Comparison with Network Flow based Methods

The proposed method is first evaluated on the popular TUD Crossing sequence [138] and ETH BAHNHOF sequence [41]. For a fair comparison, the same sequences and pre-trained pedestrian detector as in [5] are used. The quantitative metric is defined as *ID switches / total number of correct observations used in the trajectories (IDS / correctly matched detections)*, which is the same as in [5]. Table 4.1 gives the quantitative results computed on the TUD Crossing sequence, and the first 350 frames of the ETH BAHNHOF sequence. Due to the forward and panning

Table 4.1: Comparison of tracking results with other network flow based methods on TUD Crossing and ETH BAHNHOF (first 350 frames) sequences. The entries in the table are (IDS)/(correctly matched detections). Columns 1 and 2 use the pre-trained pedestrian detector of [3]. Column 3 shows the results when ground truth detections are used to generate the initial tracklets. The ground truth detections are from [6].

Algorithm	TUD Crossing	ETH	ETH (GT)
DP [6]	32/768	37/1387	25/1648
MCNF [4]	9/433	11/1057	5/922
LRMCNF [5]	14/819	23/1514	14/1783
CML	10/845	5/1728	3/1786
TSML	7/862	1/1790	0/1820

motions of the cameras, the motion affinities between tracklets are unreliable for the BAHNHOF sequence. Therefore, the online tracklet dynamics estimation is not added for the experiments in this subsection. The results of five methods are reported here: DP algorithm of [6] is our baseline work without adding the online learning of tracklet affinity models. The results are also compared to those of other network flow based methods: MCNF [4] and LRMCNF [5]. The proposed method with online target-specific metric learning is denoted as TSML. Furthermore, the results of the proposed method with only a common class metric for all tracklets, which is denoted as CML, are also reported.

Note that our method gives better results when compared with the three network flow methods [4–6]. Moreover, the noticeable improvement in ID switches indicates that our method can better deal with long-term tracking, where the traditional motion models are less reliable. Figure 4.4 shows the superiority of our method.

4.5.4.3 Comparison with State-of-the-art Methods

To show the effectiveness of our method, our method is further compared with other state-of-the-art methods on more publicly available datasets. To make fair

Figure 4.4: Columns 1 and 2 compare the tracking results of LRM CNF [5] (left) and our approach (right) respectively on the ETH BAHNHOF sequence (frames 104 - first row, 130 -second row, 187- third row). Note the detection windows pointed by red arrows. We can see that our approach maintains ID labels more reliably.

Table 4.2: Comparison of tracking results between state-of-the-art methods and ours on PETS 2009 dataset.

Method	MOTA	MOTP	Recall	Precision	FAF	GT	MT	PT	ML	Frag	IDS
Energy Minimization [142]	81.4%	76.1%	-	-	-	19	82.6%	17.4%	0.0%	21	15
DC Tracking [30]	95.9%	78.7%	-	-	-	19	100.0%	0.0%	0.0%	8	10
KSP [28]	80.3%	72.0%	-	-	-	19	73.9%	17.4%	8.7%	22	13
MTMM [143]	83.3%	71.1%	-	-	-	19	89.5%	10.5%	0.0%	45	19
UHMTGDA [144]	97.8%	75.3%	-	-	-	19	100.0%	0.0%	0.0%	8	8
HJMRMT [145]	98.0%	82.8%	-	-	-	19	100.0%	0.0%	0.0%	11	10
(MP) ² T [146]	90.7%	76.0%	-	-	-	19	-	-	-	-	-
DTLE Tracking [75]	90.3%	74.3%	-	-	-	19	78.3%	21.7%	0.0%	15	22
CEMMT [141]	90.6%	80.2%	-	-	-	19	91.3%	4.4%	4.3%	6	11
GMCP-Tracker [76]	90.3%	69.0%	-	-	-	19	89.5%	10.5%	0.0%	54	10
OMTD [29]	79.7%	56.3%	-	-	-	-	-	-	-	-	-
OMAT [147]	92.8%	74.3%	-	-	-	19	100.0%	0.0%	0.0%	11	8
PMPTCS [148]	76.0%	53.8%	-	-	-	-	-	-	-	-	-
OGOMT [149]	98.1%	80.5%	-	-	-	19	100.0%	0.0%	0.0%	16	9
CSL-VOX [150]	89.78%	-	98.28%	91.07%	-	19	-	-	-	-	6
CSL-DPT [150]	88.13%	-	97.64%	90.45%	-	19	-	-	-	-	8
CML	92.1%	86.4%	95.1%	97.6%	0.14	19	94.7%	5.3%	0.0%	26	28
TSML	93.4%	86.4%	96.0%	97.7%	0.13	19	94.7%	5.3%	0.0%	21	18
TD	93.7%	86.3%	96.6%	97.4%	0.15	19	94.7%	5.3%	0.0%	17	13
TSML+TD	94.7%	86.4%	97.2%	97.6%	0.14	19	94.7%	5.3%	0.0%	12	7
TSML+TD+WP	95.3%	86.4%	97.4%	98.0%	0.11	19	94.7%	5.3%	0.0%	11	4

comparisons, the same offline learned human detector [3] is utilized to detect pedestrians as in compared methods. The superiority of our method can be observed from the evaluation results.

PETS 2009. For a fair comparison, the same ground truth as in [141] is used for the experiments, in which all the occurring pedestrians have been annotated. The quantitative results are shown in Table 4.2. As expected, taking tracklet dynamics into account increases the overall tracking performance. Our full tracklet affinity model (TSML+TD+WP) further raises the MOTA by 0.6% and reduces the ID switches by $\approx 43\%$. This indicates that our method with full tracklet affinity model combines motion and appearance cues properly, resulting in further improvement on tracking performance. On the whole, our method with full tracklet affinity model achieves the best performance compared with 16 state-of-the-art methods in terms of MOTP, Precision, FAF, ML and IDS. For other evaluation items, our method also achieves comparative performance.

Town Centre. To show the generality of the learned weighting parameters,

our method with the learned weighting parameters is further evaluated on Town Centre dataset. The ground truth used here is provided by [139], which is the same as in the compared methods. As shown in Table 4.3, the full tracklet affinity model with the weighting parameters achieves better or nearly the same performances on all evaluation items. Compared with the tracklet affinity model without weighting parameters, the MOTA is improved by about 8.4%; recall and precision are improved by about 1.6% and 3.4% respectively; fragments and ID switches are reduced by 23.6% and 24.3% respectively. The obvious improvements in tracking performance indicate that the learned weighting parameters are applicable to new data.

For this Town Centre dataset, due to severe occlusions caused by static obstacles (such as benches) and more frequent dynamic interactions between pedestrians, many pedestrians cannot be detected by the state-of-the-art detectors, thus affecting the tracking performance of the proposed method. The average number of pedestrians per frame of the Town Centre dataset is more than 2 times larger than the other datasets. This means that the density of the Town Centre dataset is more than 2 times denser than the other datasets. All the above factors result in that the tracking accuracies on Town Centre dataset are lower compared to those for the other datasets. These factors also result in that the accuracy by the proposed method on this dataset is superior in terms of some of the performance metrics to other competing methods but not on other performance metrics.

TUD-Stadtmitte. To make a fair comparison, the experiments are conducted using the same ground truth as defined in [74]. The quantitative tracking results are shown in Table 4.4. Note that the proposed method with only target-specific metric learning (TSML) has achieved very good performance already, resulting in rather limited room of improvement. Hence, the tracklet affinity model with CML and TSML are used to do the experiments for this dataset. Compared with [74, 87],

Table 4.3: Comparison of tracking results between state-of-the-art methods and ours on Town Centre dataset.

Method	MOTA	MOTP	Recall	Precision	FAF	GT	MT	PT	ML	Frag	IDS
$(MP)^2T$ [146]	75.7%	71.6%	-	-	-	231	-	-	-	-	-
SGB Tracker [151]	71.3%	71.8%	-	-	-	231	58.6%	34.4%	7.0%	363	165
GMCP-Tracker [76]	75.6%	71.9%	-	-	-	231	-	-	-	-	-
MCNF [4]	69.1%	72.0%	-	-	-	231	53.0%	37.7	9.3%	440	243
SMT [139]	64.3%	80.2%	-	-	-	231	67.4%	26.1%	6.5%	343	222
MSBMT [89]	65.5%	71.8%	-	-	-	231	59.1%	33.9%	7.0%	499	288
OMAT [147]	69.5%	68.7%	-	-	-	231	64.7%	27.4%	7.9%	453	209
WAYWAG [152]	66.6%	71.7%	-	-	-	231	58.1%	35.4%	6.5%	492	302
OGOMT [149]	70.7%	68.6%	-	-	-	231	56.3%	36.3%	7.4%	321	157
CML	55.3%	72.6%	69.9%	86.5%	1.79	231	52.8%	36.4%	10.8%	508	327
TSML	57.3%	72.9%	72.3%	87.5%	1.58	231	60.2%	30.7%	9.1%	326	269
TD	55.7%	73.2%	71.5%	86.9%	1.71	231	55.8%	34.2%	10.0%	362	264
TSML+TD	61.6%	74.3%	74.0%	89.5%	1.38	231	61.9%	29.9%	8.2%	259	214
TSML+TD+WP	66.8%	74.4%	75.2%	92.5%	0.96	231	64.9%	28.2%	6.9%	198	162

Table 4.4: Comparison of tracking results between state-of-the-art methods and ours on TUD Stadtmitte dataset. Note that very good performance has been achieved by TSML, resulting in limited room of improvement. We thus do not employ TD and WP in the experiment.

Method	Recall	Precision	FAF	GT	MT	PT	ML	Frag	IDS
Energy Minimization [142]	-	-	-	9	60.0%	30.0%	0.0%	4	7
DC Tracking [30]	74.7%	84.2%	0.870	10	50.0%	50.0%	0.0%	8	10
PRIMPT [87]	81.0%	99.5%	0.028	10	60.0%	30.0%	10.0%	0	1
Online CRF Tracking [74]	87.0%	96.7%	0.184	10	70.0%	30.0%	0.0%	1	0
CML	95.1%	99.4%	0.030	10	100%	0.0%	0.0%	2	1
TSML	98.0%	99.3%	0.040	10	100%	0.0%	0.0%	3	0

the improvement is obvious for some metrics. Our method achieves the highest recall and the mostly tracked score (MT) among all the methods. It also achieves the lowest ID switches. Meanwhile, our method achieves competitive performance on precision, false alarms per frame and fragments compared with [74, 87].

ETH. To see the effectiveness of the proposed method, this chapter further evaluates it on the challenging ETH dataset [41]. Due to the unreliable motion affinities between tracklets of this dataset, the tracklet affinity models without TD and WP are used in the experiments. For a fair comparison, the ground truth provided by [74] is utilized. The quantitative tracking results are shown in Table 4.5. Our method achieves better or competitive performance on all the commonly

Table 4.5: Comparison of tracking results between state-of-the-art methods and ours on ETH dataset. Note that the forward and panning motions of the cameras lead to the unreliable motion affinities between tracklets of this dataset. We thus do not employ TD and WP in the experiment.

Method	Recall	Precision	FAF	GT	MT	PT	ML	Frag	IDS
PRIMPT [87]	76.8%	86.6%	0.891	125	58.4%	33.6%	8.0%	23	11
Online CRF Tracking [74]	79.0%	90.4%	0.637	125	68.0%	24.8%	7.2%	19	11
DTLE Tracking [75]	77.3%	87.2%	-	125	66.4%	25.4%	8.2%	69	57
LIMC Tracking [153]	83.8%	79.7%	-	125	72.0%	23.3%	4.7%	85	71
CML	78.7%	92.0%	0.710	125	60.0%	29.6%	10.4%	77	19
TSML	80.9%	94.2%	0.520	125	65.6%	24.0%	10.4%	26	5

used evaluation measures. Compared with [87], the most related work, the recall and precision are improved by 4.1% and 7.6% respectively; the MT is improved by 7.2%; false alarms per frame are reduced by 41.6%; and ID switches are reduced by 54.5%. The significant reduction in ID switches and false alarms indicates that the proposed target-specific appearance-based model is superior to the method by [87].

Only the ETH dataset is captured by a moving camera, all the other datasets in this chapter are acquired without camera motion. To capture the ETH dataset, the camera is fasten to a moving stroller. This stroller is mostly moving forward, but sometimes has panning motions, which makes the motion affinity between tracklets less reliable. The proposed tracking dynamics and weighting parameter scheme are only suitable for stationary camera setting. Nevertheless, in real surveillance applications, most of the cameras are static cameras. The proposed tracking dynamics and weighting parameter scheme can be used in such applications.

4.5.5 Tracking Results by Using the proposed Human Detector

In this subsection, the tracking results of the proposed tracking framework by using the semantic pedestrian part detectors presented in Chapter 3, which can

achieve better performance than the DPM detector [3], are provided. For PETS 2009 dataset and Town Centre dataset, the proposed tracking framework with full tracklet affinity model (TSML+TD+WP) is used. For TUD Stadtmitte dataset and ETH dataset, the proposed tracking framework with online target-specific metric learning (TSML) is used. The tracking results are shown in Table 4.6, 4.7, 4.8 and 4.9.

On the whole, the tracking results by using the proposed semantic pedestrian part detectors only achieve little improvements in performance on all datasets. This verifies that the proposed tracking method is robust to the detection responses, which means that the proposed method is less affected by the detectors.

Table 4.6: Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on PETS 2009 dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Using DPM detector [3]	97.4%	98.0%	0.11	90	121	19	94.7%	5.3%	0.0%	11	4
Using the proposed	97.5%	98.1%	0.11	86	117	19	94.7%	5.3%	0.0%	8	4

Table 4.7: Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on Town Centre dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Using DPM detector [3]	75.2%	92.5%	0.96	439	1791	231	64.9%	28.2%	6.9%	198	162
Using the proposed	75.3%	92.6%	0.95	430	1779	231	65.4%	27.7%	6.9%	185	156

Table 4.8: Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on TUD Stadtmitte dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Using DPM detector [3]	98.0%	99.3%	0.04	8	22	10	100%	0.0%	0.0%	3	0
Using the proposed	98.1%	99.4%	0.04	7	21	10	100%	0.0%	0.0%	2	0

Table 4.9: Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on ETH dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Using DPM detector [3]	80.9%	94.2%	0.52	517	1978	125	65.6%	24.0%	10.4%	26	5
Using the proposed	80.9%	94.3%	0.51	510	1963	125	65.6%	24.0%	10.4%	22	5

4.5.6 Computational Speed

The computation speed depends on the number of targets in a video sequence. The proposed method is implemented using MATLAB on a 3.3GHz PC with 8 GB memory. The speed of the proposed method with target-specific metric learning (TSML) is about 13, 6, 10 and 9 fps for PETS 2009, Town Centre, TUD and ETH datasets, respectively, excluding the detection time; for PETS 2009 and Town Centre datasets, the speed of the proposed method with full tracklet affinity model is 11 and 5 fps respectively. Note that speed-up can be achieved by parallel implementations with code optimization.

4.6 Concluding Remarks

This chapter has presented a novel method developed for tracking of multiple objects of interest in the scene over a longer period with the aim to maintain consistent tracking and tagging of objects, reducing identity switches in the presence of occlusions and object-to-object interactions. The proposed method processes the initial tracklets (track fragments) produced by a simple trajectory based tracking algorithm. This chapter proposes a two-step online target-specific metric learning to improve the similarity measure based on the appearance cues, and together with coherent dynamics estimation for tracklets based on the motion cues, constituting the new affinity model. The tracking of objects is accomplished by performing tracklet association with network flow optimization where the nodes in the net-

Figure 4.5: Tracking results of our method (TSML) on (top to bottom) *TUD Crossing*, *TUD Stadtmitte*, *ETH BAHNHOF* and *ETH SUNNY DAY*.

Figure 4.6: Tracking results of our method (TSML+TD+WP) on (top to bottom) *PETS 2009* and *Town Centre*.

work are tracklets. Thus, the proposed method exploiting both appearance and motion cues is capable to prevent identity switches during tracking and recover missed detections. The proposed method is found to be effective even when the appearance or motion cues fail to identify or follow the target due to occlusions or object-to-object interactions. To further improve the proposed method, the weights of these two tracking cues in the affinity model are proposed to be learned. The proposed tracking algorithm has been validated on several public datasets and the experimental results show that it outperforms several state-of-the-art tracking algorithms.

Chapter 5

Joint Learning of Convolutional Neural Networks and Temporally Constrained Metrics for Tracklet Association

5.1 Introduction

As many seminal achievements have been obtained in object detection [3, 7, 9], tracklet association-based tracking methods [27, 74, 82, 87, 92, 93, 154] have become popular recently. These methods usually include two key components: 1) The tracklet affinity model that estimates the linking probability between tracklets (track fragments), which is usually based on the combination of multiple cues (motion and appearance cues); 2) The global optimization framework for tracklet association, which is usually formulated as a maximum a posterior problem (MAP). The global optimal solution can be achieved by various optimization algorithms.

Even though some state-of-the-art methods [82, 86, 87, 155] have achieved much

progress in constructing more discriminative appearance and motion based tracklet affinity models, problems such as track fragmentation and identity switch still cannot be well handled, especially under difficult situations where the appearance or motion of an object changes abruptly and significantly. Most of state-of-the-art tracklet association-based multi-object tracking methods make use of image representations often not good enough for constructing robust appearance-based tracklet affinity models. Current methods usually utilize pre-selected features, such as HOG features [7] and local binary patterns [9], or color histograms, which are not “tailor-made” for the tracked objects in question. Recently, deep convolutional neural network architectures have been successfully applied to many challenging tasks, such as image classification [14] and object detection [8], and provide highly promising results. The core to its success is to take advantage of deep architectures to learn richer hierarchical features through multiple nonlinear transformations. Hence, the deep convolutional neural network is exploited for multi-object tracking in this work.

This chapter proposes to jointly learn the siamese convolutional neural network, which consists of two sub-networks joined at their outputs (see Figure 5.1), and appearance-based tracklet affinity models, which makes our method learn the appearance-based affinity models and the “tailor-made” hierarchical features for tracked targets simultaneously and coherently. Moreover, taking advantage of the characteristics of the sequential data stream, a novel temporally constrained multi-task learning mechanism is proposed to be added to the objective function. This makes the deep architectures more effective in tackling the tracklet association problem. Although deep architectures have been employed in single object tracking [64–68], to our best knowledge, this work is the first to exploit deep architectures in multi-object tracking. This work also proposes to embed the temporally constrained multi-task learning mechanism into the last layer of the proposed deep

Figure 5.1: Tracking framework of our method. In the Generalized Linear Assignment (GLA) graph, each node denotes a reliable tracklet; each edge denotes a possible link between two tracklets. This work jointly learns the siamese CNN and the temporally constrained metrics for tracklet affinity model, as shown in the red-dashed box, which is exploited to estimate the linking probability between two tracklets in the GLA graph. The tracking results are obtained by combinatorial optimization via the softassign algorithm.

neural network.

The proposed framework in this chapter is shown in Figure 5.1. Given a video input, the objects of interest in each frame are first detected by a pre-trained object detector, such as the popular DPM detector [3]. Then a simple dual-threshold strategy [25] is employed to generate reliable tracklets. The siamese CNN is first pre-trained on the auxiliary data offline. Subsequently, the siamese CNN and temporally constrained metrics are jointly learned for tracklet affinity models by using the online collected training samples from the reliable tracklets. Finally, the tracklet association problem is formulated as a Generalized Linear Assignment (GLA) problem, which is solved by the softassign algorithm [156]. The final trajectories of multiple objects are obtained after a trajectory recovery process.

The contributions of the work presented in this chapter can be summarized as follows:

- This work proposes a unified deep model for jointly learning “tailor-made” hierarchical features for tracked objects and temporally constrained segment-wise metrics for tracklet affinity models. With this deep model, the feature learning and the discriminative tracklet affinity model learning can efficiently interact with each other, maximizing their performance by cooperating with each other.
- A novel temporally constrained multi-task learning mechanism is proposed to be embedded into the last layer of the unified deep neural network, which makes the deep neural network more effective in tackling the tracklet association problem.

The remainder of this chapter is organized as follows: The unified deep model is introduced in Section 5.2. Section 5.3 describes the tracklet association framework. Experimental results are shown in Section 5.4. Section 5.5 concludes the whole chapter.

5.2 The Unified Deep Model

This section introduces how the unified deep model is designed for jointly learning tailor-made hierarchical features and temporally constrained metrics for tracklet association.

5.2.1 Architecture

The 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 the tracklet association problem, the objective is to estimate the tracklet affinities between two tracklets to decide whether

they belong to the same object. Hence, the “sample $\rightarrow$ label” mode deep neural network is not applicable to the tracklet association problem. To deal with this problem, a siamese deep neural network, which consists of two sub-networks working in a “sample pair $\rightarrow$ similarity” mode, is proposed.

The structure of the siamese convolution neural network (CNN) is shown in Figure 5.1 (red-dashed box). Given two target images, they are first warped to a fixed 96×96 pixel size and then sent to the siamese CNN. The siamese CNN is composed of two sub convolutional neural networks (CNNs), as shown in Figure 5.1 (red-dashed box). A novel loss function coupled with the temporally constrained multi-task learning mechanism is used for learning the siamese CNN. The temporally constrained metrics for tracklet association are obtained through this learning process. Moreover, the siamese CNN has a constraint that their two sub-CNNs share the same parameters, *i.e.*, weights and biases.

The sub-CNN in the unified deep model consists of 2 convolutional layers (C1 and C2), 2 max pooling layers (S1 and S2) and a fully connected layer (F3), as shown in Figure 5.2. The number of channels of convolutional and max pooling layers are both 96. The output of the sub-CNN is a feature vector of 512 dimensions. A cross-channel normalization unit is included in each pooling layer. The convolutional layer output has the same size as the input by zero padding of the input data. The filter sizes of C1 and C2 layers are 7×7 and 5×5 respectively. The activation function for each layer exploited in the CNN is ReLU neuron [14].

5.2.2 Loss Function and Temporally Constrained Metric Learning

Before learning the parameters of the unified deep model, we revert back to its structure again. As shown in Figure 5.1 (red-dashed box), the siamese CNN consists of two basic components: two sub-CNNs and a loss function. The loss function

Figure 5.2: The structure of 5-layer sub-CNN used in the unified deep model.

converts the difference between the input sample pair into a margin-based loss.

The relative distance between an input sample pair used in the loss function, which is parameterized as a Mahalanobis distance, is defined as:

$$\|x_i - x_j\|_M^2 = (x_i - x_j)^T M (x_i - x_j), \quad i \neq j \quad (5.1)$$

where x_i and x_j are two 512-dimensional feature vectors obtained from the last layer of the two sub-CNNs; and M is a positive semidefinite matrix.

The discriminative temporally constrained metrics for tracklet affinity models are also obtained through the learning process of the unified deep model. Before introducing the proposed loss function with the temporally constrained multi-task learning mechanism, the loss function with common metric learning is first presented. A common metric can be obtained for tracklet affinity model through the learning process. Given training samples, it is aimed to minimize the following loss:

$$\begin{aligned} \min_M \quad & \frac{\lambda}{2} \|M - I\|_F^2 + C \sum_{i,j} \max(0, b - l_{i,j} [1 - \|x_i - x_j\|_M^2]) \\ \text{s.t.} \quad & M \succeq 0, i \neq j \end{aligned} \quad (5.2)$$

where λ is a regularization parameter; $\|\cdot\|_F$ denotes the Frobenius norm of a matrix; C is the weight parameter of the empirical loss; b is a constant value satisfying $0 \leq b \leq 1$, which represents the decision margin; $l_{i,j}$ is a label that equals to 1 when x_i and x_j are of the same object and -1 otherwise; and $M \succeq 0$ means that M is a positive semidefinite matrix.

Nevertheless, object appearance can vary a lot in the entire video sequence. It is undesirable to use the same metric to estimate the tracklet affinities over the entire video sequence. In this chapter, the segment-wise metrics are proposed to be learned within each relative short-time segment known as a local segment instead of the entire sequence. Meanwhile, to capture the common discriminative information shared by all the segments, a multi-task learning mechanism is proposed to be embedded into the loss function for learning the segment-wise and common metrics simultaneously. Moreover, to make full use of the sequential information of a video sequence, a multi-task learning method incorporating temporal constraints is proposed. Hence, the learning problem is formulated as follows:

$$\begin{aligned}
 \min_{M_0, \dots, M_n} & \left(\frac{\lambda_0}{2} \|M_0 - I\|_F^2 + \sum_{t=2}^n \frac{\eta}{2} \|M_t - M_{t-1}\|_F^2 + \right. \\
 & \left. \sum_{t=1}^n \left[\frac{\lambda}{2} \|M_t\|_F^2 + C \sum_{i,j} h(x_i, x_j) \right] \right) \\
 \text{s.t. } & M_0, M_1, \dots, M_n \succeq 0, i \neq j
 \end{aligned} \tag{5.3}$$

where λ_0 and λ are the regularization parameters of M_t for $t = 0, 1, \dots, n$; n is the total number of segments; M_0 is the common metric shared by all the segments; M_t is the segment-wise metric; $\|\cdot\|_F$ denotes the Frobenius norm of a matrix; the second term of this loss function is the temporal constraint term, in which η is a regularization parameter; $h(x_i, x_j)$ is the empirical loss function; and C is the weight parameter of the empirical loss.

The empirical loss function $h(x_i, x_j)$ used in Equation (5.3) is expressed as:

$$\begin{aligned} h(x_i, x_j) &= \max(0, b - l_{i,j}[1 - \|x_i - x_j\|_{M_{tot}}^2]); \\ M_{tot} &= M_0 + M_t, \quad i \neq j, \\ \|x_i - x_j\|_{M_{tot}}^2 &= (x_i - x_j)^T (M_0 + M_t) (x_i - x_j) \end{aligned} \quad (5.4)$$

where b is a constant value, which represents the decision margin; $l_{i,j}$ is a label that equals to 1 when x_i and x_j are of the same object and -1 otherwise; x_i and x_j are two 512-dimensional feature vectors obtained from the last layer of the two sub-CNNs; and M_{tot} is the metric used for estimating the relative distance between a sample pair.

Intuitively, the common metric M_0 represents the shared discriminative information across the entire video sequence and the segment-wise metric $M_{t>0}$ specialize the metric for each local segment. In the proposed objective function, as presented in Equation (5.3), the second term is the temporal constraint term, which is based on the observation that the neighboring segments sharing more information than the non-neighboring segments. See Figure 5.3 for an illustration. In the implementation, the previous segment-wise metric M_{t-1} in temporal space is used to initialize the current segment-wise metric M_t , accounting for the fact that the neighboring segment-wise metrics are more correlated than the non-neighboring ones.

To learn the parameters of the unified deep model, back-propagation (BP) [157] is utilized. The forward propagation function to calculate the loss of the training pairs is presented in Equation (5.3). By differentiating the loss function with respect to the two input samples, the gradients can be obtained. The total gradient for back-propagation is the sum of the contributions from the two samples,

Figure 5.3: An illustration of the temporally constrained multi-task learning mechanism. n is the total number of the segments and the segments are shown in the temporal space.

which is shown as follows:

$$\nabla G_{total} = 2Cl_{i,j}(M_{tot} + M_{tot}^T)(x_i - x_j) (\mathbb{I}\{g(x_i, x_j) > 0\}); \quad (5.5)$$

where

$$M_{tot} = M_0 + M_t, \quad (5.6)$$

$$g(x_i, x_j) = b - l_{i,j}[1 - \|x_i - x_j\|_{M_{tot}}^2], \quad (5.7)$$

$$\|x_i - x_j\|_{M_{tot}}^2 = (x_i - x_j)^T (M_0 + M_t)(x_i - x_j) \quad (5.8)$$

where $\mathbb{I}\{\cdot\}$ is the indicator function.

Based on Equations (5.3) and (5.5), the parameters of the unified deep model can be learned by stochastic gradient descent via back-propagation. Moreover, the temporally constrained metrics for tracklet affinity models are obtained simultaneously by batch-based stochastic gradient descent. By differentiating the loss

function, as shown in Equation (5.3), with respect to the common metric M_0 and the segment-wise metric $M_{t>0}$, the gradients are:

$$\begin{aligned}\nabla G_0 &= \frac{\partial L}{\partial M_0} \\ &= \lambda_0(M_0 - I) + C \sum_{i,j} l_{i,j} A_{i,j} (\mathbb{I}\{g(x_i, x_j) > 0\});\end{aligned}\tag{5.9}$$

$$\begin{aligned}\nabla G_t &= \frac{\partial L}{\partial M_t} \\ &= \begin{cases} \lambda M_t + C \sum_{i,j} l_{i,j} A_{i,j} (\mathbb{I}\{g(x_i, x_j) > 0\}), & \text{if } t = 1; \\ \left(\eta(M_t - M_{t-1}) + \lambda M_t + \right. \\ \left. C \sum_{i,j} l_{i,j} A_{i,j} (\mathbb{I}\{g(x_i, x_j) > 0\}) \right), & \text{otherwise } (t > 1). \end{cases}\end{aligned}\tag{5.10}$$

where

$$A_{i,j} = (x_i - x_j)(x_i - x_j)^T,\tag{5.11}$$

$$g(x_i, x_j) = b - l_{i,j}[1 - \|x_i - x_j\|_{M_{tot}}^2],\tag{5.12}$$

$$\|x_i - x_j\|_{M_{tot}}^2 = (x_i - x_j)^T (M_0 + M_t) (x_i - x_j)\tag{5.13}$$

Online training sample collection is an important issue in the learning of the unified deep model. The assumptions similar to those as in [87] are utilized: (1) detection responses in one tracklet are from the same object; (2) any detection responses in two different tracklets which have overlaps over time are from different objects. The first one is based on the observation that the tracklets generated by the dual-threshold strategy are reliable; the second one is based on the fact that one target cannot appear at two or more different locations at the same time, known

as spatio-temporal conflict. For each tracklet, κ strongest detection responses are selected as training samples ($\kappa = 4$ in our implementation). Then arbitrary selected two different detection responses from the κ strongest responses of T_i are used as positive training samples, and two detection responses from the κ strongest responses of two spatio-temporal conflicted tracklets are used as negative training samples.

Finally, the common metric M_0 and the segment-wise metrics $M_{t>0}$ are obtained simultaneously through a gradient descent rule. The online learning algorithm is summarized in Algorithm 4.

$$M_0 = M_0 - \beta \frac{\partial L}{\partial M_0} \quad (5.14)$$

$$M_t = M_t - \beta \frac{\partial L}{\partial M_t} \quad (5.15)$$

where β is the learning rate.

5.3 Tracklet Association Framework

In this section, the tracklet association framework, which incorporates the temporally constrained metrics learned by the unified deep model to obtain robust appearance-based tracklet affinity models, is presented.

5.3.1 Tracklet Association with Generalized Linear Assignment

To avoid learning tracklet starting and termination probabilities, the tracklet association problem is formulated as a Generalized Linear Assignment (GLA) [158], which does not need the source and sink nodes as in conventional network flow optimization [4–6, 93]. Given N tracklets $\{T_1, \dots, T_N\}$, the Generalized Linear As-

Algorithm 4 Online Learning Algorithm for Temporally Constrained Metric Learning

Input:

Feature vectors of online collected training samples $\{x_i^t\}$; $i = 1, \dots, n_t$, n_t is the number of the samples within segment t ; $t = 1, \dots, n$, n is the total number of the segments; and learning rate β .

Output:

The learned metrics: $M_0, M_1, \dots, M_n$.

```

1: Initialize  $M_0 = I$  (identity matrix).
2: for  $t = 1, \dots, n$  do
3:   if  $t == 1$  then
4:     Initialize  $M_t = 0$ .
5:   else
6:     Initialize  $M_t = M_{t-1}$ .
7:   end if
8:   Randomly generate the training pairs  $\{x_i, x_j, l_{i,j}\}$  from  $\{x_i^t\}$ .  $l_{i,j} = 1$ , if  $x_i$ 
   and  $x_j$  are from one tracklet;  $l_{i,j} = -1$ , if  $x_i$  and  $x_j$  are from two different
   tracklets which have overlaps over time. A total of  $2m$  training pairs in
   an random order are generated, which includes  $m$  positive and  $m$  negative
   pairs.
9:   for  $p = 1, \dots, 2m$  do
10:    if  $l_{i,j}[1 - (x_i - x_j)^T(M_0 + M_t)(x_i - x_j)] > b$  then
11:       $M_0 = M_0$ ;  $M_t = M_t$ .
12:    else if  $l_{i,j} < 0$  then
13:      Compute  $M_0$  and  $M_t$  by Equations (5.14) and (5.15).
14:    else
15:       $M_0 = \pi_{S^+}(M_0 - \beta \nabla G_0)$ ;
16:       $M_t = \pi_{S^+}(M_t - \beta \nabla G_t)$ ;
      where  $\pi_{S^+}(A)$  projects matrix  $A$  into the positive semidefinite cone.
17:    end if
18:  end for
19: end for

```

signment (GLA) problem is formulated as:

$$\begin{aligned}
 & \max_X \sum_{i=1}^N \sum_{j=1}^N P(T_i, T_j) X_{ij} \\
 & s.t. \quad \sum_{i=1}^N X_{ij} \leq 1; \sum_{j=1}^N X_{ij} \leq 1; X_{ij} \in \{0, 1\}
 \end{aligned} \tag{5.16}$$

where $P(T_i, T_j)$ is the linking probability between T_i and T_j . The variable X_{ij} denotes that T_i is the predecessor of T_j in temporal domain when $X_{ij} = 1$ and that they may be merged during the optimization.

5.3.2 Tracklet Affinity Measurement

To solve the Generalized Linear Assignment (GLA) problem in Equation (5.16), it is needed to estimate the tracklet affinity score, or equivalently, the linking probability, $P(T_i, T_j)$, between two tracklets. The linking probability $P(T_i, T_j)$ is defined based on two cues: motion and appearance.

$$P(T_i, T_j) = P_m(T_i, T_j)P_a(T_i, T_j) \quad (5.17)$$

The motion-based tracklet affinity model $P_m(T_i, T_j)$ is defined as:

$$P_m(T_i, T_j) = \mathcal{N}(p_i^{tail} + v_i^F \Delta t; p_j^{head}, \Sigma) \cdot \mathcal{N}(p_j^{head} + v_j^B \Delta t; p_i^{tail}, \Sigma) \quad (5.18)$$

where p_i^{tail} is the position of the tail response in T_i ; p_j^{head} is the position of the head response in T_j ; v_i^F is the forward velocity of T_i ; v_j^B is the backward velocity of T_j ; and Δt is the time gap between the tail response of T_i and the head response of T_j .

In Equation (5.18), the forward velocity v_i^F is estimated from the head to the tail of T_i , while the backward velocity v_j^B is estimated from the tail to the head of T_j . It is assumed that the difference of the predicted position and the refined position follows a Gaussian distribution.

To estimate the appearance-based tracklet affinity scores, it is needed to construct the probe set, consisting of the strongest detection response in each tracklet.

The probe set is defined as $G = \{g_i\}$, $i = 1, \dots, N_s$, in which N_s is the number of tracklets in a local segment. Each T_i has only one selected g_i in G to represent itself.

The appearance-based tracklet affinity model $P_a(T_i, T_j)$ is defined based on the learned temporally constrained metrics:

$$\begin{aligned}
 d_{ij}^k &= (x_i^k - g_j)^T (M_0 + M_t) (x_i^k - g_j); \\
 d_{ji}^{k'} &= (x_j^{k'} - g_i)^T (M_0 + M_t) (x_j^{k'} - g_i); \\
 norm_i^k &= \sqrt{\sum_{j=1}^{N_s} d_{ij}^k}; \quad norm_j^{k'} = \sqrt{\sum_{i=1}^{N_s} d_{ji}^{k'}}; \\
 d_{ij} &= \left[\sum_k \left(\frac{d_{ij}^k}{norm_i^k} \right) \right] / m_i; \quad d_{ji} = \left[\sum_{k'} \frac{d_{ji}^{k'}}{norm_j^{k'}} \right] / m_j; \\
 P_a(T_i, T_j) &= (d_{ij} d_{ji})^{-1}
 \end{aligned} \tag{5.19}$$

where x_i^k denotes the feature vector of the k th detection response in T_i ; $x_j^{k'}$ denotes the feature vector of the k' th detection response in T_j ; $g_i, g_j \in G$; m_i and m_j are the numbers of detection responses of T_i and T_j respectively.

In the testing, the siamese CNN are used to generate the feature vectors of the sample instances from tracklets. Then these feature vectors and the temporally constrained metrics are utilized to generate the appearance-based tracklet affinity model via Equation (5.19), which is used to associate the tracklets.

Through Equation (5.17), the predecessor-successor matrix P can be obtained for the objective function (5.16). To achieve fast and accurate convergence, P is normalized by column and a threshold ω is introduced to ensure that a reliable

tracklet association pair has a high affinity score.

$$P(T_i, T_j) = \begin{cases} P_m(T_i, T_j)P_a(T_i, T_j), \\ \quad \text{if } P_m(T_i, T_j)P_a(T_i, T_j) \geq \omega \\ 0, \text{ otherwise} \end{cases} \quad (5.20)$$

The Generalized Linear Assignment problem in Equation (5.16) can be solved by the softassign algorithm [156]. Due to the missed detections, there may exist some gaps between adjacent tracklets in each trajectory after tracklet association. Therefore, the final tracking results are obtained through the trajectory interpolation process over gaps based on a linear motion model.

The complete step-by-step pseudo-code algorithm description of the proposed tracking method in this chapter is summarized in Algorithm 5.

5.4 Experiments

5.4.1 Datasets

To evaluate the multi-object tracking performance of the proposed method, experiments are conducted on four publicly available datasets: PETS 2009 [137], Town Centre [139], Parking Lot [12] and ETH Mobile scene [41]. In the PETS 2009 dataset [137], tracking sequence S2.L1, which consists of 795 frames and includes the non-linear motions, is selected for evaluation. The Town Centre dataset [139] is captured by a single elevated camera in a busy town centre street, which includes frequent target occlusions and target interactions. The Parking Lot sequence [12] consists of 1000 frames of a relatively crowded scene, which includes frequent target occlusions and parallel motions. In the ETH Mobile scene dataset [41], the SUNNY DAY and BAHNHOF sequences of busy street scenes captured

Algorithm 5 Joint Learning of Convolutional Neural Networks and Temporally
Constrained Metrics for Tracklet Association

Input:

Input video sequence;

Output:

Tracking results.

- 1: A state-of-the-art detector such as DPM detector is used to generate human detections.
 - 2: After tracklet generation step, it is the online training sample collection step. Based on the two assumptions described in section 5.2.2, page 99, the training samples are online collected.
 - 3: All the online collected samples are warped to a fixed 96×96 pixel size.
 - 4: The siamese CNN is first pre-trained on the auxiliary data offline with the loss function defined in Eq. 5.2.
 - 5: The siamese CNN and temporally constrained metrics are jointly learned by using the online collected training samples with the loss function defined in Eq. 5.3.
 - 6: The appearance-based tracklet affinities are obtained based on the learned temporally constrained metrics and the feature vectors generated by the siamese CNN through Eq. 5.19. Meanwhile, the motion-based affinities are estimated through Eq. 5.18.
 - 7: The linking probabilities, which is based on the tracklet affinities obtained from previous step, and the tracklets (graph nodes) are passed to the GLA graph. The tracking trajectories of multiple humans are obtained by combinatorial optimization via the softassign algorithm.
 - 8: After a trajectory recovery process, the final tracking results are obtained.
-

by a moving camera are used for evaluation.

5.4.2 Experimental Settings

The proposed siamese CNN is first pre-trained on the JELMOLI dataset [106] with the loss function in Equation (5.2). For the regularization parameters in the loss function (5.3), we set $\lambda_0 = 0.01$, $\lambda = 0.02$ and $\eta = 0.02$. The variance Σ in the motion-based tracklet affinity model in Equation (5.18) is fixed at $\Sigma = \text{diag}[625 \ 3600]$. A value of 0.5 to 0.6 for the threshold ω in Equation (5.20) works well for all the datasets. Moreover, we do two passes with the segments on the whole sequence. A segment of 50 to 80 frames works well for all the sequences.

The training samples are selected from the ground truth annotations of the JELMOLI dataset. 40 different pedestrians, 15 samples per pedestrian (600 samples in total) are selected from the ground truth data of the JELMOLI dataset for pre-training the siamese CNN. All the selected training samples are warped to a fixed 96×96 pixel size for the training.

5.4.3 Performance Evaluation

Evaluation metrics: The evaluation metrics as defined in Chapter 4, which consists of multiple evaluation measures, are utilized: Recall ($\uparrow$), Precision ($\uparrow$), False Alarms per Frame (FAF $\downarrow$), False Positives (FP $\downarrow$), False Negatives (FN $\downarrow$), the number of Ground Truth trajectories (GT), Mostly Tracked (MT $\uparrow$), Partially Tracked (PT), Mostly Lost (ML $\downarrow$), the number of Track Fragments (Frag $\downarrow$) and Identity Switches (IDS $\downarrow$). Here, $\uparrow$ denotes higher scores indicate better performance, and $\downarrow$ denotes lower scores indicate better performance. Among all the evaluation measures, IDS is the most direct measure for improvement in tracklet association. Although correct tracklet associations can lead to better overall performance, some other factors can also affect the other evaluation measures.

Evaluation: In the experiments, the popular offline learned detector based on deformable part models in [3] is used to generate the detections. To show the effectiveness of joint learning and temporally constrained metrics, two baselines for the proposed method are designed. For *Baseline 1*, the siamese CNN and the metrics are learned separately. The siamese CNN is first learned alone by using the loss function (5.2), in which the M is fixed as $M = I$. Then the common metric M is learned separately with the features obtained from the previous learned siamese CNN. For *Baseline 2*, the unified deep model without the temporally constrained multi-task learning mechanism is learned for tracklet affinity model. In Baseline 2, the loss function in Equation (5.2) instead of Equation (5.3) is used to learn

Table 5.1: Comparison of tracking results between state-of-the-art methods and ours on PETS 2009 dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Milan <i>et al.</i> [75]	-	-	-	-	-	23	78.3%	21.7%	0.0%	15	22
Milan <i>et al.</i> [141]	92.4%	98.4%	0.07	59	302	23	91.3%	4.4%	4.3%	6	11
Berclaz <i>et al.</i> [28]	83.8%	96.3%	0.16	126	641	23	73.9%	17.4%	8.7%	22	13
Andriyenko <i>et al.</i> [142]	89.5%	97.6%	0.11	88	417	23	78.3%	17.4%	4.3%	21	38
Andriyenko <i>et al.</i> [30]	90.0%	98.7%	0.06	47	396	23	82.6%	17.4%	0.0%	14	18
Pirsiavash <i>et al.</i> [6]	81.2%	97.2%	0.12	93	742	23	60.9%	34.8%	4.3%	62	57
Wen <i>et al.</i> [154]	94.4%	98.4%	0.08	62	222	23	95.7%	0.0%	4.3%	10	5
Possegger <i>et al.</i> [149]	-	-	-	-	-	19	100.0%	0.0%	0.0%	16	9
Chen <i>et al.</i> CSL-VOX [159]	98.28%	91.07%	-	-	-	19	-	-	-	-	6
Chen <i>et al.</i> CSL-DPT [159]	97.64%	90.45%	-	-	-	19	-	-	-	-	8
Baseline1	96.3%	97.7%	0.13	106	170	19	94.7%	5.3%	0.0%	18	18
Baseline2	96.6%	97.9%	0.12	94	157	19	94.7%	5.3%	0.0%	16	11
Ours	97.5%	98.4%	0.09	74	115	19	94.7%	5.3%	0.0%	8	4

the unified deep model. The siamese CNNs in Baseline 1 and Baseline 2 are also pre-trained on the JELMOLI dataset [106].

For PETS 2009, to make fair comparison, the groundtruth provided by Milan *et al.* [141] is used. Moreover, the complete annotation instead of the cropped annotation is selected for evaluation. The complete annotation includes more detection groundtruth and merge some trajectories of the cropped one, which make it more challenging. The numbers of groundtruth trajectories of the complete and cropped annotations are 19 and 23 respectively. The tracking results are shown in Table 5.1. From this table, it is found that our method achieves the best performance on Recall, FN, ML and IDS. For the other items of the evaluation metric, our method also achieves competitive performance compared with state-of-the-art methods.

The proposed method is further evaluated on the Town Centre dataset. The ground truth used here is provided by [139], which is the same as in the compared methods. The quantitative results are shown in Table 5.2. Compared with up-to-date methods, our method attains best performance on Recall, Precision, FAF, FP, FN, ML, Frag and IDS, while comparative in MT. Compared with one of the latest works [149], our method reduces the fragments and the identity switches by

Table 5.2: Comparison of tracking results between state-of-the-art methods and ours on Town Centre dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Leal-Taixe <i>et al.</i> [151]	-	-	-	-	-	231	58.6%	34.4%	7.0%	363	165
Zhang <i>et al.</i> [4]	-	-	-	-	-	231	53.0%	37.7	9.3%	440	243
Benfold <i>et al.</i> [139]	-	-	-	-	-	231	67.4%	26.1%	6.5%	343	222
Pellegrini <i>et al.</i> [89]	-	-	-	-	-	231	59.1%	33.9%	7.0%	499	288
Wu <i>et al.</i> [147]	-	-	-	-	-	231	64.7%	27.4%	7.9%	453	209
Yamaguchi <i>et al.</i> [152]	-	-	-	-	-	231	58.1%	35.4%	6.5%	492	302
Possegger <i>et al.</i> [149]	-	-	-	-	-	231	56.3%	36.3%	7.4%	321	157
Baseline1	71.1%	85.0%	1.99	895	2068	231	58.0%	31.2%	10.8%	360	268
Baseline2	72.2%	87.5%	1.63	735	1983	231	59.7%	30.3%	10.0%	325	251
Ours	75.2%	92.6%	0.95	428	1770	231	65.8%	27.7%	6.5%	173	146

Table 5.3: Comparison of tracking results between state-of-the-art methods and ours on ParkingLot dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Zamir <i>et al.</i> [76]	85.3%	98.2%	-	-	-	14	-	-	-	-	-
Shu <i>et al.</i> [12]	81.7%	91.3%	-	-	-	14	-	-	-	-	-
Andriyenko <i>et al.</i> [142]	69.3%	91.3%	0.65	162	756	14	21.4%	71.5%	7.1%	97	68
Andriyenko <i>et al.</i> [30]	86.8%	89.4%	1.01	253	326	14	78.6%	21.4%	0.0%	70	83
Pirsiavash <i>et al.</i> [6]	69.4%	97.8%	0.16	39	754	14	7.1%	85.8%	7.1%	60	52
Wen <i>et al.</i> [154]	90.8%	98.3%	0.16	39	227	14	78.6%	21.4%	0.0%	23	21
Baseline1	86.0%	91.6%	0.78	195	344	14	71.4%	28.6%	0.0%	95	39
Baseline2	89.5%	92.3%	0.74	185	258	14	78.6%	21.4%	0.0%	63	33
Ours	91.9%	93.5%	0.63	158	200	14	78.6%	21.4%	0.0%	51	8

about 46.1% and 7% respectively and improves the most tracked by about 16.9%.

For Parking Lot sequence, the groundtruth provided by Shu *et al.* [12], which is the same as that used in the compared methods, is utilized. As shown in Table 5.3, our method obtains the best performance on Recall, FN, MT, ML and IDS. Compared with the latest work [154], the FN and the IDS are reduced by 11.9% and 61.9% respectively for our method.

Our method is further evaluated on the more challenging ETH Mobile scene dataset, which is captured by a moving camera. The same ground-truth from [74] is used for evaluation. The comparison results are shown in Table 5.4. From this table, it is found that the best performance on Precision, FAF, FP, FN, and IDS is achieved by our method. Compared with one of the latest tracklet association

Table 5.4: Comparison of tracking results between state-of-the-art methods and ours on ETH dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Kuo <i>et al.</i> [87]	76.8%	86.6%	0.891	-	-	125	58.4%	33.6%	8.0%	23	11
Yang <i>et al.</i> [74]	79.0%	90.4%	0.637	-	-	125	68.0%	24.8%	7.2%	19	11
Milan <i>et al.</i> [75]	77.3%	87.2%	-	-	-	125	66.4%	25.4%	8.2%	69	57
Leal-Taixe <i>et al.</i> [153]	83.8%	79.7%	-	-	-	125	72.0%	23.3%	4.7%	85	71
Bae <i>et al.</i> [155]	-	-	-	-	-	126	73.8%	23.8%	2.4%	38	18
Baseline1	76.7%	90.7%	0.60	812	2406	125	56.8%	32.8%	10.4%	133	31
Baseline2	78.4%	91.8%	0.54	728	2236	125	60.8%	29.6%	9.6%	75	20
Ours	80.2%	94.5%	0.36	486	2050	125	68.8%	24.8%	6.4%	36	6

method [155], our method reduces the fragments and the identity switches by about 5.3% and 66.7% respectively. This indicates that the proposed appearance-based tracklet affinity model based on the unified deep model is superior to the one from [155].

On the whole, Baseline 2 achieves overall better performance than Baseline 1 on all datasets, which proves the effectiveness of the joint learning. Moreover, compared with Baseline 2, our method achieves significant improvements in performance on all datasets, which verifies the superiority of the proposed unified deep model with the temporally constrained multi-task learning mechanism. Furthermore, compared with state-of-the-art methods on all datasets, our method achieves the best performance on IDS, which is the most direct measure for tracklet association evaluation.

Computation speed: The proposed method was implemented using the Mat-ConvNet toolbox [37] on a server with a 2.60GHz CPU and a Tesla K20c GPU. The computation speed is subject to the number of targets in a video sequence. The speeds of the proposed method are about 0.38, 0.81, 0.50 and 0.60 (sec/frame) for PETS 2009, Town Centre, ParkingLot and ETH datasets, respectively, excluding the detection step. Note that speed-up can be achieved by further optimization of the codes.

5.4.4 Tracking Results by Using the proposed Human Detector

The tracking results of the proposed method by using the proposed semantic pedestrian part detectors introduced in Chapter 3, which can achieve better performance than the DPM detector [3], are presented in this subsection. The tracking results on the four publicly available datasets are shown in Table 5.5, 5.6, 5.7 and 5.8, respectively. Form the tracking results, it is found that the proposed method by using the semantic pedestrian part detectors introduced in Chapter 3 achieves little improvements in performance on all the four datasets. This demonstrates that the proposed tracking method in this chapter has strong robustness to the detectors.

Table 5.5: Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on PETS 2009 dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Using DPM detector [3]	97.5%	98.4%	0.09	74	115	19	94.7%	5.3%	0.0%	8	4
Using the proposed	97.6%	98.4%	0.09	72	112	19	94.7%	5.3%	0.0%	6	4

Table 5.6: Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on Town Centre dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Using DPM detector [3]	75.2%	92.6%	0.95	428	1770	231	65.8%	27.7%	6.5%	173	146
Using the proposed	75.3%	92.7%	0.94	422	1761	231	66.2%	27.3%	6.5%	165	145

Table 5.7: Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on ParkingLot dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Using DPM detector [3]	91.9%	93.5%	0.63	158	200	14	78.6%	21.4%	0.0%	51	8
Using the proposed	92.2%	93.8%	0.60	150	186	14	78.6%	21.4%	0.0%	45	8

Table 5.8: Comparison of tracking results by using the DPM detector [3] and the proposed human detector in Chapter 3 on ETH dataset.

Method	Recall	Precision	FAF	FP	FN	GT	MT	PT	ML	Frag	IDS
Using DPM detector [3]	80.2%	94.5%	0.36	486	2050	125	68.8%	24.8%	6.4%	36	6
Using the proposed	80.3%	94.6%	0.35	476	2036	125	68.8%	24.8%	6.4%	30	6

5.5 Concluding Remarks

In this chapter, a novel unified deep model for tracklet association is presented. This deep model can jointly learn the siamese CNN and temporally constrained metrics for tracklet affinity models. The experimental results of Baseline 1 and Baseline 2 verifies the effectiveness of the joint learning and the temporally constrained multi-task learning mechanism of the proposed unified deep model. Furthermore, extensive experimental results on four public datasets compared with state-of-the-art methods demonstrate the superiority of the proposed method.

Chapter 6

Conclusions and Future Research

6.1 Conclusions

This thesis studies the challenges in human detection and tracking in surveillance videos. Based on the investigation and analysis, some effective detection and tracking methods are proposed to handle the difficulties such as target interactions and occlusions.

For human detection, the performance is often hindered by difficulties such as partial or severe occlusions. Even part-based detectors such as DPM [3] may fail due to the lack of effective representations of pedestrian parts in such difficult situations. Moreover, usually no annotated training data for parts is available for pedestrian detection datasets. To tackle this problem, a transferred convolutional neural network (T-CNN) is proposed to learn semantic pedestrian part detectors. The proposed detection method can explicitly learn visual patterns of semantic pedestrian parts for occlusion handling. The experimental results show that our method achieves the state-of-the-art performance and outperforms other deep learning based methods on challenging datasets.

For human tracking, to better preserve human identities, a novel method based

on online target-specific metric learning and coherent dynamics estimation for tracklet (track fragment) association is proposed. The tracking results are obtained through a global network flow optimization in which the nodes in the network are tracklets. To further improve the tracking performance in difficult situations, a learning algorithm to learn the weights of motion and appearance tracking cues for tracklet affinity models is developed. The experimental results on several public datasets demonstrate the superiority of the proposed method.

Inspired by recent advances in deep architectures, a novel unified deep model, which can jointly learn “tailor-made” hierarchical features for tracked objects and temporally constrained segment-wise metrics for tracklet affinity models, is proposed. Moreover, for reliable association between tracklets, a novel loss function incorporating temporally constrained multi-task learning mechanism is developed. By employing the proposed method, tracklet association can be attained even in difficult situations. Extensive experimental results on four challenging public datasets verify the effectiveness and superiority of the proposed method.

The proposed tracking method in chapter 4 needs the information of entrance/exit areas. The proposed method assumes the information of entrance/exit based on the fact that the pedestrians tend to enter/exit near image borders or enter/exit at places such as doors. The proposed tracking method in chapter 5 does not need the entrance/exit information, which is a more generic tracking framework. The average computation speed of the proposed method in chapter 4 is about 0.1 second/frame. This is a relative high speed. However, the average computation speed of the method in chapter 5 is about 0.6 second/ frame, which is a relative slower speed. The speed of the proposed tracking method in chapter 5 is 6 times slower than the one proposed in chapter 4. Moreover, it needs powerful GPU for online fine-tuning the Siamese CNN. Overall, the two human tracking methods in chapter 4 and 5 are proposed for different application scenarios. For example, in a

surveillance application, if the entrance/exit information is available in the scene and the application requires fast computation speed, we can choose the tracking method presented in chapter 4 in this scenario. Otherwise, the human tracking method in chapter 5 can be selected.

Both the formulations in chapter 4 and chapter 5 consider the tracklet association for several frames. The cost-flow network formulation in chapter 4 takes the entrance/exit information into consideration. The source node s and sink node t in this formulation are used to model the entrance/exit points. However, the GLA formulation in chapter 5 does not model the starting or termination (entrance/exit) points, which is different from the cost-flow network formulation presented in chapter 4. The cost-flow network formulation can model the tracklet association problem better in the situation in which the entrance/exit information in a scene is available but it has more parameters to be determined in the cost function. However, solving the network flow is generally fast due to the way the video frames are organized. However, the GLA formulation has less parameters in the cost function to determine but it takes longer time to solve than the network flow. When the entry/exits information is available the method proposed in chapter 4 should be used. Otherwise, the method proposed in chapter 5 should be used.

For both two proposed tracking methods, the occlusion problem is not handled in the tracklet affinity models; it is handled in the global optimization process. The tracklets to be associated are reliable short trajectories across successive frames. These tracklets usually include few occluded instances. The proposed tracklet affinity models in the two tracking methods are used to associate broken short tracklets into long trajectories through global optimization. The associated tracklets may not be from consecutive frames. There may be small gaps in temporal space between two associated tracklets due to occlusions. These gaps between

adjacent tracklets are linearly interpolated in the global optimization. Hence, the occlusion problem can be solved in this way. Severe occlusions or even complete occlusions can be handled effectively by this way.

The performance of the two tracking methods has been included in the respective chapter (see Table 4.2 in chapter 4 and Table 5.1 in chapters 5). From these two tables, it can be seen that the method proposed in chapter 5 has better accuracy by some performance metrics and similar accuracy by other performance metrics than the method proposed in chapter 4. However, in terms of computation speed the method in chapter 4 is 6 times faster than the one proposed in chapter 5.

Both the proposed tracking algorithms presented in Chapter 4 and 5 are tracking-by-detection based methods. Hence, good performing detectors are needed for both tracking algorithms. These good performing detectors cannot perform detections in real time. The tracking stage is fast but it requires a stack of frames to be captured and processed in batch mode causing delayed output. For the proposed tracking algorithms presented in Chapter 5, powerful GPUs are needed but may still fall short of real time performance. Hence, the proposed algorithms in these two chapters are not suitable for real time performance. They are more suitable for human detection in archived surveillance videos in which real time performance is not critical. In addition, the proposed tracking algorithm presented in Chapter 4 needs user to determine the entry and exit area of a scene beforehand. However, this can be done during system installation and configuration.

6.2 Recommendations for Future Research

For human detection and tracking, there are still many challenges on its way to be successfully usable in real applications. In the following, the promising research

directions, which can be explored starting from this thesis work, are given.

Exploiting grouping cues. Grouping behaviors of humans in crowded scenes have been reported in [91] to improve tracking, the author proposes to further explore grouping cues to improve both the detection and tracking performance in crowded surveillance videos. The humans walking in groups usually show particular spatial visual patterns. These kind of patterns can be detected to help to estimate the presence and movements of multiple humans in a group. Hence, human detector can benefit from these group information to handle the occlusions within a group. A plausible way for human tracking is to use grouping cues to model the relationship between individual trajectories and group trajectories over a time interval. This relationship modeling is expected to benefit multi-person tracking significantly.

Integration with 3D information. The proposed detection and tracking methods do not utilize any 3D information or camera calibration information. However, 3D information is potentially helpful to improve detection and tracking performance. As is known to all, detecting and tracking humans in the presence of occlusions are very challenging tasks. Only relying on the 2D visual patterns is difficult to handle the occlusion problems well. With the help of 3D geometries of the humans, we can learn templates for the visible parts capturing both appearance and 3D information. Hence, humans occluded by other humans or objects in crowded scenes can be better differentiated using 3D information.

Real-time detection and tracking process. In order to address the requirement of real-time applications, fast detection and tracking algorithms are preferred. However, when some effective but computational expensive components are incorporated into the detection and tracking methods, more time is needed to handle more calculations in complex models. The proposed detection and tracking methods in this thesis still cannot meet the real-time requirement. New approaches

need to be developed, especially for online feature extraction and online learning which are the main computational bottlenecks in the current algorithms.

Multi-target tracking in surveillance network. Another promising direction of the future work is to utilize all-available information from surveillance video archives across network for multi-target tracking. This research can focus on non-overlapping surveillance networks, which are often the case in practical surveillance applications. The available information can include motion, appearance, grouping cues, time codes in each video sequence, the map of the whole surveillance network and so on. Developing methodology and algorithm that can track and re-identify each individual appeared in the network automatically is desired for many real surveillance applications. This tracking system in surveillance network may be implemented in a hierarchical manner, which includes two levels. For the lower level, the proposed tracking algorithms in this thesis can be employed for each individual camera of the surveillance network. For the higher level, the tracking trajectories from each camera can be further associated with the aid of the time codes, the actual network map of the cameras and trajectory information from the lower level.

Author's Publications

1. 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), 2014.
2. Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang, "Tracklet association in detect-then-track paradigm for long-term multi-person tracking", in Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2014.
3. Bing Wang, Kap Luk Chan, Gang Wang, and Haijian Zhang, "Pedestrian detection in highly crowded scenes using "online" dictionary learning for occlusion handling", in Proceedings of IEEE International Conference on Image Processing (ICIP), 2014.
4. Bing Shuai, Gang Wang, Zhen Zuo, Bing Wang, and Lifan Zhao, "Integrating parametric and non-parametric models for scene labeling", in Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
5. Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen, "Convolutional recurrent neural networks: learning spatial dependencies for image representation", in Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015.

References

- [1] “PETS 2009 benchmark data.” <http://www.cvg.rdg.ac.uk/PETS2009/a.html>.
- [2] M. Andriluka, S. Roth, and B. Schiele, “Monocular 3d pose estimation and tracking by detection,” in CVPR, 2010.
- [3] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627–1645, 2010.
- [4] L. Zhang, Y. Li, and R. Nevatia, “Global data association for multi-object tracking using network flows,” in CVPR, 2008.
- [5] A. Butt and R. T. Collins, “Multi-target tracking by lagrangian relaxation to min-cost network flow,” in CVPR, 2013.
- [6] H. Pirsiavash, D. Ramanan, and C. Fowlkes, “Globally-optimal greedy algorithms for tracking a variable number of objects,” in CVPR, 2011.
- [7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in CVPR, 2005.
- [8] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, “Object detection with grammar models,” in NIPS, 2011.

- [9] X. Wang, T. X. Han, and S. Yan, “An hog-lbp human detector with partial occlusion handling,” in CVPR, 2009.
- [10] D. Meger, C. Wojek, B. Schiele, and J. J. Little, “Explicit occlusion reasoning for 3d object detection,” in BMVC, 2011.
- [11] E. Hsiao and M. Hebert, “Occlusion reasoning for object detection under arbitrary viewpoint,” in CVPR, 2012.
- [12] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, “Part-based multiple-person tracking with partial occlusion handling,” in CVPR, 2012.
- [13] D. G. Lowe, “Object recognition from local scale-invariant features,” in ICCV, 1999.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in NIPS, 2012.
- [15] “ILSVRC: imagenet large-scale visual recognition challenge (2010-2012).”
<http://www.image-net.org/challenges/LSVRC/2011/>.
- [16] W. Ouyang and X. Wang, “A discriminative deep model for pedestrian detection with occlusion handling,” in CVPR, 2012.
- [17] W. Ouyang, X. Zeng, and X. Wang, “Modeling mutual visibility relationship with a deep model in pedestrian detection,” in CVPR, 2013.
- [18] W. Ouyang and X. Wang, “Joint deep learning for pedestrian detection,” in ICCV, 2013.
- [19] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning,” in CVPR, 2013.

- [20] P. Luo, Y. Tian, X. Wang, and X. Tang, “Switchable deep network for pedestrian detection,” in CVPR, 2014.
- [21] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in CVPR, 2008.
- [22] C. Huang and R. Nevatia, “High performance object detection by collaborative learning of joint ranking of granule features,” in CVPR, 2010.
- [23] O. Tuzel, F. Porikli, and P. Meer, “Human detection via classification on riemannian manifolds,” in CVPR, 2007.
- [24] B. Wu and R. Nevatia, “Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors,” in CVPR, 2005.
- [25] C. Huang, B. Wu, and R. Nevatia, “Robust object tracking by hierarchical association of detection responses,” in ECCV, 2008.
- [26] Y. Li, C. Huang, and R. Nevatia, “Learning to associate: Hybridboosted multi-target tracker for crowded scene,” in CVPR, 2009.
- [27] J. Xing, H. Ai, and S. Lao, “Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses,” in CVPR, 2009.
- [28] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, “Multiple object tracking using k-shortest paths optimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.
- [29] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool, “Online multi-person tracking-by-detection from a single, uncalibrated cam-

- era,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1820–1833, 2011.
- [30] A. Andriyenko, K. Schindler, and S. Roth, “Discrete-continuous optimization for multi-target tracking,” in *CVPR*, 2012.
- [31] A. V. Goldberg, “An efficient implementation of a scaling minimum-cost flow algorithm,” *Journal of Algorithms*, vol. 22, pp. 1–29, 1992.
- [32] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, “Tracking multiple people under global appearance constraints,” in *ICCV*, 2011.
- [33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *CVPR*, 2014.
- [34] C. Huang, H. Ai, B. Wu, and S. Lao, “Boosting nested cascade detector for multi-view face detection,” in *ICPR*, 2004.
- [35] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *CVPR*, 2012.
- [36] P. Viola, M. J. Jones, and D. Snow, “Detecting pedestrians using patterns of motion and appearance,” *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.
- [37] A. Vedaldi and K. Lenc, “Matconvnet – convolutional neural networks for matlab,” *arXiv preprint*, vol. arXiv:1412.4564, 2014.
- [38] P. Dollar, Z. Tu, P. Perona, and S. Belongie, “Integral channel features,” in *BMVC*, 2009.
- [39] S. Walk, N. Majer, K. Schindler, and B. Schiele, “New features and insights for pedestrian detection,” in *CVPR*, 2010.

- [40] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis, “Human detection using partial least squares analysis,” in ICCV, 2009.
- [41] A. Ess, K. S. B. Leibe, and L. van Gool, “Robust multiperson tracking from a mobile platform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831–1846, 2009.
- [42] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. Gavrilu, “Multi-cue pedestrian classification with partial occlusion handling,” in CVPR, 2010.
- [43] D. Hoiem, A. A. Efros, and M. Hebert, “Putting objects in perspective,” in CVPR, 2006.
- [44] A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg, “Part-based feature synthesis for human detection,” in ECCV, 2010.
- [45] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, “Latent hierarchical structural learning for object detection,” in CVPR, 2010.
- [46] D. Park, D. Ramanan, and C. Fowlkes, “Multiresolution models for object detection,” in ECCV, 2010.
- [47] H. Azizpour and I. Laptev, “Object detection using strongly-supervised deformable part models,” in ECCV, 2012.
- [48] W. Ouyang and X. Wang, “Single-pedestrian detection aided by multi-pedestrian detection,” in CVPR, 2013.
- [49] K. E. A. V. de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, “Segmentation as selective search for object recognition,” in ICCV, 2011.
- [50] E. Rahtu, J. Kannala, and M. Blaschko, “Learning a category independent object detection cascade,” in ICCV, 2011.

-
- [51] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan, “Contextualizing object detection and classification,” in CVPR, 2011.
 - [52] X. Wang, X. Bai, W. Liu, and L. J. Latecki, “Feature context for image classification and object detection,” in CVPR, 2011.
 - [53] Y. Ding and J. Xiao, “Contextual boost for pedestrian detection,” in CVPR, 2012.
 - [54] R. G. Cinbis and S. Sclaroff, “Contextual object detection using setbased classification,” in ECCV, 2012.
 - [55] C. Li, D. Parikh, and T. Chen, “Extracting adaptive contextual cues from unlabeled regions,” in ICCV, 2011.
 - [56] X. Wang, M. Yang, S. Zhu, and Y. Lin, “Regionlets for generic object detection,” in ICCV, 2013.
 - [57] “Caltech pedestrian detection benchmark.” http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/.
 - [58] H. Grabner, M. Grabner, and H. Bischof, “Real-time tracking via on-line boosting,” in BMVC, 2006.
 - [59] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008.
 - [60] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in CVPR, 2009.
 - [61] X. Mei and H. Ling, “Robust visual tracking using l1 minimization,” in ICCV, 2009.

- [62] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-n learning: Bootstrapping binary classifiers by structural constraints,” in CVPR, 2010.
- [63] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, “Robust visual tracking via multi-task sparse learning,” in CVPR, 2012.
- [64] N. Wang and D. Y. Yeung, “Learning a deep compact image representation for visual tracking,” in NIPS, 2013.
- [65] H. Li, Y. Li, and F. Porikli, “Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking,” in BMVC, 2014.
- [66] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, “Video tracking using learned hierarchical features,” *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1424–1435, 2015.
- [67] K. Zhang, Q. Liu, Y. Wu, and M. H. Yang, “Robust tracking via convolutional networks without learning,” *arXiv preprint*, vol. arXiv:1501.04505, 2015.
- [68] S. Hong, T. You, S. Kwak, and B. Han, “Online tracking by learning discriminative saliency map with convolutional neural network,” *arXiv preprint*, vol. arXiv:1502.06796, 2015.
- [69] T. Zhao and R. Nevatia, “Tracking multiple humans in complex situations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1208–1221, 2004.
- [70] J. Berclaz, F. Fleuret, and P. Fua, “Robust people tracking with global trajectory optimization,” in CVPR, 2006.

- [71] S. Khan and M. Shah, “A multiview approach to tracking people in crowded scenes using a planar homography constraint,” in ECCV, 2006.
- [72] X. Song, J. Cui, H. Zha, and H. Zhao, “Vision-based multiple interacting targets tracking via on-line supervised learning,” in ECCV, 2008.
- [73] B. Yang, C. Huang, and R. Nevatia, “Learning affinities and dependencies for multi-target tracking using a crf model,” in CVPR, 2011.
- [74] B. Yang and R. Nevatia, “An online learned crf model for multi-target tracking,” in CVPR, 2012.
- [75] A. Milan, K. Schindler, and S. Roth, “Detection- and trajectory-level exclusion in multiple object tracking,” in CVPR, 2013.
- [76] A. R. Zamir, A. Dehghan, and M. Shah, “Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs,” in ECCV, 2012.
- [77] A. Dehghan, S. M. Assari, and M. Shah, “Gmmcp-tracker:globally optimal generalized maximum multi clique problem for multiple object tracking,” in CVPR, 2015.
- [78] W. Brendel, M. Amer, and S. Todorovic, “Multiobject tracking as maximum weight independent set,” in CVPR, 2011.
- [79] H. Jiang, S. Fels, and J. Little, “A linear programming approach for multiple object tracking,” in CVPR, 2007.
- [80] V. Reilly, H. Idrees, and M. Shah, “Detection and tracking of large number of targets in wide area surveillance,” in ECCV, 2010.
- [81] Z. Qin and C. R. Shelton, “Improving multi-target tracking via social grouping,” in CVPR, 2012.

-
- [82] C. Dicle, O. Camps, and M. Sznaiier, “The way they move: tracking multiple targets with similar appearance,” in ICCV, 2013.
 - [83] A. Butt and R. Collins, “Multiple target tracking using frame triplets,” in ACCV, 2012.
 - [84] R. Collins, “Multitarget data association with higher-order motion models,” in CVPR, 2012.
 - [85] P. Ochs and T. Brox, “Higher order motion models and spectral clustering,” in CVPR, 2012.
 - [86] B. Yang and R. Nevatia, “Multi-target tracking by online learning of non-linear motion patterns and robust appearance models,” in CVPR, 2012.
 - [87] C. H. Kuo and R. Nevatia, “How does person identity recognition help multi-person tracking?,” in CVPR, 2011.
 - [88] B. Yang and R. Nevatia, “Multi-target tracking by online learning a crf model of appearance and motion patterns,” *International Journal of Computer Vision*, vol. 107, no. 2, pp. 203–217, 2014.
 - [89] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool, “You’ll never walk alone: modeling social behavior for multi-target tracking,” in ICCV, 2009.
 - [90] S. Pellegrini, A. Ess, and L. V. Gool, “Improving data association by joint modeling of pedestrian trajectories and groupings,” in ECCV, 2010.
 - [91] Z. Qin and C. R. Shelton, “Improving multi-target tracking via social grouping,” in CVPR, 2012.
 - [92] X. Chen, Z. Qin, L. An, and B. Bhanu, “An online learned elementary grouping model for multi-target tracking,” in CVPR, 2014.

-
- [93] B. Wang, G. Wang, K. L. Chan, and L. Wang, “Tracklet association with online target-specific metric learning,” in CVPR, 2014.
 - [94] I. Tosić and P. Frossard, “Dictionary learning for stereo image representation,” *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 921–934, 2011.
 - [95] P. Sprechmann, I. Ramirez, and G. Sapiro, “Classification and clustering via dictionary learning with structured incoherence and shared features,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
 - [96] S. Kong and D. Wang, “A dictionary learning approach for classification: separating the particularity and the commonality,” in *European Conference on Computer Vision (ECCV)*, 2012.
 - [97] G. Wang, D. Hoiem, and D. Forsyth, “Learning image similarity from flickr groups using fast kernel machines,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2177–2188, 2012.
 - [98] G. Wang, D. Forsyth, and D. Hoiem, “Improved object categorization and detection using comparative object similarity,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2442–2453, 2013.
 - [99] N. Wang, J. Wang, and D.-Y. Yeung, “Online robust non-negative dictionary learning for visual tracking,” in *IEEE International Conference on Computer Vision (ICCV)*, 2013.
 - [100] J. Lu, Y. P. Tan, and G. Wang, “Discriminative multi-manifold analysis for face recognition from a single training sample per person,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 39–51, 2013.

- [101] J. Lu, Y. P. Tan, and G. Wang, “Human identity and gender recognition from gait sequences with arbitrary walking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 51–61, 2014.
- [102] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
- [103] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, “Max-margin multiple-instance dictionary learning,” in *International Conference on Machine Learning (ICML)*, 2013.
- [104] S. Lazebnik, C. Schmid, Ponce, and J. Beyond, “bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [105] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: an evaluation of the state of the art,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.
- [106] A. Ess, B. Leibe, and L. van Gool, “Depth and appearance for mobile scene analysis,” in *ICCV*, 2007.
- [107] “Oxford town center dataset.” http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bбенfold_headpose/project.html\#datasets.
- [108] X. Zeng, W. Ouyang, and X. Wang, “Multi-stage contextual deep learning for pedestrian detection,” in *ICCV*, 2013.
- [109] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *ECCV*, 2010.

- [110] Y. Aytar and A. Zisserman, “Tabula rasa: model transfer for object category detection,” in ICCV, 2011.
- [111] R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: an unsupervised approach,” in ICCV, 2011.
- [112] J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li, “Robust multi-resolution pedestrian detection in traffic scenes,” in CVPR, 2013.
- [113] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” in ICLR, 2014.
- [114] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: delving deep into convolutional nets,” in BMVC, 2014.
- [115] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in ECCV, 2014.
- [116] C. L. Zitnick and P. Dollar, “Edge boxes: Locating object proposals from edges,” in ECCV, 2014.
- [117] P. Viola and M. Jones, “Robust real-time face detection,” IJCV, 2004.
- [118] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, “Pedestrian detection with spatially pooled features and structured ensemble learning,” arXiv preprint, vol. arXiv:1409.5209, 2014.
- [119] R. Benenson, M. Omran, J. Hosang, and B. Schiele, “Ten years of pedestrian detection, what have we learned?,” in ECCV-CVRSUAD, 2014.
- [120] W. Nam, P. Dollar, and J. H. Han, “Local decorrelation for improved pedestrian detection,” in NIPS, 2014.

- [121] S. Zhang, C. Bauckhage, and A. B. Cremers, “Informed haar-like features improve pedestrian detection,” in CVPR, 2014.
- [122] M. Mathias, R. Benenson, R. Timofte, and L. V. Gool, “Handling occlusions with franken-classifiers,” in ICCV, 2013.
- [123] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, “Strengthening the effectiveness of pedestrian detection,” in ECCV, 2014.
- [124] K. Okuma, A. Taleghani, N. D. Freitas, J. Little, and D. Lowe, “A boosted particle filter: multitarget detection and tracking,” in ECCV, 2004.
- [125] Y. Cai, N. de Freitas, and J. J. Little, “Robust visual tracking for multiple targets,” in ECCV, 2006.
- [126] P. Storms and F. Spieksma, “An lp-based algorithm for the data association problem in multitarget tracking,” *Computers and Operations Research*, vol. 30, no. 7, pp. 1067–1085, 2003.
- [127] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, “Multi-camera people tracking with a probabilistic occupancy map,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 267–282, 2008.
- [128] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, “Multi-commodity network flow for tracking multiple people,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.
- [129] B. Yang and R. Nevatia, “Online learned discriminative part-based appearance models for multi-human tracking,” in ECCV, 2012.
- [130] W. Ge and R. T. Collins, “Multi-target data association by tracklets with unsupervised parameter estimation,” in BMVC, 2008.

- [131] V. K. Singh, B. Wu, and R. Nevatia, “Pedestrian tracking by associating tracklets using detection residuals,” in WMVC, 2008.
- [132] W. S. Zheng, S. Gong, and T. Xiang, “Re-identification by relative distance comparison,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 653–668, 2013.
- [133] O. Camps, H. Li, M. C. Mazzaro, and M. Sznaier, “A caratheodory-fejer approach to robust multiframe tracking,” in ICCV, 2003.
- [134] T. Ding, M. Sznaier, and O. Camps, “Fast track matching and event detection,” in CVPR, 2008.
- [135] B. Ho and R. Kalman, “Effective construction of linear, statevariable models from input/output functions,” *Regelungstechnik*, vol. 14.
- [136] M. Moonen, B. D. Moor, L. Vandenberghe, and J. Vandewalle, “On- and off-line identification of linear state space models,” *International Journal of Control*, vol. 49.
- [137] J. Ferryman and A. Shahrokni, “Pets2009: Dataset and challenge,” in Winter-PETS, 2009.
- [138] M. Andriluka, S. Roth, and B. Schiele, “People-tracking-by-detection and people-detection-by-tracking,” in CVPR, 2008.
- [139] B. Benfold and I. Reid, “Stable multi-target tracking in real-time surveillance video,” in CVPR, 2011.
- [140] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The clear mot metrics,” *Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.

- [141] A. Milan, S. Roth, and K. Schindler, “Continuous energy minimization for multi-target tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 58–72, 2014.
- [142] A. Andriyenko and K. Schindler, “Multi-target tracking by continuous energy minimization,” in *CVPR*, 2011.
- [143] J. F. Henriques, R. Caseiro, and J. Batista, “Globally optimal solution to multi-object tracking with merged measurements,” in *ICCV*, 2011.
- [144] M. Hofmann, M. Haag, and G. Rigoll, “Unified hierarchical multi-object tracking using global data association,” in *PETS*, 2013.
- [145] M. Hofmann, D. Wolf, and G. Rigoll, “Hypergraphs for joint multi-view reconstruction and multi-object tracking,” in *CVPR*, 2013.
- [146] H. Izadinia, I. Saleemi, W. Li, and M. Shah, “ $(MP)^2T$: Multiple people multiple parts tracker,” in *ECCV*, 2012.
- [147] Z. Wu, J. Zhang, and M. Betke, “Online motion agreement tracking,” in *BMVC*, 2013.
- [148] J. Yang, P. A. Vela, Z. Shi, and J. Teizer, “Probabilistic multiple people tracking through complex situations,” in *PETS*, 2009.
- [149] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof, “Occlusion geodesics for online multi-object tracking,” in *CVPR*, 2014.
- [150] S. Chen, A. Fern, and S. Todorovic, “Multi-object tracking via constrained sequential labeling,” in *CVPR*, 2014.
- [151] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn, “Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker,” in *ICCV Workshops*, 2011.

-
- [152] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, “Who are you with and where are you going?,” in CVPR, 2011.
 - [153] L. Leal-Taixe, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, “Learning an image-based motion context for multiple people tracking,” in CVPR, 2014.
 - [154] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li, “Multiple target tracking based on undirected hierarchical relation hypergraph,” in CVPR, 2014.
 - [155] S. H. Bae and K. J. Yoon, “Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning,” in CVPR, 2014.
 - [156] S. Gold and A. Rangarajan, “Softmax to softassign: Neural network algorithms for combinatorial optimization,” *Journal of Artificial Neural Nets*, vol. 2, no. 4, pp. 381–399, 1995.
 - [157] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. of the IEEE*, 1998.
 - [158] D. Shmoys and E. Tardos, “An approximation algorithm for the generalized assignment problem,” *Mathematical Programming*, vol. 62, no. 1, pp. 461–474, 1993.
 - [159] S. Chen, A. Fern, and S. Todorovic, “Multi-object tracking via constrained sequential labeling,” in CVPR, 2014.