2018

Human Action Detection, Tracking and Segmentation in Videos

Yicong Tian
University of Central Florida

HUMAN ACTION DETECTION, TRACKING AND SEGMENTATION IN VIDEOS

by

YICONG TIAN
B.S. Beijing University of Posts and Telecommunications, 2011

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2018

Major Professor: Mubarak Shah

ABSTRACT

This dissertation addresses the problem of human action detection, human tracking and segmentation in videos. They are fundamental tasks in computer vision and are extremely challenging to solve in realistic videos. We first propose a novel approach for action detection by exploring the generalization of deformable part models from 2D images to 3D spatiotemporal volumes. By focusing on the most distinctive parts of each action, our models adapt to intra-class variation and show robustness to clutter. This approach deals with detecting action performed by a single person. When there are multiple humans in the scene, humans need to be segmented and tracked from frame to frame before action recognition can be performed. Next, we propose a novel approach for multiple object tracking (MOT) by formulating detection and data association in one framework. Our method allows us to overcome the confinements of data association based MOT approaches, where the performance is dependent on the object detection results provided at input level. We show that automatically detecting and tracking targets in a single framework can help resolve the ambiguities due to frequent occlusion and heavy articulation of targets. In this tracker, targets are represented by bounding boxes, which is a coarse representation. However, pixel-wise object segmentation provides fine level information, which is desirable for later tasks. Finally, we propose a tracker that simultaneously solves three main problems: detection, data association and segmentation. This is especially important because the output of each of those three problems are highly correlated and the solution of one can greatly help improve the others. The proposed approach achieves more accurate segmentation results and also helps better resolve typical difficulties in multiple target tracking, such as occlusion, ID-switch and track drifting.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Computer vision is the scientific discipline that deals with processing and analysis of images or videos to gain high-level understanding from them. With the rapidly increasing amount of images and videos taken every day, numerous applications of computer vision have emerged. In this dissertation, we focus on analyzing humans in videos, because humans are the most important subjects in most videos people are interested in.

Human action detection, tracking and segmentation in videos are fundamental tasks in computer vision and are extremely challenging to solve in realistic videos. Human action detection aims to localize and recognize action in videos. It has a variety of applications such as video surveillance, sports video analysis, video retrieval and human machine interaction. Human action detection is to match the observation, i.e. video, with previously defined patterns and assign it a label, i.e. action type. The challenges include intra-class variation, camera motion and cluttered scenes. When there are multiple humans in the scene, usually humans need to be segmented and tracked from frame to frame before action detection and recognition can be performed. The objective of multiple object tracking is to locate multiple objects, maintain their identities and yield their individual trajectories in videos. Multiple object segmentation provides pixel-level per-object masks as well as background mask. Occlusion, target interactions and articulations pose difficulty in multiple human tracking and segmentation.

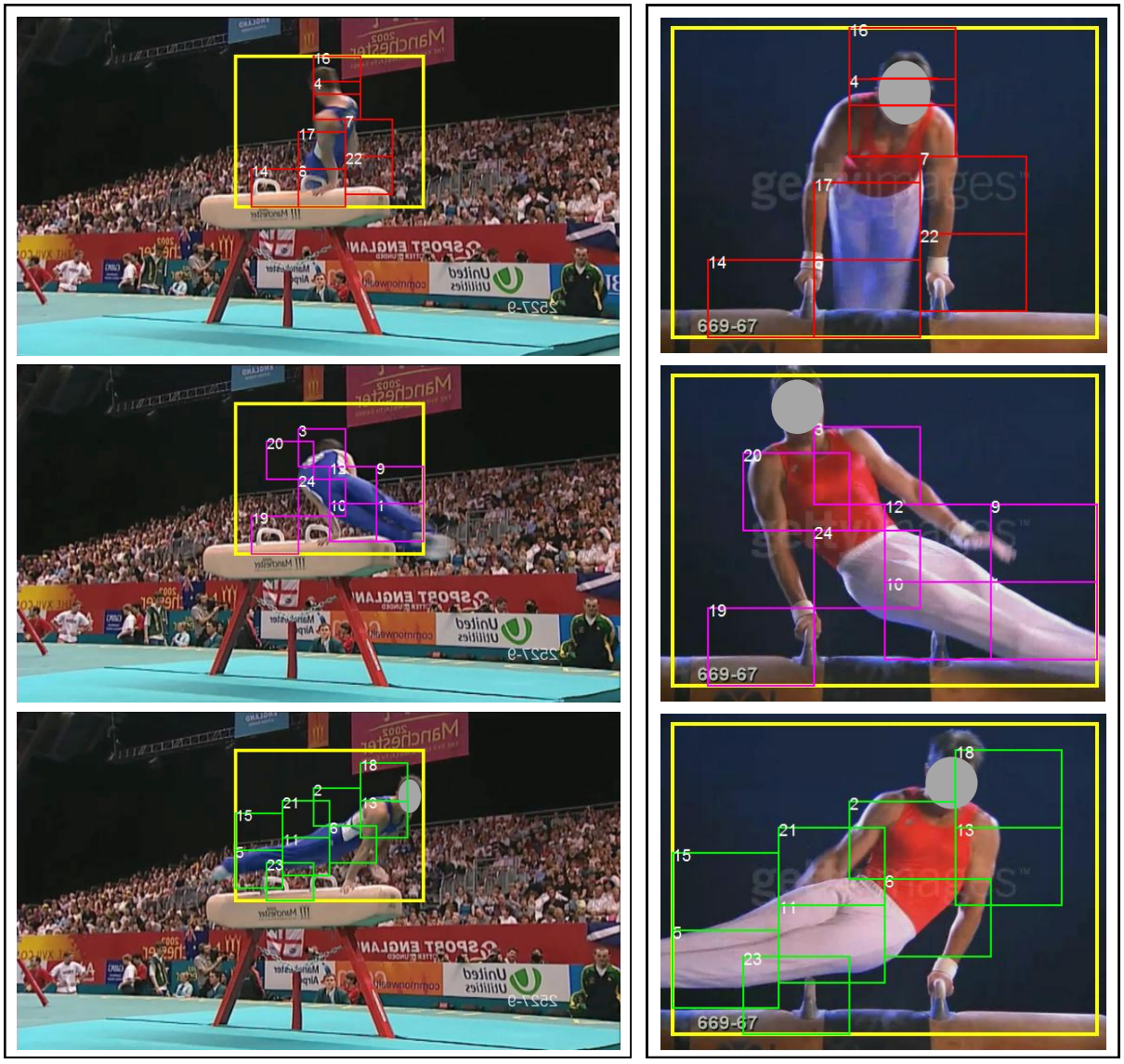
In this dissertation, we make the following contributions to deal with the above challenges in action detection, human tracking and segmentation in videos:

- We propose spatiotemporal deformable part models (SDPM) for action detection. In the model, the most discriminative 3D subvolumes are automatically selected as parts and the

spatiotemporal relations between their locations are learned. By focusing on the most distinctive parts of each action, our models adapt to intra-class variation and show robustness to clutter. Extensive experiments on several video datasets demonstrate the strength of SDPM for classifying and localizing actions.

- We propose a novel approach for multiple target tracking by formulating detection and data association in one framework. The proposed tracker does not rely on a pre-trained object detector to get the initial object hypotheses. Structured learning is used to learn a model for each target and infer the best location of all targets simultaneously. The inference of structured learning is done through a new Target Identity-aware Network Flow (TINF). We show that automatically detecting and tracking targets in a single framework helps resolve the ambiguities due to frequent occlusion and heavy articulation of targets. Experiments on challenging yet distinct datasets show that our method achieves results better than the state-of-art.

- We propose a tracker that simultaneously solves three main problems: detection, data association and segmentation. The output of each of those three problems are highly correlated and the solution of one can greatly help improve the others. TINF tracker and spatiotemporal segmentation are combined through dual decomposition. This leads to more accurate segmentation results and also helps better resolve typical difficulties in multiple target tracking, such as occlusion, ID-switch and track drifting. In addition, the final output of the proposed tracker is the fine contours of the targets rather than traditional bounding boxes. The experiments on diverse and challenging sequences show that our method achieves superior results compared to competitive approaches.



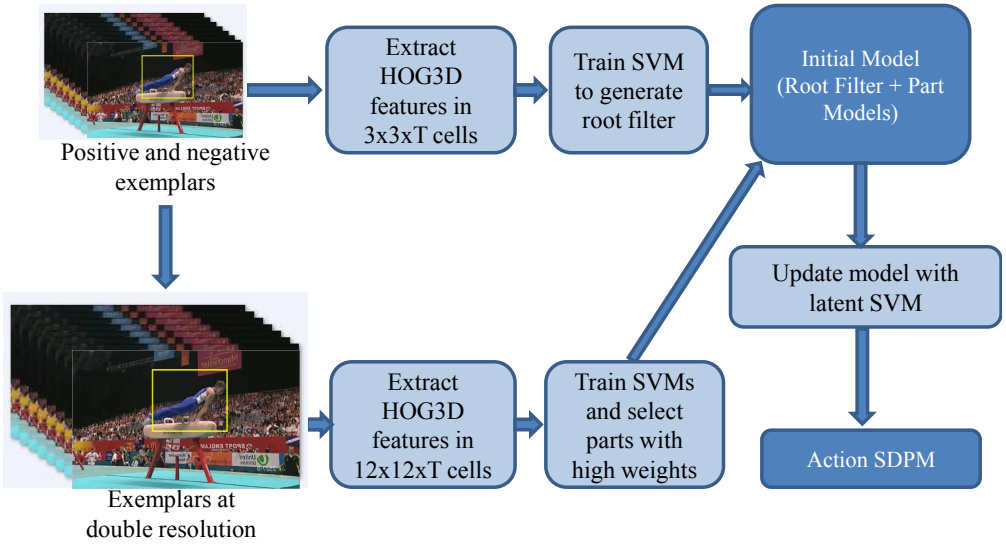
(a) training (b) testing

Figure 1.1: An example of "Swing Bench" SDPM (left) and its localization results (right) in a test video from UCF Sports. This model consists of parts across three temporal stages (middle frame of each stage shown in each row). The large yellow rectangle indicates the area under the root filter and the small red, magenta, and green ones denote parts. Although trained in videos with cluttered background at a different scale, the SDPM successfully localizes the target action in both space and time.

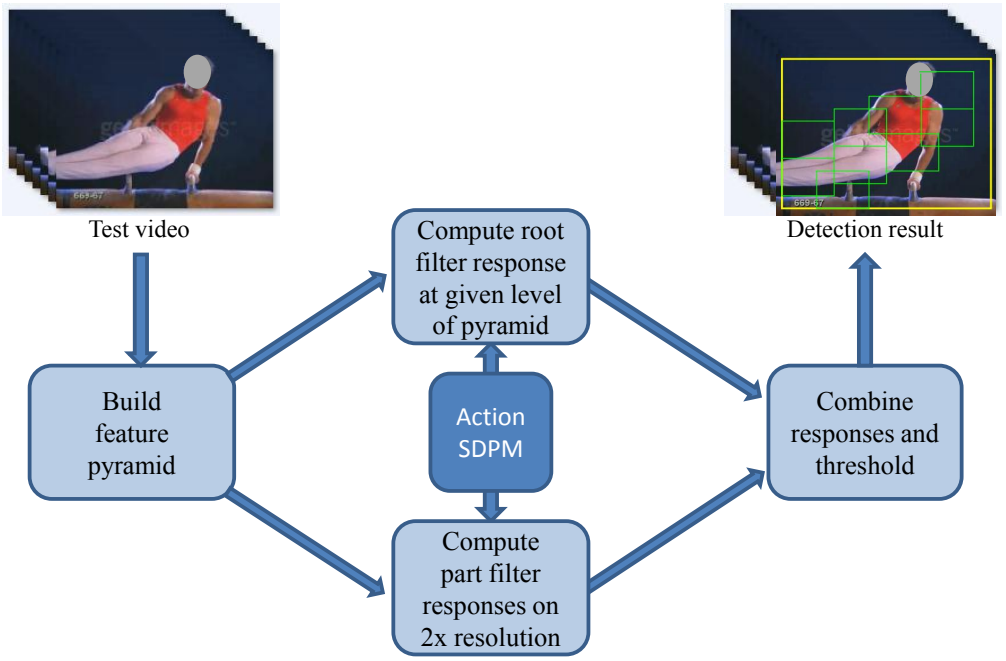
1.1 Action Detection

Action recognition in videos continues to attract significant attention from the computer vision community, with the bulk of the research focusing primarily on whole-clip video classification, where approaches derived from bag-of-words dominate [66, 46, 48, 79]. This work focuses on the related problem of action detection [39, 68], sometimes termed action localization [45] or event detection [42, 43], where the goal is to detect every occurrence of a given action within a long video, and to localize each detection both in space and time. As observed in the literature [14, 42, 92], the action detection problem can be viewed as a spatiotemporal generalization of 2D object detection in images; thus, it is fruitful to study how successful approaches pertaining to the latter could be extended to the former. Analogous to the manner in which Ke *et al*. [42] investigate spatiotemporal extensions of Viola-Jones [78], we study how the current state-of-the-art method for object detection in images, the deformable part model (DPM) [33] should best be generalized to spatiotemporal representations (see Figure. 1.1).

Deformable part models for object detection in images were proposed by Felzenszwalb *et al*. [33]. Niebles *et al*.explored a temporal (but not spatiotemporal) extension for videos [54]. Two straightforward spatiotemporal generalizations of the DPM approach to action detection in videos would be to: 1) treat action detection as a set of image-level detection problems addressed using DPMs, and 2) detect actions as spatiotemporal volumetric patterns that can be captured by a global template and set of 2D parts, each represented using the standard histograms of oriented gradients (HOG) features [25]. Unfortunately, the first is not sufficiently expressive to distinguish between similar actions and the second is unable to capture the intra-class spatiotemporal variation of many actions [45]. Clearly, a more sophisticated approach is warranted and in this work, we propose a spatiotemporal deformable part model (SDPM) that stays true to the structure of the original DPM (see Figure. 1.2), while generalizing the parts to capture spatiotemporal structure.



(a) Training: like DPM, we automatically select discriminative parts.



(b) Testing: the action is detected with root denoted by yellow rectangle and parts indicated by green rectangles.

Figure 1.2: The SDPM framework retains the overall structure of DPM but the volumetric parts are organized in temporal stages.

In SDPM, both the global (yellow rectangle) and the part (smaller green rectangles) templates employ the volumetric HOG3D descriptor [44]. Our automatically selected parts are organized into multiple temporal stages (seen in Figure. 1.1) that enable SDPM to capture how the appearance of parts changes through time during an action. A key difference between SDPM and earlier approaches is that our proposed model employs volumetric parts that displace in both time and space; this has important implications for actions that exhibit significant intra-class variation in terms of execution and also improves performance in clutter.

The primary aim of this work is to comprehensively evaluate spatiotemporal extensions of the deformable part model to understand how well the DPM approach for object detection generalizes to action detection in videos. For this reason, we restrict ourselves to HOG-like features and resist the temptation of augmenting our method with features such as person detection, dense flow, or trajectories [79, 17] or enhancements like the mixture model. Although SDPM achieves state-of-the-art performance on both controlled and real-world datasets, we stress that it was not engineered for that goal. We believe that a hybrid action detection system that incorporates our ideas could achieve further gains.

1.2 Online Discriminative Learning based Multiple Target Tracking

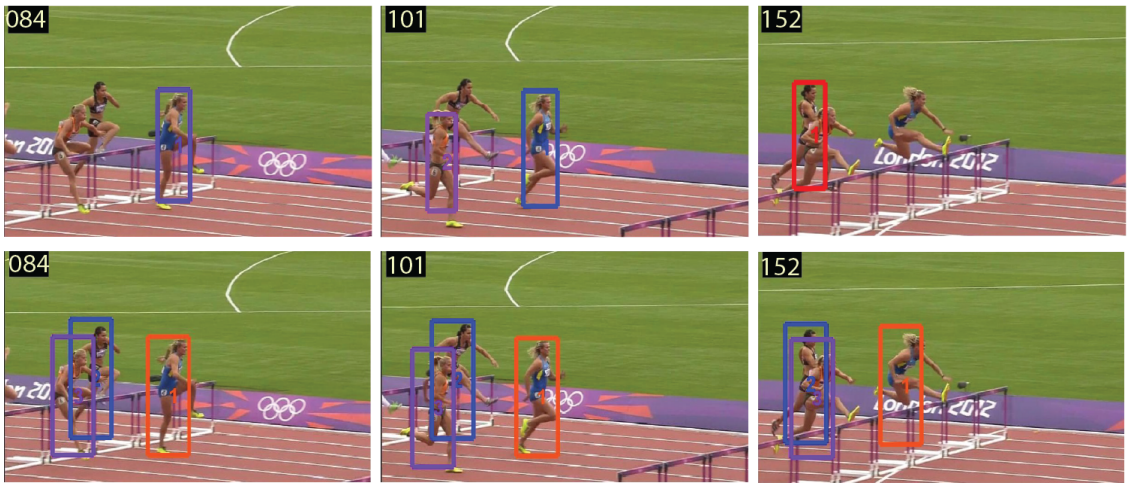
The above approach deals with detecting action performed by a single person. When there are multiple humans in the scene, humans need to be tracked from frame to frame first and then action detection and recognition can be performed. Multiple Object Tracking (MOT) is a fundamental problem in computer vision with numerous applications, ranging from surveillance, behavior analysis, to sport video analysis.

Most of the recent approaches that aim to solve the multiple object tracking problem follow two

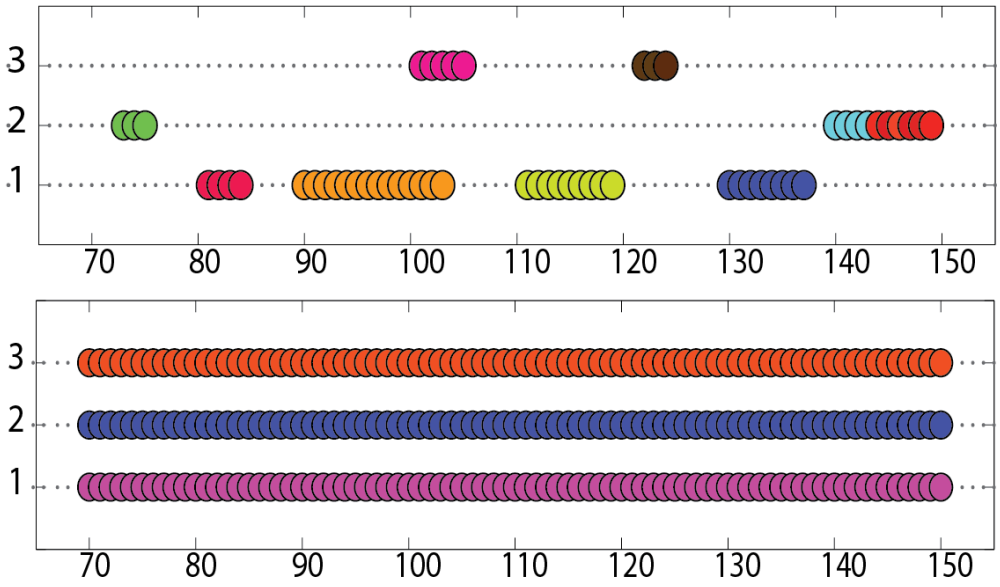
main steps: Object Detection and Data Association. In the detection phase, a pre-trained object detector is first applied to find some potential object locations in every frame of a video. Once the object candidates are found, in the data association phase the object candidates are pruned and tracks between frames are formed. In most previous work, these two steps have been considered as two separate problems and the focus of tracking is mostly on designing powerful data association techniques.

There are two main classes of data association: local association and global association. Local association base methods solve data association problem on a few frames every time. Whilst this class of methods are computationally inexpensive, their assumption of using few frames makes them prone to ID-switches and other difficulties in tracking such as long/short term occlusions, pose changes and camera motion. On the contrary, global association based approaches solve data association problem on a large number of frames and are able to take context information from more frames into consideration when forming tracks. In this way, issues caused by occlusions, pose changes, etc. are mitigated. Recent approaches have formulated the global data association as a network flow problem where a set of tracks are found efficiently by solving min-cost flow. Different solutions to min-cost flow for multiple object tracking have been proposed and demonstrated to achieve competitive tracking results.

Although data association based tracking methods have shown to be promising, still there is a major downside to such approaches. The performance is highly reliant on the performance of pre-trained object detector. If the object detector fires a lot of false alarms, or misses many true detections, the data association fails consequently. In particular, in case of articulated objects the object detector often fails when object goes under heavy articulation, because the object detector is never trained in such scenarios. This causes failure in tracking. An example is shown in Figure. 1.3.



(a)



(b)

Figure 1.3: Failure case of data association based trackers. (a) shows the tracking results of our method (bottom row) and the method proposed in [5] (top row). A pre-trained object detector fails when objects go under heavy articulation. This error is propagated to the data association step, which consequently cause failure in tracking. Differently, our method is based on online discriminative learning and solves detection and global data association simultaneously, thus handles articulated targets well. The same observation can be made from (b). Each row represents one of the three identities in the scene. Each circle shows a corresponding match in a frame and the color represents the ID that is assigned to that detection.

To overcome the above issue, recent approaches have focused on improving the performance of the generic object detector or designing better data association techniques. An alternative direction is to use online discriminative learning to learn target specific models for a given sequence. Candidates fed into data association step are obtained from target specific models, instead of generic pre-trained object detector. Target specific models are able to use video specific features, discriminate different targets, and adapt themselves as the targets appearance change. Online discriminative learning based methods have been used extensively for tracking deformable objects in the context of single object tracking. However, its extension to multiple objects remains relatively unexplored.

In this thesis we propose a tracking method based on online discriminative learning, which solves detection and global data association simultaneously by integrating a new global data association technique into the inference of a structured learning tracker. Our learning step is inspired by STRUCK [36], which is the state-of-art based on recent studies [73, 90]. We extend STRUCK to track multiple objects simultaneously. Despite other online trackers which are temporally local, our method provides the tracks across a segment of a video. The input to our tracker in every frame, is densely sampled candidate windows instead of sparse detections. This allows our tracker to infer temporal consistency between the frames and correct poor detections (mostly caused by occlusion or severe pose change), thus avoiding error propagation. We propose to do the inference through a new target identity-aware network flow graph which is a variant of multi-commodity flow graph [40].

The network used in our work is different from those in previous works [70, 83]. First, our network includes the target identities by considering more than one node per candidate location, where each node encodes the probability of assigning one of the target identities to that candidate location. Moreover, the network consists of multiple source and sink nodes, where each pair accounts for entry and exit of one of the targets. Second, the exact solution to the proposed network flow problem opens the door to using powerful structured learning algorithm and we show how the

proposed network can be used in an inner loop of structured learning which has not been explored before. Our structured learning framework allows training target specific model which eliminates the need for noisy pre-trained detectors. Third, we show that a high-quality solution to the network can be found through Lagrange relaxation of some of the hard constraints which is more efficient compared to Integer Programming (IP) or Linear Programming (LP) solutions. After relaxing the constraints, at each iteration, the problem reduces to finding the best track for each target individually, where the optimal solution can be found in linear time through dynamic programming. Thus we do not need to prune the graph as in [70, 83].

Additionally, the proposed iterative solution allows us to easily incorporate a soft spatial constraint that penalizes the score of candidate windows from different tracks that highly overlap during optimization. This helps reducing the ambiguity caused by nearby targets with similar appearance in the crowded scenes. Moreover, our spatial constraint replaces the greedy non-maximum suppression step used in most of the object detectors. Our approach, by bringing detection and data association in a single framework, not only enables us to track arbitrary multiple objects (for which there does not exist a good pre-trained detector) but also helps in better dealing with common challenges in multiple object tracking such as pose changes, miss detections and false alarms mostly caused by using a pre-trained object detector. We not only achieve results better than the state of art on sequence which pre-trained detectors perform well, but also we improve state-of-art by a significant margin on sequences for which generic detectors fail.

1.3 Detection, Data Association and Segmentation for Multiple Target Tracking

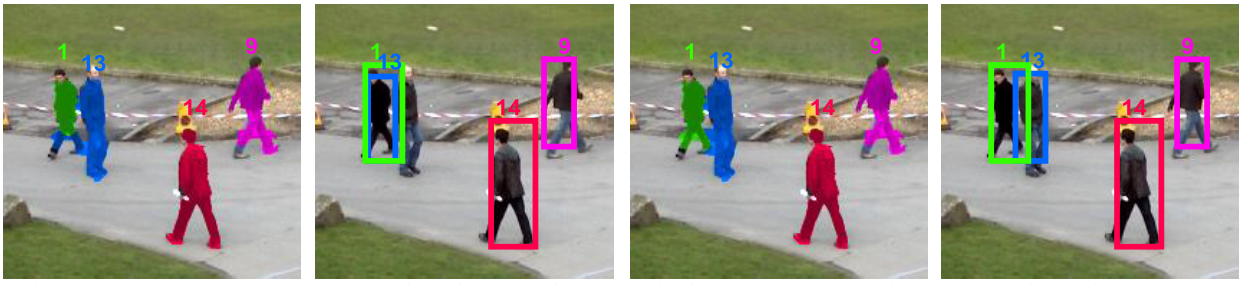
Similar to most existing tracking methods, in the tracker described above targets are represented by bounding boxes, which is a coarse representation. However, the ultimate way of detecting a target is to provide pixel-wise segmentation, so that fine contour of a target can be achieved and

tracked from frame to frame. The pixel-wise object segmentation provides fine details of targets, which is also desirable for later tasks.

Formulating tracking where each target pixel is assigned a label, requires solving three major problems: detection, data association and segmentation. Each of these problems has their own line of research, which have been active for decades in computer vision community. Most existing tracking methods limit themselves to bounding box level target representation and mainly focus on improving either the detection or the data association component of the tracker. Though convenient, bounding boxes are coarse approximations of targets. Moreover, since bounding boxes usually include non-target pixels, the features extracted from them could be contaminated by background pixels. When these features are used as target representation in tracking, they may cause drift, ID-switches and inaccurate target localization. Therefore, the ultimate goal of tracking should be to determine the pixel-wise localization of targets instead of just coarse bounding boxes.

The focus of most previous multi-target tracking algorithm is to improve the data association component of the tracker. A majority of these algorithms assume the existence of pre-trained object detector. Some of these methods heavily rely on the results of the pre-trained detector [8, 57], while some have more tolerance [27, 20, 86] toward miss-detections and false detections that commonly happen when using a pre-trained detector. One solution to address this issue is to design trackers that internally train a detector for each target, eliminating the need for a pre-trained detector. However, there is only a handful of trackers that focus on solving both detection and tracking in the context of multi-target tracking [23, 35, 30, 31]. Due to lack of a good pre-trained object detector in several scenarios, e.g those in which objects undergo heavy articulation and occlusion, and also due to heavy correlation between performance of a data association method and object detector, solving detection and data association simultaneously is very natural. An example to further motivate this approach is shown in Figure 1.3, where we show a scenario that poor detection propagates into data association and results in poor tracking performance.

(a)



(b)

Figure 1.4: Two examples of the tracking and segmentation tasks benefiting from each other (zoomed in views are shown). (a) By applying pure segmentation, the upper body of target 9 is mislabelled as target 15 due to similar color. But the tracking part is able to track target 9 correctly. After dual decomposition, the whole body of target 9 is labelled correctly and more accurate box is obtained for target 9. (b) Without incorporating segmentation, the track for target 13 drifts to target 1. However, the segmentation results for target 13 are correct using pure segmentation. After dual decomposition, target 13 is tracked successfully and the segmentation results for target 1 are also improved. Combining the two subproblems lead to both better tracking and better segmentation results.

Another problem that is highly correlated with tracking is video segmentation. Looking back at the literature, these two problems are almost always considered as separate problems. However, we argue that tracking and segmentation are actually closely related and solving them should help each other (See Figure 1.4). On one hand, the object track, which is a set of bounding boxes with one bounding box in every frame, would provide strong high-level guidance for the target/background segmentation task. Pixels within a target box are highly likely to be labelled as the target. Con-

versely, the chance that pixels far away from the box belonging to the target is quite low. On the other hand, the object segmentation would separate object from other objects and background, which will be useful for determining track locations in every frame. This will help in resolving common issues in tracking. For example, during occlusion, the bounding box based appearance score of the occluded target is typically low, posing difficulty in tracking. However, the pixel labels in the visible part of the target would guide tracker to find the correct location of the target. In addition, labels of pixels in target/background contain information about target identities and locations, thus will help in avoiding track drifting and ID-switches.

In this dissertation, we propose to combine detection, data association and segmentation in one framework. The key idea to couple these three tasks is the high correlation between them. As discussed above, poor detection results in poor data association, therefore pixel level segmentation can help further improve tracking. At the heart of our tracker lies a Lagrange dual decomposition that combines an online discriminative tracker with segmentation. Our tracker is a new online discriminate learning tracker that solves detection and data association simultaneously. This online tracker is later combined with a segmentation method through Lagrange dual decomposition. In each iteration, the two subproblems of online tracking and segmentation are solved independently with the Lagrange variables serving as a connection between them.

For the tracking subproblem in dual decomposition, we propose an algorithm based on online discriminative learning, which solves detection and global data association simultaneously by integrating a new global data association technique into the inference of a structured learning tracker. Despite other online trackers which are temporally local, our tracker provides the tracks across a segment of a video. The input to our tracker, in every frame, is densely sampled candidate windows instead of sparse detections. This allows our tracker to impose temporal consistency between the frames and correct poor detections (mostly caused by occlusion or severe pose change), thus avoiding error propagation. We propose to perform the inference through a Target Identity-aware

Network Flow graph (TINF), which is a variant of multi-commodity flow graph [40].

For the segmentation subproblem, a foreground Gaussian Mixture Model (GMM) is constructed for each target along with one universal background GMM. These GMMs are used to compute target/background confidence maps. For a segment of video (a few frames), a superpixel based spatio-temporal graph is built and multi-label CRF is applied to the graph to obtain final target/background labeling.

The tracking and segmentation subproblems are coupled through dual decomposition. We introduce a new coupling energy term, which penalizes background labels inside target bounding boxes as well as foreground labels outside target bounding boxes. Iterative optimization is applied to solve the problem. In each iteration, Lagrange variables are updated based on the inconsistency between tracking and segmentation results. The algorithm converges when tracking and segmentation results are consistent.

In Chapter 4, we introduce the online discriminative learning component of Lagrange dual decomposition. In Chapter 5, we further extends it by combining segmentation and online discriminative learning tracking through dual decomposition. To summarize, this work makes following important contributions. First, we propose a novel framework which combines multiple target tracking and segmentation in one energy function. The two tasks benefit from each other, thus leading to both better tracking and better segmentation results (See Figure 1.4). The unified energy function is optimized effectively using dual decomposition. Second, to solve the tracking subproblem, we present a new multiple object tracking method which combines discriminative learning and global data association. We introduce a new Target Identity-aware Network Flow graph (TINF) and efficiently optimize it through Lagrangian relaxation. Our soft-spatial constraint replaces the ad-hoc non-maximum suppression step of object detection methods and further improves the results. Finally, the proposed approach is able to track multiple targets in terms of finer segments (regions)

supported by corresponding target pixels rather than coarse bounding boxes, and achieve better results for both tracking and segmentation than state-of-art on challenging sequences.

1.4 Dissertation Organization

The rest of the dissertation is structured as follows: In Chapter 2, we review existing literature on human action detection, human tracking as well as segmentation in videos. In Chapter 3, we present our proposed approach for action detection using spatiotemporal deformable part models. In Chapter 4, we describe a new approach for multiple target tracking by formulating detection and data association in one framework. Chapter 5 presents our proposed approach that simultaneously solves three main problems: detection, data association and segmentation. We conclude and discuss future work in Chapter 6.

CHAPTER 2: LITERATURE REVIEW

Human action detection, human tracking and segmentation in video are three fundamental tasks in computer vision. In this chapter, we review a number of works in literature related to these tasks. First, We present early works on whole clip action video classification, template matching based approaches for action detection and existing exploration on action parts. We also describe the advantage of our spatiotemporal deformable part based action model. Second, we present traditional approaches on multiple target tracking, which consists of two main steps: object detection using a pre-trained object detector and data association to form tracks based on object detection results. We discuss the disadvantage of traditional tracking-by-detection framework as well as the motivation of our online discriminative learning based tracker, which formulates detection and data association in one framework. Last, we review works on object segmentation in video and dual decomposition, with which we propose a novel tracker that simultaneously solves detection, data association and segmentation.

2.1 Action Detection

Bag-of-words representations [66, 46, 48, 79] have demonstrated excellent results in action recognition. However, such approaches typically ignore the spatiotemporal distribution of visual words, preventing localization of actions within a video. With bag-of-words representations, Neibles *et al*. [55] and Wong *et al*. [87] apply pLSA to capture the spatiotemporal relationship of visual words. Although some examples of action localization are shown, the localization is performed in simple or controlled settings and no quantitative results on action detection are presented.

Earlier work proposes several strategies for template matching approaches to action localization.

Rodriguez *et al*. [61] generalize the traditional MACH filter to video and vector-valued data, and detect actions by analyzing the response of such filters. Kläser *et al*. [54] localize human actions by a track-aligned HOG3D action representation, which (unlike our method) requires human detection and tracking. Ke *et al*. [43] introduce the notion of parts and efficiently match the volumetric representation of an event against oversegmented spatiotemporal video volumes; however, these parts are manually specified using prior knowledge and exhibit limited robustness to intra-class variation.

There has been recent interest in learning parts directly from data. Lan *et al*.. [45] detect 2D parts frame-by-frame followed by a CRF with tracking constraints. Brendel and Todorovic [17] construct spatiotemporal graphs over tubes to represent the structure of primitive actions. Raptis *et al*. [58] embed parts obtained by grouping trajectories into graphical model. However, SDPM differs from these in the following four respects. First, SDPM includes an explicit model to capture intra-class variation as a deformable configuration of parts. By contrast, the model in [17] is not flexible enough to handle speed variation within an action. Second, both the global template and set of part templates in SDPM are spatiotemporal volumes, and we search for the best fit across scale, space and time. As a 3D subvolume, each part jointly considers appearance and motion information spanning several frames, which is better suited for actions than 2D parts in a single frame [45] that primarily capture pose. Third, we employ a dense scanning approach that matches parts to a large state space, avoiding the potential errors caused by hard decisions on video segmentation, which are then used for matching parts [58]. Finally, we focus explicitly on demonstrating the effectiveness of action detection within a DPM framework, without resorting to global bag-of-words information [45, 58], trajectories [58] or expensive video segmentation [43, 17].

2.2 Multiple Target Tracking

Most approaches for multiple target tracking (MOT) follow tracking-by-detection framework. First, a pre-trained object detector is applied to find a set of candidate locations for targets. Then these candidates are fed into a data association mechanism to form tracks. A majority of previous work on MOT focuses on designing data association techniques. There are two main classes of data association.

Local Association. These methods are temporally local, which means they consider only a few frames while solving the association problem. The best example of such approaches is bi-partite matching and its extensions [59, 69, 71, 8]. In [8], the association probabilities are computed jointly across all targets to deal with ambiguities in association. Shu et. al in [71] use a greedy approach to combine the responses of part detectors to form a joint likelihood model of multiple cues to associate detections and object hypotheses. Whilst this class of methods are computationally inexpensive, their assumption of using few frames makes them prone to ID-switches and other difficulties in tracking such as long/short term occlusions, pose changes and camera motion.

Global Association. To better deal with above problems, another set of data association techniques have recently received a lot of attentions. In global association methods, the number of frames is increased and sometimes the entire video is processed at once to determine the tracks [98, 27, 89]. Recent approaches have formulated the data association as a network flow problem, where a set of tracks are found efficiently by solving min-cost flow.

Different solutions to minimum cost flow for MOT have been proposed recently. In [100] a global optimal solution is found using push-relabel algorithm. Pirsiavash *et al.* in [57] utilize the same graph as [100], and solve the problem using a fast greedy shortest path procedure based on dynamic programming. Berclaz *et al.* in [9] introduce an efficient shortest path algorithm to solve the flow

problem. In [19], a new network flow is proposed to incorporate constant velocity motion model in the graph and the solution is found efficiently using Lagrange relaxation. Shitrit *et al.* in [70] include image appearance cues by solving multiple networks in parallel; each network representing one appearance group.

Though these methods show competitive tracking results, their performance heavily depend on object detector outputs, which are usually poor when dealing with occlusion and articulated objects. Recent approaches have focused on improving the performance of the generic object detector [71] or designing a better data association techniques [57, 7] to improve tracking. Shu *et al.* in [71] proposed an extension to deformable part-based human detector [33], which can handle occlusion up to a scale. An alternative method to overcome the drawbacks of object detector when dealing with articulated objects or arbitrary objects (when a good pre-trained detector does not exist) is online learning of the object classifier [3, 36, 82]. Online discriminative learning approaches allow training target specific classifiers for a given sequence using different features including video specific features like color histogram. Moreover, these classifiers can adapt themselves as the appearance of targets change, which is not the case in pre-trained object detector.

Online discriminative learning methods have been used extensively for tracking deformable objects in the context of *single object tracking*. However, its extension to multiple objects remains relatively unexplored and is limited to only few works. The work of Zhang and Maaten [102] is probably the first attempt to apply online discriminative learning in tracking multiple objects. In [102], the spatial constraint among the targets is modeled during tracking. It is shown that the tracker performs well when the structure among the objects remains the same (or changes very slowly). However, this is only applicable to very limited scenarios and it will perform poorly in others, specially when the targets are moving independently.

Multi-commodity network flows have been used recently for multi-target tracking [70, 83, 94,

96]. We show that multi-commodity network flows can be used in an inner loop of structured learning. Additionally the network design in our work is different from [70, 83], where our network includes the target identities by considering more than one node per candidate location and each node encodes the probability of assigning one of the target identities to that candidate location. Moreover, the network consists of multiple source and sink nodes, where each pair accounts for entry and exit of one of the targets. Also, we show that a high-quality solution to the network can be found through Lagrange relaxation of some of the hard constraints, which is more efficient compared to Integer Programming (IP) or Linear Programming (LP) solutions. Thus we do not need to prune the graph as in [70, 83].

2.3 Object Segmentation in Video

Video object segmentation [18, 49, 99] aims to segment foreground pixels belonging to the object from the background in every frame. Video Object segmentation has been used in combination with single object tracking in [93, 47, 85]. However, the videos which are typically used in this work contain only one or two main moving objects. Different from these approaches, we solve video object segmentation along with multiple target tracking. The goal is to segment multiple interacting targets and preserve targets' identities at the same time. Authors in [13, 38, 52] track contours of targets using a level-set framework. Milan *et al.* [51] propose a CRF model to jointly optimize over tracking and segmentation. First, a large number of trajectory hypotheses are generated by two trackers ([57] and [37]) using human detection results. Then the objective becomes assigning detections and superpixels to trajectory hypotheses. In contrast, we propose an energy function coupling the tracking and segmentation subproblems, which is solved using dual decomposition by taking advantage of synergies between them. In addition, we do not rely on human detection or other trackers.

2.4 Dual Decomposition

Dual decomposition is a general and powerful technique widely used in optimization. It solves a problem by decomposing the original problem into multiple subproblems, solving the subproblems separately and then merging the solutions to solve the overall problem. Using dual decomposition, Strandmark and Kahl [74] solve the max-flow/min-cut problem in parallel by splitting a large graph into multiple subgraphs. Thus, the algorithm runs much faster when multiple CPU cores are available and is also able to handle graph that is too large to fit in computer's RAM. Wu *et al.* [91] propose to incorporate both object detection and data association in a single objective function to avoid error propagation. The objective function is optimized by dual decomposition. Wang and Koller [80] construct a unified model over human poses as well as pixel-wise foreground/background segmentation and optimize the energy function using dual decomposition. To the best of our knowledge, we are the first ones to utilize dual decomposition to solve the multiple target tracking and segmentation problems.

2.5 Summary

In this chapter, we first reviewed related works in the area of action recognition and detection, including bag-of-words representations, template matching approaches and early exploration of action parts. Next, we presented commonly used tracking-by-detection framework for multiple target tracking and the motivation for our proposed online discriminative learning based tracker. Finally we reviewed works on object segmentation in video and dual decomposition, and proposed a novel tracker that simultaneously solves detection, data association and segmentation. In the next three chapters, we present our proposed methods on action detection, multiple target tracking and segmentation.

CHAPTER 3: SPATIOTEMPORAL DEFORMABLE PART MODELS FOR ACTION DETECTION

Action recognition is a challenging topic and has attracted lots of attentions in computer vision community. A significant amount of research have been done to classify a video clip into one of the action categories. However, action *classification* is always not enough since we need more information about when and where the action actually happens in many applications. In this chapter, we address the problem of action *detection*, by answering not only what action happens in a video, but also when and where it happens.

Actions can be treated as spatiotemporal volume patterns. Inspired by the popular deformable part model for object detection in image, we explore the generalization of deformable part models from 2D images to 3D spatiotemporal volumnes. Naive generalization of deformable part models from 2D to 3D will not work well. We discuss in this chapter a few design decisions driven by the inherent asymmetry between space and time. Our spatiotemporal model consists of a global template and several part templates. Both the global template and part templates are 3D volumes, capturing appearance feature as well as motion feature. Given training samples for an action, the part templates are automatically selected, which are the most discriminative 3D subvolumes. The part templates are organized into multiple temporal stages such that our model is able to capture how the appearance of parts changes through time during an action. In addition, the spatiotemporal relations between part locations are learned. To detect actions, action parts are allowed for certain displacements with costs in both time and space. The combination of root and part filters ensures good detection performance. By focusing on the most distinctive parts of each action, our models adapt to intra-class variation and show robustness to clutter. Extensive experiments on standard video datasets demonstrate the strength of SDPM for classifying and localizing actions.

3.1 Generalizing DPM from 2D to 3D

Generalizing deformable part models from 2D images to 3D spatiotemporal volumes involves some subtleties that stem from the inherent asymmetry between space and time that is often ignored by volumetric approaches. Briefly: 1) Perspective effects, which cause large variation in observed object/action size do not affect the temporal dimension; similarly, viewpoint changes affect the spatial configuration of parts while leaving their temporal orderings unchanged. 2) The units of space (pixels) and time (frames) in a video are different and should not be treated interchangeably. Additionally, we make several observations below that are specific to deformable part models.

First, consider the difference between a bounding box circumscribing an object in a 2D image and the corresponding cuboid enclosing an action in a video. In the former, unless the object is unusually shaped or wiry, the majority of pixels contained in the bounding box correspond to the object. By contrast, for actions — particularly those that involve whole-body translation, such as walking, or large limb articulations such as kicking or waving — the bounding volume is primarily composed of background pixels. This is because enclosing the set of pixels swept during even a single cycle of the action requires a large spatiotemporal box (see Figure. 3.1). The immediate consequence of this phenomenon, as confirmed in our experiments, is that a detector without parts (solely using the root filter on the enclosing volume) is no longer competitive. Finding discriminative parts is thus more important for action detection than learning the analogous parts for DPMs for 2D objects.

To quantify the severity of this effect, we analyze the masks in the Weizmann dataset and see that for nine out of ten actions, the percentage of pixels occupied by the actor in a box bounding a *single cyle* of the action is between 18% to 30%; the highest is 'pjump' with 35.7%. These are all dramatically smaller than 80%, which is the fraction of the bounding box image occupied by object parts in DPM [33]. This observation drives our decision during training to select a set of parts such

that in total they occupy 50% of the action cycle volume.[1] Naively using the same settings as DPM would force SDPM to form parts from background or unreliable regions, impairing its overall accuracy.

Second, in the construction of spatiotemporal feature pyramids that enable efficient search across scale, we treat space and time differently. This is because, unlike its size, the duration of an action does not change with its distance from the camera. The variation in action duration is principally caused by differences between actors, is relatively small and better handled by shifting parts. Thus, our feature pyramids employ multiple levels in space but not in time.

Finally, the 2D HOG features in the original DPM must be replaced with their volumetric counterparts. To maximize reproducibility, rather than proposing our own generalization of HOG, we employ Kläser *et al*.'s HOG3D [44].

3.2 Deformable Part Models

Inspired by the 2D models in [33], we propose a spatiotemporal model with deformable parts for action detection. The model we employ consists of a root filter F_0 and several part models. Each part model is defined by a part filter F_i, an anchor position (x_i,y_i,t_i) and coefficients of deformation cost $d_i = [d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5}, d_{i6}]$. Here $i \in (1, N)$, where N is the number of parts.

3.2.1 HOG3D feature descriptor

Kläser *et al*.propose the HOG3D [44] descriptor based on a histogram of oriented spatiotemporal gradients as a volumetric generalization of the popular HOG [25] descriptor. The effectiveness of

[1] Since SDPM parts are themselves rigid cuboids that contain background pixels, the total volume they occupy in the bounding volume should be higher than the fraction of pixels that correspond solely to the actor.

HOG3D as a feature is evidenced in [81]. We briefly summarize the HOG3D descriptor that we use to build fixed-length representations of each volume, along with our minor modifications.

We divide each video volume into a fixed number of non-overlapping cuboid cells. First, gradients are computed along x, y and t directions at every pixel. For each pixel, gradient orientation is quantized to a 20-dimensional vector by projecting the (dx, dy, dt) vector on to a regular icosahedron with the gradient magnitude as its weight. Then for each cell, a 3D Gaussian filter (σ is determined by the size of cell) placed at the centre of the cell is used to smooth the weighted gradients. These gradients are then accumulated into histograms with 20 bins (corresponding to the 3D gradient directions defined by the icosahedron) and normalized using L2 norm within each cell. The final descriptor is obtained by concatenating the histograms of all cells, which is different with the interest point based HOG3D descriptor in [44]. Thus, the dimension of the computed descriptor is determined by the number of cells, but is independent of the dimensions of the input volume.

This spatiotemporal feature jointly encodes both appearance and motion information, but is invariant to changes in illumination and robust to small deformations. During training, we extract HOG3D features over an action cycle volume to train root filter and part filters. During detection, HOG3D features of the whole test video volume are used to form feature maps and construct a feature pyramid to enable efficient search through scale and spatiotemporal location.

3.2.2 Root filter

We follow the overall DPM training paradigm, as influenced by the discussion in Section 3.1: During training, for positive instances, from each video we select a single box enclosing one cycle of the given action. Volumes of other actions are treated as negative examples. These negatives are supplemented with random volumes drawn at different scales from videos that do not contain the

given action to help better discriminate the given action from background.

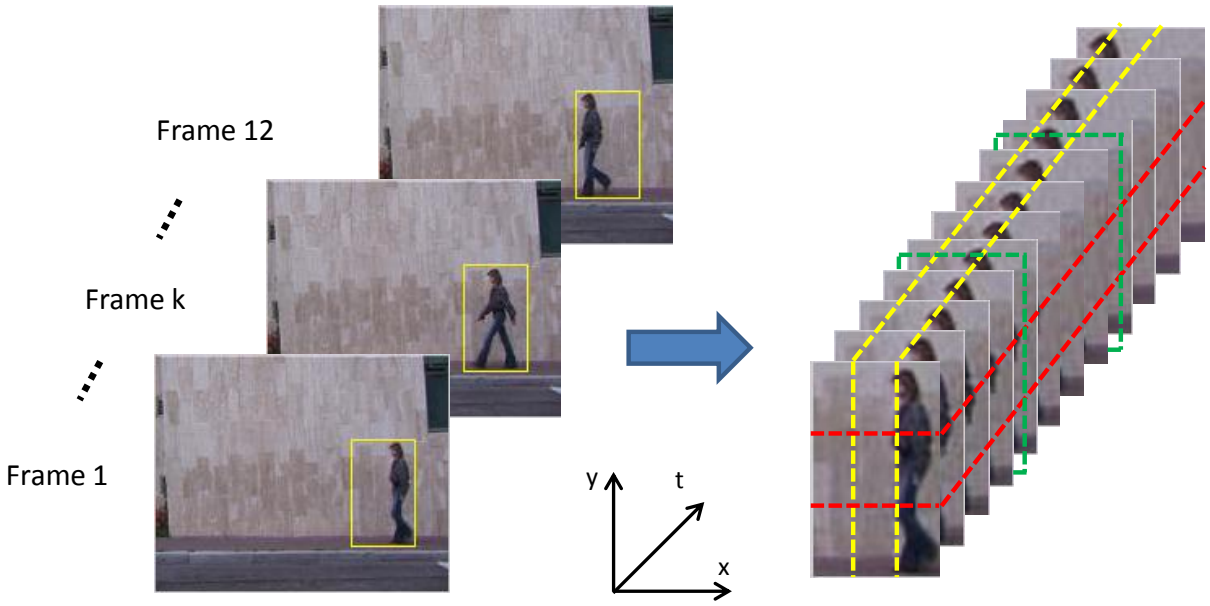


Figure 3.1: Example of computing HOG3D features for root filter. Left: 12 consecutive frames consisting one cycle of walking (annotations in yellow). Right: spatial area corresponding to the bounding volume, which (for this action type) is divided into 3 cells in x (yellow), 3 cells in y (red), 3 cells in t (green) to compute the HOG3D features for the root filter. The resulting feature descriptor is a $3\times3\times3\times20$ vector. (Part filters not shown here.)

The root filter captures the overall information of the action cycle and is obtained by applying an SVM on the HOG3D features of the action cycle volume. How to divide the action volume is important for good performance. Too few cells will decrease the distinctiveness of the feature in each cell. On the other hand, dividing the volume into too many cells, means that each cell cannot capture enough appearance or motion information since it contains too few pixels or frames. In our experiments, to train the root filter, we have experimentally determined that dividing the spatial extent of an action cycle volume into 3×3 works well. However, the temporal division is critical since cycles for different actions may vary from only 6 frames (short actions) to more than 30 frames (long actions). This is an instance of the asymmetry between space and time discussed in Section 3.1 since the observed spatial extent of an action varies greatly with camera pose but is

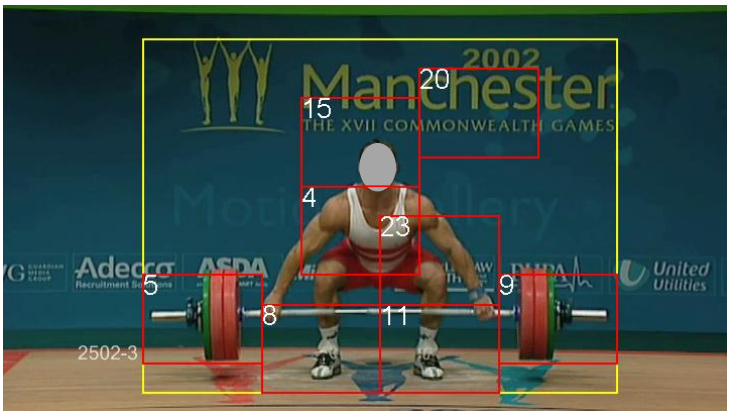
similar across actions, while temporal durations are invariant to camera pose but very dependent on the type of action.[2] Dividing all of them into the same number of temporal stages would, of course, be too brittle. Thus, the number of stages T is determined automatically for each action type according to its distribution of durations computed over its positive examples, such that each stage of the model contains 5–10 frames. In summary, we adopt a $3\times3\times T$ scheme and the resulting root filter F_0 is a vector with $3\times3\times T\times20$ weights. Figure. 3.1 shows an example root filter with $3\times3\times3$ cells.

3.2.3 Deformable parts

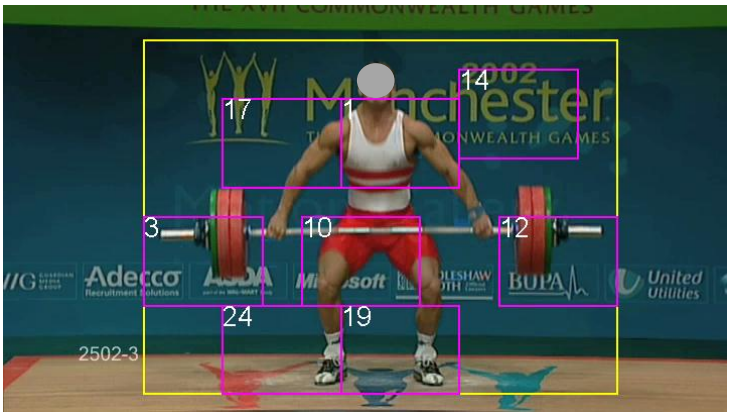
As discussed in Section 3.1 and seen in Figure. 3.1(a), only a small fraction of the pixels in a bounding action volume correspond to the actor. The majority of pixels correspond to background and can detrimentally impact detection accuracy, particularly in dynamic environments with cluttered backgrounds. As confirmed by our experiments, these issues are more serious in volumetric action detection than in images, so the role of automatically learned deformable parts in SDPM to address them is consequently crucial.

The same training examples, including random negatives, and the same number of temporal stages T is employed for training part models. Our experiments confirm that extracting HOG3D features for part models at twice the resolution and with more cells in space (but not time) enables the learned parts to capture important details; this is consistent with Felzenszwalb *et al.*'s observation [33] for DPMs in images. Analogous to the parts in DPM, we allow the parts selected by SDPM to overlap in space.

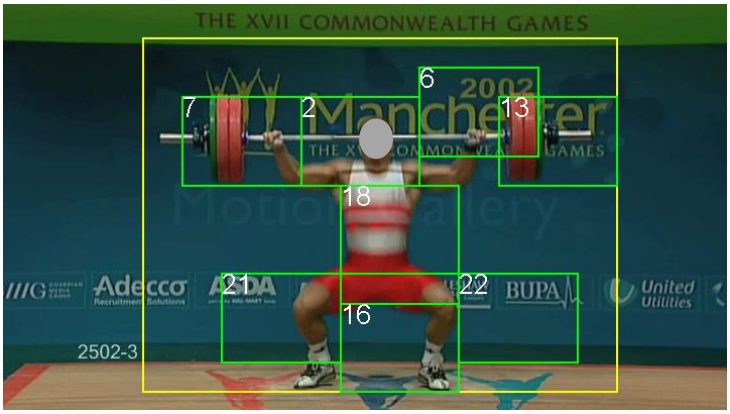
[2] As observed by [65], the correlation between action type and duration can cause researchers to overestimate the accuracy of action recognition when testing on temporally segmented video, since features inadvertently encode duration. This supports our decision to detect actions in raw video.



(a)



(b)



(c)

Figure 3.2: SDPM for "lifting" in UCF Sports, with parts learned in each of the temporal stages. There are in total 24 parts for this SDPM and the index of each part is indicated at the left top corner of corresponding small rectangle. See Figure. 1.1 for example in clutter.

After applying SVM to the extracted features, subvolumes with higher weights, which means they are more discriminative for the given action type, are selected as parts, while those with lower weights are ignored. In our setting, the action volume is divided into $12\times12\times T$ cells to extract HOG3D features and each part is a subvolume occupying $3\times3\times1$ cells. Then, we greedily select the N parts with the highest energy such that their union fills 50% of the action cycle volume. Here we define energy as the sum of positive weights in all cells of a subvolume. The weights in a subvolume are cleared after that subvolume has been selected as a part, and this process continues until all N parts are determined.

In our model, each part represents a spatiotemporal volume. It captures both appearance and motion information spanning several frames. Weights for each part filter are initialized by weights from corresponding cells forming this part. So each part filter is a vector with $3\times3\times1\times20$ weights. In addition, an anchor position (x_i,y_i,t_i) for the ith part is determined, where x_i, y_i and t_i are indices of the cell in the middle of the ith part. Anchor positions define spatiotemporal configuration of parts. For example, $x_i < x_j$ means that the ith part occurs to the left of the jth part, and $t_i < t_j$ means that the ith part occurs before the jth part in time.

Additionally, to address the high degree of intra-class variability in each action type, we allow each part of the model to shift within a certain spatiotemporal region. The cost for the ith part's deformation is a quadratic function of the distance between the placement (x_i', y_i', t_i') and the anchor position (x_i, y_i, t_i): $\varepsilon(i, X_i) = d_i \cdot X_i^T$, where $X_i = [|x_i' - x_i|, |y_i' - y_i|, |t_i' - t_i|, |x_i' - x_i|^2, |y_i' - y_i|^2, |t_i' - t_i|^2]$ records the displacement of the ith part. d_i is the learned coefficient of deformation cost for the ith part, and is initialized to $[0, 0, 0, 0.1, 0.1, 0.1]$.

Figure. 3.2 illustrates an example model for "lifting" trained on UCF Sports (on clean background for clarity). An action cycle is divided into three temporal stages, with each stage containing several frames. In this case, HOG3D features for root filter are computed by dividing the action

cycle volume into 3×3×3 cells. (a), (b) and (c) show middle frames of the first, second and third stage in time, respectively. The large yellow rectangle indicates the region covered by the root filter; the small red, magenta, and green ones are the selected parts in each temporal stage. Each part's index is shown at the top left corner of its corresponding rectangle. A low index denotes that the part was selected early and is therefore more discriminative. We observe that our learned parts cover the essential portions of the action, both in terms of appearance and motion, and that SDPM eliminates the majority of the background. Crucially, these results hold in complex scenes (*e.g.*, Figure. 1.1) because the background clutter is not consistently discriminative for the given action.

3.2.4 Model update using latent SVM

After obtaining our initial model, we train it using latent SVM with hard negative mining, as in a standard DPM. The exact position of the ith part (x'_i, y'_i, t'_i) is treated as latent information. Thus, filters and deformation cost coefficients d_i are updated to better capture action characteristics.

3.3 Action detection with SDPM

Given a test video volume, we build a spatiotemporal feature pyramid by computing HOG3D features at different scales, enabling SDPM to efficiently evaluate models in scale, space and time. As discussed in Section 3.1, the pyramid has multiple scales in space but only one in time. We denote the HOG3D features at level l of the pyramid as $\phi(l)$.

We employ a sliding window approach for template matching during detection (where the sliding window is actually a sliding subvolume). The aspect ratio of the template is determined by the mean of aspect ratios of positive training examples. Score maps for root and part filters are computed at every level of the feature pyramid using template matching. For level l, the score map

$S(l)$ of each filter can be obtained by correlation of filter F with features of the test video volume $\phi(l)$,

$$S(l,i,j,k) = \sum_{m,n,p} F(i,j,k)\ \phi(i+m, j+n, k+p, l). \tag{3.1}$$

At level l in the feature pyramid, the score of a detection volume centered at (x,y,t) is the sum of the score of the root filter on this volume and the scores from each part filter on the best possible subvolume:

$$\begin{aligned}&\text{score}(x,y,t,l) = F_0 \cdot \alpha(x,y,t,l) + \\ &\sum_{1\le i\le n} \max_{(x',y',t')\in Z} \left[F_i \cdot \beta(x_i', y_i', t_i', l) - \varepsilon(i, X_i)\right],\end{aligned} \tag{3.2}$$

where F_0 is the root filter and F_i are part filters. $\alpha(x,y,t,l)$ and $\beta(x',y',t',l)$ are features of a $3\times3\times T$ volume centered at (x,y,t) and $3\times3\times1$ volume centered at part location (x_i', y_i', t_i') respectively, at level l of the feature pyramid. Z is the set of all possible part locations and $\varepsilon(i, X_i)$ is the corresponding deformation cost. We choose the highest score from all possible placements in the detection volume as the score of each part model, and for each placement, the score is computed by the filter response minus deformation cost. If a detection volume scores above a threshold, then that action is detected at the given spatiotemporal location.

We perform a scanning search with a step stride equal to the cell size. This strikes an effective balance between exhaustive search and computational efficiency, covering the target video volume with sufficient spatiotemporal overlap.

As with DPM, our root filter expresses the overall structure of the action while part filters capture the finer details. The scores of part filters are computed with different cell size for HOG3D features and at twice the resolution compared with the root filter. This combination of root and part

filters ensures good detection performance. In experiments, we observe that the peak of score map obtained by combining root score and part scores is more distinct, stable and accurate than that of only root score map. Since the parts can ignore the background pixels in the bounding volume and focus on the distinctive aspects of the given action, the part-based SDPM is significantly more effective.

3.4 Experimental Methodology and Results

Since most of previously published results on actions in video are on whole-clip recognition rather than localization, we choose to evaluate SDPM using both criteria, while stressing that the former is not the focus of our work. Where possible, we also present direct comparisons to published localization results on standard datasets.

We present evaluations on three standard datasets, Weizmann, UCF Sports and MSR-II. The main advantage of the first is that the controlled conditions under which actions are performed and the availability of pixel-level actor masks enable us to directly assess the impact of design choices and better understand how SDPM root and part filters work in spatiotemporal volumes. SDPM achieves 100% recognition (without use of masks) and localizes every action occurrence correctly, which is an excellent sanity check.

The second dataset is much more challenging and is drawn from broadcast videos with realistic actions performed in dynamic, cluttered environments. Our results on UCF Sports demonstrate that SDPM achieves state-of-the-art localization in challenging video.

The third dataset contains videos recorded in complex environments and is particularly well suited for cross-dataset experiments. We evaluate action detection on MSR-II Dataset using SDPMs trained solely on the KTH Dataset. Our results on MSR-II confirm that parts are critical for action

detection in crowded and complex scenes.

For action detection (spatiotemporal localization), we employ the usual "intersection-over-union" criterion, generate ROC curves when overlap criterion equals 0.2 and also summarize ROC curves with different overlap criterions by the area-under-curve (AUC) measure when necessary for space constraints. For MSR-II, we show precision-recall curves following [21].

For action recognition (whole clip, forced-choice classification), we apply an SDPM for each action class to each clip and assign the clip to that class with the highest number of detections. We provide action recognition results mainly to show that SDPM is also competitive on this task, even though detection is our primary goal.

3.4.1 Experiments on Weizmann Dataset

The Weizmann dataset [14] is a popular action dataset with nine people performing ten actions. This dataset is considered easy because the actor in each clip is filmed against a static background, with little variation in viewpoint, scale and illumination. We use it primarily to understand the relative contribution of SDPM root vs. part filters.

Weizmann does not come with occurrence-level annotations so we annotate a single action cycle from each video clip to provide positive training instances; as usual, negatives include such instances from other classes augmented with randomly-sampled subvolumes from other classes.

For recognition, we follow the experimental methodology from [14]. SDPM achieves 100% recognition accuracy. While perfect recognition has also recently been achieved by others (*e.g.*, [32, 76, 84]), these all perform recognition through silhouettes. To the best of our knowledge, we are the first to achieve 100% recognition on Weizmann in a detection-based framework that operates only on raw video. When SDPM is learned using root filter alone, recognition accuracy drops to 92.4%,

confirming our hypothesis in Sec. 3.1 that parts are important, even under "easy" conditions. The feature pyramid does not contribute much on this dataset since actions are roughly at the same scale.

On detection, SDPM also achieves perfect results, correctly localizing every occurrence with no false positives. But, SDPM without parts performs poorly: only 66.7% of occurrences are correctly localized! Table 3.1 compares the detection rate for SDPM with and without parts.

Table 3.1: Detection rate on Weizmann, showing impact of parts.

Method	bend	jack	jump	pjump	run	side	skip	walk	wav1	wav2
SDPM	100	100	100	100	100	100	100	100	100	100
w/o parts	100	75	43.8	78.6	80	95.7	27.3	67.5	85	52.9

3.4.2 Experiments on UCF Sports Dataset

The UCF Sports Dataset [61] consists of videos from sports broadcasts, with a total of 150 videos from 10 action classes, such as golf, lifting and running. Videos are captured in realistic scenarios with complex and cluttered background, and actions exhibit significant intra-class variation. From the provided frame-level annotations, we create a new large bounding volume that circumscribes all of the annotations for a given action cycle. We train the SDPM using these bounding boxes.

Following Lan *et al.*'s experimental methodology [45], we split the dataset into disjoint training and testing sets. For action recognition (not our primary goal), SDPM's forced-choice classification accuracy, averaged over action classes is 75.2%, which is between 73.1% from [45] and 79.4% in [58]. Our recognition results are competitive, considering that we restrict ourselves to HOG-like features and do not employ trajectories or bag-of-words [45, 58]. When SDPM is trained without parts, the recognition accuracy drops to 64.9%; the drop of 10.3% is greater than the 7.6% observed

on Weizmann, supporting our hypothesis that parts are more important in complex videos. The per-class classification accuracy comparison among all of these methods is summarized in Figure. 3.3.

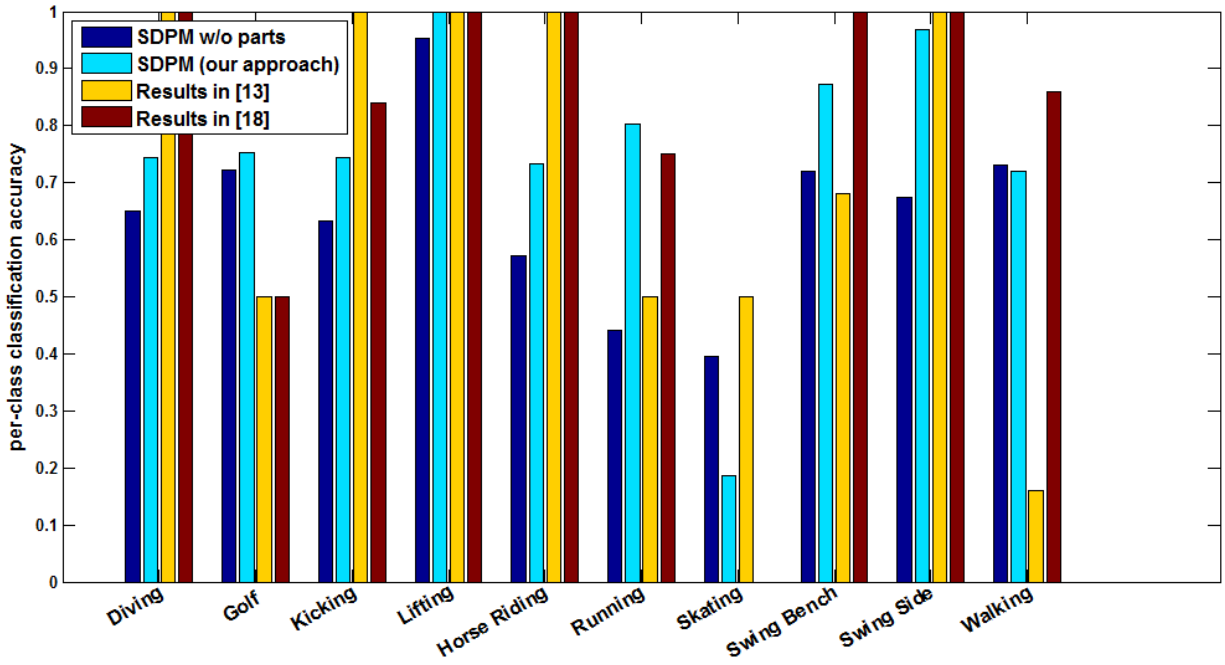
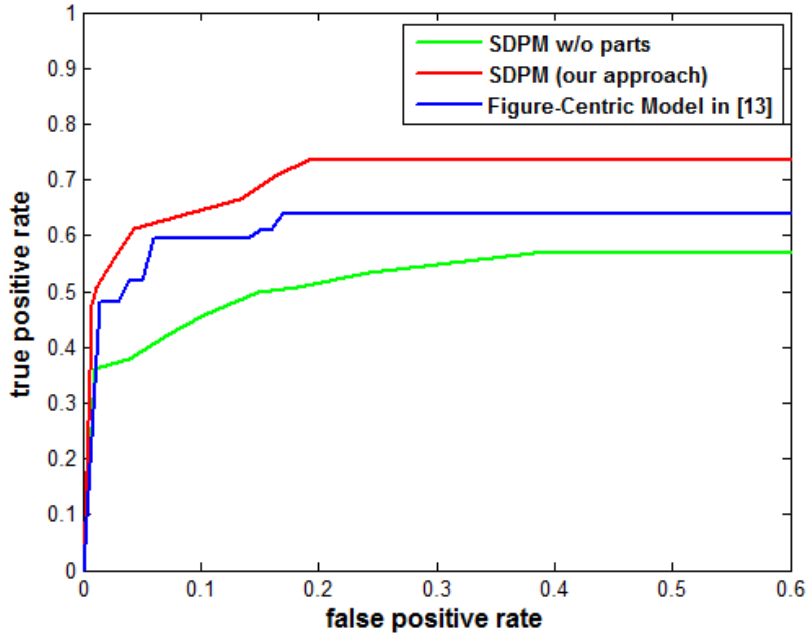


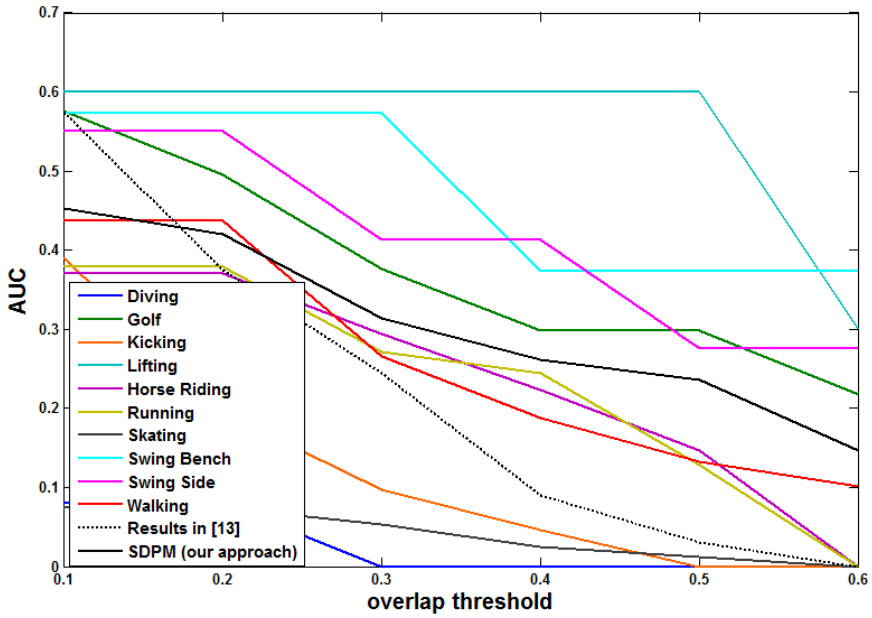
Figure 3.3: Classification performance comparison on UCF Sports vs. [45, 58].

We evaluate action localization using the standard "intersection-over-union" measure. Following [45], an action occurrence is counted as correct when the measure exceeds 0.2 and the predicted label matches. Figure. 3.4(a) shows the ROC curve for overlap score of 0.2; Figure. 3.4(b) summarizes results (as AUC) for overlap scores ranging from 0.1 to 0.6. In direct comparisons, SDPM clearly outperforms Lan *et al*. [45] on action detection; we are unable to directly compare detection accuracy against Raptis *et al*. [58] because they do not provide bounding-box level evaluations.

Figure. 3.5 shows several sample detections from UCF Sports and MSR-II datasets in a diverse set of complex scenes.

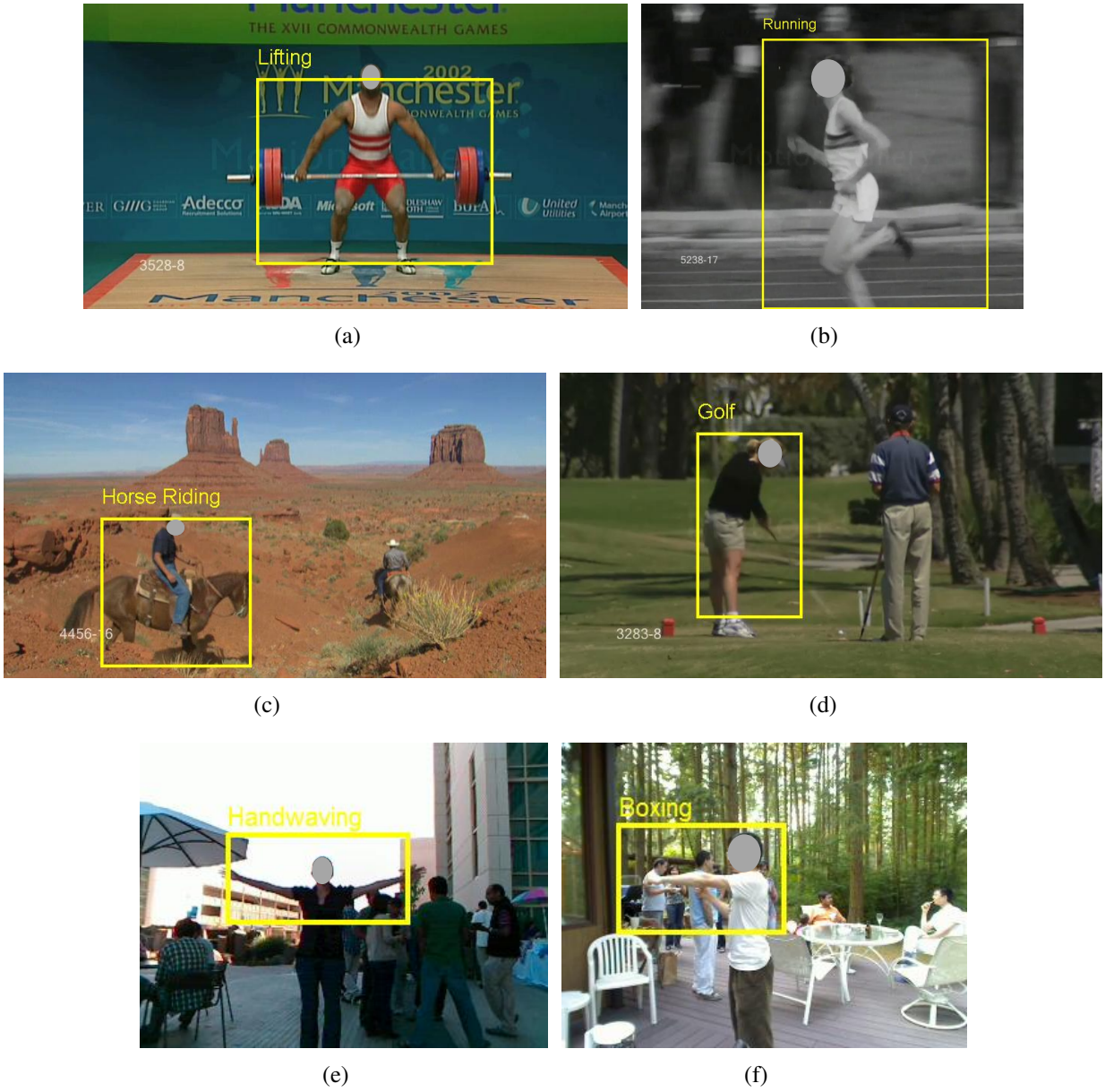


(a)



(b)

Figure 3.4: Detection performance comparisons on UCF Sports vs. [45, 58]. (a) ROC at overlap threshold of $\theta = 0.2$; (b) AUC for θ from 0.1 to 0.6. The black solid curve shows the average performance of SDPM and the black dotted curve shows the average performance of [45]. Other curves show SDPM results for each action. (Best viewed in color.)



(a) (b) (c) (d) (e) (f)

Figure 3.5: Detection examples on UCF Sports and MSR-II. (a)–(d) are examples with lifting, running, horse riding and golf SDPMs, respectively. (e) and (f) are examples with handwaving and boxing SDPMs. Actions are detected correctly even in complex scenarios.

3.4.3 Experiments on MSR-II Dataset

MSR-II [21] includes 54 video sequences recorded in crowded and complex scenes, with each video containing several instances of boxing, handclapping and handwaving. Following the cross-dataset paradigm in [21], we train on actions from KTH and test on MSR-II. For each model, the training set consists of a single action cycle from each KTH clip (positives) and instances from the other two classes (negatives). Fig 3.6 shows a direct comparison[3] between SDPM and Cao *et al*. [21]. Surprisingly, SDPM outperforms [21] even though we perform no explicit domain adaptation. We attribute this robustness to SDPM's ability to capture the intrinsic spatiotemporal structure of actions.

3.5 Summary

In this chapter, we present SDPM for action detection by extending deformable part models from 2D images to 3D spatiotemporal volumes. Naive approaches to generalizing DPMs fail, while we design SDPM by taking the asymmetry between space and time into consideration. SDPM parts are automatically selected. We show that SDPM parts are critical, both to focus on important regions in the volume as well as to handle the significant intra-class variation in real-world actions. We are the first to demonstrate perfect recognition and localization results on Weizmann in an unconstrained detection setting and achieve state-of-the-art recognition and localization results on both UCF Sports as well as MSR-II datasets. We conclusively demonstrate that DPMs (when extended correctly) can achieve state-of-the-art results in video, even with simple HOG-like features.

[3]We note that the description of precision and recall in [21] is reversed. In our evaluation, we employ the correct expression.

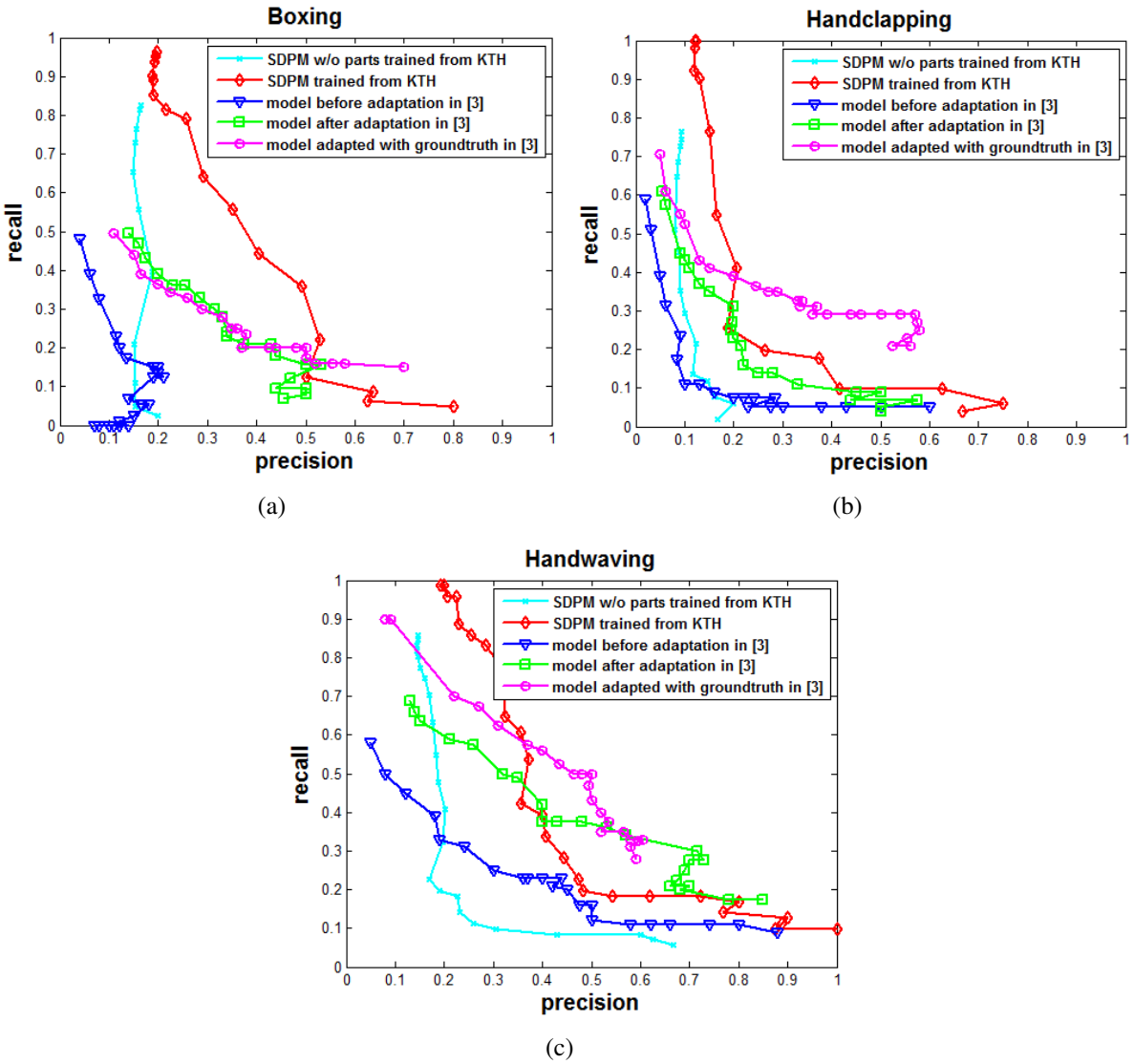


Figure 3.6: Action detection on MSR-II. SDPM outperforms model w/o parts as well as baselines in [21]. Comparison of average precision by SDPM and the best baseline in [21]: 0.3886 vs. 0.1748 (Boxing), 0.2391 vs. 0.1316 (handclapping), 0.4470 vs. 0.2671 (handwaving).

The benefits of SDPMs are two-fold. First, parts are selected automatically according to their contributions to the distinctiveness of an action. The model is represented by the most essential parts so that the negative influence caused by dynamic and cluttered environment is suppressed. Second, parts are in a deformable configuration to capture intra-class variations. We study the

generalization with straightforward features and several aspects that are not directly transferable from 2D to 3D are discussed. Extensive experimental results validate the significance of utilizing parts and SDPMs' effectiveness in detecting actions regardless of intra-class variation, scale and complex environment.

SDPM works well when detecting action performed by a single person. However, for cases where there are multiple humans in the scene, it is desirable to first track humans from frame to frame and then apply action detection and recognition algorithms. In next chapter, we present a novel approach for multiple target tracking.

CHAPTER 4: TARGET IDENTITY-AWARE NETWORK FLOW FOR ONLINE MULTIPLE TARGET TRACKING

The approach described in Chapter 3 solves the problem of detecting action performed by a single person. When there are multiple humans in the scene, humans need to be tracked from frame to frame first and then action detection and recognition can be performed. In this chapter, we present a novel approach for multiple target tracking. It differs with traditional data association based MOT trackers in that it does not rely on a pre-trained object detector to get the initial object hypotheses.

Multiple target tracking is, undoubtedly, one of the fundamental problems in computer vision, with a variety of applications ranging from surveillance to sports video analysis and medical image analysis. The goal of tracking is to detect targets and associate them across sequence of frames. Most approaches for multiple target tracking follow tracking-by-detection framework. First, a pre-trained object detector is applied to find a set of candidate locations for targets. Then these candidates are fed into a data association mechanism to form tracks. Though these methods show competitive tracking results, their performance heavily depend on object detector outputs, which are usually poor when dealing with occlusion and articulated objects. In this work we propose a tracking method based on online discriminative learning, which solves detection and global data association simultaneously by integrating a new global data association technique into the inference of a structured learning tracker. The proposed tracker overcomes the confinements of traditional tracking-by-detection based MOT approaches and is able to better handle cases where a generic object detector does not perform well. We conduct experiments on several standard sequences and two new sequences where targets experience heavy articulation. We show that our approach not only achieves results better than the state-of-art on sequences where pre-trained detectors perform well, but also improves state-of-art by a significant margin on sequences for which generic

detectors fail.

Figure 4.1: Tracking steps for one person in batch of frames. (a) shows the union of dense candidate windows used in a batch of frames in our method. (b) illustrates the union of human detection results of [33], where center of each detection is shown by "+". (c) shows the most violated constraint found through TINF to update the classifier and in (d) we show the tracking result of our method.

4.1 Proposed Approach

Given the initial bounding boxes for the objects entering the scene in the first few frames (from annotation or using an object detector), our method starts by training a model for each of the objects through structured learning (section 4.2). During learning, the most violated constraints are found by searching for a set of tracks that minimize the cost function of our target identity-aware network flow. Later, the same network is used to find the best tracks in the next temporal span (segment)

of a sequence (section 4.3). The new tracks are later used to update the model through passive aggressive algorithm [24]. An example is show in Figure 4.1.

4.2 Target-specific Model

Given a set of τ training images, $X = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^\tau\} \subset \mathcal{X}$, along with labels $Y = \{\mathbf{y}_1^1, \mathbf{y}_2^1, ..., \mathbf{y}_K^1, ..., \mathbf{y}_{K-1}^\tau, \mathbf{y}_K^\tau\} \subset \mathcal{Y}$, where $\mathbf{y}_k^t$, defines the bounding box location of object k in frame t, the target models are obtained through structured learning [77]. The aim of learning is to find a prediction function $f : \mathcal{X} \mapsto \mathcal{Y}$, which directly predicts the locations of all the objects in a set of frames. The task of structured learning is to learn a prediction function of the form

$$f_w(X) = \arg\max_{Y \in \mathcal{Y}} \sum_{t=1}^{\tau} \sum_{k=1}^{K} \mathbf{w}_k^T \phi(\mathbf{x}^t, \mathbf{y}_k^t), \tag{4.1}$$

where $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K\}$ is the concatenation of the models for all the K objects. $\phi(\mathbf{x}^t, \mathbf{y}_k^t)$ is the joint feature map which represents the feature extracted at location $\mathbf{y}_k^t$ in frame t. The optimal parameter vector $\mathbf{w}^*$ is obtained by solving the following optimization problem:

$$\begin{gathered} \min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \qquad s.t. \quad \xi \geq 0 \\ \sum_{t=1}^{\tau} \sum_{k=1}^{K} \mathbf{w}_k^T \left(\phi(\mathbf{x}^t, \mathbf{y}_k^t) - \phi(\mathbf{x}^t, \bar{\mathbf{y}}_k^t)\right) \geq \Delta(Y, \bar{Y}) - \xi \\ \forall \bar{Y} \in \mathcal{Y} \setminus Y. \end{gathered} \tag{4.2}$$

The loss function is defined based on the overlap between groundtruth label Y and prediction $\bar{Y}$

$$\Delta(Y, \bar{Y}) = \frac{1}{\tau} \sum_{t=1}^{\tau} \sum_{k=1}^{K} (1 - (\mathbf{y}_k^t \cap \bar{\mathbf{y}}_k^t)). \tag{4.3}$$

Due to exponential number of possible combinations of bounding boxes in $\mathcal{Y}$, exhaustive verification of constraint in 4.2 is not feasible. However [77, 53] showed that high quality solution can be obtained in polynomial time by using only the *most-violated constraints*, i.e a set of bounding boxes that maximize the sum of scores and loss functions. Once the model parameters are learned ($\mathbf{w}$), we use the same inference that we used for finding the *most-violated constraints* to find the best set of tracks for all the K objects in next segment of the video.

4.3 Track Inference

Given the model parameters, $\mathbf{w}$, and *dense overlapping bounding boxes* in each frame, the goal is to find a sequence of candidate windows, called a track, for each object which maximizes the score in Eq. 4.1. This maximization requires searching over exponentially many configurations. We propose to formulate the inference as a global data association which helps reducing the search space by enforcing some temporal consistency across the candidates in consecutive frames. Recently, such global data association has been formulated using network flow [101, 56], for which there exists an exact solution. In order to be able to use such networks as inference of our structured learning, the solution to the network needs to maximize the score function in Eq. 4.1. This requires the nodes in the graph to encode the probability of assigning each of the target identities to them using the learned parameters $\mathbf{w}_k$. This is not possible through traditional network flow methods.

We propose a new network called Identity-Aware network, which is shown in Figure. 4.2. The black circles represent all possible candidate locations in each frame (densely sampled across the entire frame). Each candidate location is represented with a pair of nodes that are linked through K *observation edges*; one *observation edge* for each identity. This is different from traditional network flow for which there is only one *observation edge* connecting a pair of nodes. Another major difference between our network with traditional network flow is that, our network has K

sources and K sinks, each belonging to one object. The rest of the network is similar to that of traditional network flow. Transition edges that connect nodes from different frames, represent a potential move of an object from one location to the other and there is a transition cost associated with that. There is an edge between the start/sink node and every other node in the graph which takes care of persons entering/leaving the scene. (For simplicity we are only showing some of the entry/exit edges).

The flow is a binary indicator which is 1 when a node is part of a track and 0 otherwise. A unit of flow is pushed through each source and the tracks for all the objects are found by minimizing the cost assigned to the flows. In addition, we will show later that by setting the upper bound of flows passing through *observation edges* of one bounding box, we will ensure that at most one track will claim one candidate location. In the following subsections we will first present formulation of the problem as a Lagrangian relaxation optimization and later we will introduce our spatial constraint which replace the greedy non-maximum suppression in object detectors.

4.3.1 Target Identity-aware Network Flow

First we need to build our graph $G(V, E)$. For every candidate window in frame t we consider a pair of nodes which are linked through K different *observation edges*, each belonging to one identity. For every node v_p, in frame t and v_q in frame $t + 1$, there has to be a transition edge between the two if v_q belongs to the neighborhood of v_p. Neighborhood of the node v_p is defined as

$$v_q^{t+1} \in N_\sigma(v_p^t) \Leftrightarrow \left\| v_p^t - v_q^{t+1} \right\|_2 \leq \sigma,$$

we consider a neighboring area within σ distance of node v_p that connects two candidate windows in two consecutive frames. In addition, we have source/sink edges which connect all the candidate windows to the source and sink nodes.

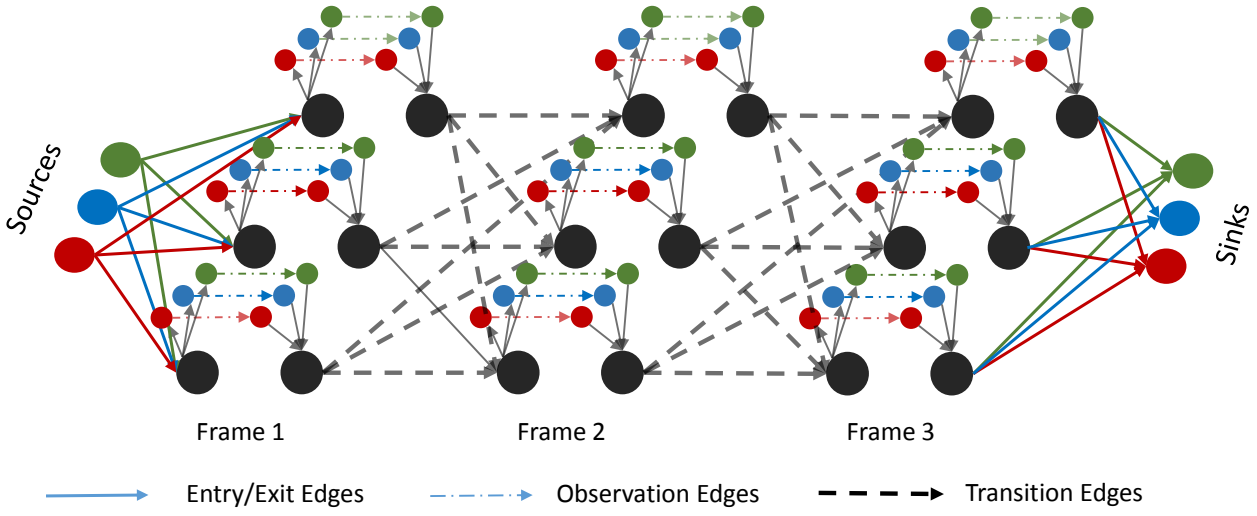


Figure 4.2: Shows the network used in our inference for three identities. Each identity is shown with a unique color. The flow entering each node can take only one of the three observation edges depending on which source (identity) does it belong to. The constraint in Eq. 4.8 ensures that one candidate can belong to only one track, so the tracks will not overlap.

Different edges in our graph are assigned costs that take into account different characteristics of objects during tracking. Each pair of nodes which represents a candidate window will be assigned K different costs defined by the K target-specific models. Considering $\mathbf{w}_k$ to be the linear weights learned for the k^{th} object, the cost assigned to k^{th} *observation edge* representing the candidate location $\mathbf{y}_p^t$ in frame t is computed as follow:

$$c_{mn}^k = -\mathbf{w}_k^T \phi(\mathbf{x}^t, \mathbf{y}_p^t).$$

Transition edges which connect the nodes in consecutive frames are assigned costs which incorporate both appearance and motion direction. The cost of a transition edge (c_{mn}^k) which connects two

candidate windows $\mathbf{y}_p^t$ and $\mathbf{y}_q^{t+1}$ in two consecutive frames is computed as:

$$c_{mn}^k = -\alpha_1 H(\phi_c(\mathbf{x}^t, \mathbf{y}_p^t), \phi_c(\mathbf{x}^{t+1}, \mathbf{y}_q^{t+1})) - \alpha_2 \frac{V_{pq} V_{ref}^k}{\|V_{pq}\| \left\|V_{ref}^k\right\|}, \tag{4.4}$$

where $H(\phi_c(\mathbf{x}^t, \mathbf{y}_p^t), \phi_c(\mathbf{x}^{t+1}, \mathbf{y}_q^{t+1}))$ is the histogram intersection between the color histograms extracted from the location $\mathbf{y}_p^t$ and $\mathbf{y}_q^{t+1}$. $\frac{V_{pq} V_{ref}^k}{\|V_{pq}\| \left\|V_{ref}^k\right\|}$ is the cosine similarity between the reference velocity vector V_{ref}^k for the k^{th} object[1] and the velocity vector between the two candidate windows V_{pq}.

Once the graph $G(V, E)$ is constructed, our aim is to find a set of K flows (tracks) by pushing a unit of flow through each source node. The flow $f_{m,n}^k$, is found by minimizing the following cost function:

$$E_{track}(F) = \sum_{k=1}^{K} \sum_{(m,n) \in E} c_{mn}^k f_{mn}^k. \tag{4.5}$$

The flow passing through these edges need to satisfy some constraints to ensure that it can actually represent a track in a real world. The set of constraints that we define in our graph are as follow:

$$\sum_n f_{mn}^k - \sum_n f_{nm}^k = \begin{cases} 1 & \text{if } m = s_k \\ -1 & \text{if } m = t_k \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

$$f_{mn}^k \in \{0, 1\} \quad \forall (m, n) \in E \ \ and \ \ 1 \leq k \leq K \tag{4.7}$$

[1] Average velocity vector for the k^{th} identity in previous batch

$$\sum_{k=1}^{K} f_{mn}^{k} \leq 1 \tag{4.8}$$

The constraint in Eq. 4.6 is the supply/demand constraint, enforcing the sum of flows arriving at one node to be equal to the sum of flows leaving that node. Constraint in Eq. 4.8 is the bundle constraint, ensuring that the tracks of different identities will not share a node by setting the upper bound of sum of flows passing through each node to be one.

One can formulate Eq. 4.5 as an Integer Program (IP). Since IP is NP-Complete, in practice, the problem can be relaxed to Linear Program (LP) in which the solution can be found in polynomial time. However, our experiments show that without pruning steps like the one in [70, 83], which reduces the number of candidate windows, it is intractable to find a solution for a large number of people in a long temporal span (one should note that the input to our tracker is dense candidate windows sampled from the entire frame). Instead, we propose a Lagrange relaxation solution to this problem. We show that after relaxing the hard constraints, the problem in each iteration, reduces to finding the best track for each target separately. The global solution to this can be found in linear time through dynamic programming. Moreover, our iterative optimization allows us to incorporate spatial constraint which further improves the tracking results.

4.3.2 Lagrange Relaxation Solution to TINF

The key idea of Lagrange relaxation is relaxing the hard constraints and moving them into the objective function in order to generate a simpler approximation. We start by relaxing the bundle constraints in Equation. 4.8, where we introduce the non-negative Lagrange multiplier λ_{mn}. λ is a vector of Lagrange multipliers that has the same dimension as the number of edges in the graph.

After relaxing the bundle constraint the new objective function becomes:

$$E_{track}(F) = \sum_{k=1}^{K} \sum_{(m,n)\in E} c_{mn}^{k} f_{mn}^{k} + \sum_{(m,n)\in E} \lambda_{mn} (\sum_{k=1}^{K} f_{mn}^{k} - 1), \tag{4.9}$$

We can further simplify this and write it as follow:

$$E_{track}(F) = \sum_{k=1}^{K} \sum_{(m,n)\in E} (c_{mn}^{k} + \lambda_{mn}) f_{mn}^{k} - \sum_{(m,n)\in E} \lambda_{mn}, \tag{4.10}$$

Subject to:

$$\sum_{n} f_{mn}^{k} - \sum_{n} f_{nm}^{k} = \begin{cases} 1 & \text{if } m = s_k \\ -1 & \text{if } m = t_k \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

$$f_{mn}^{k} \in \{0, 1\} \quad \forall (m,n) \in E \ \ and \ \ 1 \leq k \leq K \tag{4.12}$$

The second term in Eq. 4.10 is a constant for any given choice of Lagrange multipliers, therefore we can ignore it. The new objective function has a cost of $c_{mn}^{k} + \lambda_{mn}$ associated with every flow variable f_{mn}^{k}. Since none of the constraints in this problem contains the flow variables for more than one of the identities, we can decompose the problem into separate **minimum cost flow** problem for each identity. Since only one unit of flow is pushed through each source, the solution to minimum cost flow can be found optimally through dynamic programming in $O(N)$. Thus the complexity of our optimization in each iteration is $O(KN)$, where K is the number of targets and N is the number of frames in the temporal span.

Consequently, to apply the sub-gradient optimization to this problem, we alternate between the following two steps:

- For a fixed value of Lagrange multipliers we would solve the minimum cost flow for each identity separately considering the cost coefficients $c^k_{mn} + \lambda_{mn}$.

- Update the Lagrange multipliers according to Eq. 4.13.

$$\lambda^{q+1}_{mn} = \left[\lambda^q_{mn} + \theta^q (\sum_{k=1}^{K} f^k_{mn} - 1) \right]^+, \tag{4.13}$$

where λ^q is the Lagrange multipliers at iteration q, θ^q is the step size defining how far we would like to move from current solution and $[\alpha]^+ = max(0, \alpha)$.

4.3.3 Spatial Constraint

One major difference between our tracking algorithm and other data association based trackers is that, the input to our tracker is dense candidate windows instead of human detection output. When pedestrians with similar appearance and motion are walking next to each other, it is very likely to have ID-Switches in tracking results. Also when a pedestrian becomes partially occluded, the track for that person tend to pick candidates that highly overlap with other nearby pedestrians (see Figure. 4.3). This issue is addressed by non-maximum suppression in human detection [33] or by using other techniques like the one in [103], where the objects are forced to keep the spatial configurations between consecutive frames. Instead we introduce a soft-spatial constraint which penalizes the tracks that highly overlap.

Our spatial constraint can be easily integrated into our iterative optimization. Similar to our Lagrange multipliers, we introduce a new set of variables that penalizes the cost of *observation edges* that highly overlap. Now the cost associated to each *observation edge* becomes $c^k_{mn} + \lambda_{mn} + \rho_{mn}$. ρ is a vector which has the same size as the number of *observation edges* in the graph. It is initialized

with a zero vector in the first iteration and is updated according to Eq. 4.14.

$$\rho_{mn}^{q+1} = \left[\rho_{mn}^{q} + \theta^{q}[(y_m^t \cap y_n^t) - 0.5]^{+} \exp^{((y_m^t \cap y_n^t)-0.5)/2}\right]^{+}, \tag{4.14}$$

where $y_m^t \cap y_n^t$ is the overlap between neighboring bounding boxes in the same frame. ρ_{mn} penalizes the observation node which is associated with the cost c_{mn}. One should note that the spatial constraint only penalizes the bounding boxes that overlap more than 50% and the penalty increases exponentially as the overlap increases.

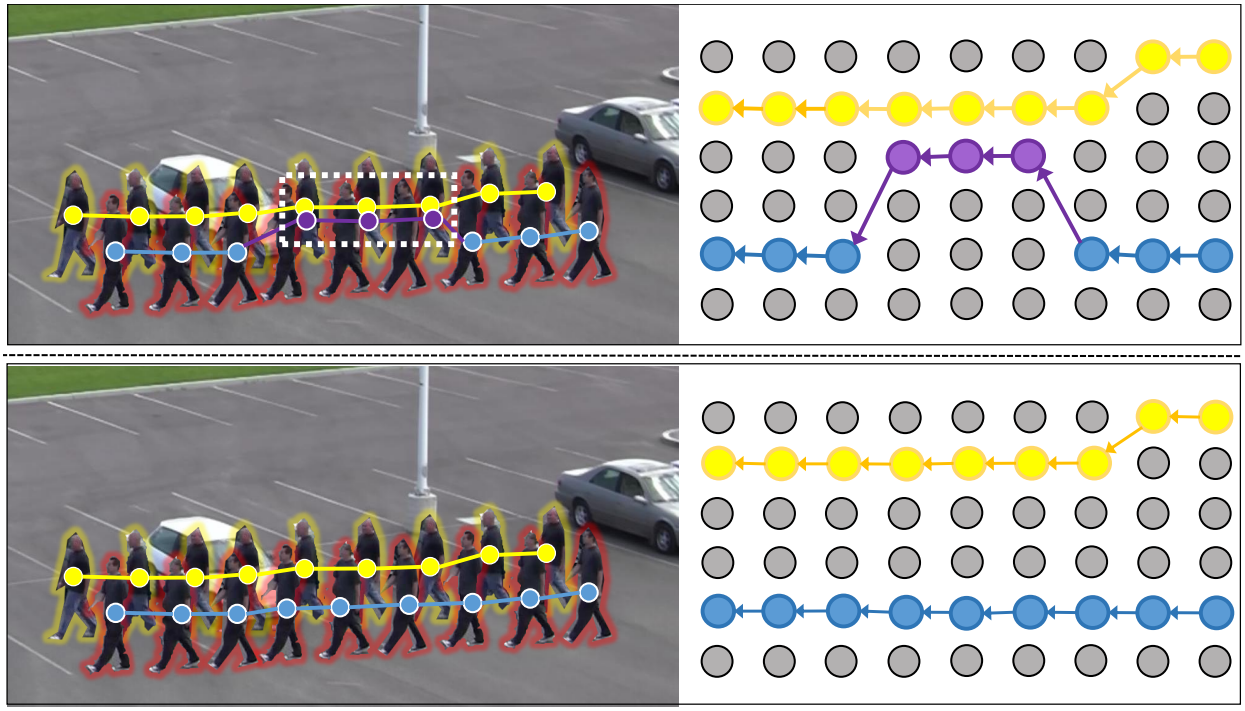


Figure 4.3: In top row the tracks of two pedestrians get confused due to their appearance similarity. This issue is fixed when the spatial constraint is enforced (bottom row). Images on the right show nodes in TINF graph and images on the left show the selected nodes mapped to real video frame.

After adding the spatial constraint the cost of the nodes are updated at each iteration according to

the following:

$$c_{mn}^{q+1,k} = c_{mn}^{k} + \lambda_{mn}^{q+1} + \rho_{mn}^{q+1}. \tag{4.15}$$

We observed that penalizing both nodes that highly overlap, sometimes lead to inaccurate bounding boxes for one of the tracks. Therefor, we only penalize the observation nodes of the track that have lower score according to the score function in Eq. 4.1. The algorithm of our Lagrangian relaxation solution, including the spatial constraint, is shown in Algorithm 1.

Algorithm 1: Lagrangian Relaxation Solution to TINF.

Input: candidate windows in T frames
model parameters for each identity ($\mathbf{w}_k$)
Output: Tracking result for K identities

- *build the TINF graph*
  $G(V, E)$
- *Initialize the lagrange multipliers and spatial constraint multipliers*
  $\lambda = 0,\ \rho = 0,\ \theta = 1,\ q = 1$

while *do not converge* **do**

-Solve the minimum cost flow for each identity ($\mathbf{f}^k$)
-Update Lagrange multiplies;
$\lambda_{mn}^{q+1} = \left[\lambda_{mn}^{q} + \theta^q(\sum_k^K f_{mn}^k - 1)\right]^+$
-Update spatial constraint multipliers;
$\rho_{mn}^{q+1} = \left[\rho_{mn}^{q} + \theta^q[(y_m^t \cap y_n^t) - 0.5]^+ \exp^{((y_m^t \cap y_n^t)-0.5)/2}\right]^+$
-Update edge costs
$c_{mn}^{q+1,k} = c_{mn}^{k} + \lambda_{mn}^{q+1} + \rho_{mn}^{q+1}$
-Update step size
$\theta^{q+1} = \frac{1}{q}$
$q = q + 1$

end

4.4 Experimental Results

In our evaluation, we focus on tracking humans, due to its importance. But our method can be used for tracking any object. We conducted two sets of experiments. First we compare our method with the state of the art trackers on publicly available sequences. For those sequences where the object detection performs well, excellent results are already reported. However, we show that, using our method, one can further improve the performance. Second, we evaluated our method on two new sequences where targets experience heavy articulation and we show that we can significantly improve the performance of data-association based trackers as well as online trackers. *Parking Lot 1* [71], *Parking Lot 2* [72], *TUD Crossing* [5] and PETS [34] are the four publicly available sequences used in our experiments and the two new sequences are called *Running* and *Dancing*.

Setup. To initialize the target, similar to [103, 22] we used manual annotation. We annotated four initial bounding boxes for each object entering the scene. We also report results where targets are initialized automatically using a pre-trained object detector. For manual annotation the target is initialized only once and there is no re-initialization of targets. We use histogram-of-oriented gradient [26] and color histogram [29] as our features. We found the combination of both features to be important. HOG captures the edge information of target and is helpful in detecting target from the background, while color histogram is a video specific features and helps in distinguishing different targets from each other. The sequence is divided into segments of 20 frames each. At the end of each temporal span we check if a track is valid or not by comparing its score with a pre-defined threshold. If the track is valid then it is used to update the model.

Comparison. We quantitatively and qualitatively compare our method with two main sets of trackers: data-association based trackers and online trackers. On sequences for which no other tracking results are reported, we compare our method with three data-association based trackers for which we have access to their code, CET [5], DCT [6] and GOG [56]. We used Deformable

Part based model [33] as our human detector. The input to the data-association methods is the DPM output with different thresholds ranging from -1 to 0. We agree that these trackers have parameters to tune to achieve the best performance for each sequence. However, we stayed with the default parameter suggested by the authors and the only parameter we changed was the human detector threshold. The numbers reported are for a threshold that gave us the best performance. In addition to these three trackers, we quantitatively compared our results with other trackers which have used the same sequences in their experiments. For online discriminative learning-based trackers we selected STRUCK [36] as well as structure preserve multi-object tracking (SPOT) approach [103]. For STRUCK, we train one structured SVM per target given the annotation of humans in the first frame. For SPOT, the manual annotation is used to initialize the tracking. Whenever a new object enters the scene we re-initialize the tree to get the track for the new target. In SPOT the spatial relationship between the targets are modeled during tracking. This model is updated according to a weight ,γ, every frame. The weight was set originally to 0.05 and we found the weight to be important in final results. The reported results for SPOT are based on the best value that we found for γ.

For quantitative analysis we utilized two sets of metrics. CLEAR MOT metrics [10] as well as Trajectory Based Metrics (TBM) [101]. CLEAR metrics (MOTA-MOTP) look at the entire video as a whole while TBM consider the behavior of each track separately. Each of these metrics captures different characteristics of a tracker and it is important to look at both of them while comparing different tracking algorithms to better capture strength and weakness of each tracker. [2] The quantitative comparison of our approach is shown in Table. 4.1. Since the candidate windows are not sample at all possible scales, the final tracking bounding boxes might not be as accurate as when a human detector is used. Thus to be fair, we used 30% overlap threshold for our quantitative evaluation (for all trackers) when computing CLEAR or TBM metrics.

[2]For more information please visit: `http://crcv.ucf.edu/projects/TINF/`

Table 4.1: Quantitative comparison of our method with competitive approaches of LPD [75], LDA [67], DLP [41], H2T [86], GMCP [97], PF [16],CET [5], DCT [6], GOG [56], STRUCK [36] and SPOT [103].

	Method	MOTA	MOTP	MT	ML	IDS
Running	CET	0.463	0.508	0.67	0	0
	DCT	0.376	0.504	0	0	0
	GOG	0.03	0.6945	0	1	0
	SPOT	0.661	0.662	0.67	0	0
	STRUCK	0.799	0.643	1	0	0
	Ours	**0.987**	**0.665**	**1**	**0**	**0**
Dancing	CET	0.366	0.62	0.57	0	64
	DCT	0.363	0.636	0	0.14	81
	GOG	0.249	0.64	0	0.14	96
	SPOT	0.554	0.659	0.43	0	16
	STRUCK	0.691	0.671	0.71	0.14	9
	Ours	**0.899**	**0.659**	**0.86**	**0**	**1**
Parking Lot 2	CET	0.717	0.558	0.6	0	59
	DCT	0.736	0.565	0.8	0	48
	GOG	0.4827	0.598	0.2	0.1	96
	Ours	**0.893**	**0.663**	**1**	**0**	**0**
Parking Lot 1	LPD	0.893	0.777	NR	NR	NR
	GMCP	0.9043	0.741	NR	NR	NR
	H2T	0.884	0.819	0.78	0	21
	Ours	**0.907**	**0.693**	**0.86**	**0**	**3**
TUD Crossing	PF	0.843	0.71	NR	NR	2
	GMCP	0.9163	0.756	NR	NR	0
	Ours	**0.929**	**0.692**	**1**	**0**	**0**
PET 2009	LDA	0.9	0.75	0.89	NR	6
	DLP	0.91	0.7	NR	NR	5
	GMCP	0.903	0.6902	NR	NR	8
	Ours	**0.904**	**0.6312**	**0.95**	**0**	**3**

Initialization. For initialization, besides manual annotation, we use human detection to automatically initialize the targets. During each segment a new track is initialized if there are at least four confident detections in consecutive frames that highly overlap and are not associated to any other tracks. We tested automatic initialization of targets on publicly available sequences where human detection performs reasonably well. As can be seen in Table. 4.2, the performance of our method doesn't change much when using automatic initialization. The main difference is that some of the tracks in some sequences will start late compared to manual annotation which cause a small drop in MOTA due to the added false negatives.

Table 4.2: This table shows the performance of our method with automatic and manual initialization of targets. For automatic initialization of targets a pre-trained human detector is used [33].

Method	MOTA	MOTP	MT	ML	IDS
PL1-Auto	0.905	0.652	0.857	0	5
PL1-Manual	0.907	0.693	0.8571	0	3
TUD-Auto	0.908	0.688	0.9167	0.083	0
TUD-Manual	0.929	0.692	1	0	0
PL2-Auto	0.834	0.632	0.7	0	5
PL2-Manual	0.893	0.663	1	0	0

Effect of Spatial Constraint. In order to clearly see the effect of our spatial constraint, we ran our method on different sequences with and without the spatial constraint. As can be seen in Table. 4.3, when spatial constraint is added, the performance increases, specially for sequences which involve interaction between objects.

Run Time and Convergence. In order to compare the complexity of the proposed Lagrangian relaxation method with the one of IP and LP, we implemented the IP and LP version of our method as well. We used CPLEX [1] as the optimization toolbox.The performance of IP and LP is within $1-2\%$ performance of our Lagrange relaxation formulation when no spatial constraint is used.

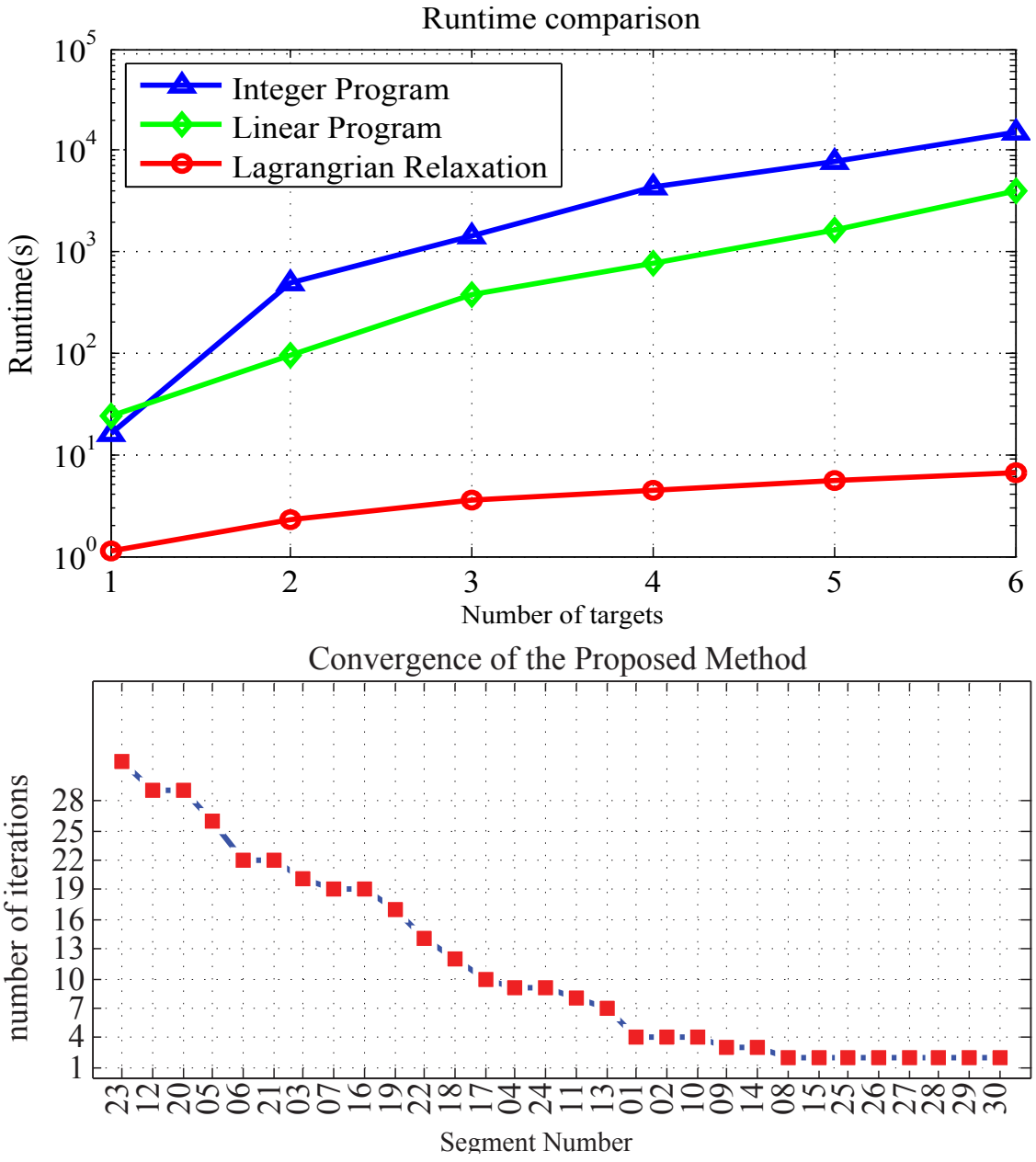


Figure 4.4: The top figure shows the run time comparison of the proposed Lagrangian solution vs IP and LP. The bottom figure shows the convergence of the proposed method on PL2 sequence.

Table 4.3: This table shows the performance of our method with and without spatial constraint. The improvement from spatial constraint is evident from this evaluation.

	MOTA	MOTP	MT	ML	IDS
Running	0.972	0.681	1	0	0
Running-SP	0.987	0.665	1	0	0
Dancing	0.88	0.649	0.86	0	2
Dancing-SP	0.899	0.659	0.86	0	1
PL1	0.88	0.629	0.79	0	4
PL1-SP	0.907	0.693	0.86	0	3
PL2	0.822	0.656	0.9	0	2
PL2-SP	0.893	0.663	1	0	0
TUD	0.866	0.698	0.92	0	1
TUD-SP	0.929	0.692	1	0	0

The runtime for a selected segment of PL2 sequence with different number of targets is shown in the top row in Figure. 4.4. Note that the curves are shown with logarithmic coordinates. As can be observed, the proposed optimization is a lot more efficient compared to the IP and LP solutions. Finally, the bottom row in Figure. 4.4 shows the number of iterations that the Lagrangian optimization takes to converge in PL2 sequence. In Figure. 4.4 the horizental axes shows the segment number in PL2 sequence.

4.5 Summary

In this chapter, we introduce a new tracker which brings in online discriminative learning and global data association method in a unified framework. At the core of our framework lies a structured learning which learns a model for each target. The inference is formulated as global data

association problem which is solved through a proposed target identity-aware network flow. Our experiments show that the proposed method outperforms traditional online trackers in difficult scenarios. Our work is one of the very few attempts that aims to solve tracking multiple objects by solving detection and tracking simultaneously. We hope that our results encourage other researcher to discover this direction more.

The output tracks from the proposed TINF tracker are coarse bounding boxes around targets. In next chapter, we will present a new framework to couple TINF tracker with spatial temporal target segmentation, which outputs tracks as pixel-wise target masks.

CHAPTER 5: ON DETECTION, DATA ASSOCIATION AND SEGMENTATION FOR MULTI-TARGET TRACKING

Most existing trackers, including TINF tracker described in Chapter 4 outputs target tracks via a sequence of bounding boxes, which is a coarse representation. However, the ultimate way of labeling targets in a sequence is to tag every pixel in an image instead of just providing a coarse bounding box around the target. The pixel-wise object segmentation would provide fine details of targets, which is also desirable for later tasks.

Traditionally, object detection, tracking and segmentation are treated as separate problems and solved independently. However, they are actually closely related and solving one should help the others. In chapter 4, we presented how object detection and data association are combined within one framework by TINF. In this chapter, we propose to further couple TINF tracker with spatiotemporal pixel wise segmentation through Lagrange dual decomposition, so that detection, data association and segmentation can be solved simultaneously in one framework. Spatiotemporal target segmentation is performed by applying multi-label Conditional Random Field (CRF) to a superpixel based spatiotemporal graph in a segment of video to assign background or target labels to every superpixel. Dual decomposition serves as the bridge to connect tracking and segmentation parts. It tries to make sure the results from tracking and segmentation parts are consistent through iterative optimization. By taking advantage of the synergies between tracking and segmentation, the proposed dual decomposition based approach achieves more accurate segmentation results and also helps resolve typical difficulties in multiple target tracking, such as occlusion, ID-switch and track drifting. In addition, the final output of our tracker is the fine contours of the targets rather than traditional bounding boxes. We evaluate the proposed approach on diverse and challenging sequences and achieve competitive results on tracking, segmentation as well as detection.

The organization of the rest of this chapter is as follows: In the next section, we describe our proposed approach for spatiotemporal segmentation and how it is coupled with TINF tracker through dual decomposition, to solve detection, data association and segmentation simultaneously. In Section 5.2, we present extensive and thorough experimental results on tracking, segmentation and detection respectively. In Section 5.3, We summarize this chapter.

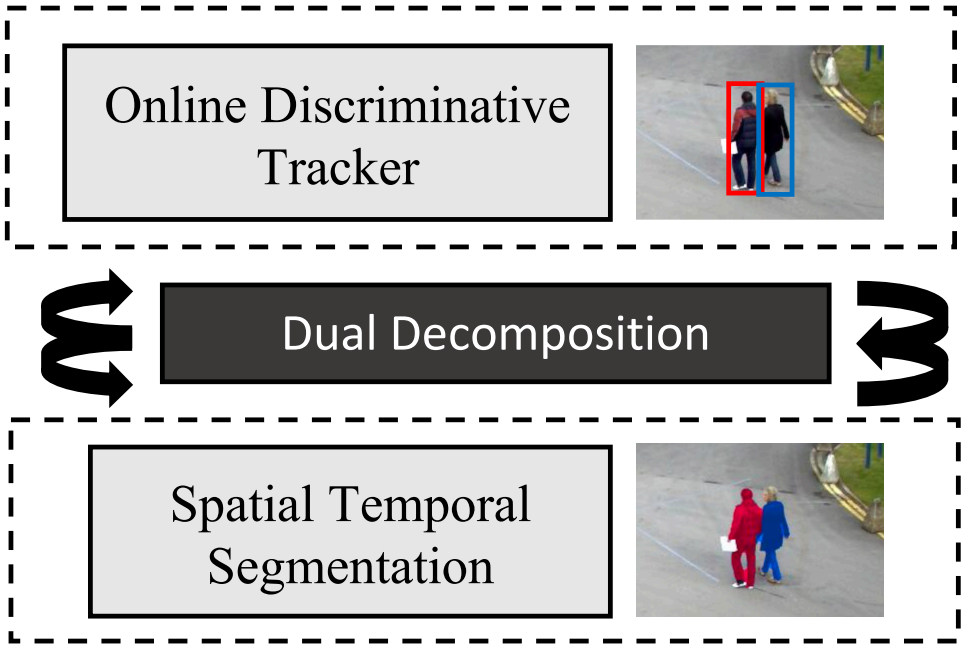


Figure 5.1: This figure shows pipeline of the proposed method. The two main components of our algorithm, online multi-target tracking and spatial-temporal segmentation, are combined through dual decomposition.

5.1 Proposed Approach

Our pipeline is shown in Figure 5.1. Two main components of our framework are an online discriminative learning based tracker and a GMM(Gaussian Mixture Model)-based spatial temporal video segmentation algorithm. These two components collaborate through a Lagrange dual decom-

position to help improve performance of each task of detection, data association or segmentation. In the following subsections, we first present the spatial temporal segmentation algorithm used in our approach, and then explain how dual decomposition is used to combine the online discriminative learning based tracker (from Chapter 4) with spatial temporal video segmentation.

5.1.1 Spatiotemporal Segmentation

In this section, we describe the procedure to get foreground/background segmentation for all targets in a segment of video. The main aim of segmentation is to find foreground pixels corresponding to each target, so that precise object contour separating it from the background can be determined, instead of typical bounding box representation.

We determine the segmentation mask of target, k, in its first frame automatically from its initial box $\bar{\mathbf{y}}_k$. GrabCut algorithm [62] is applied to target k's small surrounding region, by initializing pixels within box $\bar{\mathbf{y}}_k$ as foreground, while pixels outside box $\bar{\mathbf{y}}_k$ as background. GrabCut starts from this initial segmentation and iteratively refines foreground/background boundary.

Then based on the foreground pixels obtained by GrabCut, we build a pixel-level foreground GMM model, $\mathbf{w}_{fg(k)}$, for target k. In addition, a background image, obtained by averaging frames in the video, is used to build a universal background GMM model $\mathbf{w}_{bg}$. $CIELAB$ color space is used. A foreground confidence map, $S_{fg(k)}$, for target k and a background confidence map, S_{bg}, are computed by applying $\mathbf{w}_{fg(k)}$ and $\mathbf{w}_{bg}$ to every pixel in a new frame respectively. An example is shown in Figure 5.2.

Given K targets in the scene, the goal of segmentation is to assign one of $K+1$ labels (K targets or background) to every pixel. The segmentation problem in upcoming frames is solved by multi-label CRF. Since superpixels naturally preserve the boundary of objects and are computationally

efficient for processing, we build a superpixel based spatio-temporal graph. Simple Linear Iterative Clustering (SLIC) [2] is employed to generate N superpixels in every frame. There are two types of edges in the graph: spatial edges, ε_S, and temporal edges, ε_T. Spatial edges connect all neighboring superpixels in a frame. Two superpixels s_m and s_n are considered as spatial neighbors if they share an edge in image space. Temporal edges connect all neighboring superpixels in two consecutive frames. Superpixels s_m and s_n are considered as temporal neighbors if at least $1/3$ of the pixels in s_m move to s_n in the next frame as predicted by optical flow. Temporal edges help preserve segmentation consistency across frames.

With the spatio-temporal graph, the multi-label Conditional Random Field (CRF) energy function is defined as

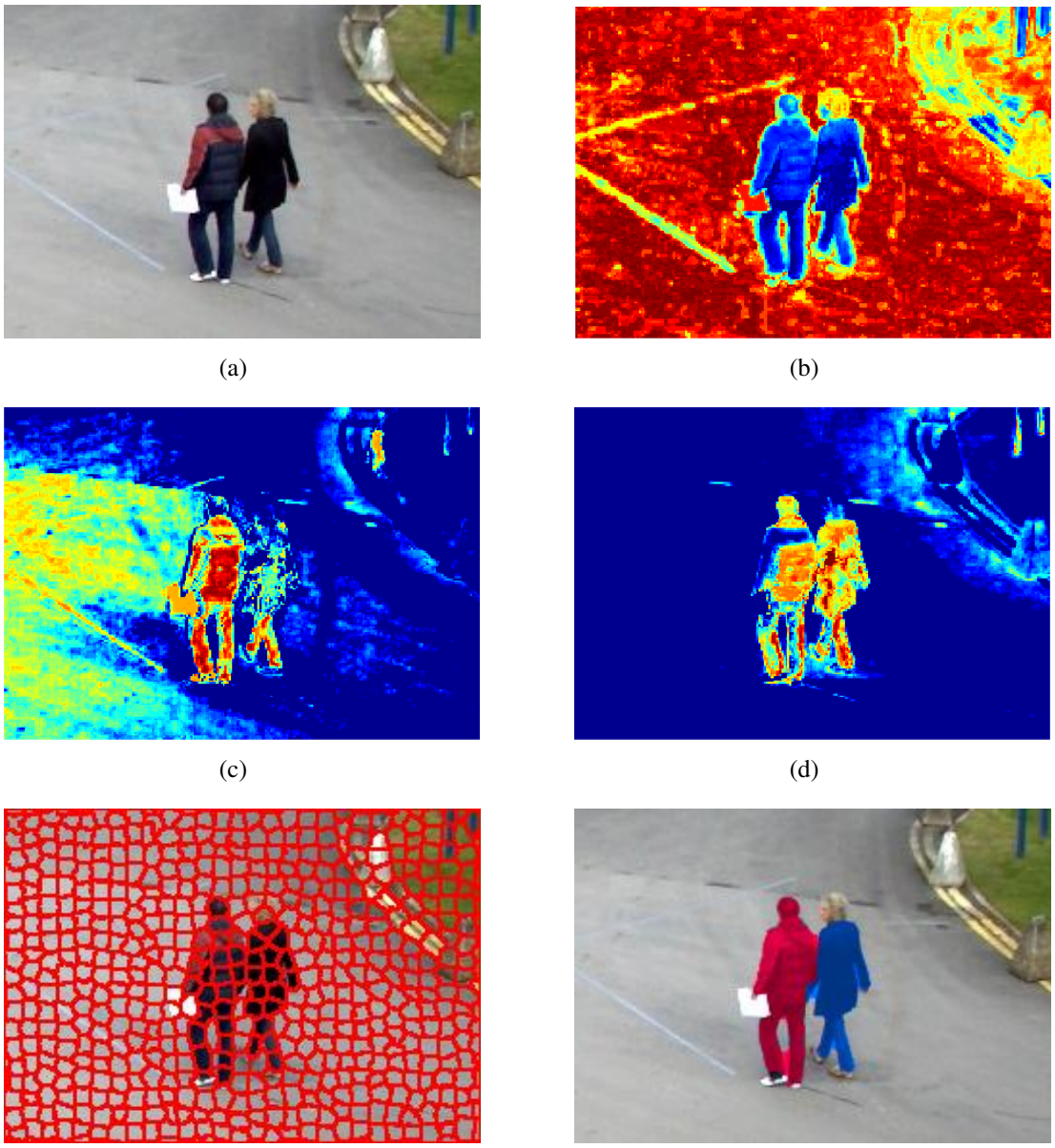
$$\begin{aligned} E_{seg}(Z) = & \sum_{s_m} Q(s_m, z_{s_m}) + \beta_1 \sum_{(s_m,s_n)\in\varepsilon_S} D(s_m, s_n) \\ & + \beta_2 \sum_{(s_m,s_n)\in\varepsilon_T} D(s_m, s_n), \end{aligned} \tag{5.1}$$

where Z denotes the target/background labeling of all superpixels in a segment of video. z_{s_m} is the labeling of superpixel s_m. $z_{s_m} = k$ if s_m is labelled as target k and $z_{s_m} = 0$ if s_m is labelled as background. The energy function is optimized using graph cuts with α-expansion [15].

The unary term $Q(s_m, z_{s_m})$ in Eq. 5.1 is the cost of labeling superpixel s_m:

$$Q(s_m, z_{s_m}) = \begin{cases} -log(S_{fg(k)}(s_m)), & \text{if } z_{s_m} = k \\ -log(S_{bg}(s_m)), & \text{if } z_{s_m} = 0 \end{cases} \tag{5.2}$$

Here $S_{fg(k)}(s_m)$ represents the probability of superpixel s_m belonging to target k. It is computed as the average confidence value of $S_{fg(k)}$ over all pixels in s_m. $S_{bg}(s_m)$ denotes the probability that superpixel s_m belongs to the background.



(a) (b) (c) (d) (e) (f)

Figure 5.2: An illustration of target/background confidence maps and segmentation results. (a) A new frame (part of the frame is shown for clarity). (b) Background confidence map. Red represents higher confidence value while blue represents lower value. (c) and (d) show confidence maps for the target on the left and the target on the right respectively. (e) Superpixels in the part of the frame. (f) The final segmentation results after applying CRF to the superpixel based spatio-temporal graph. Red and blue masks represent foreground pixels for the two targets respectively.

The pairwise terms in Eq. 5.1 incorporate pairwise constraints by combining color similarity and the mean flow direction similarity between two neighboring superpixels. The pairwise potential $D(s_m, s_n)$ between two spatial/temporal neighboring superpixels s_m and s_n is defined as

$$\begin{aligned} D(s_m, s_n) =& \mathbf{1}(z_{s_m} \neq z_{s_n}) \cdot D_c(s_m, s_n) \cdot D_f(s_m, s_n), \\ D_c(s_m, s_n) =& \frac{1}{1 + \|LAB(s_m) - LAB(s_n)\|}, \\ D_f(s_m, s_n) =& \frac{V_{s_m} V_{s_n}}{\|V_{s_m}\| \, \|V_{s_n}\|}, \end{aligned} \tag{5.3}$$

where $\mathbf{1}(\cdot)$ is the one-zero indicator function. $LAB(s_m)$ is the average LAB color of superpixel s_m and $D_c(s_m, s_n)$ defines the color similarity between superpixels s_m and s_n. V_{s_m} denotes the mean optical flow of superpixel s_m and $D_f(s_m, s_n)$ is the direction similarity between the mean flows of superpixels s_m and s_n.

5.1.2 Dual Decomposition

As discussed previously, and we later show quantitative results in our experiment section, the two tasks: online discriminative tracker (Sec. 4.1) and spatial temporal target segmentation (Sec. 5.1.1) are highly correlated. To take advantage of synergies between them, dual decomposition is employed to couple these two tasks. We aim at minimizing the following energy function:

$$\min_{F,Z} E(F, Z) = \min_{F,Z} (E_{track}(F) + E_{couple}(F, Z) + E_{seg}(Z)), \tag{5.4}$$

where $E_{track}(F)$ and $E_{seg}(Z)$ are defined as in Eq. 4.5 and Eq. 5.1 respectively. F denotes the set of bounding boxes found by the tracking procedure in Sec. 4.1 and Z denotes target/background segmentation obtained in Sec. 5.1.1. The coupling term contains both bounding boxes and seg-

mentation information:

$$E_{couple}(F, Z) = \sum_{k,m} (\mathbf{1}(m \in f_k, z_m \neq k)\theta^m_{f_k} + \mathbf{1}(m \notin f_k, z_m = k)\varphi^m_{f_k}). \tag{5.5}$$

This energy introduces penalties for background labels inside target bounding boxes as well as foreground labels outside target bounding boxes. k denotes a target and m denotes a pixel. The first term penalizes pixels that are not labelled as target k, but are in target k's tracking boxes. f_k denotes the bounding boxes for target k, and $\mathbf{1}(m \in f_k, z_m \neq k)$ represents pixels in f_k which are not labelled as target k. Since a target's bounding box is highly likely to include some non-target pixels near the border of box, but not at the center of box, the resulting penalty is weighted by a human shape prior θ_{f_k}. Thus, background pixels at the center of box induce higher penalty while those close to the border of box result in lower penalty. The same human shape prior θ is used as in [51]. The second term penalizes pixels that are labelled as target k but are outside target k's boxes. $\mathbf{1}(m \notin f_k, z_m = k)$ represents pixels outside f_k which are labelled as target k. The corresponding penalty is weighted by φ_{f_k}, which has a zero weight within f_k and uniform non-zero weight outside f_k.

By introducing an equality constraint, Eq. 5.4 can be rewritten as

$$\min_{F^0,F^1,Z} E(F^0, F^1, Z) = \min_{F^0,F^1,Z} (E_{track}(F^0) + E_{couple}(F^1, Z) + E_{seg}(Z)) \quad s.t. \quad F^0 = F^1. \tag{5.6}$$

Now, the energy function is separable. We form the Lagrangian dual form of the above problem

by introducing Lagrange multipliers λ'

$$\begin{aligned} L(\lambda') &= \min_{F^0,F^1,Z}(E_{track}(F^0) + E_{couple}(F^1, Z) + E_{seg}(Z) + \lambda'(F^0 - F^1)), \\ &= \min_{F^0}(E_{track}(F^0) + \lambda' F^0) + \min_{F^1,Z}(E_{couple}(F^1, Z) + E_{seg}(Z) - \lambda' F^1). \end{aligned} \tag{5.7}$$

Here λ' has the same dimension as F^0 and F^1.

Eq. 5.7 can be further decomposed into two independent subproblems:

$$g(\lambda') = \min_{F^0}(E_{track}(F^0) + \lambda' F^0), \tag{5.8}$$

$$h(\lambda') = \min_{F^1,Z}(E_{couple}(F^1, Z) + E_{seg}(Z) - \lambda' F^1). \tag{5.9}$$

The first subproblem (Eq. 5.8) is equivalent to a set of network flow problems, thus $g(\lambda')$ can be solved efficiently using dynamic programming. The second subproblem (Eq. 5.9) involves both tracking boxes and segmentation. When F^1 is fixed, $E_{couple}(F^1, Z)$ becomes a unary term on Z, thus $h(\lambda')$ can be solved by graph-cut. When Z is fixed, $h(\lambda')$ can be optimized by evaluating all candidate boxes. So a two-step procedure is employed to optimize $h(\lambda')$.

We use a sub-gradient method to optimize the Lagrangian dual problem. The algorithm works by repeating the following steps:

1. Get F^0 by solving the tracking subproblem $g(\lambda')$ (Eq. 5.8).
2. Get F^1 by solving the segmentation subproblem $h(\lambda')$ (Eq. 5.9).
3. Stop if $F^0 = F^1$.
4. Otherwise, update dual variable λ' by $\lambda' \leftarrow \lambda' + \alpha_p(F^0 - F^1)$, where α_p is the step size in iteration p and is computed as $\alpha_p = 1/(10 + p)$.

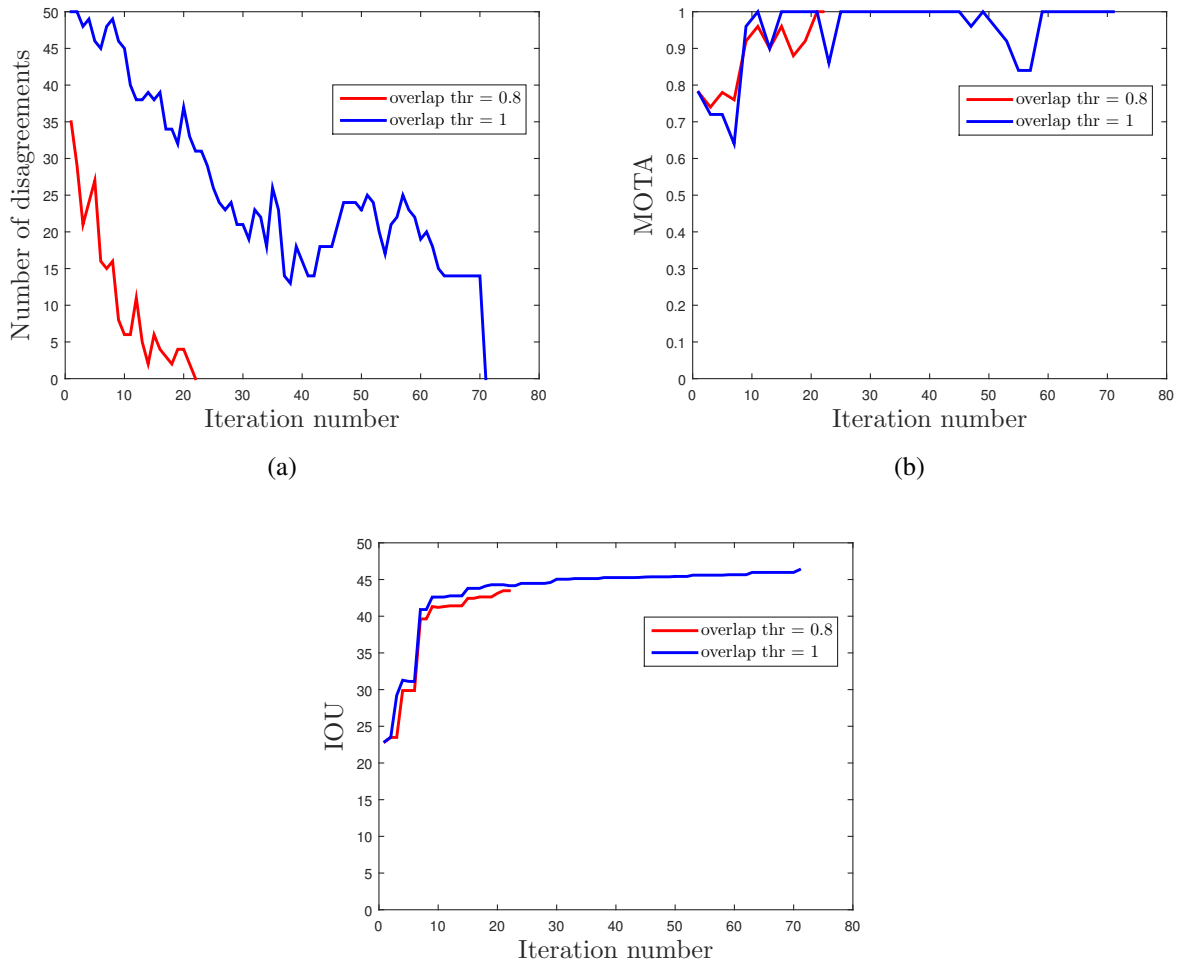


Figure 5.3: Number of Disagreements, MOTA and IOU as function of number of iterations. The curves are generated based on a 10-frame segment in TUD-Crossing with 5 persons in the scene. (a) The number of disagreements between tracking and segmentation solutions drops over iterations. The algorithm converges when the two solutions are consistent. (b) The MOTA increases over iterations and reaches the best value at convergence. (c) The IOU (metric detailed in Sec. 5.2.2.2) increases over iterations. Since the segmentation annotations are available in every 10 frame, IOU is evaluated on the one frame in the 10-frame segment which has segmentation annotations.

In each iteration, we check the consistency between solutions of the two subproblems. The dual variable λ' changes, based on the inconsistent parts among F^0 and F^1, thus adjusting F^0 and F^1 accordingly to make them to be more and more consistent. Suppose in some iteration, boxes f_k are selected for target k by the tracking subproblem, but the segmentation subproblem selects another set of boxes. Then the corresponding element in λ' will increase, such that the penalty of selection of f_k by the tracking subproblem would increase and the penalty of selection of f_k by the segmentation subproblem would decrease. When F^0 and F^1 achieve agreement, λ' will not change and the optimal solution is found.

The spatial constraint described in Section 4.3.3 replaces the non-maximum suppression step in object detection methods and penalizes tracks that highly overlap. When two bounding boxes are highly overlapping, it adds cost to both observation nodes that are involved or the one with lower detection score. In some cases, this scheme leads to inaccurate tracks since it would push both tracks away, no matter if any of the tracks are actually correct or the detection score may not be very accurate. However, we can now utilize the segmentation results to make better decision on the spatial constraint. Assume that from the tracking results in iteration $p-1$, a box $\mathbf{y}_k$ is selected for target k. If the overlap between $\mathbf{y}_k$ and any box in other tracks is larger than 50%, and no pixel in $\mathbf{y}_k$ is labelled as target k from the segmentation results, there is a large chance that box $\mathbf{y}_k$ does not correspond to target k. Thus the cost of the observation node corresponding to $\mathbf{y}_k$ is updated as in Eq. 4.15. In this way, box $\mathbf{y}_k$ will introduce larger penalty and be less likely selected in tracking in iteration p. However, if there are pixels in $\mathbf{y}_k$ labelled as target k, then the observation node corresponding to $\mathbf{y}_k$ will not be penalized, no matter if it has a large overlap with other boxes. Note that the segmentation results are considered along with the tracking results, so the spatial constraint introduces penalty only if a box is too close to another box and is not supported by the segmentation results. This happens when two targets are close to each other, and the track of one target incorrectly jumps to the other target. On the contrary, when one target is occluded by another

target, even though their boxes are close, they both have supporting pixels from the segmentation results, therefore spatial constraint is not applicable.

Due to the dense and overlapping candidate boxes used in our approach, we observe it is not necessary to have F^0 and F^1 to be exactly the same for convergence. In most cases, the results in early iterations are already good enough, though some boxes found by the two subproblems may shift a little. In our experiments, boxes returned by the two subproblems are considered consistent if their overlap is larger than 0.8 and the corresponding element in λ' would not be updated. This greatly reduces the number of iterations to solve the Lagrangian dual problem, with almost no performance loss. As shown in Figure 5.3(a), when overlap threshold of 0.8 is used, the number of disagreements drops more quickly compared to that case when overlap threshold is 1. The number of iterations to solve the Lagrangian dual problem is reduced by more than three times. Meanwhile, the performance remains almost the same as illustrated in Figure 5.3(b) and 5.3(c). Coupling tracking and segmentation lead to both better tracking and better segmentation results as demonstrated in experiments. It can also be observed in Figure 5.3 that both MOTA and IOU are increasing over iterations.

In summary, on one hand, the object tracks provide strong high-level guidance for target/background segmentation. On the other hand, segmentation helps resolve typical difficulties encountered in multiple target tracking in a couple of ways. First, in traditional tracking-by-detection approach, the tracking results highly depend on the detection performance. Miss-detections are common especially when there is occlusion. So special scheme, such as dummy nodes in network flow, needs to be designed in order to handle them. However, our approach does not rely on pre-trained object detector. We assume densely sampled candidate boxes instead of sparse detection boxes, so the tracker is able to infer temporal consistency between frames naturally. In addition, when target gets occluded, its visible part is segmented correctly even though its overall appearance score may be low. The segmentation results guide tracker to find correct box for the target. Second, the seg-

mentation result provides more information about target location and target identity. Therefore, it helps tracker avoid drifting and ID-switch.

5.2 Experiments

In our evaluation, we focus on tracking humans, due to its importance. But our method can be used for tracking any object. We evaluate our proposed TINF tracker and the proposed approach that couples multiple target tracking and segmentation on a set of standard multiple target tracking sequences. Along with tracking, we also provide both segmentation and detection results on a few sequences.

5.2.1 Experimental Setup

To initialize the target, similar to [102, 22], we use manual annotation. We annotate four initial bounding boxes for each object entering the scene. We also report results where targets are initialized automatically using a pre-trained object detector. For manual annotation, the target is initialized only once and there is no re-initialization of targets. We use histogram-of-oriented gradient [26] and color histogram [29] as our features. We found the combination of both features to be important. HOG captures the edge information of target and is helpful in detecting target from the background, while color histogram is a video specific feature and helps in distinguishing different targets from each other. The sequence is divided into segments of 20 frames each. At the end of each temporal span we check if a track is valid or not by comparing its appearance score from structural SVM ($\mathbf{w}_k^T \phi(\mathbf{x}^t, \mathbf{y}_k^t)$) with a pre-defined threshold. If the track is valid then it is used to update the model. When a target is close to the scene border and its velocity is towards outside of the scene, that target is treated as exiting the scene and the algorithm stops tracking that

target. In this way, our approach is able to handle a variable number of targets in the scene.

5.2.2 Experimental Results

In this section, we conduct three sets of experiments. First we compare our approach with the state-of-the-art methods on publicly available sequences. For those sequences, where the object detection performs well, excellent results are already reported. However, we show that, using our approach, one can further improve the performance. Second, we evaluate our approach on two new sequences of [28] where targets experience heavy articulation and we show that we can significantly improve the performance of data-association based trackers as well as online trackers. Third, we test our approach on the popular and complex MOT16 Benchmark. *Parking Lot 1* [71], *Parking Lot 2* [72], *TUD Crossing* [5], *TUD-Stadtmitte* [4] and PET [34] are the five publicly available sequences used in our experiments and the two new sequences are called *Running* and *Dancing*.

5.2.2.1 Tracking

To quantitatively evaluate the tracking performance of our approach, both popular CLEAR MOT metrics [11] and Trajectory Based Metrics (TBM) [88] are employed. CLEAR metrics (MOTA-MOTP) consider the entire video as a whole, while TBM consider the behavior of each track separately. Each of these metrics captures different characteristics of a tracker and it is important to look at both of them, while comparing different tracking algorithms to better capture strength and weakness of each tracker. MOTA considers the number of misses, false positives and mismatches; while MOTP measures the estimated object locations accuracy. MT, ML and IDS respectively are percentage of mostly tracked trajectories, percentage of mostly lost trajectories and number of identity switches.

Table 5.1: Quantitative tracking results comparison of our methods ("TINF" and "TINF + Seg") with competitive approaches of LPD [75], LDA [67], DLP [41], H2T [86], GMCP [98], PF [16], SegTrack [51], CET [5], DCT [6], GOG [57], STRUCK [36] and SPOT [102] using tracking metrics.

Dataset	Method	MOTA	MOTP	MT	ML	IDS
Running	CET	46.3	50.8	0.67	0	0
	DCT	37.6	50.4	0	0	0
	GOG	3	69.5	0	1	0
	SPOT	66.1	66.2	0.67	0	0
	STRUCK	79.9	64.3	1	0	0
	TINF	98.7	66.5	1	0	0
	TINF + Seg	**99.1**	**68.3**	**1**	**0**	**0**
Dancing	CET	36.6	62	0.57	0	64
	DCT	36.3	63.6	0	0.14	81
	GOG	24.9	64	0	0.14	96
	SPOT	55.4	65.9	0.43	0	16
	STRUCK	69.1	67.1	0.71	0.14	9
	TINF	89.9	**65.9**	0.86	0	1
	TINF + Seg	**91.2**	65.7	**0.86**	**0**	**0**
Parking Lot 1	LPD	89.3	77.7	-	-	-
	GMCP	90.4	74.1	-	-	-
	H2T	88.4	81.9	0.78	0	21
	TINF	90.7	**69.3**	0.86	0	3
	TINF + Seg	**91.5**	67.4	**0.86**	**0**	**0**
Parking Lot 2	CET	71.7	55.8	0.6	0	59
	DCT	73.6	56.5	0.8	0	48
	GOG	48.3	59.8	0.2	0.1	96
	TINF	89.3	66.3	1	0	0
	TINF + Seg	**90.5**	**68.7**	**1**	**0**	**0**
TUD Crossing	SegTrack	59.2	73.1	0.67	0	8
	PF	84.3	71	-	-	2
	GMCP	91.6	75.6	-	-	0
	TINF	92.9	**69.2**	1	0	0
	TINF + Seg	**93**	68.2	**1**	**0**	**0**
TUD Stadtmitte	SegTrack	68	55.9	0.6	0	3
	GMCP	77.7	63.4	-	-	0
	TINF	81.6	75.4	0.8	0	0
	TINF + Seg	**83.8**	**78.7**	**0.8**	**0**	**0**
PETS	SegTrack	85.3	77.5	1	0	9
	LDA	90	75	0.89	-	6
	DLP	91	70	-	-	5
	GMCP	90.3	69	-	-	8
	TINF	90.4	63.1	0.95	0	3
	TINF + Seg	**92.5**	**68.2**	**0.95**	**0**	**0**

First, we evaluate and compare the proposed TINF tracker with two main sets of trackers: data-association based trackers and online trackers. On sequences for which no other tracking results are reported, we compare our method with three data-association based trackers for which we have access to their code, CET [5], DCT [6] and GOG [57]. We use Deformable Part based model [33] as the human detector. The input to the data-association methods is the DPM output with different thresholds ranging from -1 to 0. We agree that these trackers have parameters to tune to achieve the best performance for each sequence. However, we stayed with the default parameter suggested by the authors and the only parameter we changed was the human detector threshold. The numbers reported are for a threshold that gives us the best performance. In addition to these three trackers, we quantitatively compare our results with other trackers which have used the same sequences in their experiments. For online discriminative learning-based trackers, we selected STRUCK [36] as well as structure preserve multi-object tracking (SPOT) approach [102]. For STRUCK, we train one structured SVM per target given the annotation of humans in the first frame. For SPOT, the manual annotation is used to initialize the tracking. Whenever a new object enters the scene we re-initialize the tree to get the track for the new target. In SPOT the spatial relationship between the targets are modeled during tracking. This model is updated according to a weight, γ, for every frame. The weight was set originally to 0.05 and we found the weight to be important in final results. The reported results for SPOT are based on the best value that we found for γ. The results comparison is shown in Table 5.1.

The results of our proposed approach that couples multiple target tracking and segmentation are shown in Table 5.1, denoted as "TINF + Seg". The coupled approach achieves better tracking results compared to TINF. In particular, the number of ID-switches is substantially reduced compared to other methods and TINF.

MOT16 Benchmark. We also test our approach on the popular MOT16 Benchmark [50], which is a standardized benchmark for evaluating multiple object tracking. It contains 7 test sequences

and is challenging due to sequences' diversity. The benchmark includes sequences with different crowd density levels, captured by moving or static camera, captured from different viewpoints and under different whether conditions (such as sunny, cloudy and night conditions).

Table 5.2: Tracking performance comparison on MOT16 Benchmark.

Method	MOTA	MOTP	FP	FN	MT	ML	IDS
[63]	52.5	78.8	4407	81223	0.19	0.35	910
[12]	59.8	79.6	8698	63245	0.25	0.23	1423
[95]	66.1	79.5	5061	55914	0.34	0.21	805
[64]	47.2	75.8	2,681	92,856	0.14	0.42	774
Ours	57.6	77.9	12121	64401	0.3	0.22	733

Due to the large number of targets, human detection is used to automatically initialize targets. The results comparison is shown in Table 5.2. We compare our results with other published on-line trackers that use non-standard detections [63, 12, 95] as well as a top performer that uses the standard detections [64]. All better results reported on MOT16 use deep learning based human detection or deep learning based data association. Considering that our approach does not need training and does not involve deep learning features, the results are quite competitive. In addition, our approach achieves low number of ID-switches compared to most state-of-the-art methods. In particular it is interesting to mention that our approach, using simple hand-crafted features, out-performs the top performer which uses standard publicly available detections along with powerful deep pipeline. Finally the large number of FPs in our approach is mainly due to the way we sample dense candidates. This sometimes leads to inaccurate bounding boxes. (The number of FPs will significantly reduce if we lower the overlap threshold for computing the metrics.)

5.2.2.2 Segmentation

Besides improving the tracking performance, our proposed dual decomposition based approach is able to track multiple targets with pixel-level target/background labeling. In order to evaluate the segmentation performance, we use the segmentation annotations for TUD-Crossing from [38] and manually annotate pixel-level target masks every 10 frames in the other sequences. The segmentation annotations will be released to facilitate future research in this area.

Table 5.3: A quantitative comparison of segmentation results of our method with competitive approaches in Milan et al. [51] and Horbert et al. [38].

Dataset	Method	Identity -based IOU	Overall err.	Avg. err.	Over -seg.
PETS	[51]	54.82	0.78	40.08	1.65
	Seg Only	19.51	1.68	66.86	1
	TINF + Seg	**73.51**	**0.43**	**17.79**	**1**
TUD Crossing	[51]	25.35	6.68	63.87	2.23
	[38]	46.50	4.13	35.88	3.23
	Seg Only	15.64	7.96	71.85	1
	TINF + Seg	**55.36**	**3.88**	**26.93**	**1**
TUD Stadtmitte	[51]	27.33	6.10	48.59	1.09
	Seg Only	18.87	6.85	56.48	1
	TINF + Seg	**41.62**	**3.35**	**23.65**	**1**
Parking Lot 1	**Seg Only**	20.97	5.12	49.38	1
	TINF + Seg	**68.27**	**1.36**	**20.57**	**1**
Parking Lot 2	**Seg Only**	14.79	8.91	59.54	1
	TINF + Seg	**58.66**	**4.94**	**26.09**	**1**
Running	**Seg Only**	24.67	5.35	58.56	1
	TINF + Seg	**67.18**	**2.31**	19.57	1
Dancing	**Seg Only**	15.5	12.93	67.8	1
	TINF + Seg	**58.17**	**7.88**	**17.33**	**1**

For evaluation, the segments are optimally assigned to ground truth masks and multiple segments can be assigned to the same ground truth mask (pixel-wise labeled segmentation). Following met-

rics are used for evaluation. Identity-based IOU is the average intersection-over-union overlap with target identity information incorporated. Traditional IOU used in video segmentation evaluation [47] computes the mean IOU of foreground regions over all frames. However, it has no notion of target identities. Therefore, in order to better evaluate the segmentation performance for multiple targets, we extend the traditional foreground IOU to **identity-based IOU**. Identity-based IOU computes the intersection-over-union overlap between ground truth mask and segments assigned to it for every target in every frame and then takes the average over all of them. Overall error is the percentage of wrongly labelled pixels, while average error computes the percentage of misclassified pixels per ground truth mask. Over-segmentation counts the number of segments merged to cover the ground truth masks.

We compare the above four metrics with [51][1] and [38][2] in Table 5.3. The proposed approach achieves much higher identity-based IOU and much lower overall error as well as average error compared to previous methods. "TINF + Seg" outperforms "Seg Only", by a large margin, demonstrating that incorporating tracking leads to more accurate segmentation results. Some qualitative results are shown in Figure 5.5. Note that targets are segmented and tracked correctly even when being occluded or when they are close to other targets.

Moreover, we show number of extracted objects with varying threshold α on ratio of correctly labelled pixels per ground truth mask in Figure 5.4. An object is extracted if more than α of its ground truth mask is correctly covered. On all the seven sequences, our approach ("TINF + Seg") is able to extract more objects for all different thresholds compared to previous methods and "Seg Only".

[1]We test the code available on the author's website with default parameters on TUD-Crossing. The results on the other two sequences are obtained from the author.

[2]Note that the identity-based IOU of Horbert et al.'s [38] results is computed using the segmentation results provided by the author, while the IOU reported in [38] is the traditional foreground IOU without notion of target identities.

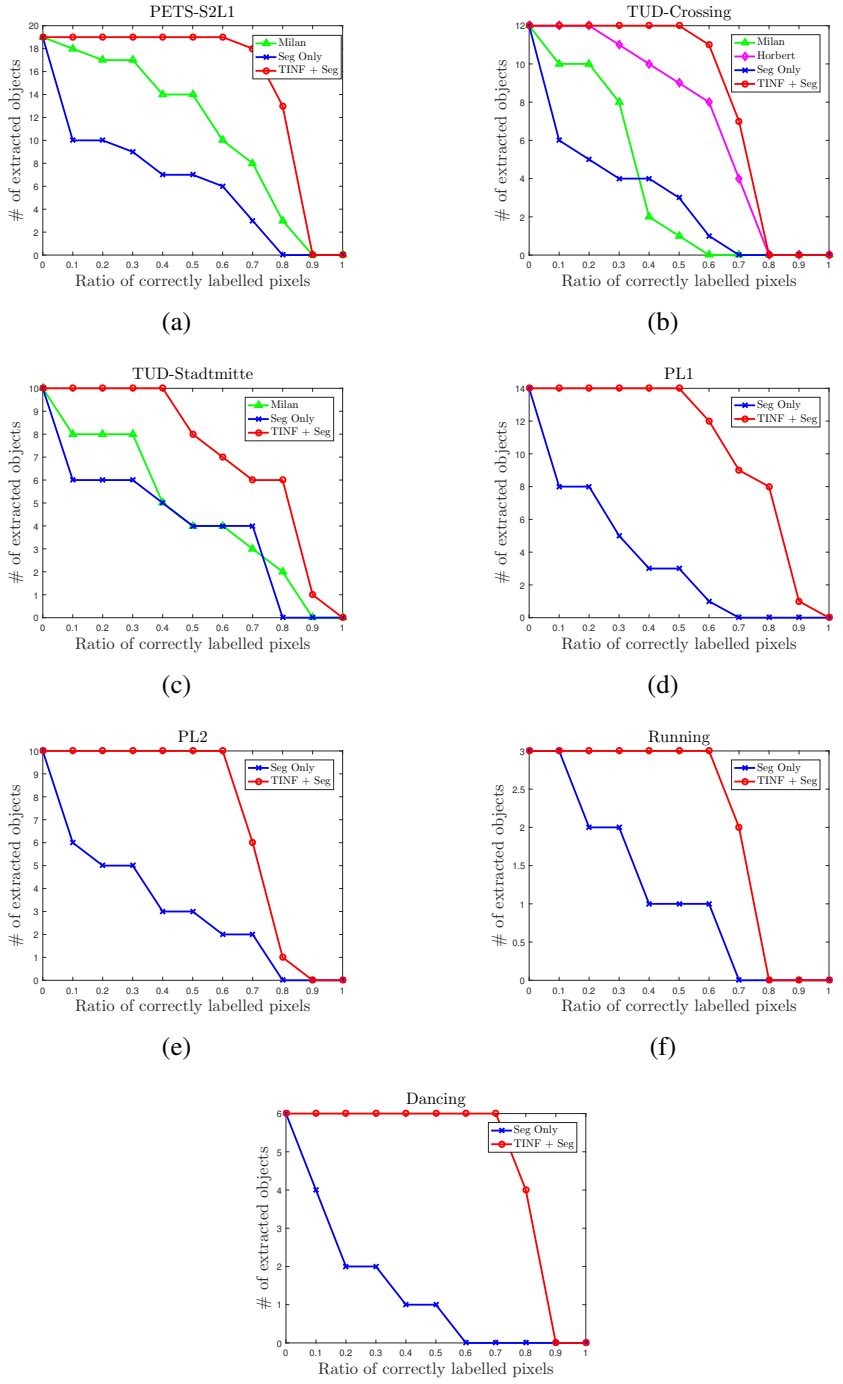


Figure 5.4: The curves show the number of extracted objects as a function of correctly labeled pixels per ground truth mask for different sequences.

Figure 5.5: Examples of segmentation and tracking results on PETS-S2L1, TUD-Crossing, TUD-Stadtmitte and MOT16 (from top row to bottom row). Each target is shown by a unique color.

Since MOT16 Benchmark is designed for evaluating multiple object tracking performance, there are no segmentation annotations available. Qualitative results on one sequence in MOT16 Benchmark are shown in Figure 5.5.

5.2.2.3 Detection

We also present the detection performance of our proposed approach. The comparison with DPM [33] is shown in Table 5.4. Our detector is much simpler compared to DPM, while coupled with segmentation and data association, it can achieve better performance on almost all the sequences.

Table 5.4: This table shows the detection performance comparison between our detector and DPM [33] in terms of average precision.

Seq	PL1	PL2	PETS	TUD Crossing	TUD Stadmitte
DPM	87.61	74.61	68.54	85.95	77.62
Ours	88.12	81.77	84.23	83.78	79.99

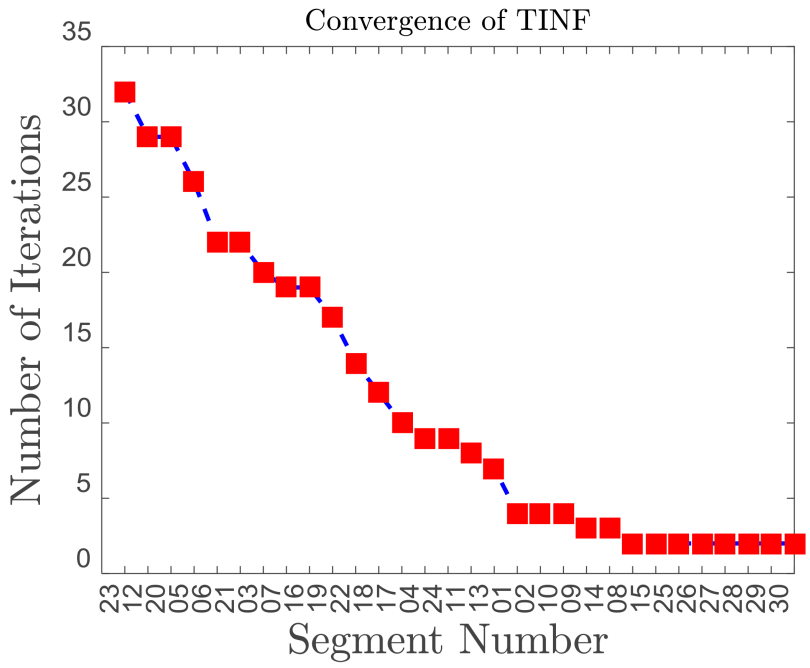
In addition, we evaluate our approach on MOT17DET Benchmark. The detection performance comparison is summarized in Table 5.5. With the help from tracking and segmentation, our approach outperforms DPM [33] and Faster R-CNN [60] on the videos of complex scenes.

Table 5.5: Detection performance comparison with DPM [33] and Faster R-CNN [60] on MOT17DET Benchmark.

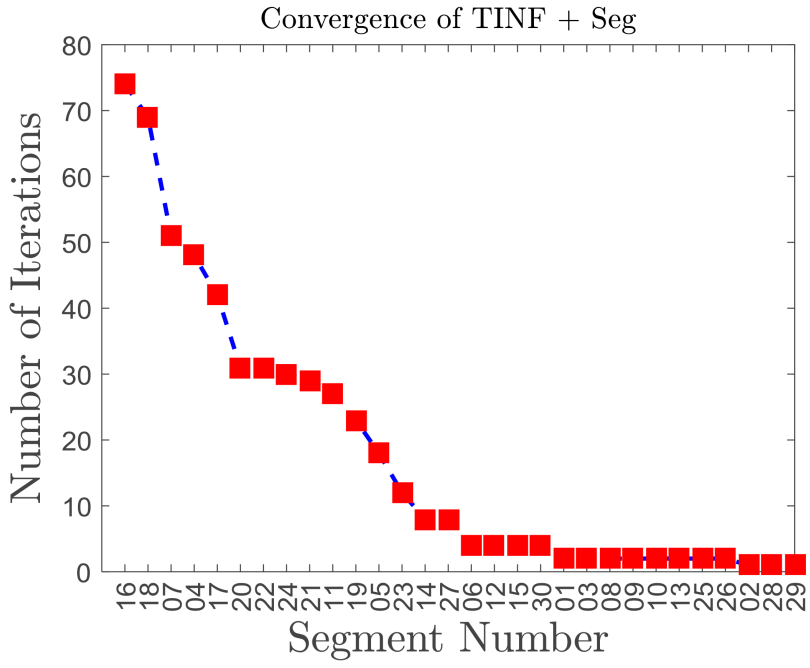
Method	AP	Prec.	Rec.	TP	FP	FN
DPM	0.61	64.8	68.1	78007	42308	36557
Faster R-CNN	0.72	89.8	77.3	88601	10081	25963
Ours	0.74	89.3	83.4	95506	11435	19058

5.2.3 Convergence

In Figure 5.6, we demonstrate the convergence of the proposed TINF and TINF + Seg trackers on PL2 sequence. The horizontal axes shows the segment number in PL2 sequence and the vertical axes represents the number of iterations taken for convergence.



(a)



(b)

Figure 5.6: The figures show convergence of TINF and TINF + Seg on PL2 sequence.

The number of iterations taken for convergence varies depending on the complexity of the segments. For example, for TINF + Seg, it takes only a few iterations to converge for segments near the beginning or the end of PL2 sequence, since the scene is simpler and it is easy to reach agreement between tracking and segmentation results. While some segments in middle of the PL2 sequence take 40 to 75 iterations to converge. That is because the scene is more complex, there are more interacting targets and a lot of occlusions.

5.3 Summary

In this chapter, we present a novel framework that combines two main components of most existing trackers, detection and data association, along with segmentation in a single framework. The three tasks are closely related, and solving one helps improve the others. Detection and data association are combined through a structured learning framework, using a novel network flow graph. Additionally, the online discriminative tracking algorithm and segmentation are jointly optimized using dual decomposition, which leads to more accurate segmentation results and also helps resolve typical difficulties in tracking, such as occlusion handling, ID-switch and track drifting. Moreover, more detailed representation of targets - pixel-level target foreground labeling, is obtained rather than coarse bounding boxes.

CHAPTER 6: CONCLUSION

In this dissertation, we address three fundamental and related problems in computer vision: human action detection, human tracking and segmentation in video. They have a variety of applications such as video surveillance, sports video analysis, video retrieval and self-driving car. Though they are fundamental problems and have been attracting significant attention from the computer vision community, these problems are extremely challenging in realistic video. We propose novel approaches to resolve typical difficulties in these problems.

6.1 Summary

First, we propose spatiotemporal deformable part models for action detection. Actions are treated as spatiotemporal patterns and a deformable part model is generated for each action from a collection of examples. For each action model, the most discriminative 3D subvolumes are automatically selected as parts and the spatiotemporal relations between their locations are learned. By focusing on the most distinctive parts of each action, our models adapt to intra-class variation and show robustness to clutter. Extensive experiments on several video datasets demonstrate the strength of spatiotemporal DPMs for classifying and localizing actions.

Second, we formulate multiple target tracking in a framework where the detection and data-association are performed simultaneously. Our method allows us to overcome the confinements of data association based MOT approaches; where the performance is dependent on the object detection results provided at input level. At the core of our method lies structured learning which learns a model for each target and infers the best location of all targets simultaneously in a video clip. The inference of our structured learning is done through a new Target Identity-aware Net-

work Flow (TINF), where each node in the network encodes the probability of each target identity belonging to that node. The proposed Lagrangian relaxation optimization finds the high quality solution to the network. During optimization a soft spatial constraint is enforced between the nodes of the graph which helps reducing the ambiguity caused by nearby targets with similar appearance in crowded scenarios. We show that automatically detecting and tracking targets in a single framework can help resolve the ambiguities due to frequent occlusion and heavy articulation of targets. Our experiments involve challenging yet distinct datasets and show that our method can achieve results better than the state-of-art.

Finally, we propose a novel tracker that simultaneously solves three main problems: detection, data association and segmentation. This is especially important because the output of each of those three problems are highly correlated and the solution of one can greatly help improve the others. The proposed algorithm consists of two main components: structured learning and Lagrange dual decomposition. The first component - structured learning based tracker is achieved by TINF. The second component is Lagrange dual decomposition, which combines the structured learning tracker with a segmentation algorithm. For segmentation, multi-label Conditional Random Field (CRF) is applied to a superpixel based spatio-temporal graph in a segment of video, in order to assign background or target labels to every superpixel. We show how the multi-label CRF is combined with the structured learning tracker through our dual decomposition formulation. This leads to more accurate segmentation results and also helps better resolve typical difficulties in multiple target tracking, such as occlusion handling, ID-switch and track drifting. The experiments on diverse and challenging sequences show that our method achieves superior results compared to competitive approaches for detection, multiple target tracking as well as segmentation.

6.2 Future Work

This section explores some of the possible directions for future work.

In the spatiotemporal deformable part based action detection model, simple HOG-like feature is used to demonstrate the effectiveness of the proposed model. With the fast growing of powerful deep learning models, a natural direction for future work would be to integrate the SDPM framework with deep learning based features. Another line of research for future work is to extend SDPM model to solve interaction detection problem, by treating the whole interaction as root and modeling action performed by each person as a part.

For the target identity-aware network flow based tracker, its performance suffers on some sequences due to the lack of an effective mechanism to terminate and re-initialize a track when the track drifts. This would lead to both false positive and false negative at the same time for that track. One direction for future work is to explore automatic ways to terminate and re-initialize tracks to avoid drift. This will allow one to utilize the proposed algorithm in scenarios where the camera angle is low and frequent long-term intra object occlusions occur. These sequences are not common in surveillance scenarios. However, recent multi-object tracking dataset contains these types of sequences.

For the multiple target detection, tracking and segmentation work, the tracking and segmentation are purely based on non deep learning based features. It is not robust enough to handle complex and dynamic scenes well. It is important to explore the use of more powerful discriminative features, such as deep learning based features, to further improve the performance. In addition, a regressor can be added on the top to improve bounding box accuracy.

LIST OF REFERENCES

[1] Ibm ilog cplex optimizer, `www.ibm.com/software/integration/optimization/cplex-optimizer`.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 2012.

[3] Z. K. andKrystian Mikolajczyk and J. Matas. Tracking-Learning-Detection. In *PAMI*, 2010.

[4] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *CVPR*, 2010.

[5] A. Andriyenko and K. Schindler. Multi-target Tracking by Continuous Energy Minimization. In *CVPR*, 2011.

[6] A. Andriyenko, K. Schindler, and S. Roth. Discrete-Continuous Optimization for Multi-Target Tracking. In *CVPR*, 2012.

[7] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, 2014.

[8] Y. Bar-Shalom, T. Fortmann, and M. Scheffe. Joint probabilistic data association for multiple targets in clutter. In *Information Sciences and Systems*, 1980.

[9] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple Object Tracking Using K-Shortest Paths Optimization. In *PAMI*, 2011.

[10] K. Bernardin and R. Stiefelhagen. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*, 2008.

[11] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008.

[12] A. Bewley, G. Zongyuan, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *ICIP*, 2016.

[13] C. Bibby and I. Reid. Real-time tracking of multiple occluding objects using level sets. In *CVPR*, 2010.

[14] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[15] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001.

[16] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle Filter. In *ICCV*, 2009.

[17] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011.

[18] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. *ECCV*, 2010.

[19] A. Butt and R. Collins. Multi-target Tracking by Lagrangian Relaxation to Min-Cost Network Flow. In *ICCV*, 2013.

[20] A. A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *CVPR*, 2013.

[21] L. Cao, Z. Liu, and T. S. Huang. Cross-dataset action detection. In *CVPR*, 2010.

[22] S. Chen, A. Fern, and S. Todorovic. Online multi-person tracking-by-detection from a single, uncalibrated camera. In *CVPR*, 2014.

[23] K. Cheng-Hao, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010.

[24] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, , and Y. Singer. Online passive-aggressive algorithms. In *Journal of Machine Learning Research*, 2006.

[25] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[26] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[27] A. Dehghan, S. Modiri, and M. Shah. GMMCP-Tracker:Globally Optimal Generalized Maximum Multi Clique Problem for Multiple Object Tracking. In *CVPR*, 2015.

[28] A. Dehghan, Y. Tian, P. H. Torr, and M. Shah. Target identity-aware network flow for online multiple target tracking. In *CVPR*, 2015.

[29] J. Domke and Y. Aloimonos. Deformation and viewpoint invariant color histograms. In *BMVC*, 2006.

[30] M. D. B. et al. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009.

[31] S.-I. Y. et al. The Solution Path Algorithm for Identity-Aware Multi-Object Tracking. In *CVPR*, 2016.

[32] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, 2008.

[33] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010.

[34] J. Ferryman and A. Shahrokni. Dataset and challenge. *International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[35] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, 2006.

[36] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[37] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[38] E. Horbert, K. Rematas, and B. Leibe. Level-set person segmentation and tracking with multi-region appearance models and top-down shape information. In *ICCV*, 2011.

[39] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *ICCV*, 2009.

[40] G. Karakostas. Faster approximation schemes for fractional multicommodity flow problems. In *ACM-SIAM*, 2002.

[41] A. K. K.C. and C. D. Vleeschouwer. Discriminative Label Propagation for Multi-Object Tracking with Sporadic Appearance Features. In *ICCV*, 2013.

[42] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005.

[43] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV*, 2007.

[44] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[45] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011.

[46] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[47] F. Li, T. Kim, A. Humayun, D. Tsai, and J. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[48] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009.

[49] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012.

[50] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[51] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid. Joint tracking and segmentation of multiple targets. In *CVPR*, 2015.

[52] D. Mitzel, E. Horbert, A. Ess, and B. Leibe. Multi-person tracking with sparse detection and continuous segmentation. *ECCV*, 2010.

[53] C. nam Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009.

[54] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[55] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 2008.

[56] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In *CVPR*, 2011.

[57] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

[58] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012.

[59] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automated Control*, 1996.

[60] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[61] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[62] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, 2004.

[63] R.Sanchez-Matilla, F.Poiesi, and A.Cavallaro. Online multi-target tracking with strong and weak detections. In *ECCV*, 2016.

[64] A. A. A. Sadeghian and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *CVPR*, 2017.

[65] S. Satkin and M. Hebert. Modeling the temporal extent of actions. *ECCV*, 2010.

[66] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004.

[67] A. V. Segal and I. Reid. Latent Data Association: Bayesian Model Selection for Multi-target Tracking. In *ICCV*, 2013.

[68] H. J. Seo and P. Milanfar. Action recognition from one example. *PAMI*, 2011.

[69] K. Shafique and M. Shah. A noniterative greedy algorithm formultiframe point correspondence. In *PAMI*, 2005.

[70] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Multi-commodity network flow for tracking multiple people. In *PAMI*, 2013.

[71] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based Multiple-Person Tracking with Partial Occlusion Handling. In *CVPR*, 2012.

[72] G. Shu, A. Dehghan, and M. Shah. Improving an Object Detector and Extracting Regions using Superpixels. In *CVPR*, 2013.

[73] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual Tracking: an Experimental Survey. In *PAMI*, 2013.

[74] P. Strandmark and F. Kahl. Parallel and distributed graph cuts by dual decomposition. In *CVPR*, 2010.

[75] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele. Learning people detectors for tracking in crowded scenes. In *ICCV*, 2013.

[76] D. Tran and A. Sorokin. Human activity recognition with metric learning. *ECCV*, 2008.

[77] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. In *JMLR*, 2005.

[78] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[79] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[80] H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *CVPR*, 2011.

[81] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[82] S. Wang, H. Lu, F. Yang, and M. Yang. Superpixel tracking. In *CVPR*, 2011.

[83] X. Wang, E. Turetken, F. Fleuret, and P. Fua. Tracking interacting objects optimally using integer programming. In *ECCV*, 2014.

[84] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *CVPR*, 2008.

[85] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. Jots: Joint online tracking and segmentation. In *CVPR*, 2015.

[86] 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.

[87] S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. In *CVPR*, 2007.

[88] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR*, 2006.

[89] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *IJCV*, 2007.

[90] Y. Wu, J. Lim, and M. H. Yang. Online Object Tracking: A Benchmark. In *CVPR*, 2013.

[91] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *CVPR*, 2012.

[92] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, 2005.

[93] Z. Yin and R. T. Collins. Shape constrained figure-ground segmentation and tracking. In *CVPR*, 2009.

[94] D. M. W. Z. Yu, Shoou-I. and A. Hauptmann. The solution path algorithm for identity-aware multi-object tracking. In *CVPR*, 2016.

[95] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *ECCV*, 2016.

[96] Y. Y. X. L. Yu, Shoou-I. and A. G. Hauptmann. Long-term identity-aware multi-person tracking for surveillance video summarization. In *arXiv:1604.07468*, 2016.

[97] A. R. Zamir, A. Dehghan, and M. Shah. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. In *ECCV*, 2012.

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

[99] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013.

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

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

[102] L. Zhang and L. Maaten. Structure preserving object tracking. In *CVPR*, 2013.

[103] L. Zhang and L. van der Maaten. Structure Preserving Object Tracking. In *CVPR*, 2013.