Spatiotemporal Graphs for Object Segmentation and Human Pose Estimation in Videos

2016

Dong Zhang

University of Central Florida

SPATIOTEMPORAL GRAPHS FOR OBJECT SEGMENTATION AND HUMAN POSE ESTIMATION IN VIDEOS

by

DONG ZHANG
B.E. Zhejiang University, 2007

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

Summer Term
2016

Major Professor: Mubarak Shah

ABSTRACT

Images and videos can be naturally represented by graphs, with spatial graphs for images and spatiotemporal graphs for videos. However, for different applications, there are usually different formulations of the graphs, and algorithms for each formulation have different complexities. Therefore, wisely formulating the problem to ensure an accurate and efficient solution is one of the core issues in Computer Vision research. We explore three problems in this domain to demonstrate how to formulate all of these problems in terms of spatiotemporal graphs and obtain good and efficient solutions.

The first problem we explore is video object segmentation. The goal is to segment the primary moving objects in the videos. This problem is important for many applications, such as content based video retrieval, video summarization, activity understanding and targeted content replacement. In our framework, we use object proposals, which are object-like regions obtained by low-level visual cues. Each object proposal has an object-ness score associated with it, which indicates how likely this object proposal corresponds to an object. The problem is formulated as a directed acyclic graph, for which nodes represent the object proposals and edges represent the spatiotemporal relationship between nodes. A dynamic programming solution is employed to select one object proposal from each video frame, while ensuring their consistency throughout the video frames. Gaussian mixture models (GMMs) are used for modeling the background and foreground, and Markov Random Fields (MRFs) are employed to smooth the pixel-level segmentation.

In the above spatiotemporal graph formulation, we consider the object segmentation in only single video. Next, we consider multiple videos and model the video co-segmentation problem as a spatiotemporal graph. The goal here is to simultaneously segment the moving objects from multiple videos and assign common objects the same labels. The problem is formulated as a regulated maximum clique problem using object proposals. The object proposals are tracked in adjacent frames to generate a pool of candidate tracklets. Then an undirected graph is built with the nodes corresponding to the tracklets from all the videos and edges representing the similarities between the tracklets. A modified Bron-Kerbosch Algorithm is applied to the graph in order to select the prominent objects contained in these videos, hence relate the segmentation of each object in different videos.

In online and surveillance videos, the most important object class is the human. In contrast to generic video object segmentation and co-segmentation, specific knowledge about humans, which is defined by a pose (i.e. human skeleton), can be employed to help the segmentation and tracking of people in the videos. We formulate the problem of human pose estimation in videos using the spatiotemporal graph. In this formulation, the nodes represent different body parts in the video frames and edges represent the spatiotemporal relationship between body parts in adjacent frames. The graph is carefully designed to ensure an exact and efficient solution. The overall objective for the new formulation is to remove the simple cycles from the traditional graph-based formulations. Dynamic programming is employed in different stages in the method to select the best tracklets and human pose configurations

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Mubarak Shah, and committee members, Dr. Ulas Bagci, Dr. Guo-Jun Qi and Dr. Hae-Bum Yun, for their guidance throughout the research process. I feel very fortunate for having been part of Center for Research in Computer Vision (CRCV) and for being given the opportunity to be involved in and learn from a wide range of projects. I would like to thank all of the past and present members of CRCV, particularly Dr. Omar Javed, Dr. Omar Oreifej, Dr. Guang Shu, Dr. Haroon Idrees, Dr. Afshin Dehghan, Gonzalo Vaca, Waqas Sultani, Yicong Tian, Nasim Souly, Shayan Modiri Assari, Khurram Soomro, Mahdi Kalayeh, Sarfaraz Hussein, Cherry Place and Tonya LaPrarie for their support, the good times, and the great memories. Lastly, I want to thank my father and mother for their endless support. This journey has been long but I am so thankful for those that assisted me through it.

TABLE OF CONTENTS

LIST OF FIGURES xi

LIST OF TABLES xxiii

CHAPTER 1: INTRODUCTION 1

1.1 Spatiotemporal Graphs, Relational Graphs, and Hypothesis Graphs 3

1.2 Video Object Segmentation 6

1.3 Video Object Co-Segmentation 8

1.4 Human Pose Estimation in Videos 10

1.5 Dissertation Organization 13

CHAPTER 2: LITERATURE REVIEW 14

2.1 Video Segmentation 15

2.1.1 Graph-based Segmentation 15

2.1.2 Clustering Segmentation 17

2.1.3 Motion Segmentation 18

2.2 Video Object Segmentation 20

2.2.1 Supervised Methods 20

2.2.2 Semi-Supervised Methods 21

2.3 Video Object Co-Segmentation 24

2.4 Human Pose Estimation 27

2.4.1 Image-based Methods 27

2.4.2 Video-based Methods 28

2.5 Summary 29

CHAPTER 3: SPATIOTEMPORAL GRAPHS FOR VIDEO OBJECT SEGMENTATION 31

3.1 Introduction 31

3.2 Layered DAG based Video Object Segmentation 34

3.2.1 Unary Edges 36

3.2.2 Binary Edges 38

3.3 Dynamic Programming Solution 39

3.4 Per-pixel Video Object Segmentation 40

3.5 Object Proposal Generation & Expansion 41

3.6 Experiments 43

3.6.1 SegTrack Dataset 43

3.6.2 GaTech Segmentation Dataset 47

3.6.3 Persons and Cars Segmentation Dataset 47

3.7 Summary 48

CHAPTER 4: SPATIOTEMPORAL GRAPHS FOR VIDEO OBJECT CO-SEGMENTATION 51

4.1 Introduction 51

4.2 The Framework 53

4.3 Object Tracklets Generation 54

4.3.1 Efficient Object Proposal Tracking 54

4.3.2 Non-Maximum Suppression for Object Proposal Tracks 55

4.4 Multiple Object Co-Segmentation by Regulated Maximum Weight Cliques 58

4.4.1 Clique Problems 58

4.4.2 Problem Constraints 58

4.4.3 Graph Structure 59

4.4.4 Regulated Maximum Weight Clique Extraction 60

4.5 Experiments 62

4.5.1 MOViCS Dataset 62

4.5.1.1 Experimental Setup 62

4.5.1.2 Comparisons with State-of-the-art Methods 64

4.5.1.3 Advantages of Video Co-Segmentation Method 66

4.5.1.4 Advantages over Single Video Object Segmentation Method . . . 66

4.5.2 Safari Dataset 68

4.6 Summary 71

CHAPTER 5: SPATIOTEMPORAL GRAPHS FOR HUMAN POSE ESTIMATION IN VIDEOS 72

5.1 Introduction 72

5.2 Real Body Parts vs. Abstract Body Parts 76

5.3 Body Part Hypotheses in a Single Frame 77

5.3.1 N-Best Hypotheses 77

5.3.2 DCNNGM Hypotheses 78

5.3.3 Abstract Body Part Hypotheses 80

5.4 Single Part Tracklets 80

5.5 Coupled Part Tracklets 82

5.6 Optimal Pose Estimation using Part Tracklets 85

5.7 Limb Alignment and Refinement 86

5.7.1 Limb Alignment 88

5.7.2 Limb Refinement 89

5.8 Experiments 91

5.8.1 Datasets 91

5.8.2 Evaluation Metrics 92

5.8.3 Comparison of Results 92

5.8.4 Computation Time 106

5.8.5 Limitations and Failure Cases 107

5.9 Summary 109

CHAPTER 6: CONCLUSION AND FUTURE WORK 110

6.1 Conclusion 110

6.2 Future Work 112

LIST OF REFERENCES 114

LIST OF FIGURES

Figure 1.1: Relational Graph vs. Hypothesis Graph. (a) shows the relational graph for a problem. (b) shows the corresponding hypothesis graph. In this example, each entity has three hypotheses and corresponding to the structure of (a), edges are added between the hypotheses which have relationship. 3

Figure 1.2: An Illustration for Video Object Segmentation. 6

Figure 1.3: An Illustration for Video Object Co-Segmentation. 9

Figure 1.4: An Illustration for Human Pose Estimation in Videos. 11

Figure 3.1: Primary object region selection in the object proposal domain. The first row shows frames from a video. The second row shows key object proposals (in red boundaries) extracted by [60]. "?" indicates that no proposal related to the primary object was found by the method. The third row shows primary object proposals selected by the proposed method. Note that the proposed method was able to find primary object proposals in all frames. The results in row 2 and 3 are prior to per-pixel segmentation. In this dissertation we demonstrate that temporally dense extraction of primary object proposals results in significant improvement in object segmentation performance. Please see Table 3.1 for quantitative results and comparisons to state of the art.[Please Print in Color] 32

Figure 3.2: Object proposals from a video frame employing the method in [21]. The left side image is one of the video frames. Note that the monkey is the object of interest in the frame. Images on the right show some of the top ranked object proposals from the frame. Most of the proposals do not correspond to an actual object. The goal of the proposed work is to generate an enhanced set of object proposals and extract the segments related to the primary object from the video. 33

Figure 3.3: The Video Object Segmentation Framework 33

Figure 3.4: Spatiotemporal Graph for Video Object Segmentation. Node “s” and “t” are source and sink nodes respectively, which have zero weights for their edges connecting with other nodes in the graph. The yellow nodes and the green nodes are “beginning nodes” and “ending nodes” respectively and they are paired such that each yellow-green pair represents an object proposal. All the beginning nodes in the same frame are arranged in a layer and the same is the case for the ending nodes. The green edges are the unary edges and red edges are the binary edges. 35

Figure 3.5: Optical Flow Gradient Magnitude Motion Scoring. In row 1, column 1 shows the original video frame, column 2 shows one of the object proposals and column 3 shows dilated boundary of the object proposal. In row 2, column 1 shows the forward optical flow of the frame, column 2 shows the optical flow gradient magnitude map and column 3 shows the optical flow gradient magnitude response for the specific object proposal around the boundary. . . 37

Figure 3.6: Object Proposal Expansion. For each optical flow warped object proposal in frame $i-1$, we look for object proposals in frame i which have high overlap ratios with it. If multiple object proposals have high overlap ratios with the warped proposal, they are merged into a new large object proposal. This process will produce the right object proposal if it is not discovered by in frame i, but will be discovered by our method in frame $i-1$. 42

Figure 3.7: SegTrack dataset results. The regions within the red boundaries are the segmented primary objects. 45

Figure 3.8: Comparison of object region selection methods. The regions within the red boundaries are the selected object regions. "?" means there is no object region selected by the method. Numbers above are the frame indices.[Please Print in Color] 46

Figure 3.9: Object Segmentation Results on GaTech Video Segmentation Dataset. Row 1: orignial frame, Row 2: Segmentation results by the bottom-up segmentation method [40]. Row 3: Video object segmentation by the proposed method. The regions within the red or green boundaries are the segmented primary objects. [Please Print in Color] 49

Figure 3.10Sample Results on Persons and Cars Dataset. Please go to http://crcv.ucf.edu for more details. 50

Figure 4.1: (a) Shows the framework of the proposed method. (b) Shows the spatiotemporal graph for video object co-segmentation. In this example, we generate 'object proposal' tracklets for two videos and use weighted nodes to represent them. Edges are built between similar nodes, and there are two types of edges (intra-video edges: red; inter-video edges: orange). In this example, the first two maximal cliques, which have highest weights are obtained ($C1$ and $C2$). $C1$ contains all the segments for 'chicken' from two videos and $C2$ contains all the segments for 'turtle' in video 1. 52

Figure 4.2: Object Proposal Tracking. (a) Shows the similarity matrices between F1 (frame 1) and F2, F2 and F3, and F3 and F4. It also shows an example for tracking a specific object proposal (the 4th in F1): first, the largest item from row 4 of similarity matrix F1 and F2 (the 1st item in this example) is found; then, the largest item from row 1 of similarity matrix F2 and F3 is found; and so on. Note that, only 10 object proposals (the matrices are 10 by 10) are shown in this figure for simplicity, but hundreds of objects proposals are used in the experiments. (b) Shows some object proposal tracks. In this example, several object proposals are generated for frame 31, and the object proposal shown in red box is tracked backward and forward to form a track throughout all the video frames. The same process is repeated for other object proposals (in orange and purple boxes as another two examples). This process is repeated for all the frames. 56

Figure 4.3: Tracklet Splitting. In this example, after the Non-Maximum Suppression, there are T object proposal tracks selected. Track i is shown as an example to generate the object proposal tracklets. There are several adjacent frames which are not very similar compared to other adjacent frame pairs, therefore, they are split and several tracklets are generated (red, orange and purple). . . 57

Figure 4.4: Video Co-Segmentation Results on MOViCS Dataset. Each row shows the results of a video in MOViCS dataset. Column 1 is one original frame from the video; column 2 ('GT') is the ground truth for co-segmentation, red regions correspond to the first object in the video set and green regions correspond to the second object in the video set; column 3 ('Ours') is the results of the proposed method, red and green regions correspond to the first and second objects in the video set and blue region corresponds to the third object; column 4 ('VCS') and 5 ('ICS') are the results of video co-segmentation method from [16] and [50] respectively. Row 1 and 2 are for 'chicken&turtle' video set, row 3-6 are for 'lion&zebra' video set, row 7 and 8 are for 'giraffe&elephant' video set and row 9-11 are for 'tigers' video set. . 63

Figure 4.5: Advantages of the Proposed Video Co-Segmentation Method. Row 1 and row 2 show sample frames from two videos respectively. Row 3 and 4 are the video co-segmentation results of the proposed method for these two videos. Red regions correspond to the first object and green regions correspond to the second object. Row 5 and 6 are the segmentation results of applying the method separately to each video. Blue and dark red regions correspond to the first objects, and pink and orange regions correspond to the second objects. . 65

Figure 4.6: Comparison between the proposed method (VOCS) and Single Video Object Segmentation (SVOS) method ([134]). Three groups of results are shown here. In each of them, the first rows (row 1, 4 and 7) show sample frames from the videos; the second rows (row 2, 5 and 8) show results of the proposed method; and the third rows (row 3, 6 and 9) show results of the single video object segmentation method. For the results, the red regions correspond to the first objects and green regions correspond to the second objects. Since the single video object segmentation method only extract primary objects from the videos, only red regions could be shown in the results. 67

Figure 4.7: The structure of Safari dataset. 69

Figure 4.8: The ground truth and our results on Safari dataset. Row 1 shows one frame from each of the video. Row 2 shows the ground truth annotations, in which each object class is shown in different color. And row 3 shows our results, in which each object class is also shown in different color. Please note that, there is no relationship between the colors of row 2 and row 3. 70

Figure 5.1: An abstract high-level illustration of the proposed method aiming at removing simple cycles from the commonly employed graph optimization framework for video based human pose estimation problem. All of the above graphs are relational graphs for the problems. In (a), a few sample frames of a video are shown. (b) shows the spatiotemporal graph for human pose estimation in videos. Each body part in each frame is represented by a node. Green and blue edges represent relationships between different body parts in the same frame (green ones are commonly used edges in the literature, and blue ones are new edges for symmetric parts we introduced in this dissertation); red edges represent the consistency constraints for the same body part in adjacent frames. Note that this is only an illustration and not all edges are shown. In the 'Abstraction' stage, symmetric parts are combined together, and the simple cycles within each single frame are removed (shown in (c)); and in the 'Association' stage, the simple cycles between adjacent frames are removed (shown in (d)). 73

Figure 5.2: Real body parts vs. abstract body parts. The left side shows a commonly used body part definitions in the literature, and we call these body parts (nodes) 'real body parts', and the graph 'real body part relational graph'. The right side shows the proposed new definition of body parts, basically we combine a pair of symmetric body parts to be one body part, we call these body parts (nodes) 'abstract body parts', since these parts are some abstract concepts of parts, not real body parts, and the graph as 'abstract body part graph'. 74

Figure 5.3: An outline of the proposed method. (a) shows the original video frames; in (b), pose hypotheses in each frame are generated by N-Best method [78] or DCNNGM [12]; in (c), by using the results from (b), real body part hypotheses are generated for each body part in each frame and propagated to the adjacent frames; in (d), real body parts are combined into abstract body parts and the hypotheses are also combined accordingly in order to remove the intra-frame simple cycles (i.e. the simple cycles with blue and green edges in Fig.5.1(b)); in (e), tracklets are generated for abstract body parts (including single body parts and coupled body parts) using the abstract body part hypotheses generated in (d); in (f), the pose hypotheses graph is built, each node is a tracklet corresponding to the abstract body part, and the best pose estimation is obtained by selecting the best hypotheses for the parts from the graph; and in (g), the correct limbs are inferred by limb alignment and refinement schemes. 75

Figure 5.4: An Illustration for DCNNGM Pose Hypotheses Generation. The top row shows the pose estimation results using DCNNGM [12], and the bottom row shows the tracked “right ankle” for the neighboring frames and the combined detection heat-map. In this example, in frame 44, the right leg was not correctly localized; however, in some of its neighboring frames, the right leg was localized correctly. Therefore, by tracking using neighboring frames, we can generate good detection hypotheses for ’right ankle’ and thus a better detection map. 79

Figure 5.5: **Spatiotemporal Graph for Tracklet Generation**. (a) Shows the relational graph for the abstract body part tracklet generation. (b) Shows the tracklet hypothesis graph for single body parts. Each node represents one hypothesis location of the body part in a specific frame, and edges show the similarity between the connected body part hypotheses in adjacent frames. (c) Shows the tracklet hypothesis graph for coupled parts. Each node represents a coupled body part hypothesis, which is the combination of the corresponding symmetric body parts (that is why each node is colored into two halves). The edges represent the similarities between connected coupled body parts in adjacent frames. Note that, (b) and (c) are only illustrations, and for simplicity, not all edges are shown. 82

Figure 5.6: **Spatiotemporal Graph for Pose Estimation in Videos**. (a) is the pose relational graph. Each node represents one abstract body part and edges represent the relationship between the connected body parts. (b) is the pose hypothesis graph. Each node is a tracklet for the part, and edges represent the spatial compatibility of connected nodes. 84

Figure 5.7: An Illustration for Limb Alignment and Refinement. (b) is an illustration for matching a single limb with its reference. $J1, J2$ are the two joints corresponding to the limb detection, and $J1^*$, $J2^*$ are the two joints for the reference limb. (a) shows the limb alignment process: by matching with the limb part configurations from DCNNGM results, the limb part configurations (lower legs) can be inferred correctly. (c) shows the limb refinement process: by matching with the limb configurations from neighboring frames, the wrong limb configuration (please see frame 10, lower legs) can be corrected. 87

Figure 5.8: An Illustration for Limb Alignments and Refinements. “DCNNGM” shows the initial pose estimation; “Ours Initial” shows the results after “Abstraction” and “Association”; “Ours after Alignment” and “Ours after Refinement” show the results after these two steps respectively. 90

Figure 5.9: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from N-Best Dataset. Body parts are shown in different colors. 96

Figure 5.10Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from N-Best Dataset. Body parts are shown in different colors. 97

Figure 5.11Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from HumanEva I Dataset. Body parts are shown in different colors. 98

Figure 5.12Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors. 99

Figure 5.13Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCN-NGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors. 100

Figure 5.14Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCN-NGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors. 101

Figure 5.15Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCN-NGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors. 102

Figure 5.16Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCN-NGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors. 103

Figure 5.17Detailed Comparisons of Results from a Video Sequence. Here we want to show the consistency of the results for the proposed method. Please note the pose estimations for the legs: the proposed method can localize the two ankles accurately and smoothly, while “HPEV” results are also smooth but cannot localize the feet correctly in several frames, and “DCNNGM” results are quite noisy and the performance vary a lot through the frames. 104

Figure 5.18The Detailed Comparisons of the Results Obtained by Symmetric Tracking (Sym. Trk. [85]) and the Proposed Method. This example shows that the body part location estimation could be considered 'correctly' by PCP [26], but would induce larger error using KLE [85]. For example, for the arms in the top figures and legs in the bottom figures, results for both methods satisfy the PCP criteria; however, our method would have much smaller KLE since the estimated body parts are much closer to the ground truth. 107

Figure 5.19Failure Cases. Some of the failure cases are caused by occlusion (1-3), and some are caused by failed detection of joints (4-6). 108

LIST OF TABLES

Table 3.1: Quantitative results and comparison with the state of the art on SegTrack dataset 44

Table 3.2: Quantitative Results on Persons and Cars dataset 47

Table 4.1: Quantitative comparisons with the state of the art on MOViCS dataset. ‘Ours1’ shows results of using a single threshold (0.65) for removing the edges, and ‘Ours2’ shows results of using different thresholds for each video sets (the thresholds are [0.65 0.86 0.45 0.65] for these four video sets respectively.) . . 64

Table 4.2: Quantitative Comparison of Single Video Object Segmentation (SVOS) with Video Object Co-Segmentation (VOCS) 68

Table 4.3: Quantitative results on Safari dataset 69

Table 5.1: Comparisons with the state-of-the-art methods and intermediate stages of the proposed method, on Outdoor Dataset [85] . Note that PCP is an accuracy measure, so the larger the better, with a max of 1; and KLE is an error measure, so the smaller the better. Please refer to Section 5.8.3 for a detailed discussion. 93

Table 5.2: Comparisons with the state-of-the-art methods and intermediate stages of the proposed method, on Human Eva-I Dataset [100]. Note that PCP is an accuracy measure, so the larger the better, with a max of 1; and KLE is an error measure, so the smaller the better. Please refer to Section 5.8.3 for a detailed discussion. 94

Table 5.3: Comparisons with the state-of-the-art methods and intermediate stages of the proposed method, on N-Best Dataset [78]. Note that PCP is an accuracy measure, so the larger the better, with a max of 1; and KLE is an error measure, so the smaller the better. Please refer to Section 5.8.3 for a detailed discussion. 95

CHAPTER 1: INTRODUCTION

Videos contain much more information compared to images, thus it is beneficial to use videos in computer vision applications, such as object segmentation, human pose estimation, tracking, and action recognition. Compared to still images, the most important benefit from videos is the temporal information. It is crucial for object segmentation, pose estimation, etc. However, it is difficult to use this information effectively, and there are two main reasons. First, it is natural to formulate an image- or video-based problems into graph related problems; however, there can be many different formulations for a specific problem, and how to formulate it without losing useful information is important. Second, for video-based graph, there will be many more nodes and edges, thus the complexity of the formulations would be much higher. Therefore, how to select a proper formulation in order to get a computational efficient solution is also important. In this dissertation, we focus on three important computer vision problems which can be formulated as spatiotemporal graphs: video object segmentation, video object co-segmentation, and human pose estimation in videos.

Video object segmentation aims to obtain a pixel-level segmentation for the foreground moving object in a video. Object proposals, which are potential foreground regions, are commonly employed in this problem. There are two important points about the object proposals, we need to keep in mind. First, we need to sample a lot of object proposals for each video frame in order to obtain a high recall rate. Second, we want to keep the selected proposals consistent through all the video frames. Spatiotemporal graph is a perfect solution for this problem, with the object proposals represented by nodes and spatiotemporal relationships between the object proposals represented by edges. The problem can be solved efficiently by dynamic programming.

While video object segmentation can only handle a single video, video object co-segmentation is able to segment multiple videos simultaneously and label the same object class present in different videos with the same label. We also propose a spatiotemporal graph formulation for this problem. In this formulation, the nodes represent object proposal tracklets in the videos, and the edges represent the consistencies between the tracklets within one video or across videos. The solution is to extract multiple maximum cliques from the graph, where each clique corresponds to one foreground object.

Human, as the most important object class in videos, is very difficult to be segmented from the background due to the deformations of body parts. Therefore, the last problem we investigate is human pose estimation in videos, which deals with the estimation of the joint locations of human body in videos. This knowledge can potentially benefit the video object segmentation. We also use spatiotemporal graph for this problem. The joints can be naturally represented by nodes, and the spatiotemporal relationships between the body parts can be modeled by edges. A common problem for this graph formulation is the computational complexity, due to the simple cycles in the graph. Our major contribution is re-formulating the spatiotemporal graph to be a tree, in order to reduce the computational complexity.

In the next Sections, we first introduce some general concepts which are important for this dissertation: spatiotemporal graphs, relational graphs and hypothesis graphs in Section 1.1. Then, we introduce the three problems which we explore in this dissertation: video object segmentation in Section 1.2, video object co-segmentation in Section 1.3, and human pose estimation in videos in Section 1.4.

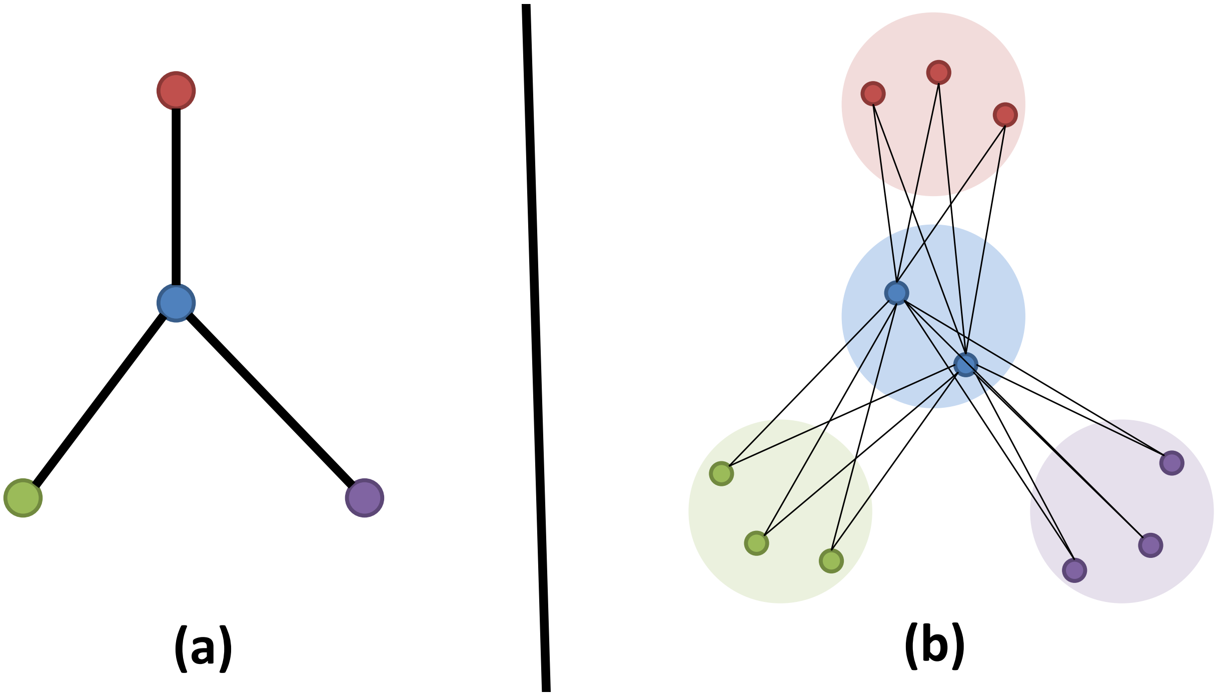


Figure 1.1: Relational Graph vs. Hypothesis Graph. (a) shows the relational graph for a problem. (b) shows the corresponding hypothesis graph. In this example, each entity has three hypotheses and corresponding to the structure of (a), edges are added between the hypotheses which have relationship.

1.1 Spatiotemporal Graphs, Relational Graphs, and Hypothesis Graphs

In the context of computer vision research for videos, *spatiotemporal graphs* are defined as the graphs which are designed to model video-based problems, in which the nodes and edges encode both spatial (pixels within a video frame) and temporal (pixels across video frames) information. The fundamental theme of this dissertation is to explore the applications of spatiotemporal graphs in video object segmentation, video object co-segmentation, and human pose estimation in videos.

In computer vision, and several other disciplines, many problems can be abstracted as follows. Assume there is a set of entities $\mathcal{E} = \{e^i|_{i=1}^N\}$, where each of them can stay in only one of the many states $\mathcal{S} = \{s^k|_{k=1}^M\}$, with the unary scoring functions $\{\Phi(e^i, s^k)|e^i \in \mathcal{E}, s^k \in \mathcal{S}\}$, which gives the likelihood that an entity e^i stays in state s^k. And there is a binary compatibility function for each pair of entities $\{\Psi(e^i, e^j, s^k, s^l)|e^i, e^j \in \mathcal{E}, s^k, s^l \in \mathcal{S}\}$, which represents the compatibility of entity e^i in state s^k and entity e^j in state s^l. The goal is to decide the best states for each entity such that these entities have high unary scores and are also compatible with each other. This problem can be modeled as a graph optimization problem formulated by relational and hypothesis graphs, which is described next.

A relational graph, $G_r = (V_r, E_r)$, represents the relationship of a set of entities which are represented by entity nodes $\{v_r^i\ |_{i=1}^{|V_r|}\}$, and the relationships between pairs of entities are represented by edges E_r. Fig.1.1(a) shows a simple relational graph which has 4 entities and 3 edges to represent their relations. Corresponding to a relational graph G_r, a hypothesis graph, $G_h = (V_h, E_h)$, can be built. Fig.1.1(b) is the hypothesis graph for the relational graph in Fig.1.1(a). For an entity node v_r^i in V_r, a *group* of hypothesis nodes $V_{h(i)} = \{v_{h(i)}^k\ |_{k=1}^{|V_{h(i)}|}\}$ are generated to form the hypothesis graph, so $V_h = \bigcup_{i=1}^{|V_r|} V_{h(i)}$. The hypothesis nodes represent the possible states of each entity, and in this dissertation they represent possible locations of body parts. Hypothesis edges, $E_h = \{(v_{h(i)}^k, v_{h(j)}^l)|v_{h(i)}^k \in V_{h(i)}, v_{h(j)}^l \in V_{h(j)}, (v_r^i, v_r^j) \in E_r\}$, are built between each pair of hypothesis nodes from different *group*s following the structure of G_r. An unary weight, Φ, can be assigned to each hypothesis node, which measures the likelihood for the corresponding entity to be in the state of this hypothesis node; and a binary weight, Ψ, can be assigned to each hypothesis edge, which measures the compatibility of the pair of hypothesis nodes connected by the edge. The methodology is to select one hypothesis node for each entity, in order to maximize the combined unary and binary weights. This is a graph optimization problem and the general form is NP-hard; however, if the relational graph is a tree (including the degenerated case of a single branch), the problem is no longer NP-hard and efficient dynamic programming based polynomial time solutions exist.

For a tree-based relational graph, G_r, and the corresponding hypothesis graph, G_h, the objective function for a set of arbitrary selected nodes (following the structure of G_r) $s = \{s^i|_{i=1}^{|V_r|}, s^i \in V_h\}$ is:

$$\mathcal{M}(s) = \sum_{s^i \in V_h} \Phi(s^i) + \lambda \cdot \sum_{(s^i, s^j) \in E_h} \Psi(s^i, s^j), \tag{1.1}$$

in which λ is the parameter for adjusting the binary and unary weights, and the goal is to maximize $\mathcal{M}(s)$: $s^* = \arg\max_s(\mathcal{M}(s))$. Let the algorithm proceed from the leaf nodes to the root, and let $\mathcal{F}(i,k)$ be the maximum achievable combined unary and binary weights of kth hypothesis for ith entity. $\mathcal{F}(\cdot,\cdot)$ satisfies the following recursive function:

$$\begin{aligned} \mathcal{F}(i,k) = {} & \Phi(v^k_{h(i)}) + \\ & \sum_{v^j_r \in kids(v^i_r)} \max_l \big(\lambda \cdot \Psi(v^k_{h(i)}, v^l_{h(j)}) + \mathcal{F}(j,l)\big). \end{aligned} \tag{1.2}$$

Based on this recursive function, the problem can be solved efficiently by dynamic programming, with a computation complexity of $\mathcal{O}(|V_r| \cdot N)$, where N is the max number of hypotheses for each node in V_r.

The concept of relational and hypothesis graphs can be applied to different problems. For example, for the video object segmentation (Section 1.2), the relational graph encodes the structure of the video frames, and the hypothesis graph encodes the object proposals. We aim to select one object proposal from each video frame using these graphs. Another example is the human pose estimations in videos, and in this problem, the relational graph encodes the human body structures in the video frames, and the hypothesis graph encodes the human pose candidates. We aim to select one human pose candidate from each video frame using these graphs.

1.2 Video Object Segmentation

The goal for video object segmentation is to detect the primary moving object in the video and to delineate it from the background in all frames. Video object segmentation is a well-researched problem in computer vision and is a prerequisite for a variety of high-level vision applications, including content based video retrieval, video summarization, activity understanding and targeted content replacement. Fig.1.2 shows an illustration for this problem.

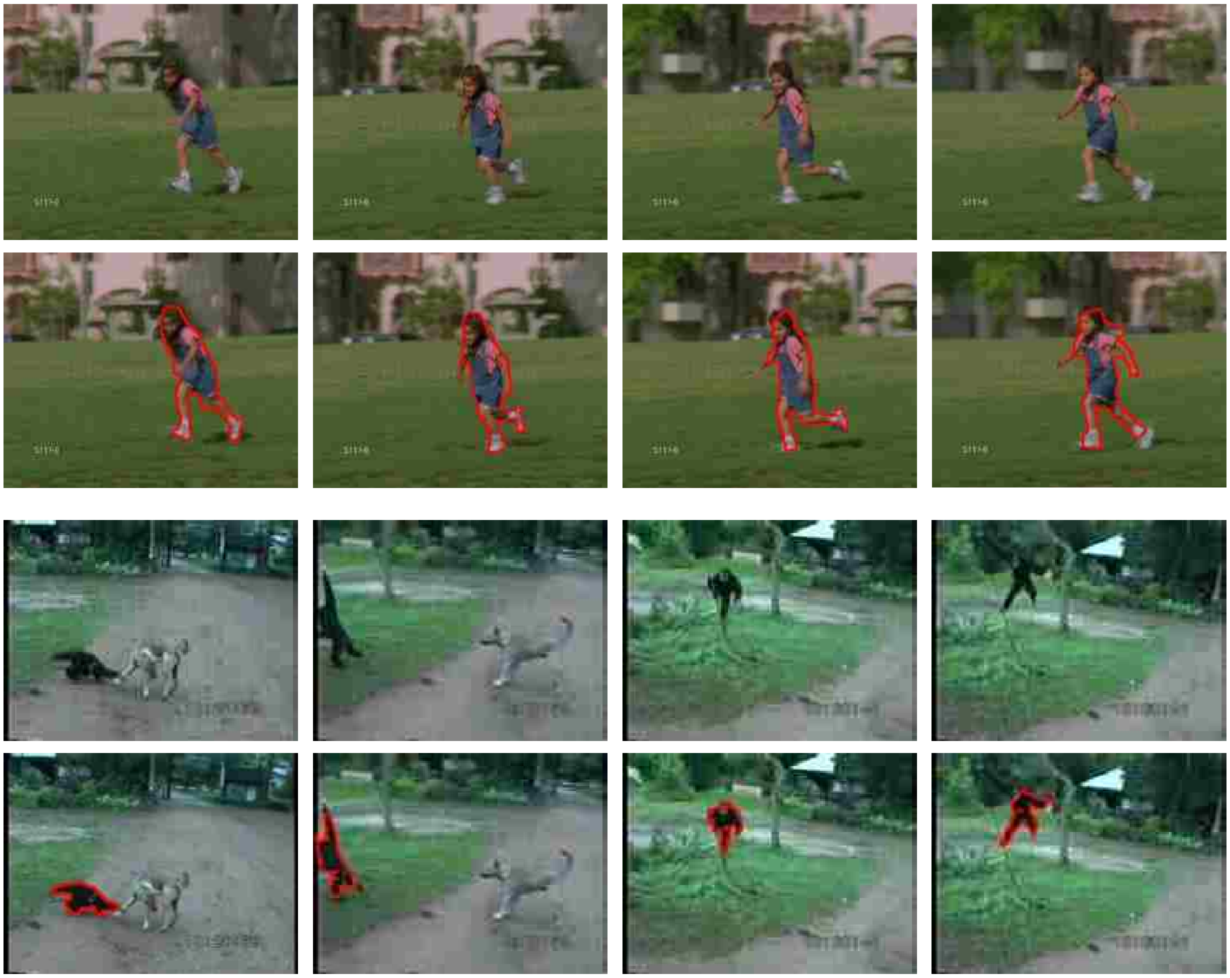


Figure 1.2: An Illustration for Video Object Segmentation.

In the past, several methods have been proposed to tackle this problem, however, most of them are supervised (e.g. [5, 82, 133]), which means that the methods use manual segmentation annotations in the first frame and prorogate them to the following frames. A fully automatic method is highly preferable and there are several methods (e.g. [60, 68, 134]) proposed recently to attack the problem in this way. A most successful technique to tackle this problem is to use the object proposals [21]. Object proposals are regions in images which are likely to be the foreground objects. For each video frame, there can be multiple object proposals, therefore how to select the correct object proposals is the most important problem. Just picking the top object proposals will not work in this case, since even though the object proposals are ranked, the rankings are very noisy and the top proposals are usually not the proposals we want. Another method to handle this problem is to use some clustering method to select the most relevant object proposals, however, in this case it is very difficult to guarantee that exactly one proposal is selected from each frame. In this case, graph-based method can be employed to solve this problem. The idea is to use nodes to represent proposals in each frame and build a fully connected graph where the edges corresponding to the similarities between the object proposals. The typical constraint for this graph is that only one object proposal can be selected from each video frame, and the objective it to get a set of object proposals which have the highest combined score.

Although the graph-based method mentioned above for video object segmentation is straightforward, it is not a good option in real world applications due to its high computational complexity. The problem is NP-hard and the optimal solution cannot be obtained in a reasonable time if the videos are long. Approximate solutions can be applied, however, a global optimum may not be guaranteed and the performance would be sacrificed. Therefore, we propose a new graph formulation to solve this problem. The intuition behind this formulation is that: introducing edges between proposals from all different frames is reasonable but not necessary, and sometimes may be harmful if the edge weights are not designed properly. For video object segmentation, the appearance of the object may change through the video frames, so connecting proposals from frames which are far away would only help the performance marginally. Furthermore, if one really wants to handle the appearance changes through the video frames, the edges which connect distant frames should be penalized, this mechanism further increases the complexity of the algorithm.

We propose a new method [134], in which compared to fully connected graphs, the object proposals are only locally connected. This method significantly makes the formulation simpler, and more importantly, leads to an efficient and exact solution. In the proposed method, the object proposals are represented by nodes, and an unary score is assigned to each object proposal which is a combination of object-ness score and motion score. The binary edges are designed to connect object proposals from neighboring frames (within 3 frames in our implementation), and their weights correspond to the similarity of color histogram and a location adjacency measure. The final graph is obtained by a directed acyclic graph, and the solution for this graph is a linear time dynamic programming algorithm. We also propose pre- and post-processing to refine the results. The pre-processing handles the incomplete object proposals. Quite often, there are no complete object proposals in some frames, we use pre-processing to combine some of the incomplete object proposals in order to have good object proposals in each frame. For post-processing, we use a video-based Markov Random Field (MRF) to smooth out the object segmentation from all the frames to get a pixel-level segmentation.

1.3 Video Object Co-Segmentation

In the previous Section, we consider the object segmentation problem in a single video. In this Section, we investigate how to benefit the object segmentation by using multiple videos, which inspires us to formulate a new problem: video object co-segmentation using spatiotemporal graph.

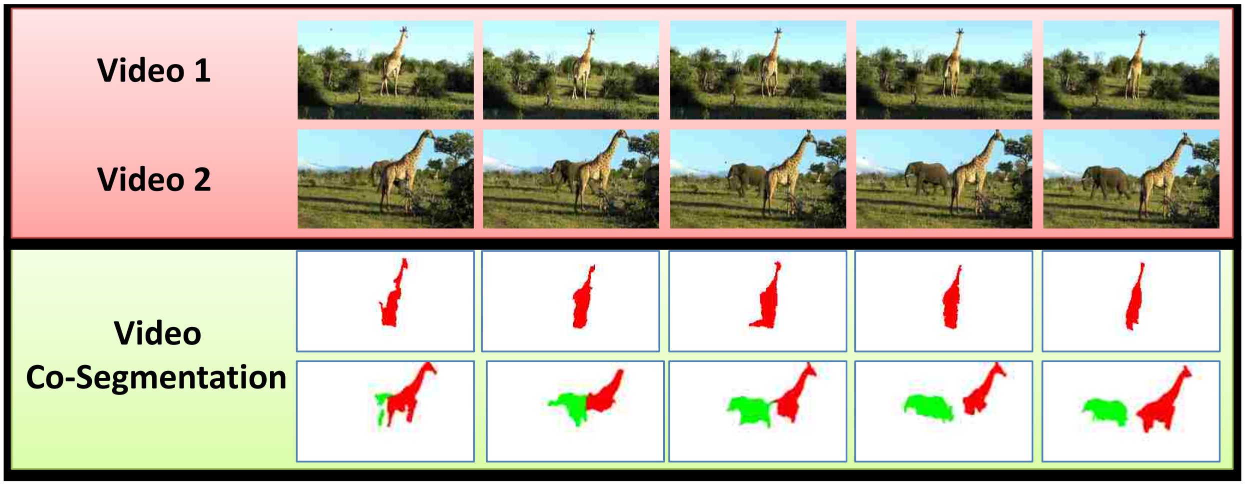


Figure 1.3: An Illustration for Video Object Co-Segmentation.

The goal of video object co-segmentation is to discover and segment objects from multiple videos in an un-supervised manner. Unsupervised segmentation of object classes from multiple videos has applications in large scale video tagging and retrieval, generation of training sets for supervised learning, and forensic video analysis. Fig.1.3 shows an example for video object co-segmentation.

Video co-segmentation is a natural extension of image co-segmentation (e.g. [93, 49, 71]), which uses multiple input images to segment the common objects from them. In contrast, for video object co-segmentation, the input is multiple videos which contain some common objects. The expected results are the segmentations of every object, such that the same labels are assigned to the common objects.

This becomes simple if each video only contains one object. Since the video object segmentation methods can be applied to extract the object, and a post-processing step can be integrated to match objects from different videos. However, if there are potentially multiple objects in each video, the problem becomes more difficult and video object segmentation method may not work. One reason is that video object segmentation methods cannot segment multiple objects. Another reason is that if multiple objects are adjacent to each other and have similar motion patterns, it is very likely for video object segmentation method to segment them into a single object.

We formulate this problem as spatiotemporal graph and obtain the exact solution [135]. The graph is built using tracklets of object proposals. Each object proposal in every video frame is tracked backward and forward throughout the whole video. If some tracks drift, the tracks are split into smaller tracklets to avoid the drift. Several top object proposal tracklets are selected from each video based on the sum of the object-ness scores. A graph is built for the object proposal tracklets from different videos, where each node represents one object proposal tracklet. Edges between nodes represent the similarity between tracklets. Then the problem becomes to group the nodes such that nodes in a group are similar to each other. The problem is a Maximum Weight Cliques Problem, and we solve it by a modified version of Bron-Kerbosch Algorithm. The complexity of the algorithm is $O(3^{n/3})$ with an exact solution. Finally, a multi-label MRF model is applied to the initial segmentation results and pixel-level segmentations are obtained.

1.4 Human Pose Estimation in Videos

The previous two Sections discussed the problem of object segmentation and co-segmentation in videos. We found that the important class of objects in the online videos, or in surveillance camera videos is human. In contrast to generic object segmentation, specific knowledge about the human body (e.g. body poses) can be employed in video object segmentation, tracking, etc. This leads us to use spatiotemporal graph in another interesting problem: human pose estimation in videos.

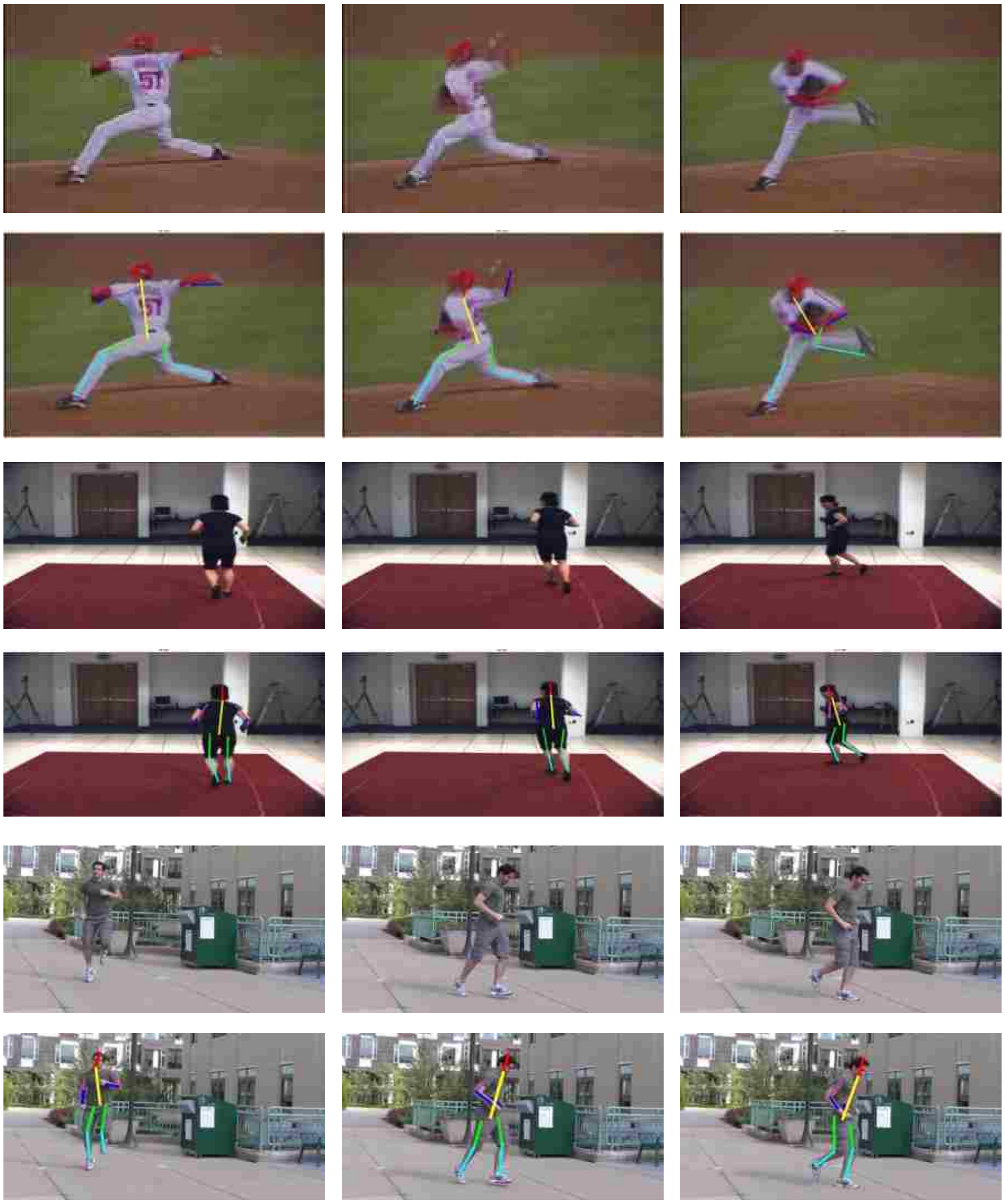


Figure 1.4: An Illustration for Human Pose Estimation in Videos.

The goal of human pose estimation in videos is to estimate the human poses from each video frame and ensure the poses are consistent throughout the video frames. Fig.1.4 shows an illustration of the problem of human pose estimation in videos. This is a very important problem for human-computer interaction, video surveillance and tracking. An obvious approach is to apply the single image based pose estimation method to each frame. However, in this dissertation, we demonstrate that by using temporal information contained in a video, the performance can be improved. Due to body part occlusions in the videos, the image based pose estimation methods (e.g. [130, 111, 18, 117]) normally cannot predict the body part locations accurately. The double counting problem is also serious for image based pose estimations, in which the left and right body parts are detected at the same location. These two problems can be potentially avoided once the temporal information in videos is considered. For example, to deal with occlusion, the pose estimation results from adjacent frames can be utilized to predict the body part locations once they are occluded. For the double counting problem, if in most of the video frames the body parts can be detected correctly, these "correct poses" can be used to avoid the double counting problem in other frames. Furthermore, even though the human poses in video frames can be estimated within a reasonable error range, the poses may not be consistent throughout the video frames. The human poses in the frames can be made more consistent by using the temporal information.

An initial idea is to generate some hypothesis poses in each video frame, using the image-based pose estimation methods, and select the best one from each frame such that the poses are consistent throughout the video. A baseline method with this idea does not give very good results. Since there are many body parts of the human body, there would be an exponential number of combinations of human poses for these body parts, therefore it is very difficult to generate accurate pose hypotheses in each frame. Another idea is to generate body part hypotheses rather than human pose hypotheses, then select the best body part locations considering the spatiotemporal relationships between the body parts. This idea is perfect if the problem can be solved efficiently. However, it is computational prohibitive since there are a lot of simple cycles in the corresponding graphs using this formulation. Therefore, how to simplify the graph without losing too much information is an important issue.

We propose two key ideas [136] to tackle this issue, which reduces the original fully connected graph model into a simplified tree-based graph model. The first idea is **Abstraction**: in contrast to the standard tree-based representation of body parts, we introduce a new concept, *abstract body parts*, to conceptually combine the symmetric body parts. In this way, we take the advantage of the symmetric nature of the human body parts without inducing simple cycles into the formulation. The second idea is **Association**, using which we generate optimal tracklets for each abstract body part to ensure the temporal consistency. Since each abstract body part is processed separately, it does not induce any temporal simple cycles into the graph. The proposed method can be solved efficiently to get accurate pose estimation results in videos.

1.5 Dissertation Organization

This dissertation is organized as follow: In Chapter 2 we introduce the related works for object segmentation and human pose estimation in videos; in Chapter 3 we introduce the proposed method for video object segmentation and show improved results on publicly available datasets; in Chapter 4, we introduce our video object co-segmentation method and show improved results on MOViCS dataset and Safari datasets; in Chapter 5, we introduce the proposed human pose estimation method for videos and show results on three public available datasets; and we conclude and discuss future works in Chapter 6.

CHAPTER 2: LITERATURE REVIEW

In the previous Chapter, we introduced the three computer vision problems: video object segmentation, video object co-segmentation, and human pose estimation in videos, which we investigate in this dissertation using spatiotemporal graphs. Before we start introducing each problem with detailed solutions and experiments in different Chapters, in this Chapter, we give a review of the related works in order to give the readers a big picture and some context for these topics.

We start with the introduction of a fundamental problem in computer vision: video segmentation. In this problem, the goal is to segment the video sequences into different spatial temporal regions which are perceptually consistent (in the context of appearance, motion, etc.). This problem is a generalization of image segmentation to multiple frames in a video. We introduce the related works for video segmentation in Section 2.1

There are many sub-problems of video segmentation, for example, video object segmentation, video object co-segmentation, and motion segmentation, etc. Video object segmentation aims to segment the single foreground moving object from the background in different frames in the video. This is different from background difference, since in background difference each moving object which is different from background is identified in the frames. However, in video object segmentation, the correspondences of object regions in different video frames are also determined. Also, in most background difference methods, it assume the camera capturing the video is stationary. However, in video object segmentation, the camera does not need to be static. Video object segmentation is a very hot topic in recent computer research and we introduce its related works in Section 2.2.

Video object co-segmentation is also a new problem which has only been investigated recently. Compared to video object segmentation, multiple videos are considered and the aim is to segment and assign the same label to the object appearing in multiple videos. In this formulation, information from multiple videos is simultaneously used to segment multiple objects. Video object segmentation methods can be applied to solve video object co-segmentation if and only if there is one object in each video. We will introduce its related works in Section 2.3.

It is very difficult to get accurate segmentation of humans in videos due to their body part deformations. However, human is one of the most important object classes in videos. We believe human pose estimation can aid in solving human object segmentation. In pose estimation the aim is to estimate the joint locations of the human body in video frames. This can be very useful for the segmentation of human in videos. Pose estimation in a single image has been very active area of research. However, research related to human pose estimation from videos has been just started in recent years. We introduce its related works in Section 2.4.

2.1 Video Segmentation

In this dissertation, video segmentation refers to the segmentation of videos into different spatiotemporal regions. It has been an active research topic in last a few years, and a large dataset [37] is available to compare different video segmentation algorithms. The video segmentation methods can be classified into three categories: graph-based segmentation, clustering segmentation, and motion segmentation.

2.1.1 Graph-based Segmentation

Graph-based segmentation methods are commonly employed for video segmentation, which employ nodes to represent the super-pixels or super-voxels and edges to represent the similarities.

Hierarchical graph-based method (e.g. [40, 105]) is the most intuitive methodology for video segmentation. In [40], the method begins by over-segmenting a video into super-voxels using appearance. A region graph over the obtained super-voxels is built, and these super-voxels are iteratively grouped over multiple levels to create a tree of spatiotemporal segmentations. To extend its application to dynamic video scene segmentation, another formulation of the hierarchical graph-based method is proposed in [105]. Custom spatiotemporal filters with texture and motion cues are used, and a novel metric-learning framework is employed to optimize the representation for specific objects and scenes.

Bottom-up video segmentation methods (e.g. [61, 115, 13]) which do not have hierarchical formulations are also proposed. Conditional Random Field model (CRF) is the traditional solution for segmentation, and in [13], CRF is employed to handle disconnected spatiotemporal segments, and to ensure consistent segmentation. The labeling decision is postponed until a high confidence is gained from the region. CRF can also be applied on top of initial rough segmentation, for example, a multiple hypothesis framework is proposed in [115], which begins by generating several segmentations for each frame and enumerating many possible trajectories of regions within some frames. Each trajectory is assigned a score, and these trajectories are used to get an initial segmentation of the video. A MAP labeling of a higher-order random field is obtained to determine the final segmentation. In contrast to CRF, holistic models can also be employed: for instance, the concept of spatiotemporal closure is proposed in [61] for video segmentation, which treats the spatiotemporal volume as a single entity, and extracts contiguous tubes whose overall surface is supported by strong appearance and motion discontinuities.

Different methods (e.g. [126, 64, 63, 127, 36]) have been proposed to make the video segmentation algorithms computational efficient. A streaming hierarchical video segmentation framework is proposed in [126], which is motivated by data streaming principal that each video frame is processed only once and does not change the segmentation of previous frames. It aims to save the computational resources and process the videos faster. Similarly, a Sub-Optimal Low-Rank Decomposition method is introduced in [64, 63] for the graph-based streaming video segmentation. The representation coefficient matrix with the fixed rank is decomposed into two sub-matrices of low rank, and then these matrices are iteratively optimized with closed-form solutions. To further improve the efficiency of streaming video segmentation, a reduced graph formulation is introduced in [36], which is re-weighted in a way such that the resulting segmentation is equivalent to that of the full graph under certain assumptions. For online video segmentation, a framework is introduced in [127], which consists of a probabilistic segment label propagation method and a temporally consistent hierarchical label merging scheme.

Many features and feature fusion methods (e.g. [131, 34, 55]) have also been proposed to boost the performance. A multi-cue fusion method is proposed in [131] within the Markov Random Field model for unconstrained video segmentation, which combines contour cues, temporally smoothness cues, and global structure cues from a video. A analysis of different within- and between-frame affinities (including motion cues) is performed for video segmentation in [34]. A classifier based graph construction method is proposed in [55] which combines features and calibrated classifier outputs as edge weights, and defines the graph topology by an edge selection scheme.

Graph-based methods can model the structure of the video sequence well, however, it is not a "direct" method since it builds the graph first then optimize it. In contrast, clustering segmentation method group the pixels directly and we will introduce these methods in the next Section.

2.1.2 Clustering Segmentation

Clustering methods, for example mean-shift [19, 77], spectral clustering [54], and Gaussian Mixture Models [39], are also employed for video segmentation. They explore the problem in a different perspective and are deeply rooted in the unsupervised learning methodology.

Mean-shift [19, 77] is the most commonly used clustering method for video segmentation. A hierarchical clustering method is adopted in [19] for video segmentation, which operates by repeatedly applying mean shift analysis over increasingly large ranges, with the combined 7-dimension color-motion features. Another mean-shift based video segmentation method is proposed in [77] with bilateral filtering and anisotropic diffusion, which preserves the edges and enforces the smoothness. For spectral clustering, must-link constraints are combined in [54] for video segmentation. Guassian mixture models have been used in [39] to extract coherent space-time regions in feature space (with color features), and then to get the corresponding coherent segments (video-regions) in the videos. Different from traditional clustering methods, a co-clustering method is introduced in [113] for semantic video segmentation. It is formalized as a Quadratic Semi-Assignment Problem which has a linear programming relaxation that makes effective use of information from hierarchies. An iterative multi-resolution video segmentation algorithm is obtained within this framework. For a special application, textured object segmentation in videos, feature fusion and global convex continuous optimization are employed in [81].

Most of the graph-based segmentation and clustering segmentation methods take only appearance information (pixel values) into account. However, motion cues can also be used for video segmentation and we will introduce its related works in the next Section.

2.1.3 Motion Segmentation

Motion segmentation is another important techniques for video segmentation. Motion boundaries and figure-ground segmentation (e.g. [57, 27, 104]) used to be the dominant techniques for motion segmentation, however, point and region trajectory based methods (e.g. [7, 73, 62, 29, 53, 35]) have become more popular recently.

Motion boundaries and figure-ground segmentation (e.g. [57, 27, 104]) are traditional ways for foreground object segmentation. Low level cues (e.g. flows, occlusions) are used for motion segmentation. A two-step motion segmentation method is proposed in [57], which consists of an algorithm for obtaining the initial estimate of the model by dividing the scene into rigidly moving components using efficient loopy belief propagation; and a second step to refine the initial estimate using $\alpha\beta$-swap and α-expansion algorithms. Long-term occlusion relations are analyzed in [104] and used within a convex optimization framework to segment the video frames. In comparison, object-level cues can also be integrated for motion segmentation, for instance, a learning based method is proposed for moving object segmentation in [27]. In this method, segment proposals are obtained by multiple figure-ground segmentations on motion boundaries. A Moving Object-ness Detector which was trained on image and motion fields is used to detect moving objects in the video.

Recently, point and region trajectories (e.g. [7, 73, 62, 29, 53, 35]) have been employed more often for motion segmentation. Long term point trajectories can be employed in many ways for motion segmentation. A basic version [7] is to use the long term point trajectories of dense optical flow. This method uses pair-wise distances between these trajectories for clustering. Temporally consistent segmentations of moving objects are obtained in a video shot. One step further, in [62], a new track clustering cost function that includes occlusion reasoning is proposed to cluster of point tracks into coherent motions. Several methods have been proposed to embed the trajectories. A variational method is proposed in [73] to get dense segmentations from sparse trajectory clusters. A hierarchical, nonlinear diffusion process which take advantage of superpixels to propagate the motion segmentation is used. A new trajectory embedding method is proposed in [29], which uses a discretization process to recover from the over-segmentations by merging clusters according to discontinuity evidence along the boundaries. A minimum cost multi-cut formulation is proposed in [53] to segment the long-term trajectories, which not only assigns cluster labels, but also decides the number of clusters while allowing for varying cluster sizes. Compared to point trajectories, a region trajectory based segmentation method is introduced in [35]: first, hierarchical image segments are obtained and a directed acyclic graph for these segments in video frames is built; then, a dynamic programming-based optimisation is applied to get the region trajectories and measures of confidence from the graph.

As a special problem in video segmentation, video object segmentation has gain its popularity in recent years. We will introduce its related works in next Section.

2.2 Video Object Segmentation

Video object segmentation aims to detect the primary object in a video and to delineate it from the background in all frames. Both supervised and fully automatic methods have been proposed.

2.2.1 Supervised Methods

Some of the supervised video object segmentation methods (e.g. [5, 82, 133]) need annotations of object segments in key frames for initialization. Optimization techniques employing motion and appearance constraints are then used to propagate the segments to all frames.

Other methods (e.g. [90, 112]) only require accurate object region annotation for the first frame, then employ region tracking to segment the rest of frames into object and background regions. A region-based particle filter method is proposed in [114], in which the particles are defined as the regions in the current image partition. The prediction step uses co-clustering between the object segmentation in the previous frame with the current frame, which allows to tackle the shape changes of non-rigid objects. Patch seams are also used in [83] to tackle the problem, which are connected paths of low energy in the video frames. With the patch seams, the proposed energy function combines the similarity of patches, temporal consistency of motion and spatial coherency of seams. Segmentations are propagated with high fidelity in the critical boundary regions. Tracking and segmentation are integrated together in [125] to get improved results for both. The segmentation problem is treated as a pixel-level label assignment task with regularization according to the part models, and the tracking problem is considered as estimating the part models based on the segmentations, which in turn is used to refine the model.

2.2.2 Semi-Supervised Methods

Motion grouping (e.g. [97, 96, 7, 76, 38, 80, 11]), as an initial step, has been employed for automatic video object segmentation for a long time. A fast object segmentation method is proposed in [76], a rough estimate of which pixels are within the object region based on motion boundaries is first obtained; then the result is refined by a spatiotemporal extension of GrabCut over all the video frames. Similarly, in [38], motion segmentation is applied to get a coarse foreground segmentation, then the motion regions are refined by optimizing an energy function based on appearance and perceptual organization. Another method [80] introduces VabCut, a video extension of GrabCut, to tackle the unsupervised video foreground object segmentation task. It combines the RGB frames with a motion layer. Dense trajectories from optical flow are employed in [11] for motion segmentation. A new affinity measurement method incorporating both global and local information of point trajectories is proposed to cluster trajectories into groups and a graph-based method is applied to obtain the final segmentation.

Visual saliency is introduced to handle the video object segmentation problem in [123, 23, 128]. In [123] the saliency is considered as a prior for object via the computation of robust geodesic measurement. Two discriminative visual features are employed to infer the object regions: spatial edges and temporal motion boundaries. The video object segmentation problem is cast as a voting scheme [23] on the graph of similar regions in the video sequence. The method starts from simple saliency votes at each pixel, and iteratively corrects the votes by consensus voting of re-occurring regions across the video sequence. Furthermore, saliency fusion is employed in [128] to boost the performance. The temporal coherent salient region throughout the whole video can be obtained as the first step, then a discriminative model will be learnt to represent the appearance of the object against the background. As an important component, a self-adaptive saliency map fusion method is employed by learning the reliability of saliency maps from different cues.

Object detections are also used for video object segmentation (e.g. [121, 118, 137]). A graph transduction method is introduced in [121, 118] for object segmentation. It learns object proposals densely over space-time, with both appearance models learned from rudimentary detections of sparse object-like regions. In contrast, object detection is employed in [137] for semantic video object segmentation. The authors propose a video segmentation-by-detection framework. They first apply object and region detectors pre-trained on images to generate a set of detection and segmentation proposals; then track several objects and solve a joint binary optimization problem with min-cost flow for segmentation.

Co-segmentation is also employed (e.g. [66, 67]) for video object segmentation. The co-segmentation algorithm learns the model of the primary object by representing the frames/videos as a graphical model. In order to handle longer videos in [67], the object segmentation should be performed within video shots. In the valid video shots, a graph is constructed to model the video object detection and the final segmentation is obtained within the detection boxes.

More recently, object proposal methods (e.g. [21, 9, 1]) are employed for video object segmentation in [60, 68, 104, 8, 134]. Lee et al. [60] proposed to detect the primary object by collecting a pool of object proposals from the video, and applied spectral graph clustering to obtain multiple binary inlier/outlier partitions. Each inlier cluster corresponds to a particular object's regions. Both motion and appearance based cues are used to measure the ‘objectness’ of a proposal in the cluster. Ma et al. [68] attempt to mitigate this issue by utilizing relationships between object proposals in adjacent frames. The object region selection problem is modeled as a constrained Maximum Weight Cliques problem in order to find the true object region from all the video frames simultaneously. However, this problem is NP-hard [68] and an approximate optimization technique is used to obtain the solution. The probabilistic graphical model is built across a set of videos based on an object proposal algorithm. Long-term occlusion cues are utilized in [104] to infer the moving objects from the videos. A shorted path formulation in [8] is employed to select the video objects.

Compared to pre-trained image-based object proposal methods, there are some other variations of object proposals which are designed for video object segmentation (e.g. [14, 74]). In [14], foreground/backround object-like pool is obtained by using the pixel-level optical flow and binary mask features. A super-pixel level conditional random field can be built to label the foreground and background based on the fact that the appearance and motion features of the moving object are temporally and spatially coherent. Compared to image based object proposals, the spatiotemporal object proposals [74] are also used for video object segmentation.

In summary, video object segmentation has been investigated intensely, and object proposal based methods (e.g. [60, 68]) have gain their popularity in recent years. However, these methods [60, 68] still have some drawbacks which inspired our proposed method. In [60], a drawback is that the clustering process ignores the order of the proposals in the video, and therefore, cannot model the evolution of object’s shape and location with time. And the shortcoming of [68] is that its formulation is NP-hard and an approximate optimization technique is used to obtain the solution. Both of the two approaches [60, 68] have two additional limitations. First, in both approaches, object proposal generation for a particular frame doesn’t directly depend on object proposals generated for adjacent frames. Second, both approaches do not actually predict the shape of the object in adjacent frames when computing region similarity, which degrades segmentation performance for fast moving objects. In this dissertation, we propose a directed acyclic graph based method which overcomes the drawbacks of the state-of-the-art method and improves the performance. The details of the proposed method and the experimental results are given in Chapter 3.

As a further step, compared to single object segmentation in videos, there are also some methods proposed to simultaneously segment objects from multiple videos. We will introduce these related works in next Section.

2.3 Video Object Co-Segmentation

Video object co-segmentation is a relatively new problem. Given multiple videos as input, the aim is to segment the objects, and assign the common objects with same labels.

Bottom-up methods with super-pixels as their building blocks have been proposed in [16, 119, 122]. Chiu and Fritz [16] proposed multi-class video object co-segmentation and also provided a publicly available dataset (MOViCS) with ground truth. In this work, a non-parametric Bayesian model for co-segmentation is used, which is based on a video segmentation prior. Simultaneous object discovery and segmentation is also proposed in [119], and this formulation uses a spatiotemporal auto-context model, and combines with the appearance modeling for superpixel labeling. For object discovery, the superpixel-level labels are propagated to the frame level through a multiple instance boosting algorithm with spatial reasoning. Multiple cues (intra-frame saliency, inter-frame consistency, and across-video similarity) have been incorporated into an energy optimization framework [122] for robust object co-segmentation.

Bottom-up methods with super-voxels have also been demonstrated effective for video object co-segmentation in [10, 116]. The method in [10] attempts to segment the common region from a pair of videos and model the problem as a common foreground and background separation. In this method, the video pair is represented by super-voxels and a motion-based video grouping method is used to find common foreground regions. Gaussian mixture models are employed to characterize the common object appearance. A Markov Random Field model with a Quadratic Pseudo-Boolean Optimization is proposed in [116] for video object co-segmentation. A subspace is used to segment the videos into consistent spatio-temporal regions with multiple classes, while assigning the common foreground consistent labels. The motion features are designed to better differentiate regions within each video, making accurate extraction of object boundaries.

Motion and saliency cues are used to guide video object co-segmentation in [41, 44, 65, 94]. Li et al. has proposed a method in [65], which employs a multi-search strategy to extract each target individually and an adaptive decision criterion is used to decide whether the segmented regions are the correct targets. Object-ness and motion are also employed by Guo et al. [41, 44]. First, a figure/ground segmentation method is employed to generate some seed foreground segmentations; then, linkage constraints between the super-pixels are added based on whether they exhibit the characteristics of common fate or not; and finally, an iterative constrained clustering algorithm is employed to trim away the incorrect and accidental linkage relationships. The work by Rubio et al. [94] aims at segmenting the same object (or objects belonging to the same class) moving in a similar manner from two or more videos. The method starts with grouping the pixels in video frames at two levels: the higher levels consists of space-time tubes and the lower level consists of within frame region segments; an initial foreground and background labeling is generated to construct the probabilistic distribution of the feature vectors of tubes and regions; and a probabilistic framework is employed to get the final co-segmentation results.

Object proposals are also used in video object co-segmentation (e.g. [129, 30, 31, 32]). An enriched object proposal set (proposal stream) is used for object co-segmentation in [129]. The problem is again formulated as a graphical model to select a proposal stream for each object in which the pairwise potentials consist of the appearance dissimilarity between different streams in the videos. The object-based co-segmentation methods proposed in [30, 31] formulate the problem as a co-selection graph in which the segments that are likely to be the foreground objects are favored while the intra-video and inter-video coherence also considered. For multiple objects, the co-selection graph model is extended into a proposed multi-state selection graph model (MSG) that optimizes the segmentations of different objects jointly. To improve the robustness, a multi-component foreground model is developed in [32] to handle appearance variations (viewpoint/pose/color) of the objects. In order to learn the parameters of the multi-component model, a transfer learning algorithm is employed to propagate the information of the labeled frames to the unlabeled ones in a tree structured model.

In summary, the methods [10, 94, 16] which are based on pixels or image patches have relatively good performance for video object co-segmentation. However, there are still some shortcomings in them. For example, both [10] and [94] use strong assumptions of a single class of object common to all videos. And the disadvantage of [16] is that it groups dense image patches to obtain segments, which can potentially yield noisy results. Inspired by the success of object proposals used in video object segmentation [60, 68], we propose a Regulated Maximum Weight Clique (RMWC) based method to attack the problem. It uses object proposal tracklets as its building blocks, and Bron-Kerbosch Algorithm to solve it efficiently. The details about the proposed method and the results are given in Chapter 4.

The video object segmentation and co-segmentation methods introduced in this section and the previous section are all about generic objects. One natural question would be, how to model a specific class of objects using spatiotemporal graphs? We found that human is one of the most important objects in videos and human pose estimation is one of the most difficult problems in this domain. Therefore, we investigate the problem of human pose estimation in videos and we will introduce its related works in next Section.

2.4 Human Pose Estimation

A large body of work in human pose estimation have been reported in recent years. Early works are focused on human pose estimation and tracking in controlled environment [100]; there is also some important work using depth images [99]. Single image based human pose estimation [130, 111, 18, 117] in unconstrained scenes has progressed dramatically in the last a few years; however, video based human pose estimation in unconstrained scenes is still in a very early stage, and some pioneer research (e.g. [85, 78, 108, 15]) has been conducted only recently. We briefly introduce related works of image based human pose estimation, and focus on video based full body human pose estimation methods.

2.4.1 Image-based Methods

For image based human pose estimation in unconstrained scenes, most work has been focused on pictorial structure models (e.g. [3, 4, 47, 87]) for quite long time and the performance has been promising. In [130], a flexible mixture-of-parts model was proposed to infer the pose configurations, which showed very impressive results. A new scheme was introduced in [48] to handle a large number of training samples which resulted in significant increase in pose estimation accuracy. Authors in [102, 132, 84] attempted to estimate 3D human poses from a single image.

The high order dependencies of body parts are exploited in [59, 45, 46, 103, 101, 124, 52, 107, 79, 86]. The authors in [107] proposed a hierarchical spatial model with an exact solution, while in [79], a conditional model is defined. The rich spatial interactions among body parts are explored in [86] with an inference machine formulation. A novel, non-linear joint regressor model was proposed in [18], which handles typical ambiguities of tree based models quite well.

More recently, deep learning [111, 110, 75, 12] has also been introduced for human pose estimation. Both color and motion features are combined into a deep learning framework [42] for pose estimation. Pose estimation is formulated as a regression problem [111] towards body joints, and a cascade of Deep Neural Networks DNN regressors are trained to estimate the poses. A hybrid architecture [110], which consists of a deep Convolutional Network and a Markov Random Field is proposed to infer the human poses.

The image-based method only uses appearance information to estimate the human poses, which is not able to handle body part occlusions, motion blurs, etc. Videos contain much more information and temporal/motion features can be very helpful in handling these issues. We will introduce some recent works for human pose estimation in videos in the next Section.

2.4.2 Video-based Methods

For video based human pose estimation in unconstrained scenes, some early research adopted the tracking-by-detection framework (e.g. [2, 70, 88]). More recently, some methods [15, 108, 139, 28, 91, 95] have mainly focused on upper body pose estimation and other methods [78, 85] have focused on full body pose estimation.

For upper body pose estimation: A tracking-by-selection framework is proposed in [108] to simplify the graph optimization problem, which makes the exact inference possible. In [15], the poses are decomposed into limbs, and recomposed together to obtain pose estimations in the video. The authors in [95] decompose the full model of body parts into many tree-based sub-models which enables them to get the exact inference for the sub-models. To handle the loopy graph in video-based pose estimation, an approximation of the full model [138] is proposed by introducing an ensemble of two tree-structured sub-models: Markov networks for spatial parsing and Markov chains for temporal parsing. Both models can be trained jointly using the max-margin techniques, and an iterative parsing process is proposed to achieve the ensemble inference.

In context of full body pose estimation, in [78], many pose candidates are generated in each frame, and the most consistent ones which have high detection scores are selected through the frames. Ramakrishna *et al.* [85] model the symmetric structures of the body parts and proposed an effective approximate solution to the problem. A 2-level region-based tracking scheme is proposed in [72] which refines the joint locations by a Kalman filter. Bayes theorem using Extreme learning machine (ELM) is employed in [89] to estimate body part likelihood scores, and a voting scheme that uses inter-frame detected segments to filter out errors and detect body parts in the current frame. Inspired by the successful pictorial structures model, a new framework is proposed in [106] which combines an image conditioned model that incorporates higher order dependencies of multiple body parts with poselet features. The authors propose in [98] to propagate the detection in each frame using the global motion estimation. The method first produces reasonable trajectory hypotheses for each body part. Then, in the optimization framework, body part trajectories rather than body part candidates are obtained to infer the human pose.

In summary, all of the above video-based full body pose estimation methods [89, 78, 98] are insightful, however, none of them has simultaneously exploited the important constraints between body parts (e.g. symmetry of parts) and has an efficient exact solution. In this dissertation, we propose a novel method which reduces the video-based human pose estimation problem into a tree-based optimization problem which can be solved efficiently. The details of the proposed method and experimental results are introduced in Chapter 5.

2.5 Summary

In this Chapter, we have reviewed the literatures for the problems we investigate: video object segmentation, video object co-segmentation, and human pose estimation in videos. The related works give us a lot of insights about the problems, and based on these insights about the limitations of current state of art approaches, we propose new ideas in this dissertation to improve the performance on publicly available datasets.

Compared to the existing methods, we have developed three new methods to solve these problems using spatiotemporal graphs. First, for video object segmentation, object proposals are used intensely in current literature; however, the main issue for the state-of-the-art methods ([60, 68]) is that they are not able to guarantee a selection of good proposals for all the video frames. We employ spatiotemporal graph to model the object proposal structure in the video, and ensure one good object proposal is selected for every video frame. The results are improved dramatically using this idea. Second, for video object co-segmentation, rather than using low-level features [16], we embed the object proposal tracklets into a spatiotemporal graph which encode the motion and appearance cues in different videos to delineate the objects. Finally, for human pose estimation in videos, the major issue is the computational complexity of the standard graph formulation. We re-formulate the spatiotemporal graph to be a tree, and employ dynamic programming to reduce the computational complexity. We will introduce our proposed methods in detail in the next three Chapters.

CHAPTER 3: SPATIOTEMPORAL GRAPHS FOR VIDEO OBJECT SEGMENTATION

3.1 Introduction

In this Chapter, our goal is to detect the primary object in a video and to delineate it from the background in all frames. Video object segmentation is a well-researched problem in the computer vision and is a prerequisite for a variety of high-level vision applications, including content based video retrieval, video summarization, activity understanding and targeted content replacement.

We propose a novel method which is inspired by the object proposal based methods [60, 68]. In general, an object's shape and appearance varies slowly from frame to frame. Therefore, our idea is that the object proposal sequence in a video with high 'objectness', and high similarity across frames is likely to be the primary object. To this end, we use optical flow to track the evolution of object shape, and compute the difference between predicted and actual shape (along with appearance) to measure similarity of object proposals across frames. The 'objectness' is measured using appearance and a motion based criterion that emphasize high optical flow gradients at the boundaries between objects proposals and the background. The primary object proposal selection problem is formulated as the longest path problem for Directed Acyclic Graph (DAG), for which (unlike [68]) an optimal solution exists in linear time. Note that, if the temporal order of object proposals locations (across frames) is not used ([60], then it can result in no proposals being associated with the primary object for many frames (please see Figure 3.1). The proposed method not only uses object proposals from a particular frame (please see Figure 3.2), but also expands the proposal set using predictions from proposals of neighboring frame. The combination of proposal expansion, and the predicted shape based similarity criteria results in temporally dense and spatially accurate primary object proposal extraction. We have evaluated the proposed approach using several challenging benchmark videos and it outperforms both unsupervised and supervised state-of-the-art methods

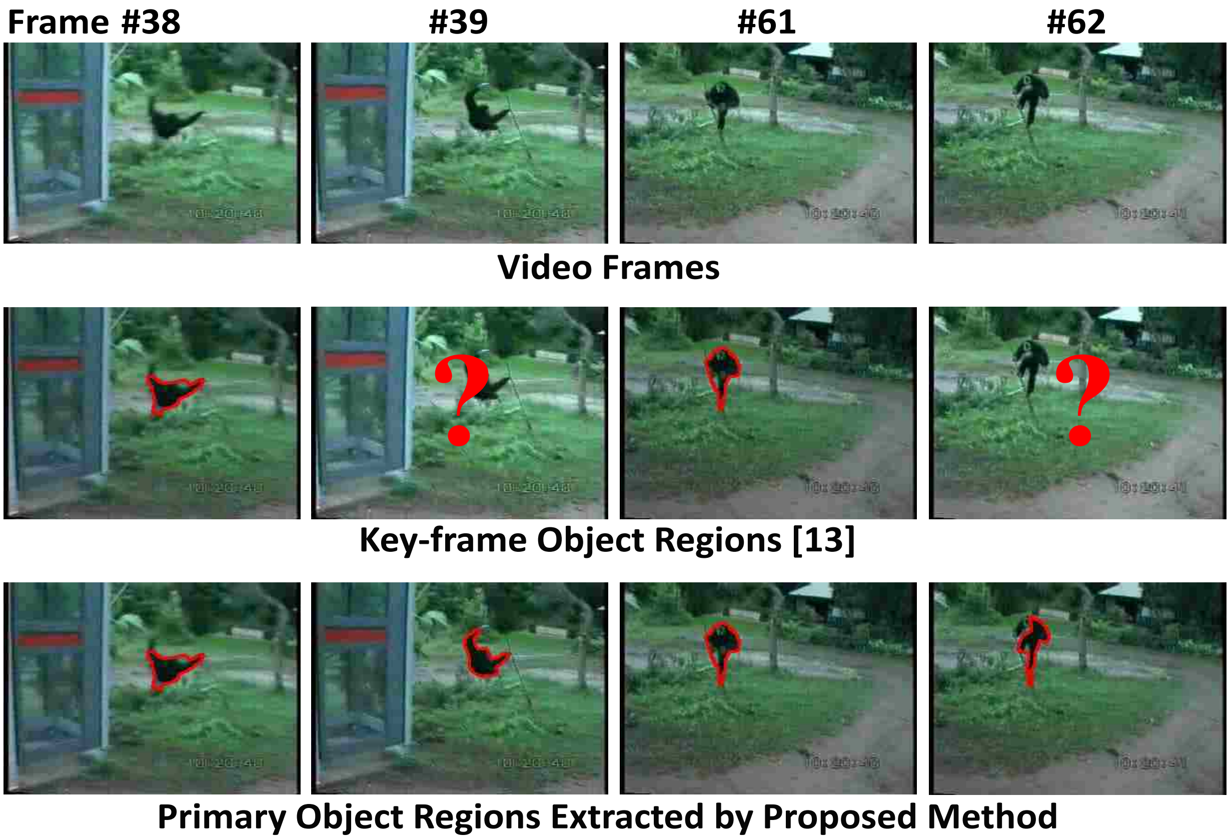

Frame #38
#39
#61
#62
Video Frames
?
?
Key-frame Object Regions [13]
Primary Object Regions Extracted by Proposed Method

Figure 3.1: Primary object region selection in the object proposal domain. The first row shows frames from a video. The second row shows key object proposals (in red boundaries) extracted by [60]. "?" indicates that no proposal related to the primary object was found by the method. The third row shows primary object proposals selected by the proposed method. Note that the proposed method was able to find primary object proposals in all frames. The results in row 2 and 3 are prior to per-pixel segmentation. In this dissertation we demonstrate that temporally dense extraction of primary object proposals results in significant improvement in object segmentation performance. Please see Table 3.1 for quantitative results and comparisons to state of the art.[Please Print in Color]

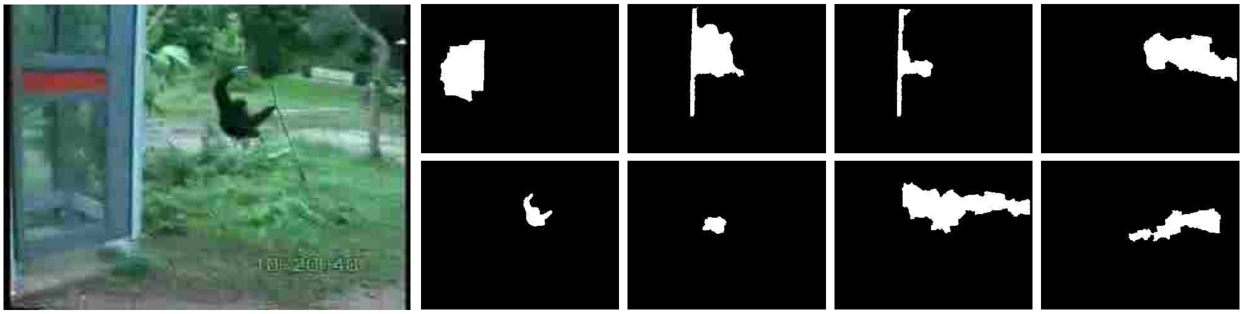


Figure 3.2: Object proposals from a video frame employing the method in [21]. The left side image is one of the video frames. Note that the monkey is the object of interest in the frame. Images on the right show some of the top ranked object proposals from the frame. Most of the proposals do not correspond to an actual object. The goal of the proposed work is to generate an enhanced set of object proposals and extract the segments related to the primary object from the video.

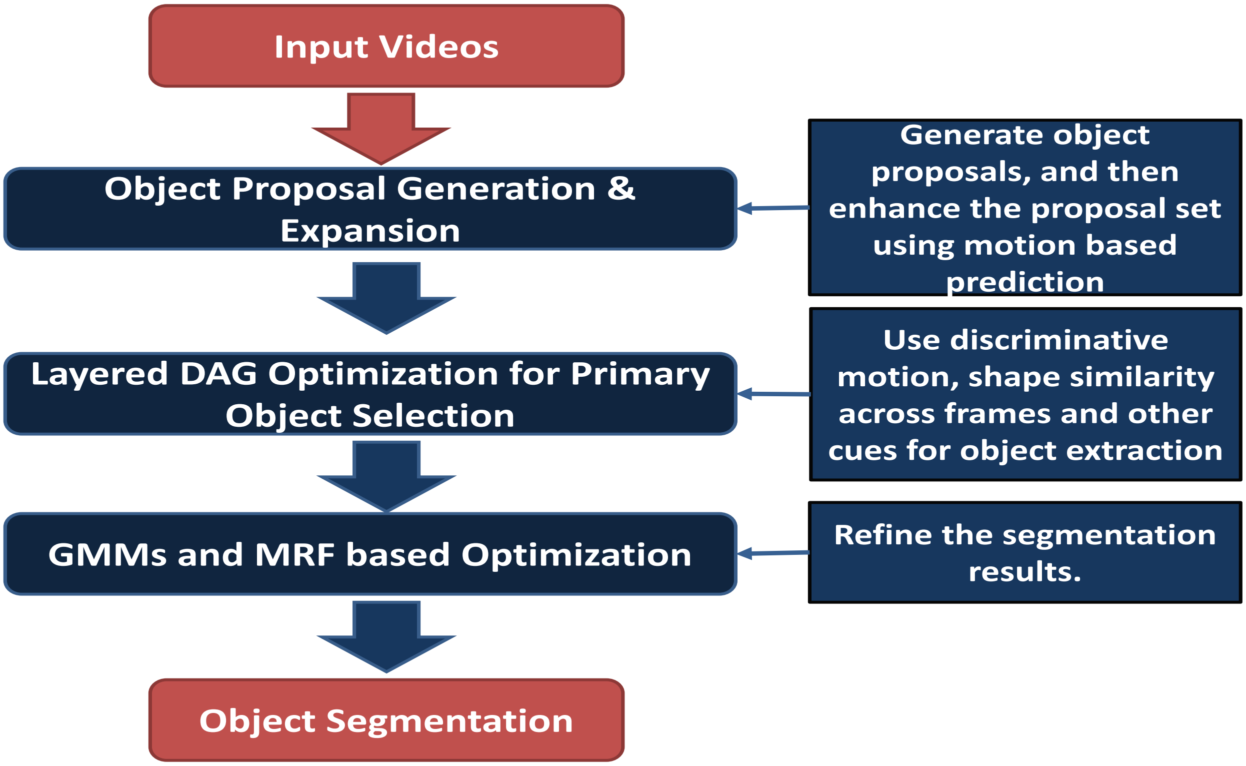

Input Videos
Object Proposal Generation & Expansion
Generate object proposals, and then enhance the proposal set using motion based prediction
Layered DAG Optimization for Primary Object Selection
Use discriminative motion, shape similarity across frames and other cues for object extraction
GMMs and MRF based Optimization
Refine the segmentation results.
Object Segmentation

Figure 3.3: The Video Object Segmentation Framework

The organization of the rest of the Chapter is as follows. In Section 3.2, the structure of the spatiotemporal is introduced, and the unary and binary weights are formulated; In Section, 3.3, the dynamic programming solution is given and analyzed; In Section 3.4, the Gaussian Mixture Models and Markov Random Field optimization are introduced for the pixel-level segmentation; In Section 3.5, the object proposal generation and expansion methods are discussed; In Section 3.6, both qualitative and quantitative experimental results for two publicly available datasets (SegTrack [112] and GaTech [40]) and some other challenging videos are shown; In Section 3.7, the Chapter is concluded.

3.2 Layered DAG based Video Object Segmentation

The proposed framework consists of 3 stages (as shown in Figure 3.3): **1**. Generation of object proposals per-frame and then expansion of the proposal set for each frame based on object proposals in adjacent frames. **2**. Generation of a layered DAG from all the object proposals in the video. The longest path in the graph fulfills the goal of maximizing objectness and similarity scores, and represents the most likely set of proposals denoting the primary object in the video. **3**. The primary object proposals are used to build object and background models using Gaussian mixtures, and a graph-cuts based optimization method is used to obtain refined per-pixel segmentation. Since the proposed approach is centered around layered DAG framework (each layer corresponds to one video frame) for selection of primary object regions, we will start with its description.

We want to extract object proposals with high objectness likelihood, high appearance similarity and smoothly varying shape from the set of all proposals obtained from the video. Also, since we want to extract the primary object only, we want to extract at most a single proposal per frame. Keeping these objectives in mind, the layered DAG is formed as follows. Each object proposal is represented by two nodes: a ‘beginning node’ and an ‘ending node’ and there are two types of edges: unary and binary edges. The unary edges have weights which measure the objectness of a proposal. The details of the function for unary weight assignments (measuring objectness) are given in Section 3.2.1. All the beginning nodes in the same frame form a layer, so as the ending nodes. A directed unary edge is built from beginning node to ending node. Thus, each video frame is represented by two layers in the graph. Directed binary edges are built from any ending node to all the beginning nodes in latter layers. The binary edges have weights which measure the appearance and shape similarity between the corresponding object proposals across frames. The binary weight assignment functions are introduced in Section 3.2.2.

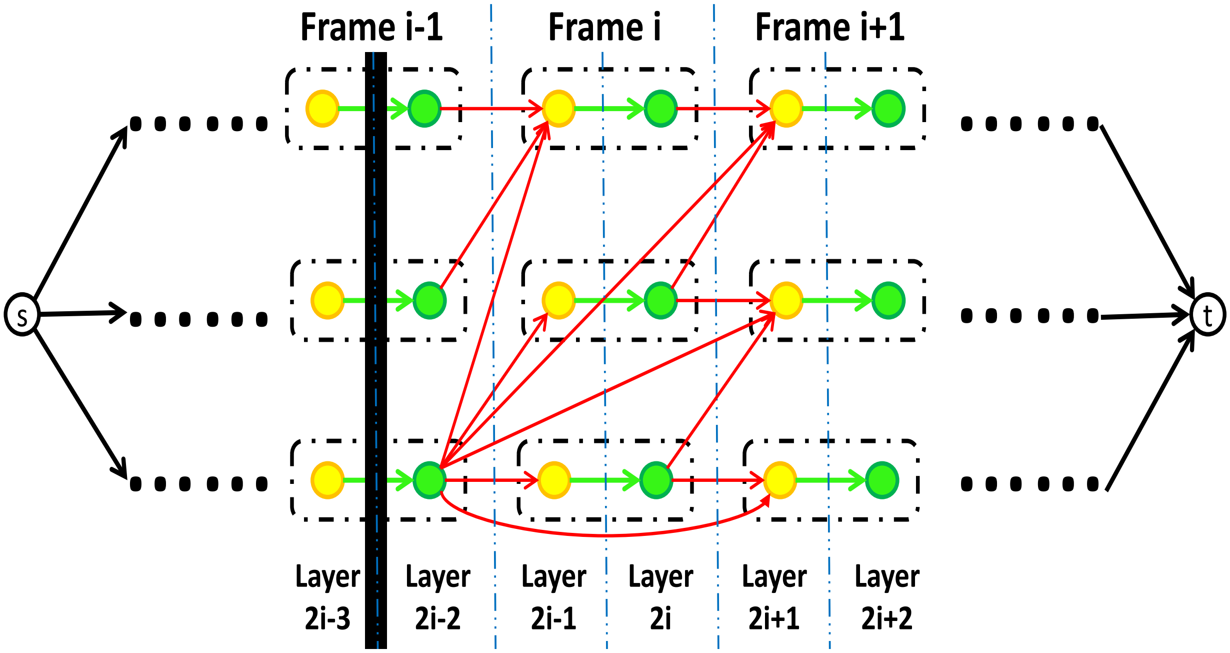


Figure 3.4: Spatiotemporal Graph for Video Object Segmentation. Node “s” and “t” are source and sink nodes respectively, which have zero weights for their edges connecting with other nodes in the graph. The yellow nodes and the green nodes are “beginning nodes” and “ending nodes” respectively and they are paired such that each yellow-green pair represents an object proposal. All the beginning nodes in the same frame are arranged in a layer and the same is the case for the ending nodes. The green edges are the unary edges and red edges are the binary edges.

Figure 3.4 is an illustration of the graph structure. It shows frame $i-1$, i and $i+1$ of the graph, with corresponding layers of $2i-3$, $2i-2$, $2i-1$, $2i$, $2i+1$ and $2i+2$. Note that, only 3 object proposals are shown for each layer for simplicity, however, there are usually hundreds of object proposals for each frame and the number of object proposals for different frames are not necessary the same. There is also a virtual source node "s" and a sink node "t" with 0 weighted edges (black edges) in the graph. Note that, it is not necessary to build binary edges from an ending node to all the beginning nodes in latter layers. In practice, only building binary edges to the next three subsequent frames is enough for most of the videos.

3.2.1 Unary Edges

Unary edges measure the objectness of the proposals. Both appearance and motion are important to infer the objectness, so the scoring function for object proposals is defined as $S_{unary}(r) = A(r) + M(r)$, in which r is any object proposal, $A(r)$ is the appearance score and $M(r)$ is the motion score. We define $M(r)$ as the average Frobenius norm of optical flow gradient around the boundary of object proposal r. The Frobenius norm of optical flow gradients is defined as:

$$\left\| U_X \right\|_F = \left\| \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \right\|_F = \sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2}, \tag{3.1}$$

where $U = (u, v)$ is the forward optical flow of the frame, u_x, v_x and u_y, v_y are optical flow gradients in x and y directions respectively.

The intuition behind this motion scoring function is that, the motions of foreground object and background are usually distinct, so boundary of moving objects usually implies discontinuity in motion. Therefore, ideally, the gradient of optical flow should have high magnitude around foreground object boundary (this phenomenon could be easily observed from Figure 3.5). In equation

3.1, we use the Frobenius norm to measure the optical flow gradient magnitude, the higher the value, the more likely the region is from a moving object. In practice, the maximum of optical flow gradient magnitude does not coincide exactly with the moving object boundary due to underlying approximation of optical flow calculation. Therefore, we dilate the object proposal boundary and get the average optical flow gradient magnitude as the motion score. Figure 3.5 is an illustration of this process. The appearance scoring function $A(r)$ is measured by the objectness ([21]).

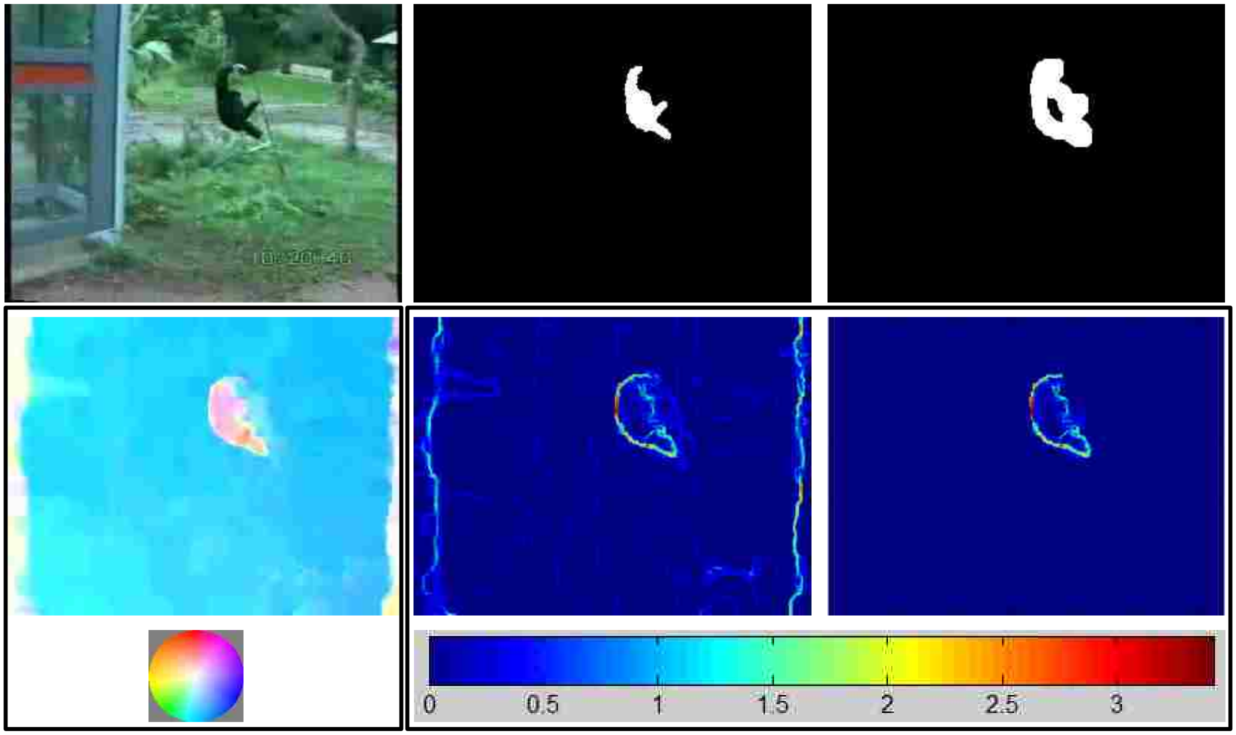


Figure 3.5: Optical Flow Gradient Magnitude Motion Scoring. In row 1, column 1 shows the original video frame, column 2 shows one of the object proposals and column 3 shows dilated boundary of the object proposal. In row 2, column 1 shows the forward optical flow of the frame, column 2 shows the optical flow gradient magnitude map and column 3 shows the optical flow gradient magnitude response for the specific object proposal around the boundary.

3.2.2 Binary Edges

Binary edges measure the similarity between object proposals across frames. For measuring the similarity of regions, color, location, size and shape are the properties to be considered. We define the similarity between regions as the weight of binary edges as follows:

$$S_{binary}(r_m, r_n) = \lambda \cdot S_{color}(r_m, r_n) \cdot S_{overlap}(r_m, r_n), \tag{3.2}$$

wherer_m and r_n are regions from frame m and n, λ is a constant value for adjusting the ratio between unary and binary edges, $S_{overlap}$ is the overlap similarity between regions and S_{color} is the color histogram similarity:

$$S_{color}(r_m, r_n) = hist(r_m) \cdot hist(r_n)^T, \tag{3.3}$$

where $hist(r)$ is the normalized color histogram for a region r,

$$S_{overlap}(r_m, r_n) = \frac{|r_m \cap warp_{mn}(r_n)|}{|r_m \cup warp_{mn}(r_n)|}, \tag{3.4}$$

where $warp_{mn}(r_n)$ is the warped region from r_n by optical flow to frame m. It is clear that S_{color} encodes the color similarity between regions and $S_{overlap}$ encodes the size and location similarity between regions. If two regions are close, and the sizes and shapes are similar, the value would be higher, and vice versa. Note that, unlike prior approaches [60, 68], we use optical flow to predict the region (i.e. encoding location and shape), and therefore we are better able to compute similarity for fast moving objects.

3.3 Dynamic Programming Solution

Until now, we have built the layered DAG and the objective is clear: to find the highest weighted path in the DAG. Assume the graph contains $2F + 2$ layers (F is the frame number), the source node is in layer 0 and the sink node is in layer $2F + 2$. Let N_{ij} denotes the jth node in ith layer and $E(N_{ij}, N_{kl})$ denotes the edge from N_{ij} to N_{kl}. Layer i has M_i nodes. Let $P = (p_1, p_2, ..., p_{m+1}) = (N_{01}, N_{j_1 j_2}, ..., N_{j_{m-1} j_m}, N_{(2n+2)1})$ be a path from source to sink node. Therefore,

$$P_{max} = arg \max_{P} \sum_{i=1}^{m} E(p_i, p_{i+1}). \tag{3.5}$$

P_{max} forms a Longest (simple) Path Problem for DAG. Let $OPT(i, j)$ be the maximum path value for N_{ij} from source node. The maximum path value satisfies the following recurrence for $i \geq 1$ and $j \geq 1$:

$$OPT(i, j) = \max_{k=0...i-1, l=1...M_k} [OPT(k, l) + E(N_{kl}, N_{ij})]. \tag{3.6}$$

This problem can be solved by dynamic programming in linear time [56]. The computational complexity for the algorithm is $O(n + m)$, in which n is the number of nodes and m is the number of edges. The most important parameter for the layered DAG is the ratio λ between unary and binary edges. However, in practice, the results are not sensitive to it, and in the experiments λ is simply set to be 1.

3.4 Per-pixel Video Object Segmentation

Once the primary object proposals are obtained in a video, the results are further refined by a graph-based method to get per-pixel segmentation results. We define a spatiotemporal graph by connecting frames temporally with optical flow displacement. Each of the nodes in the graph is a pixel in a frame, and edges are set to be the 8-neighbors within one frame and the forward-backward 18 neighbors in adjacent frames. We define the energy function for labeling $f = [f_1, f_2, ..., f_n]$ of n pixels with prior knowledge of h (from GMMs):

$$E(f, h) = \sum_{i \in S} D_i^h(f_i) + \lambda \sum_{(i,j) \in N} V_{i,j}(f_i, f_j), \tag{3.7}$$

where $S = \{p_i, ..., p_n\}$ is the set of n pixels in the video, N consists of neighboring pixels, and i,j index the pixels. p_i could be set to 0 or 1 which represents background or foreground respectively. The unary term D_i^h defines the cost of labeling pixel i with label f_i which we get from the Gaussian Mixture Models (GMM) for both color and location:

$$D_i^h(f_i) = -log(\alpha U_i^c(f_i, h) + (1 - \alpha) U_i^l(f_i, h)), \tag{3.8}$$

where $U_i^c(.)$ is the color-induced cost and $U_i^l(.)$ is the location cost.

For the binary term $V_{i,j}(f_i, f_j)$, we follow the definitions in [92]:

$$V_{i,j}(f_i, f_j) = [f_i \neq f_j] exp^{-\beta (C_i - C_j)^2}, \tag{3.9}$$

where $[.]$ denotes the indicator function taking values 0 and 1, $(C_i - C_j)^2$ is the Euclidean distance between two adjacent nodes in RGB space, and $\beta = (2 \sum (C_i - C_j)^2)^{-1}|_{(i,j) \in N}$.

We use the graph-cuts based minimization method in [33] to obtain the optimal solution for equation 3.7, and thus get the final segmentation results. Next, we describe the method for object proposal generation that is used to initialize the video object segmentation process.

3.5 Object Proposal Generation & Expansion

In order to achieve our goal of identifying image regions belonging to the primary object in the video, it is preferable (though not necessary) to have an object proposal corresponding to the actual object in each frame in which object is present. Using only appearance or optical flow based cues to generate object proposals is usually not enough for this purpose. This phenomenon could be observed in the example shown in Figure 3.6. For frame i in this figure, hundreds of object proposals were generated using method in [21], however, no proposal is consistent with the true object, and the object is fragmented between different proposals.

We assume that an object's shape and location changes smoothly across frames and propose to enhance the set of object proposals for a frame by using the proposals generated in its adjacent frames. The object proposal expansion method works by the guidance of optical flow (see Figure 3.6). For the forward version of object proposal expansion, each object proposal r_{i-1}^{k} in frame $i-1$ is warped by the forward optical flow to frame i, then a check is made if any proposal r_i^j in frame i has a large overlap ratio with the warped object proposal, i.e.,

$$o = \frac{|warp_{i-1,i}(r_{i-1}^{k}) \cap r_i^j|}{|r_i^j|}. \tag{3.10}$$

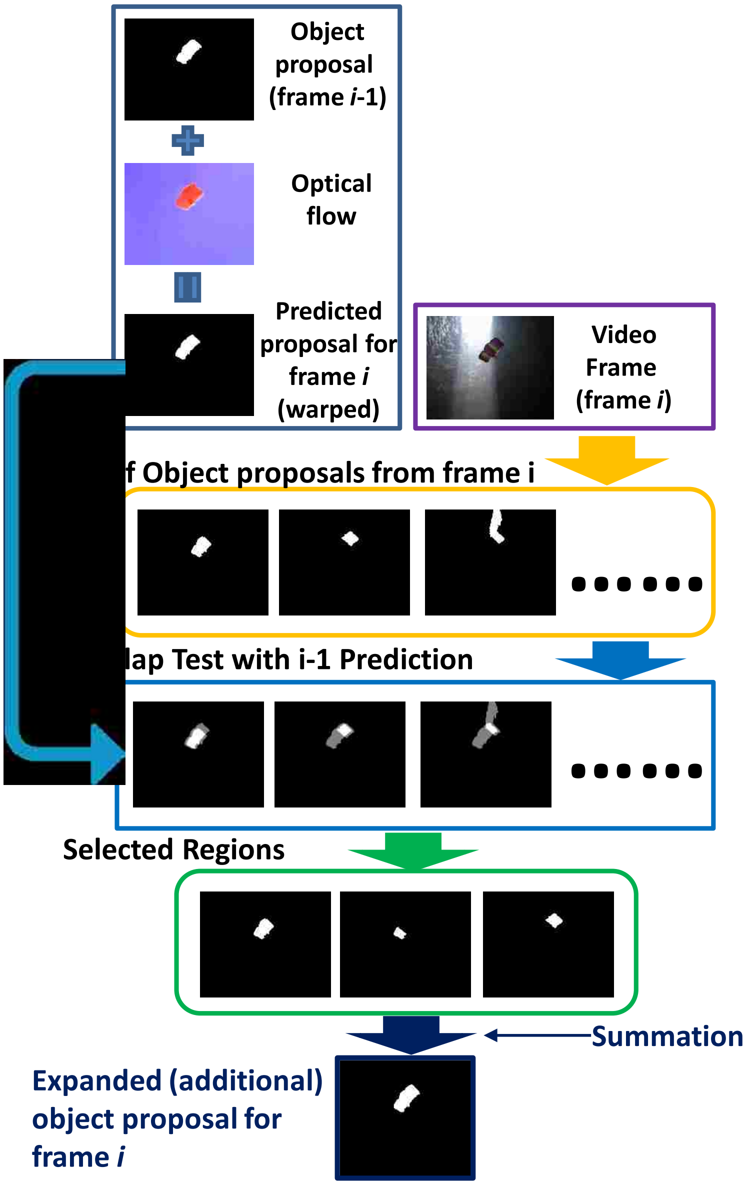

Object proposal (frame *i*-1)
Optical flow
Predicted proposal for frame *i* (warped)
Video Frame (frame *i*)
f Object proposals from frame i
....
lap Test with i-1 Prediction
....
Selected Regions
Summation
Expanded (additional) object proposal for frame *i*

Figure 3.6: Object Proposal Expansion. For each optical flow warped object proposal in frame $i-1$, we look for object proposals in frame i which have high overlap ratios with it. If multiple object proposals have high overlap ratios with the warped proposal, they are merged into a new large object proposal. This process will produce the right object proposal if it is not discovered by in frame i, but will be discovered by our method in frame $i-1$.

The contiguous overlapped areas, for regions in i+1 with o greater than 0.5, are merged into a single region, and are used as additional proposals. Note that, the old original proposals are also kept, so this is an ‘expansion’ of the proposal set, and not a replacement. In practice, this process is carried out both forward and backward in time. Since it is an iterative process, even if suitable object proposals are missing in consecutive frames, they could potentially be produced by this expansion process. Figure 3.6 shows an example image sequence where the expansion process resulted in generation of a suitable proposal.

3.6 Experiments

The proposed method was evaluated using two well-known segmentation datasets: SegTrack dataset [112] and GaTech video segmentation dataset [40]. Quantitative comparisons are shown for SegTrack dataset since ground-truth is available for this dataset. Qualitative results are shown for GaTech video segmentation dataset. We also evaluated the proposed approach on additional challenging videos.

3.6.1 SegTrack Dataset

There are 6 videos in this dataset, and also a pixel-level segmentation ground-truth for each video is available. We follow the setup in the literature ([60, 68]), and use 5 (birdfall, cheetah, girl, monkeydog and parachute) of the videos for evaluation (since the ground-truth for penguin is not useable). We use an optical flow magnitude based model selection method to infer the camera motion: for static cameras, a background subtraction cue is also used for moving object extraction; for all the results shown in this Section, the static camera model was only selected (automatically) for the “birdfall” video.

Table 3.1: Quantitative results and comparison with the state of the art on SegTrack dataset

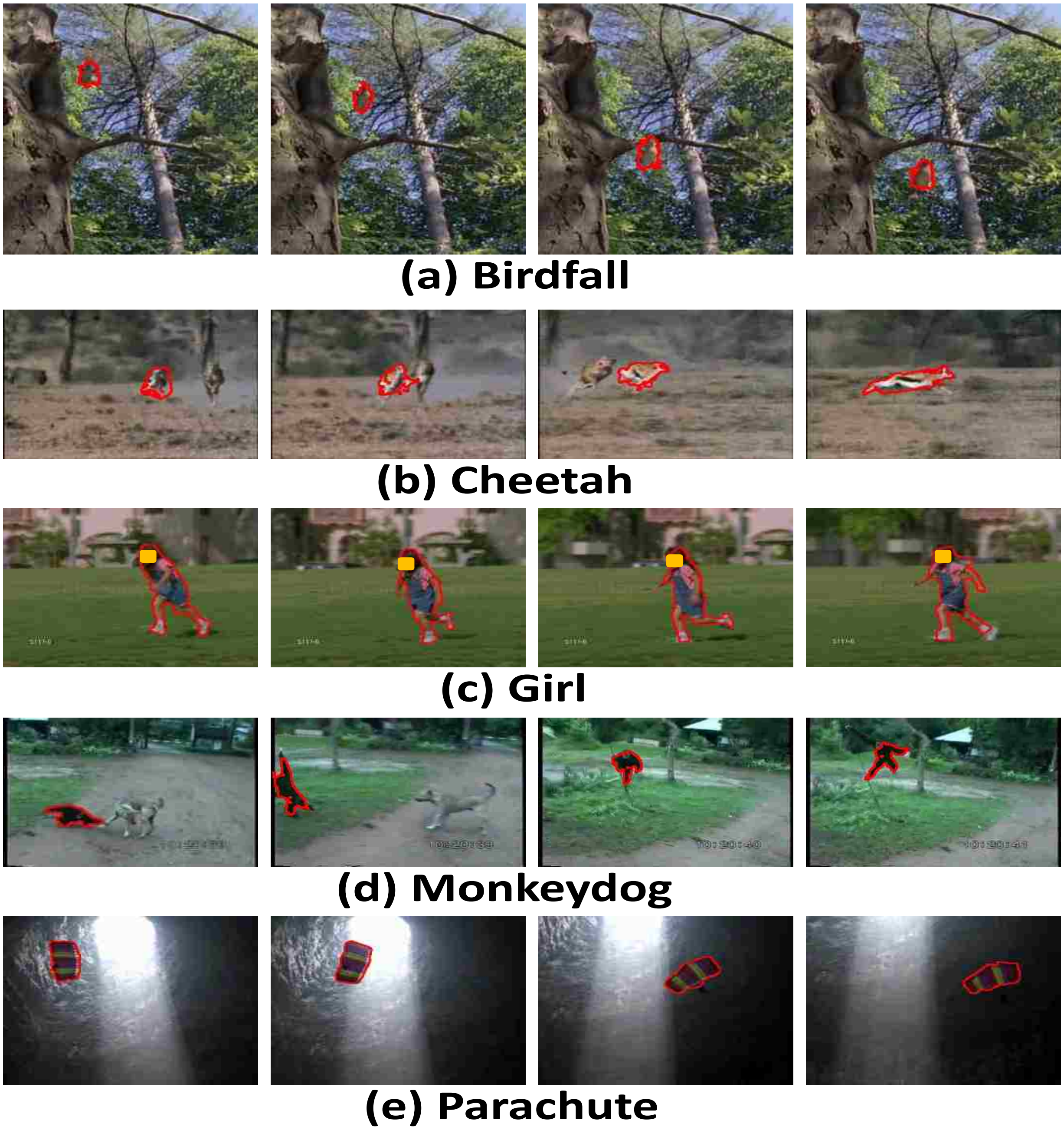
Video	Ours	[68]	[60]	[112]	[17]
birdfall	**155**	189	288	252	454
cheetah	**633**	806	905	1142	1217
girl	**1488**	1698	1785	1304	1755
monkeydog	**365**	472	521	563	683
parachute	220	221	**201**	235	502
Avg.	**452**	542	592	594	791
supervised?	N	N	N	Y	Y

We compare our method with 4 state-of-the-art methods [68], [60], [112] and [17] shown in Table 3.1. Note that our method is unsupervised, and it outperforms all the other unsupervised methods except for the parachute video where it is a close second. Note that [112] and [17] are supervised methods which need an initial annotation for the first frame. The results in Table 3.1 are the average per-frame pixel error rate compared to the ground-truth. The definition is [112]:

$$error = \frac{XOR(f, GT)}{F}, \tag{3.11}$$

where f is the segmentation labeling results of the method, GT is the ground-truth labeling of the video, and F is the number of frames in the video. Figure 3.7 shows qualitative results for the videos of SegTrack dataset.

Figure 3.8 is an example that shows the effectiveness of the proposed layered DAG approach for temporally dense extraction of primary object regions. The figure shows consecutive frames (frame 38 to frame 43) from “monkeydog” video. The top 2 rows show the results of key-frame object extraction method [60], and the bottom 2 rows show our object region selection results. As one can see, [60] detects the primary object proposal in only one of the frames, however, by using the proposed approach, we can extract the primary object region from all the frames.


(a) Birdfall

(b) Cheetah

(c) Girl

(d) Monkeydog

(e) Parachute

Figure 3.7: SegTrack dataset results. The regions within the red boundaries are the segmented primary objects.

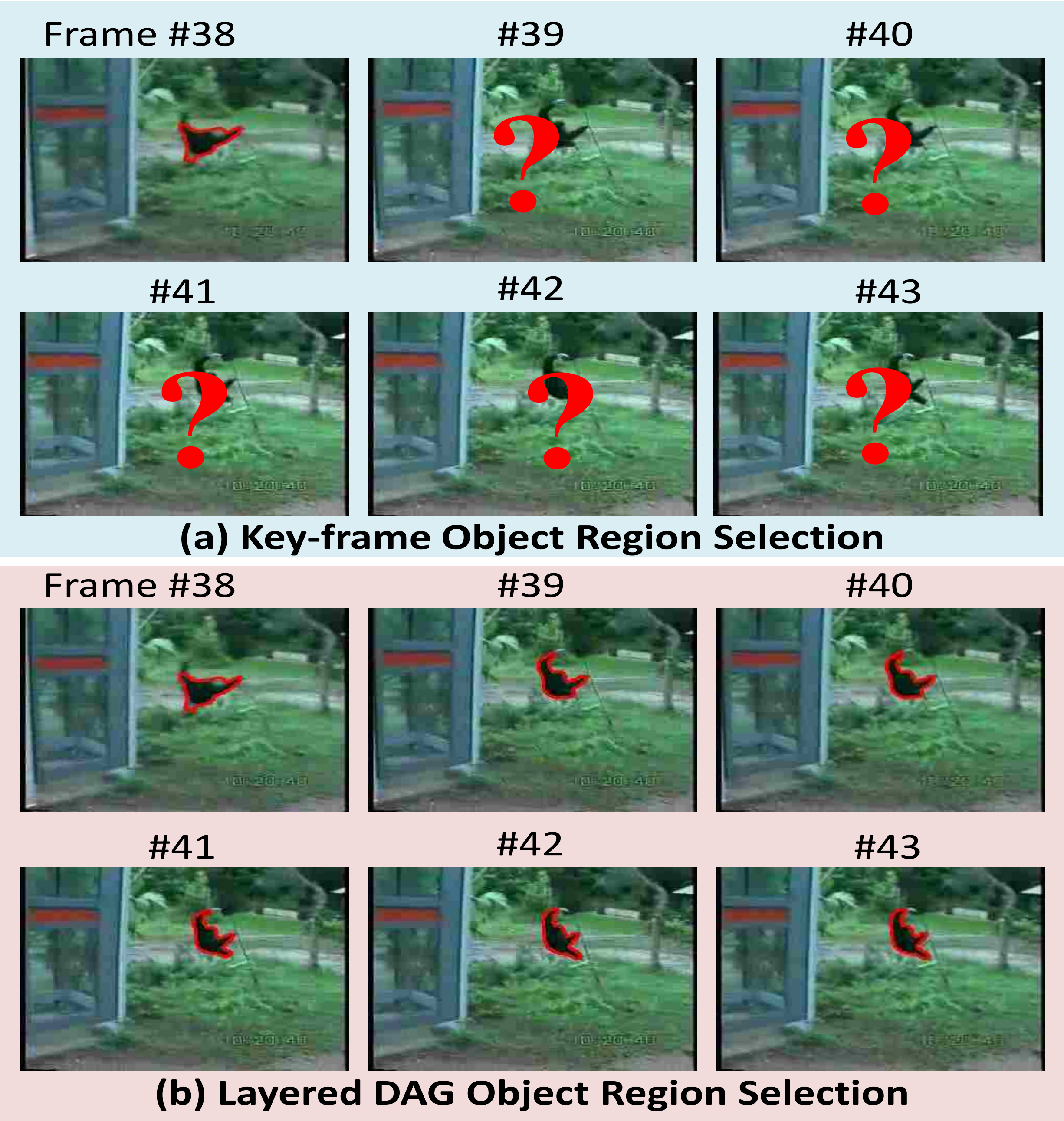

Frame #38
#39
#40
#41
#42
#43
(a) Key-frame Object Region Selection
Frame #38
#39
#40
#41
#42
#43
(b) Layered DAG Object Region Selection

Figure 3.8: Comparison of object region selection methods. The regions within the red boundaries are the selected object regions. “?” means there is no object region selected by the method. Numbers above are the frame indices.[Please Print in Color]

Table 3.2: Quantitative Results on Persons and Cars dataset

Video	Average per-frame pixel error
Surfing	**1209**
Jumping	**835**
Skiing	**817**
Sliding	**2228**
Big car	**1129**
Small car	**272**

3.6.2 GaTech Segmentation Dataset

We also evaluated the proposed method on GaTech video segmentation dataset. We show qualitative comparison of results between the proposed approach and the original bottom-up method in Figure 3.9. As one can observe, our method is able to segment the true foreground object from the background. The method in [40] doesn't use an object model which induces over-segmentation (although the results are very good for the general segmentation problem).

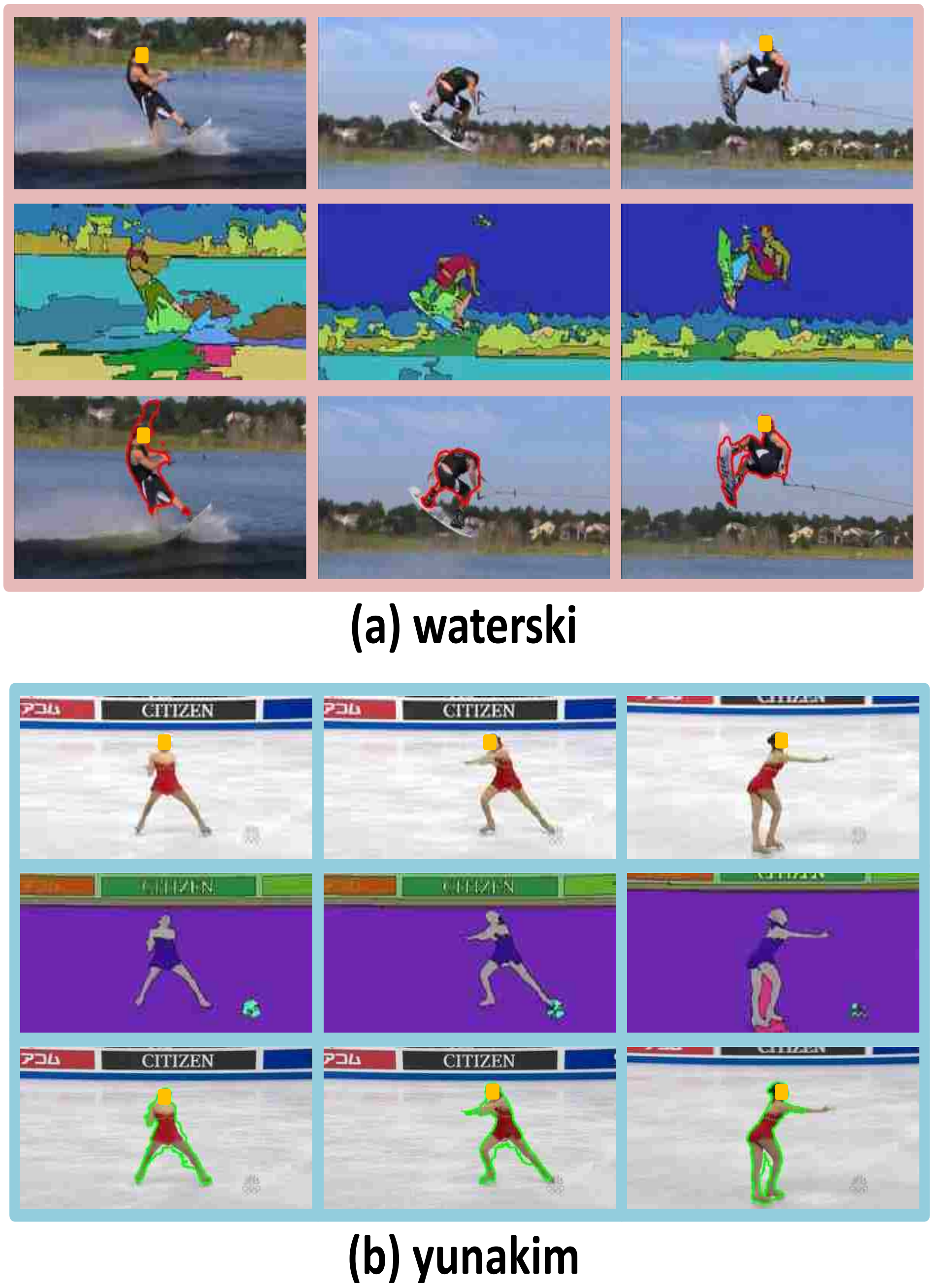
3.6.3 Persons and Cars Segmentation Dataset

We have built a new dataset for video object segmentation. The dataset is challenging: persons are in a variety of poses; cars have different speeds, and when they are slow, it is very hard to do motion segmentation. We generate ground truth for those videos. Figure 3.10 shows some sample results from this dataset, and Table 3.6.3 shows the quantitative results for this dataset (the average per-frame pixel error is defined as the same as SegTrack dataset [112]). Please visit http://crcv.ucf.edu for more details.

3.7 Summary

We have proposed a novel and efficient layered DAG based approach to segment the primary object in videos. The major contribution is that it formulate the problem into a directed acyclic graph, for which the inference can be efficiently done by dynamic programming. The result from the graph inference are consistent object proposals from all the video frames. This approach also uses innovative mechanisms to compute the 'objectness' of a region and to compute similarity between object proposals across frames. A final step with GMM and MRF ensures an accurate pixel-level video segmentation. The proposed approach outperforms the state of the art on the well-known SegTrack dataset. We also demonstrate good segmentation performance on additional challenging data sets. Similar ideas can potentially be applied to other problems, such as object tracking, image registration, etc.

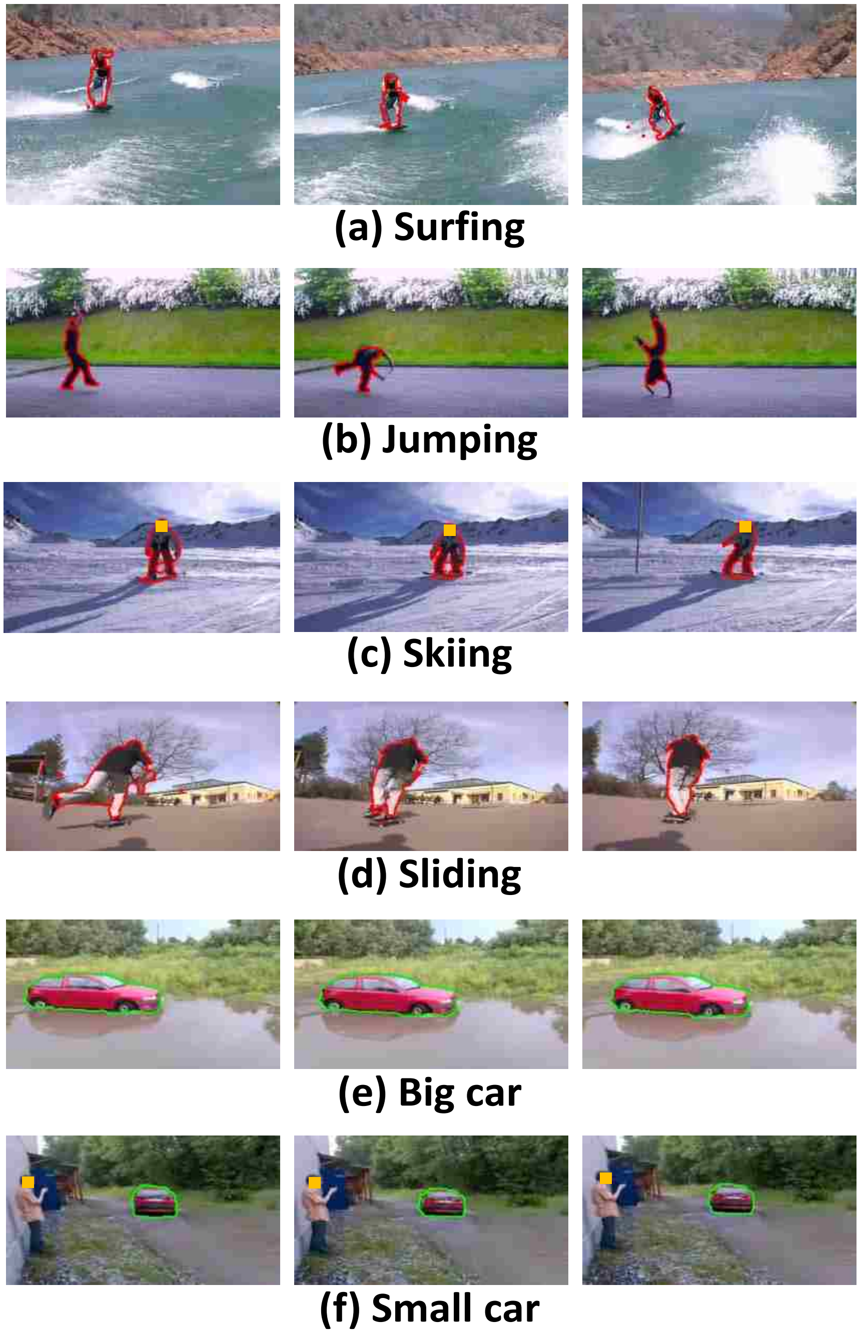
In order to extend the spatiotemporal graph formulation to handle multiple videos, in next Chapter we will employ it to solve the video object co-segmentation problem.



(a) waterski

(b) yunakim

Figure 3.9: Object Segmentation Results on GaTech Video Segmentation Dataset. Row 1: orignial frame, Row 2: Segmentation results by the bottom-up segmentation method [40]. Row 3: Video object segmentation by the proposed method. The regions within the red or green boundaries are the segmented primary objects. [Please Print in Color]



(a) Surfing

(b) Jumping

(c) Skiing

(d) Sliding

(e) Big car

(f) Small car

Figure 3.10: Sample Results on Persons and Cars Dataset. Please go to http://crcv.ucf.edu for more details.

CHAPTER 4: SPATIOTEMPORAL GRAPHS FOR VIDEO OBJECT CO-SEGMENTATION

4.1 Introduction

In the previous Chapter, we have investigated the problem of segmenting an object from a single video. A natural question would be: can multiple videos be used together to improve the segmentation for each video? And this question leads to a new problem: video object co-segmentation. Our goal is to discover and segment objects from a video collection in an unsupervised manner. We compensate for the lack of supervision by exploiting commonality of objects in video collection (if it exists) to build better object segmentation models. Unsupervised segmentation of objects from a video collection has applications in large scale video tagging and retrieval, generation of training sets for supervised learning, and forensic video analysis.

The proposed method first generates object proposal tracklets in different videos. It then groups the tracklets into Maximum Weight Cliques by a modified version of Bron-Kerbosch Algorithm. Finally it employs Gaussian Mixture Models and Markov Random Fields to obtain the pixel-level segmentations. Fig.4.1 shows an illustration of the proposed approach. Compared to the existing video co-segmentation methods, the proposed approach has the following advantages:

1. The proposed method employs object tracklets to obtain spatially salient and temporally consistent object regions for co-segmentation, while most of previous co-segmentation methods simply use pixel-level or region-level features to perform clustering. The perceptual grouping of pixels before matching reduces segment fragmentation and leads to a simpler matching problem.

2. The proposed approach does not rely on approximate solutions for object groups. The grouping

problem is modeled as a Regulated Maximum Weight Clique (RMWC) problem for which an optimal solution is available. The use of only the salient object tracklets for grouping keeps the computational cost low.

3. Unlike the state-of-the-art single video object segmentation method ([134]), the proposed method can handle occlusions of objects, or objects going in and out of videos because the object tracklets are temporally local and there is no requirement for the object to continuously remain in the field of view of the video. Furthermore, there is no limitation on the number of object classes in each video and the number of common object classes in the video collection. Therefore, the approach can be used to extract objects in an unsupervised fashion from general video collections.

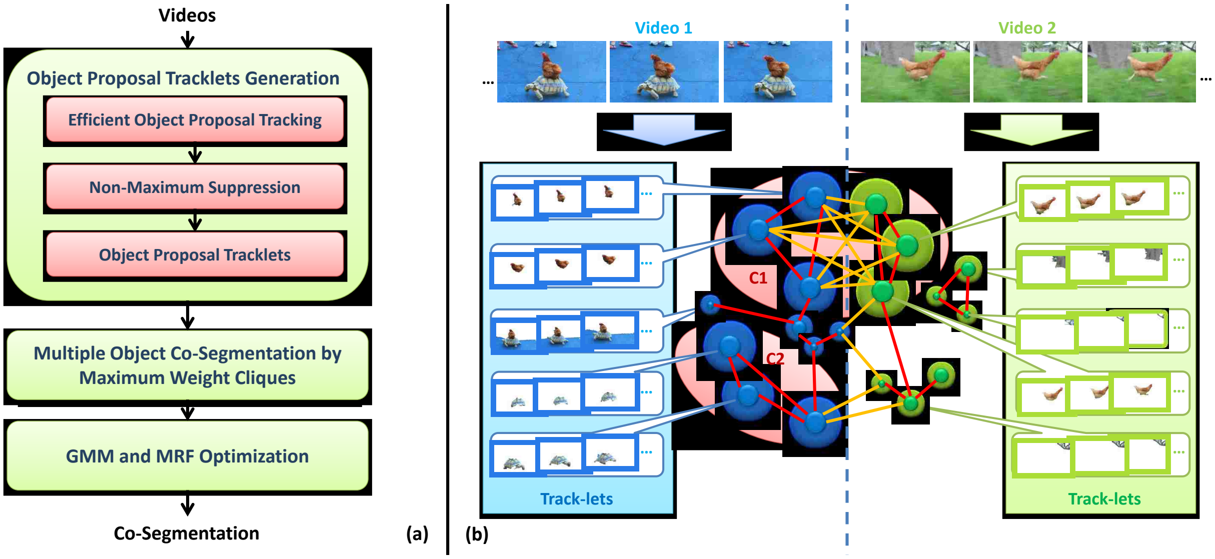

Videos
Object Proposal Tracklets Generation
Efficient Object Proposal Tracking
Non-Maximum Suppression
Object Proposal Tracklets
Multiple Object Co-Segmentation by Maximum Weight Cliques
GMM and MRF Optimization
Co-Segmentation
(a)
(b)
Video 1
Video 2
C1
C2
Track-lets
Track-lets

Figure 4.1: (a) Shows the framework of the proposed method. (b) Shows the spatiotemporal graph for video object co-segmentation. In this example, we generate 'object proposal' tracklets for two videos and use weighted nodes to represent them. Edges are built between similar nodes, and there are two types of edges (intra-video edges: red; inter-video edges: orange). In this example, the first two maximal cliques, which have highest weights are obtained ($C1$ and $C2$). $C1$ contains all the segments for 'chicken' from two videos and $C2$ contains all the segments for 'turtle' in video 1.

4. The proposed method is different from Maximum Weight Clique Problem, which has already been explored in video object segmentation [69], in a way that the clique weights of the proposed method are not simply defined as the summation of node weights, but regulated by the intra-clique consistency term. Therefore, the extracted cliques have more global consistency, and similar objects from different videos are accurately grouped.

The organization of the rest of the Chapter is as follows: In Section 4.2, we describe the framework of the proposed method. In Section 4.3 we introduce how to generate the object proposal tracklets accurately. In Section 4.4 we introduce the proposed Regulated Maximum Weighted Clique (RMWC) formulation for video object co-segmentation. In Section 4.5, we present the performance evaluation of the proposed algorithm. In Section 4.6., the Chapter is concluded.

4.2 The Framework

The proposed method consists of two stages: **(1)** Object Tracklets Generation: In this stage, we generate a number of object proposals ([21]) for each frame and use each of them as a starting point, and track the object proposals backward and forward throughout the whole video sequence. We generate reliable tracklets from the track set (those with high similarity over time) and perform non-maxima suppression to remove noisy or overlapping proposals. **(2)** Multiple Objects Co-Segmentation by Regulated Maximum Weight Cliques: A graph is generated by representing each tracklet as a node from all videos in the collection. The nodes of the graph are weighted by their appearance and motion scores, and edges are weighted by tracklet similarity. Edges with weight below a threshold are removed. A Regulated Maximum Weight Clique extraction algorithm is used to find objects ranked by score which is a combination of intra-group consistency and *Video Object Scores*. The object regions obtained from the video sets are used to initialize per-pixel segmentation [33] to get the final co-segmentation results.

4.3 Object Tracklets Generation

In this stage, the method in [21] is employed to generate a number of object proposals (which are likely to be 'object regions' in each frame). The method in [21] generates many candidate object regions for an image. It extracts multiple features to infer whether a candidate region is an object region, and assigns a score to each region. For each of the object proposals we define a ***Video Object Score***, S^{object}, which is a combination of motion and appearance information:

$$S^{object}(x) = A(x) + M(x), \tag{4.1}$$

where x is an object proposal, and $A(x)$ is the appearance score (which is the objectness score defined by [21]). The appearance objectness score is high for regions that have a well defined closed boundary in space, different appearance from its surrounds and is salient [21], and $M(x)$ is the motion score (which is defined in [134] as the average Frobenius norm of optical flow gradient around the boundary of object proposal).

4.3.1 Efficient Object Proposal Tracking

We track every object proposal from each frame backward and forward to form a number of tracks for the object proposals (please see Fig.4.2).

A combined color, location, and shape similarity function is employed for object proposal tracking:

$$S^{simi}(x_m, x_n) = S^{app}(x_m, x_n) \cdot S^{loc}(x_m, x_n) \cdot S^{shape}(x_m, x_n), \tag{4.2}$$

where x_m and x_n are object proposals from frame m and n respectively, S^{app} is the appearance similarity, S^{loc} is the location similarity which computes the overlap ratio between two regions, and S^{shape} is the shape similarity between the object proposals. Color histograms are used to model appearance. The descriptor for estimating shape similarity is computed by representing the contour of a region in normalized polar coordinates and sampling it from 0 to 360 degrees to form a vector. Dot products of descriptors are used for computing both shape similarity and appearance similarity.

Once the similarity function is defined, a simple greedy tracking method is employed to track large number of object proposals. By using the similarity scores defined in Eq.4.2, for a specific object proposal in the frame, the most similar object proposal in adjacent frame is selected to be the tracked proposal. The reason for using this method is mainly due to efficiency. As shown in Fig.4.2, the similarity matrices between all object proposals in adjacent frames are pre-computed. Based on the greedy method, tracking a specific object proposal to the next frame amounts to finding the index of max value in a specific row of the similarity matrix. Thus this tracking process is computationally economical.

4.3.2 Non-Maximum Suppression for Object Proposal Tracks

One can sample a large number of proposals per frame and, therefore, generate a larger number of tracks for an input video. Specifically, for a video that has F frames and each frame has N object proposals, $F \times N$ tracks could be obtained, since we generate tracks for each proposal. However, many of the object samples are overlapping and therefore their tracks are similar. A non-maximum suppression (NMS) ([25]) scheme is used to prune near duplicate tracks. For each object proposal track $X = \{x_1, ..., x_i, ...x_F\}$, an overall *Video Object Score* is computed as:

$$S^{object}(X) = \sum_{i=1}^{F}(S^{object}(x_i)), \tag{4.3}$$

where i is the frame index, and F is the number of frames.

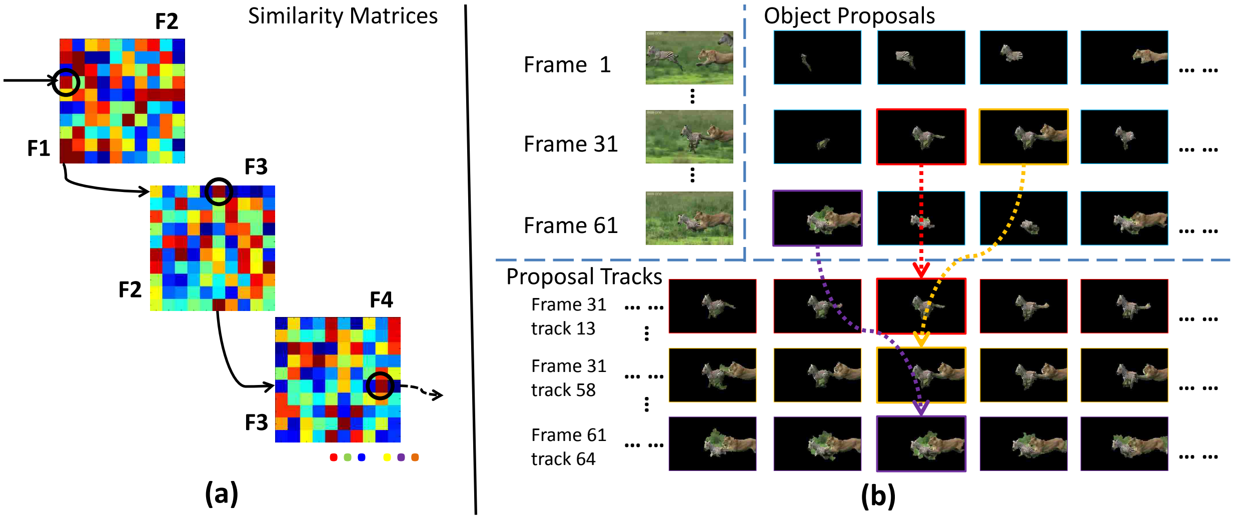


Figure 4.2: Object Proposal Tracking. (a) Shows the similarity matrices between F1 (frame 1) and F2, F2 and F3, and F3 and F4. It also shows an example for tracking a specific object proposal (the 4th in F1): first, the largest item from row 4 of similarity matrix F1 and F2 (the 1st item in this example) is found; then, the largest item from row 1 of similarity matrix F2 and F3 is found; and so on. Note that, only 10 object proposals (the matrices are 10 by 10) are shown in this figure for simplicity, but hundreds of objects proposals are used in the experiments. (b) Shows some object proposal tracks. In this example, several object proposals are generated for frame 31, and the object proposal shown in red box is tracked backward and forward to form a track throughout all the video frames. The same process is repeated for other object proposals (in orange and purple boxes as another two examples). This process is repeated for all the frames.

Next, the track that has the highest score is selected and all other tracks which have high overlap ratio $R^{overlap}$ with the selected track are removed. The value 0.5 is used for $R^{overlap}$, as suggested in ([25])). After that, the track with the second highest score among the surviving tracks is selected and the process is repeated. The process is continued iteratively until all tracks have been processed. The overlap ratio between two tracks X and Y is defined as:

$$R^{overlap}(X,Y) = \frac{\sum_{i=1}^{F}(x_i \cap y_i)}{\sum_{i=1}^{F}(x_i \cup y_i)}, \tag{4.4}$$

where x_i and y_i are object proposals in the track X and Y respectively, and F is the number of frames for the video.

After the non-maximum suppression, typically only a small percentage of the total tracks (prior to NMS) are retained. To ensure validity of the track associations, we remove associations that are 1.5 standard deviations away from the mean track similarity (shown in Fig.4.3). This reduces the likelihood of a single track containing different objects.

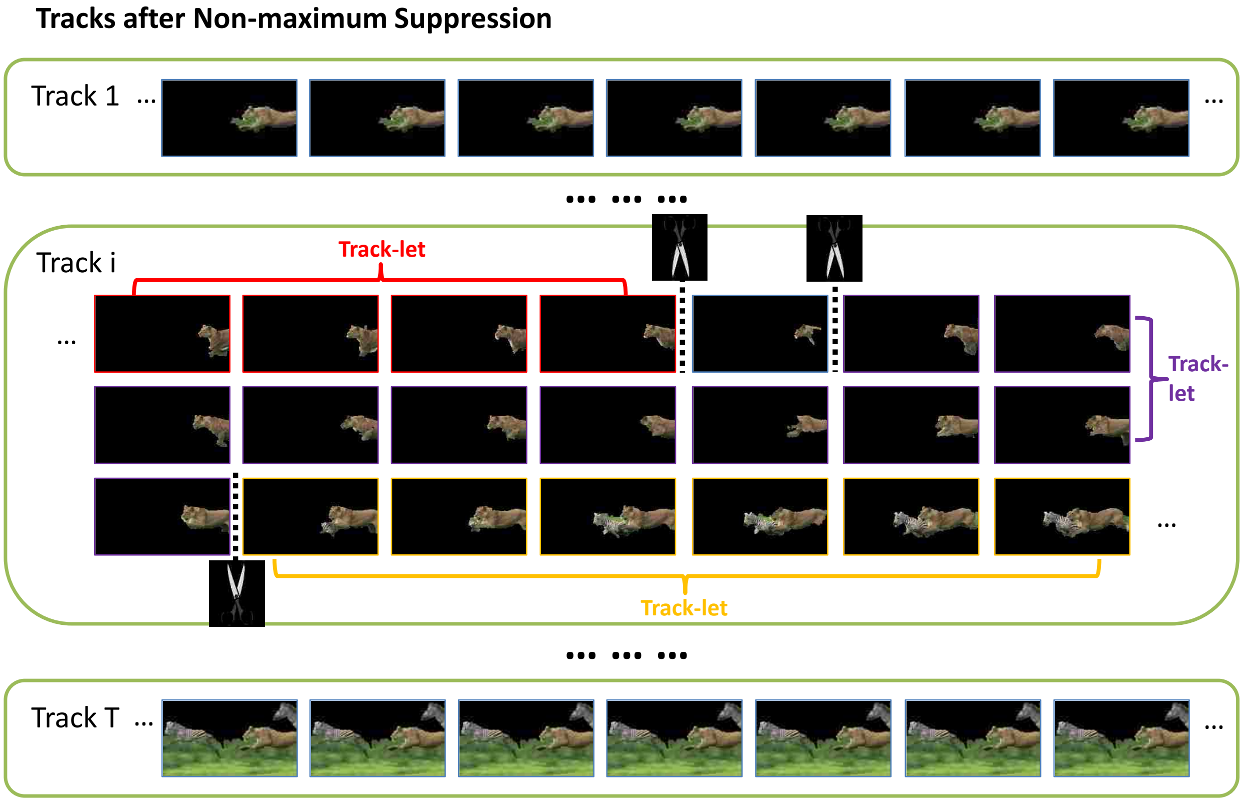


Figure 4.3: Tracklet Splitting. In this example, after the Non-Maximum Suppression, there are T object proposal tracks selected. Track i is shown as an example to generate the object proposal tracklets. There are several adjacent frames which are not very similar compared to other adjacent frame pairs, therefore, they are split and several tracklets are generated (red, orange and purple).

4.4 Multiple Object Co-Segmentation by Regulated Maximum Weight Cliques

Once object tracklets from the video collection have been obtained (Section 4.3), the next step is to discover salient object groupings in the video collection. We formulate the grouping problem as a Regulated Maximum Weight Clique Problem.

4.4.1 Clique Problems

Let $G = (V, E, W)$ be an undirected graph, where V is the set of vertices, E is the set of edges and W is a set of weights for each vertex. A clique is a complete subgraph of G, i.e. one whose vertices are pairwise adjacent. A **Maximal Clique** is a complete subgraph that is not contained in any other complete subgraph [58]. **Finding All Maximal Cliques** from a graph is NP-hard and Bron-Kerbosch Algorithm [6], which has the worst case time complexity $O(3^{(n/3)})$, is known to be the most efficient algorithm in practice ([109]). The **Maximum Clique** Problem is to find maximum complete subgraph of G. The **Maximum Weight Clique** Problem deals with finding the clique which has maximum weight.

4.4.2 Problem Constraints

We use the following constraints for co-segmenting the objects from videos:

1. The object proposal tracklets for the same class of objects should have similar appearance both within a video and across videos; however, due to the illumination differences across videos, for building color histograms in LAB space ([22]), the L channel (which represents the brightness) is only used for tracklets from the same video (intra-video edges), but a, b channels are used for tracklets from both same (intra-video edges) and different videos (inter-

video edges).

2. The shape of the object in the same video would not change significantly, so the shape similarity is also used for building the edges for tracklets of the objects in a video.

3. The dominant objects should have high *Video Object Scores*.

4. The tracklets generated by an object should have low appearance variation. Based on these constraints, the graph is built as illustrated in Fig.4.1.

4.4.3 Graph Structure

The co-segmentation problem is formulated into a Regulated Maximum Weight Cliques Problem by representing the object proposal tracklets to be the nodes. Based on constraints 1 and 2, edges are built between tracklets. There are two types of edges: intra-video edges and inter-video edges. The intra-video edge values are computed as a combined color histogram similarity in LAB color space and shape similarity:

$$\begin{aligned} E(X,Y) =&(shape(X) \cdot shape(Y)^T) \cdot \\ &\prod_{i=\{L,a,b\}} (hist(LAB_i(X)) \cdot hist(LAB_i(Y))^T), \end{aligned} \tag{4.5}$$

where $shape(X)$ and $shape(Y)$ are the shape descriptors (Sec.4.3.1) for object proposal tracklet X and Y respectively. The nearest two object proposals in the two tracklets are selected to represent the shapes of the tracklets.

And the inter-video edge values are computed as color histogram similarity of $\{a, b\}$ channels in

LAB color space:

$$E(X,Y) = \prod_{i=\{a,b\}} (hist(LAB_i(X)) \cdot hist(LAB_i(Y))^T). \tag{4.6}$$

After computing the edges, the weak edges are removed (by a threshold).

4.4.4 Regulated Maximum Weight Clique Extraction

Based on constraint 3 and according to Equation 4.1, the weight of a node (object proposal tracklet) is computed as:

$$W(X) = \sum_{i=1}^{f} (S^{object}(x_i)), \tag{4.7}$$

where f is the number of object proposals in this tracklet. $W(X)$ is the sum up of the *Video Object Score* of all object proposals contained in this tracklet.

Based on constraint 4, the weight of a clique is defined as:

$$W(C) = \Gamma_{hist}(C) \cdot \sum_{i=1}^{n(C)} (W(X_i)), \tag{4.8}$$

where $C = \{X_1, ..., X_{n(C)}\}$ is a clique, X_i is a node (tracklet) contained in this clique, $n(C)$ is the number of nodes in this clique, and $\Gamma_{hist}(C)$ is the color histogram consistency regulator which computes the mean color histogram consistency of all the object proposals contained in the clique:

$$\Gamma_{hist}(C) = \frac{\sum_{i=1}^{f(C)} \sum_{(j=1 \wedge j \neq i)}^{f(C)} (hist(x_i) \cdot hist(x_j)^T)}{f^2(C) - f(C)}, \tag{4.9}$$

where x_i and x_j are object proposals in clique C, $f(c)$ is the number of object proposals in this clique, and $hist(\cdot)$ is the $\{a, b\}$ channel color histogram in LAB space.

By this formulation, the clique that has the highest score represents the object with largest combined score of inter-object consistency and objectness. This problem is different from Maximum Weight Clique problem and can not be solved by standard methods ([58, 43]), because the clique weights are not simply defined as the summation of node weights and the weights varies over iterations as we extract objects one by one. Therefore, we call this as **Regulated Maximum Weight Cliques Problem**. Note that, we want to retrieve all Regulated Maximum Weighted Cliques as possible objects. This is achieved through iteratively finding and removing the Regulated Maximum Weight Cliques from the graph to get a ranked list of cliques (i.e. objects).

A modified version of Bron-Kerbosch Algorithm ([6]) which also has a worst-case complexity of $O(3^{(n/3)})$ is proposed to solve this problem:

1. Apply Bron-Kerbosch Algorithm to find all the maximal cliques from the graph.
2. Compute the weight of each clique in linear time.
3. Find the clique with the highest weight and remove all the nodes associated with this clique, update the clique structures and recompute the weights. This process could be performed for multiple times in order to extract multiple object groupings from the videos.

Note that, the high-complexity doesn't prohibit the use of this algorithm. The object tracklets generation stage removes most of the spurious tracklets. For videos evaluated in this dissertation, the maximum clique extraction process took less than a second on a standard laptop. The object regions obtained from the video sets are used to initialize per-pixel segmentation [33] to get the final co-segmentation results.

4.5 Experiments

The proposed method was tested on the video co-segmentation dataset (MOViCS dataset ([16])) and was compared with several other methods. The results show that it performs better both qualitatively and quantitatively. Detailed analysis is presented to show that the co-segmentation method produces better segmentation results by using information from multiple videos. Results also show that the proposed method could handle occlusions for which the state-of-the-art single video object segmentation method fails.

4.5.1 MOViCS Dataset

To the best of our knowledge, MOViCS dataset ([16]) is the only video co-segmentation dataset which has the ground truth annotations for quantitative analysis. It contains 4 video sets which totally has 11 videos, 5 frames of each video have pixel-level annotations for the object labels.

4.5.1.1 Experimental Setup

Following the setup in [16], the intersection-over-union metric is employed to quantify the results: $M(S, G) = \frac{S \cup G}{S \cap G}$, where S is a set of segments and G is the ground truth. The co-segmentation score for a set of video is defined as $Score_j = \max_i M(S_i, G_j)$, where S_i denotes all segments grouped into an object class i. And a single average score is defined for all object classes as: $Score = \frac{1}{C} \sum_j Score_j$, where C is the number of object classes in the ground truth.

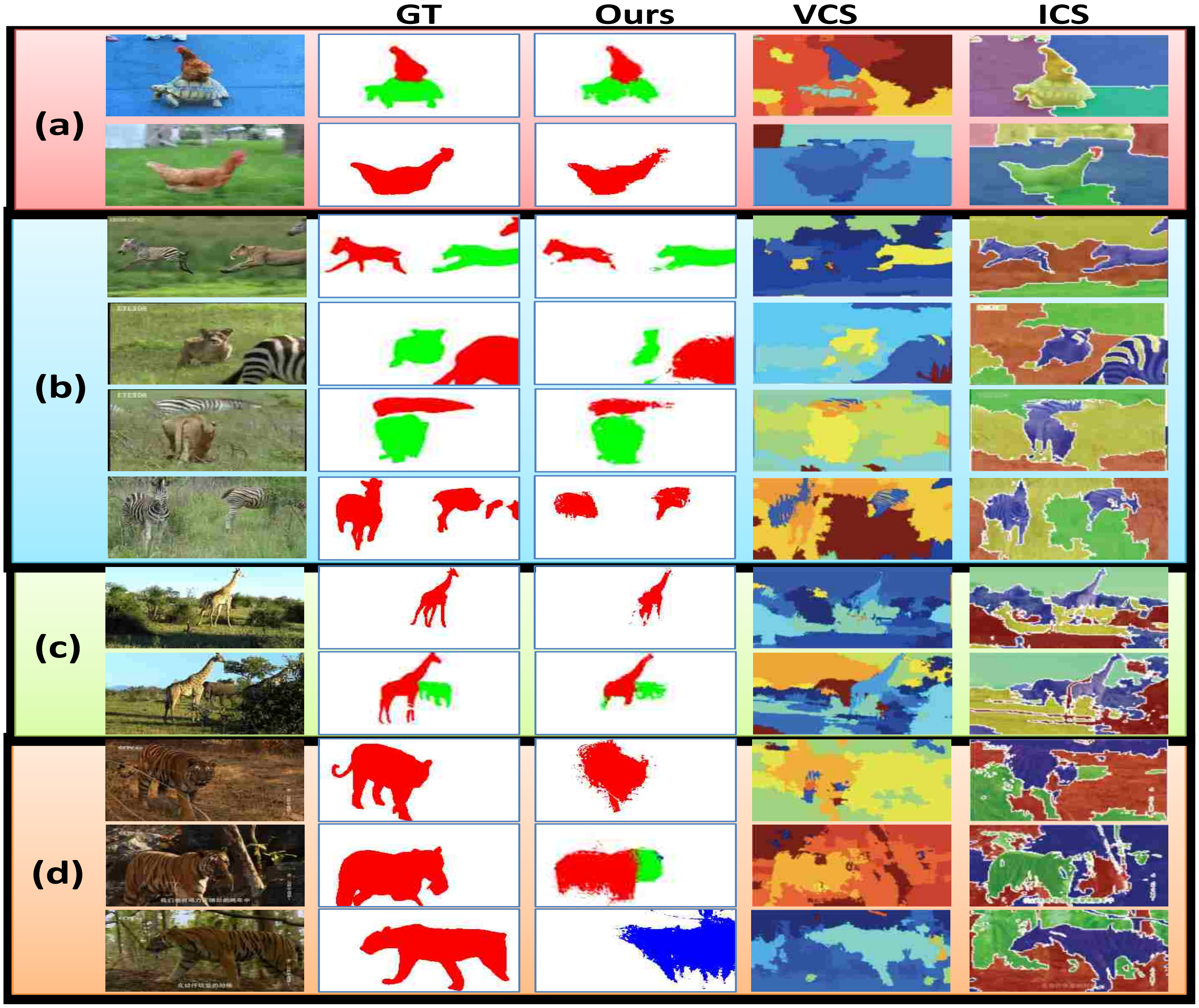


Figure 4.4: Video Co-Segmentation Results on MOViCS Dataset. Each row shows the results of a video in MOViCS dataset. Column 1 is one original frame from the video; column 2 ('GT') is the ground truth for co-segmentation, red regions correspond to the first object in the video set and green regions correspond to the second object in the video set; column 3 ('Ours') is the results of the proposed method, red and green regions correspond to the first and second objects in the video set and blue region corresponds to the third object; column 4 ('VCS') and 5 ('ICS') are the results of video co-segmentation method from [16] and [50] respectively. Row 1 and 2 are for 'chicken&turtle' video set, row 3-6 are for 'lion&zebra' video set, row 7 and 8 are for 'giraffe&elephant' video set and row 9-11 are for 'tigers' video set.

4.5.1.2 Comparisons with State-of-the-art Methods

The proposed method is compared with several state-of-the-art Co-Segmentation methods, see Table 4.1 (the results of VCS [16] and ICS [50] are obtained from [16]). As mentioned in Section 4.4.3, we use a threshold to remove the weak edges, here we show the results by using a single threshold for all video sets (see column 'Ours1'), and also using optimal thresholds for different video sets (in column 'Ours2'). Qualitative results on this dataset are shown in Fig.4.4.

The evaluation shows that the proposed method improves on the state of the art. The average improvement is more than 20%. From Fig.4.4, we can see that ours is the only result that looks visually very similar to the ground truth. Unlike prior methods, our method does not have the propensity for breaking objects into a number of fragments and the method also produces better contours for the objects. The only video in which the object regions are not accurately segmented is the 3rd video in video set 'tigers'.This is due to the large difference in appearance of animals from other two videos and qualitatively our method is still the best for this video set.

Table 4.1: Quantitative comparisons with the state of the art on MOViCS dataset. 'Ours1' shows results of using a single threshold (0.65) for removing the edges, and 'Ours2' shows results of using different thresholds for each video sets (the thresholds are [0.65 0.86 0.45 0.65] for these four video sets respectively.)

Video Set	Ours1	Ours2	VCS [16]	ICS [50]
Chicken&turtle	**0.860**	**0.860**	0.65	0.08
Zebra&lion	**0.588**	**0.636**	0.48	0.23
Giraffe&elephant	**0.528**	**0.639**	0.52	0.07
Tiger	**0.336**	**0.336**	0.30	0.30
Overall	**0.578**	**0.617**	0.49	0.17

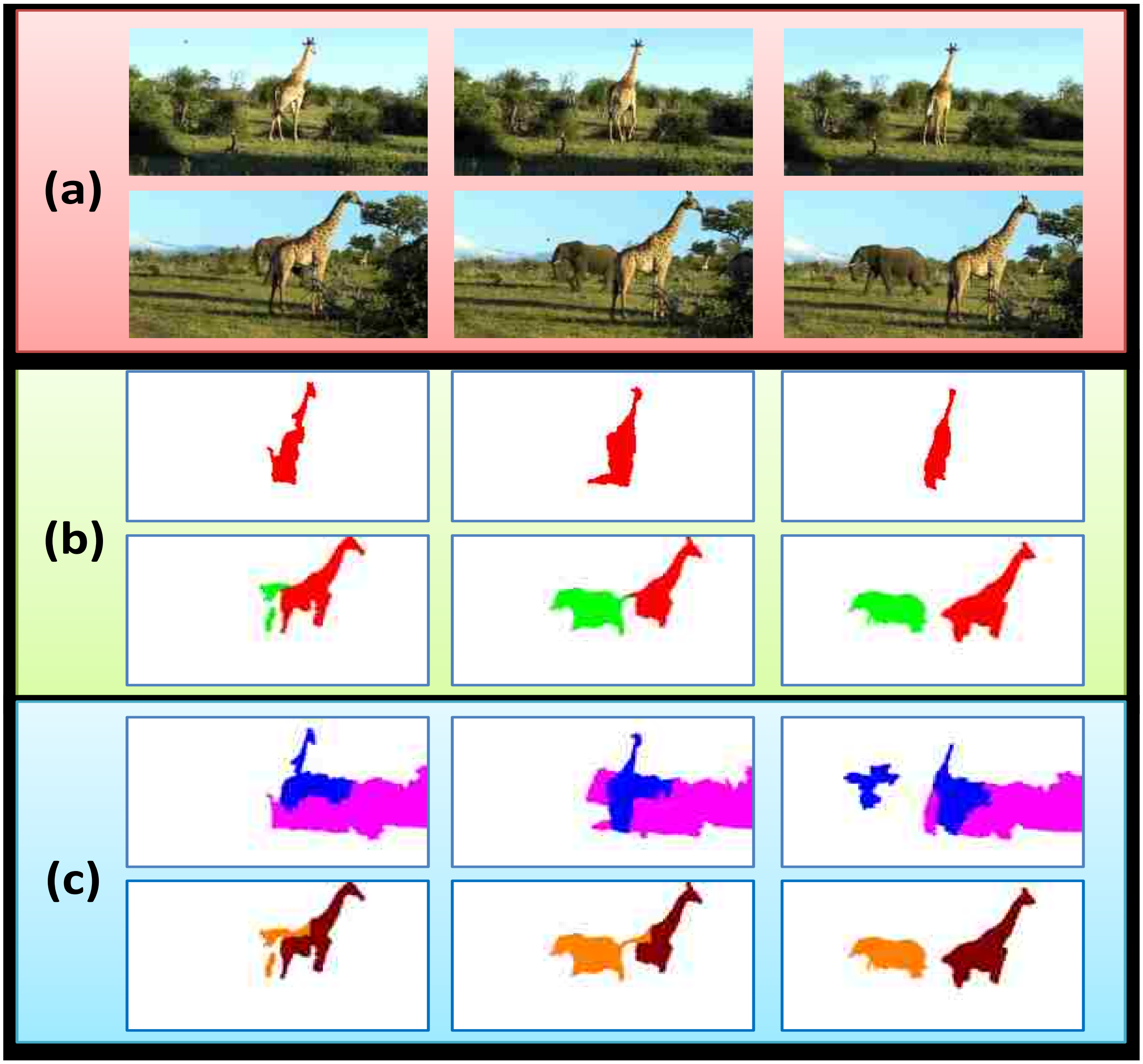


Figure 4.5: Advantages of the Proposed Video Co-Segmentation Method. Row 1 and row 2 show sample frames from two videos respectively. Row 3 and 4 are the video co-segmentation results of the proposed method for these two videos. Red regions correspond to the first object and green regions correspond to the second object. Row 5 and 6 are the segmentation results of applying the method separately to each video. Blue and dark red regions correspond to the first objects, and pink and orange regions correspond to the second objects.

4.5.1.3 Advantages of Video Co-Segmentation Method

Fig.4.5 shows how the Co-Segmentation framework helps the segmentation results for each video. In this example, we have two videos, if the proposed method is simultaneously applied to these two videos, the segmentation results are shown in row 3 and 4; if the proposed method is applied separately for each video, the segmentation results are shown in row 5 and 6. It is quite clear that the Co-Segmentation method not only helps to relate the object labels (red regions for the giraffe in row 3 and 4), but also helps to get more accurate segmentation results (video 2 helps video 1 to get better segments for giraffe in row 3; without video 2, it could only get poor segmentation of giraffe in row 5).

4.5.1.4 Advantages over Single Video Object Segmentation Method

Fig.4.6 shows the comparisons between the proposed method (VOCS) and the state-of-the-art s-ingle video object segmentation (SVOS) method ([134]). Results show that the proposed method could segment objects by using information from other videos (row 2 and 5 in the figure), in contrast, single video object segmentation method mistakenly merges two objects together if they have similar motions (row 3 and 6 in the figure). Also, the proposed method is able to handle occlusions well (row 8 in the figure), while the single video object segmentation method generates wrong labels when there are occlusions of the objects (row 9 in the figure). We compared video object segmentation results quantitatively on MOViCS dataset in Table 4.2. We observe that, if there are two or more objects appearing in the video, or there are occlusions (e.g. ‘elephant_giraffe_all2’) of the objects, or the objects do not appear in all the frames (e.g. ‘lion_zebra_all1’), the proposed method works much better than single video object segmentation method; if there is only one object in the video, the single video object segmentation method sometimes works better (e.g. ‘tiger1_all8’ results).

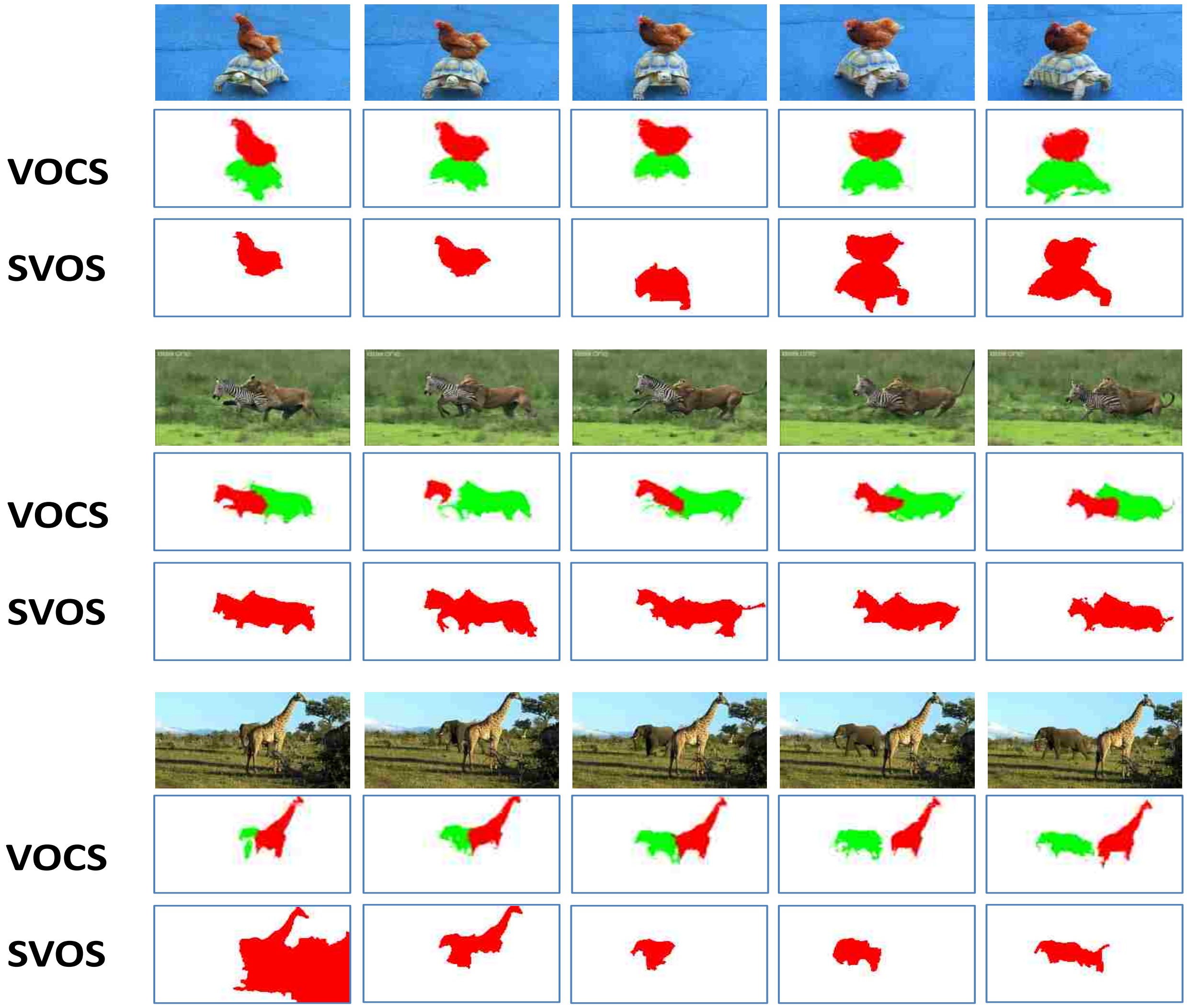


Figure 4.6: Comparison between the proposed method (VOCS) and Single Video Object Segmentation (SVOS) method ([134]). Three groups of results are shown here. In each of them, the first rows (row 1, 4 and 7) show sample frames from the videos; the second rows (row 2, 5 and 8) show results of the proposed method; and the third rows (row 3, 6 and 9) show results of the single video object segmentation method. For the results, the red regions correspond to the first objects and green regions correspond to the second objects. Since the single video object segmentation method only extract primary objects from the videos, only red regions could be shown in the results.

Table 4.2: Quantitative Comparison of Single Video Object Segmentation (SVOS) with Video Object Co-Segmentation (VOCS)

Video Name (Object)	SVOS ([134])	VOCS
ChickenNew (chicken)	0.740	**0.857**
Chicken_on_turtle (chicken)	0.306	**0.823**
Chicken_on_turtle (turtle)	0.563	**0.807**
Elephant_giraffe_all1 (giraffe)	0.570	**0.680**
Elephant_giraffe_all2 (giraffe)	0.122	**0.557**
Elephant_giraffe_all2 (elephant)	0.085	**0.557**
Lion_zebra2 (lion)	0.254	**0.817**
Lion_zebra2 (zebra)	0.510	**0.619**
Lion_zebra_all1 (lion)	0.391	**0.727**
Lion_zebra_all1 (zebra)	**0.529**	0.361
Lion_zebra_all2 (lion)	**0.883**	0.830
Lion_zebra_all2 (zebra)	0.000	**0.547**
Zebra_grass (zebra)	0.403	**0.508**
Tiger1_all8 (tiger)	**0.494**	0.428
Tiger1_all9 (tiger)	**0.841**	0.522
Tiger1_all10 (tiger)	0.384	**0.637**

4.5.2 Safari Dataset

Since video object co-segmentation problem is new and there is only one publicly available dataset with ground truth, we collected another challenging dataset (named 'Safari dataset'[1]) by getting new videos and also reusing some videos from MOViCS dataset. We annotated the key frames. This Safari dataset is challenging, since the Safari contains 5 classes of animals and a total of 9 videos. For each animal class, Safari dataset has a video which only contains this class. Other videos contain two of the animal classes. The goal is to input the 9 videos together and do co-segmentation simultaneously for all of them. We show the ground truth and our co-segmentation results in Fig.4.8 and show quantitative results in Table 4.3.

[1] `http://crcv.ucf.edu/projects/video_object_cosegmentation/`

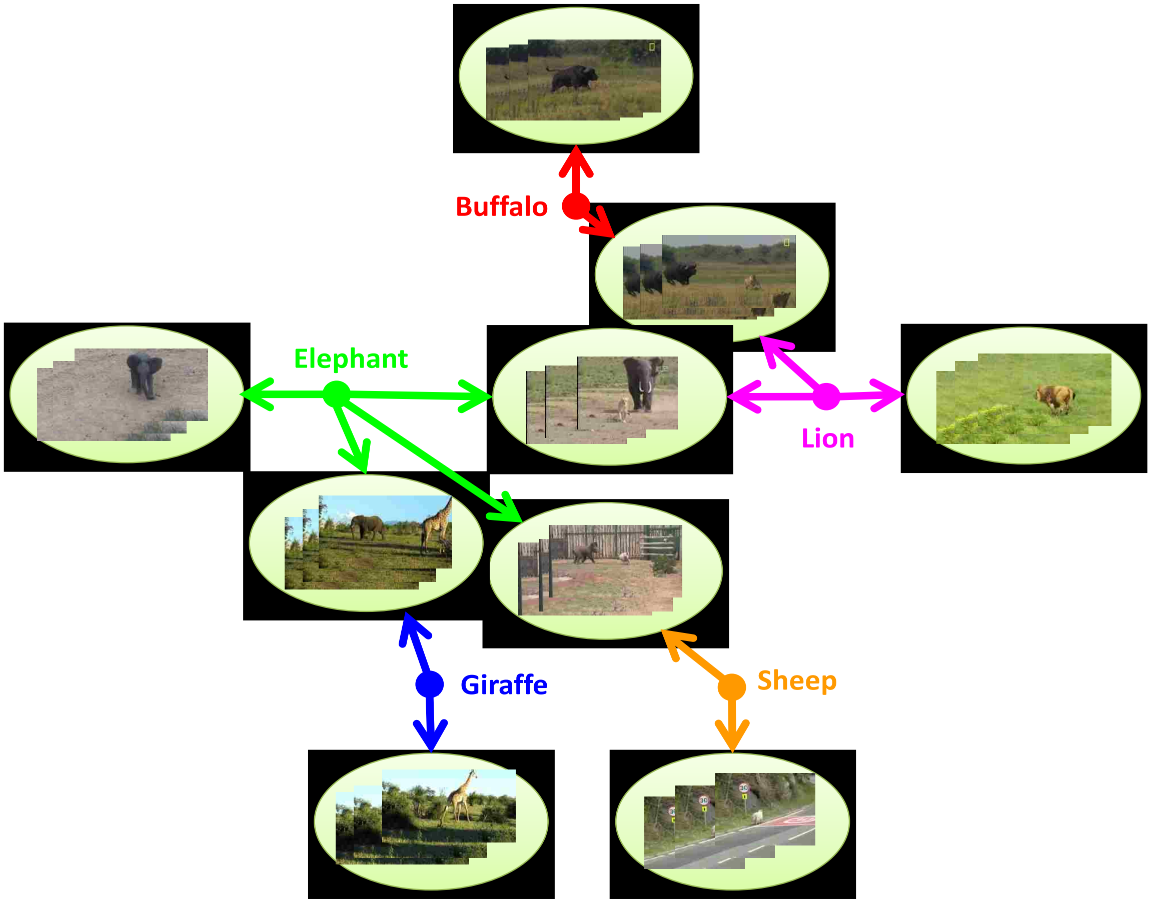


Figure 4.7: The structure of Safari dataset.

Table 4.3: Quantitative results on Safari dataset

Object:	Buffalo	Elephant	Giraffe	Lion	Sheep
Baseline [16]	0.686	0.266	0.024	0.302	0.048
Ours	**0.869**	**0.353**	0.024	**0.317**	**0.363**

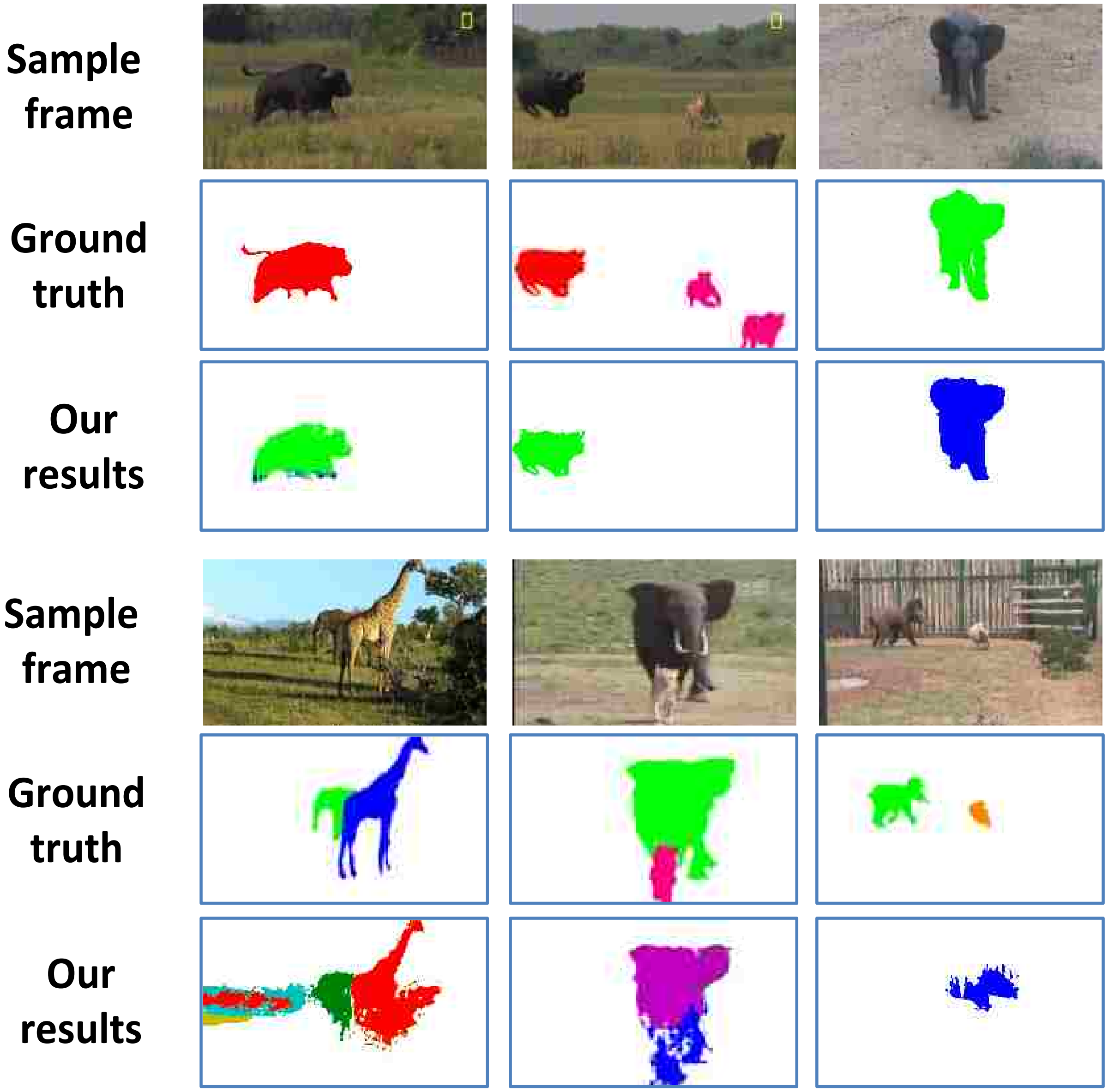


Figure 4.8: The ground truth and our results on Safari dataset. Row 1 shows one frame from each of the video. Row 2 shows the ground truth annotations, in which each object class is shown in different color. And row 3 shows our results, in which each object class is also shown in different color. Please note that, there is no relationship between the colors of row 2 and row 3.

4.6 Summary

We formulated the video object discovery and co-segmentation problem into a Regulated Maximum Weight Clique (RMWC) Problem and solved it using a modified version of Bron-Kerbosch Algorithm. The success of the proposed method relies on i) use of the objectness measure to obtain spatially coherent region proposals, ii) tracking of region proposals, which selects proposals with consistent appearance and smooth motion over time, and iii) using different weighting functions for within video and across video matching for graph construction, which results in improved grouping. Experimental results shows that the method outperforms the state-of-the-art video co-segmentation methods.

We presented solution for how to solve video object segmentation/co-segmentation problems in Chapter 3 and this Chapter. Since human is the most important object class in videos, we explore a different problem: human pose estimation in videos in next Chapter, to investigate if it is easy to extend the spatiotemporal graph formulation to other Computer Vision problems.

CHAPTER 5: SPATIOTEMPORAL GRAPHS FOR HUMAN POSE ESTIMATION IN VIDEOS

5.1 Introduction

We have investigated the problem of video object segmentation and co-segmentation in the previous two Chapters. One of the most important object classes in the videos, both for online videos and surveillance videos, is human. The knowledge of this specific object class (e.g. human poses and skeleton) can be used in solving video object segmentation problem. Therefore, in this Chapter, we explore the applications of spatiotemporal graphs for human pose estimation in videos.

Human pose estimation is crucial for many computer vision applications, including human computer interaction, activity recognition and video surveillance. It is a very challenging problem due to the large appearance variance, non-rigidity of the human body, different viewpoints, cluttered background, self occlusion, etc.

One major issue for human pose estimation in videos is: How to exploit the spatial constraints between the body parts in each frame and temporal consistency through frames to the greatest possible extent, with an efficient exact solution? Since the inference of a tree-based optimization problem has a polynomial time solution [130, 134], the issue becomes (please refer to Fig.5.1): How to formulate the problem in order to model the useful spatial and temporal constraints between body parts in the frames without inducing simple cycles? We propose two key ideas to tackle this issue, which approximate the original fully connected model into a simplified tree-based model. The first idea is **Abstraction**: in contrast to the standard tree representation of body parts, we introduce a new concept, *abstract body parts*, to conceptually combine the symmetric body parts (please refer to Fig.5.2, and details are introduced in Section 5.2). It takes advantage of the symmetric nature of the human body parts without inducing simple cycles into the formulation. The second idea is **Association**, using which we generate optimal tracklets for each abstract body part to ensure the temporal consistency. Since each abstract body part is processed separately, it does not induce any temporal simple cycles into the graph.

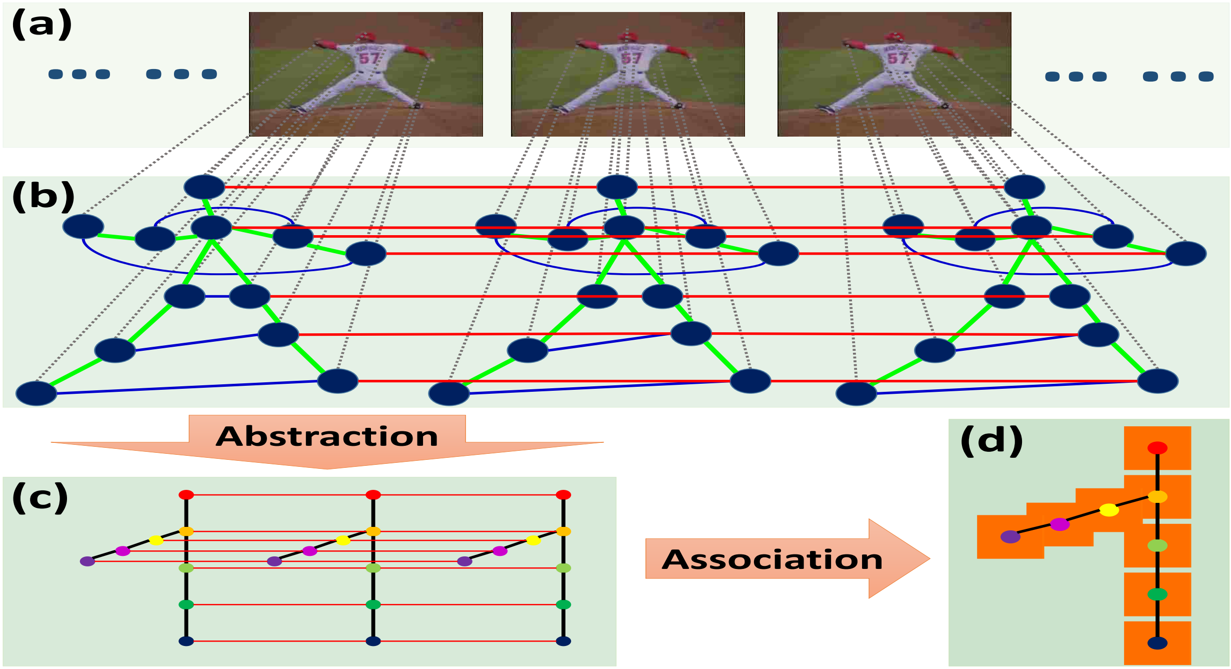


Figure 5.1: An abstract high-level illustration of the proposed method aiming at removing simple cycles from the commonly employed graph optimization framework for video based human pose estimation problem. All of the above graphs are relational graphs for the problems. In (a), a few sample frames of a video are shown. (b) shows the spatiotemporal graph for human pose estimation in videos. Each body part in each frame is represented by a node. Green and blue edges represent relationships between different body parts in the same frame (green ones are commonly used edges in the literature, and blue ones are new edges for symmetric parts we introduced in this dissertation); red edges represent the consistency constraints for the same body part in adjacent frames. Note that this is only an illustration and not all edges are shown. In the 'Abstraction' stage, symmetric parts are combined together, and the simple cycles within each single frame are removed (shown in (c)); and in the 'Association' stage, the simple cycles between adjacent frames are removed (shown in (d)).

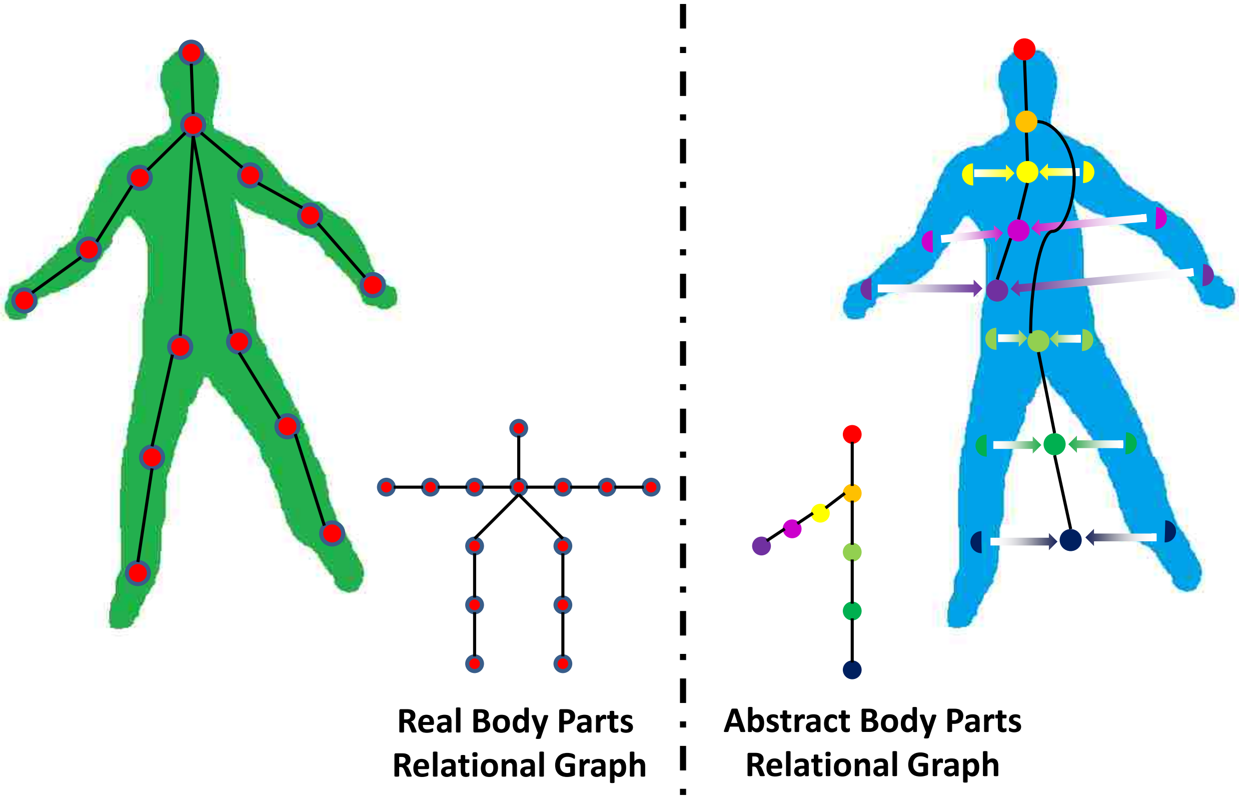


Figure 5.2: Real body parts vs. abstract body parts. The left side shows a commonly used body part definitions in the literature, and we call these body parts (nodes) ‘real body parts’, and the graph ‘real body part relational graph’. The right side shows the proposed new definition of body parts, basically we combine a pair of symmetric body parts to be one body part, we call these body parts (nodes) ‘abstract body parts’, since these parts are some abstract concepts of parts, not real body parts, and the graph as ‘abstract body part graph’.

The proposed method is different from the state-of-the-art methods [85, 108, 15, 95] in the following ways: [85] exploits the symmetric nature of body parts, however, the problem is formulated as a multi-target tracking problem with mutual exclusions, which is NP-complete and only approximate solutions can be obtained by relaxation; the method in [108] is designed to remove the temporal simple cycles from the graph shown in Fig.5.1(b) to track upper body parts, however, the employed junction tree algorithm will have much higher computational complexity if applied to full-body pose estimation, since there are many more simple cycles induced by symmetric body parts; compared to [15], the proposed method has no temporal simple cycles; and in contrast to [95], the proposed method can model symmetric body part structure more accurately rather than settling for the sub-models. Therefore, the proposed method ensures both spatial and temporal constraints, without inducing any simple cycles into the formulation, and an exact solution can be efficiently found by dynamic programming.

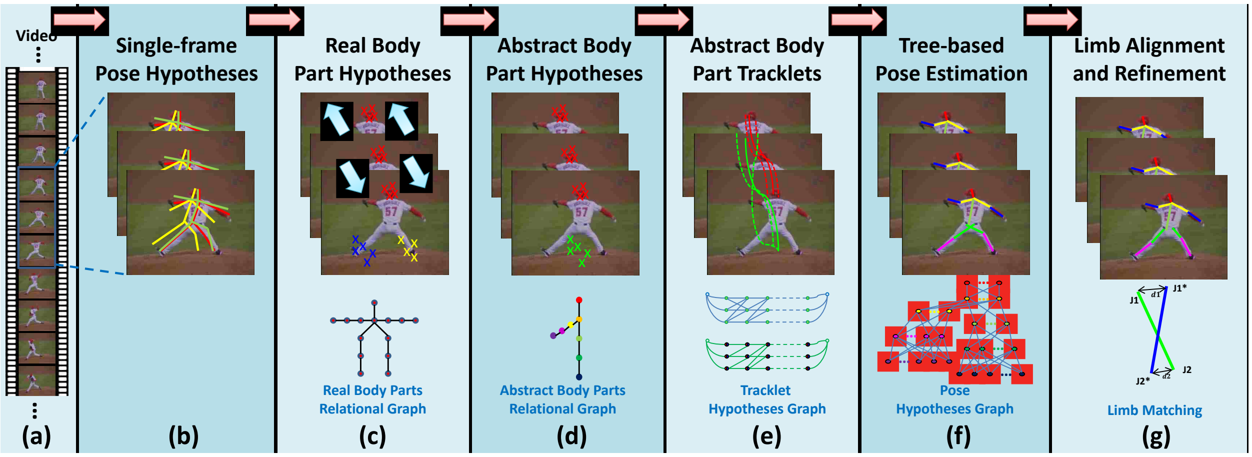

Video
Single-frame Pose Hypotheses
Real Body Part Hypotheses
Abstract Body Part Hypotheses
Abstract Body Part Tracklets
Tree-based Pose Estimation
Limb Alignment and Refinement
Real Body Parts Relational Graph
Abstract Body Parts Relational Graph
Tracklet Hypotheses Graph
Pose Hypotheses Graph
J1
d1
J1*
J2*
d2
J2
Limb Matching
(a)
(b)
(c)
(d)
(e)
(f)
(g)

Figure 5.3: An outline of the proposed method. (a) shows the original video frames; in (b), pose hypotheses in each frame are generated by N-Best method [78] or DCNNGM [12]; in (c), by using the results from (b), real body part hypotheses are generated for each body part in each frame and propagated to the adjacent frames; in (d), real body parts are combined into abstract body parts and the hypotheses are also combined accordingly in order to remove the intra-frame simple cycles (i.e. the simple cycles with blue and green edges in Fig.5.1(b)); in (e), tracklets are generated for abstract body parts (including single body parts and coupled body parts) using the abstract body part hypotheses generated in (d); in (f), the pose hypotheses graph is built, each node is a tracklet corresponding to the abstract body part, and the best pose estimation is obtained by selecting the best hypotheses for the parts from the graph; and in (g), the correct limbs are inferred by limb alignment and refinement schemes.

The organization of the rest of the Chapter is as follows. We formulate the video based human pose estimation problem into a *unified* tree-based optimization framework, which can be solved efficiently by dynamic programming. Please refer to Fig.5.3 for the major steps. The main steps of the method (as show in Fig.5.3 (b - g)) are: **1)** for each frame, generate many pose hypotheses by the N-Best method [78] or DCNNGM [12]; **2)** based on step 1, generate hypotheses for each real body part and prorogate them to adjacent frames (We use the term *real body parts* to represent body parts which are commonly used in the literature, and *abstract body parts* as a new concept which will be introduced later to facilitate the formulation of the proposed method); **3)** combine the symmetric real body part hypotheses and obtain the abstract body part hypotheses; **4)** build the tracklet hypotheses graph for each abstract body part and select the top trackles for each; **5)** build the pose hypotheses graph for the abstract body part tracklets and select the best pose configuration. **6)** use limb alignment and refinement schemes to infer the correct limb configurations.

In next a few Sections, we discuss the new concept: 'abstract body parts' in comparison to 'real body parts' in Section 5.2 and show how to generate body part hypotheses in each frame in Section 5.3; we introduce tracklets generation in Section 5.4 and 5.5; we show how to extract the optimal poses in Section 5.6, and we introduce the limb alignment and refinement schemes in Section 5.7. We present experimental results in Section 5.8.

5.2 Real Body Parts vs. Abstract Body Parts

We use the term *real body parts* to represent body parts which are commonly used in the literature. And we use **abstract body parts**, which is a new concept introduced here to facilitate the formulation of the proposed method (as shown in Fig.5.2). In contrast to the real body part definitions, there are two types of the abstract body parts in this dissertation: **single part** and **coupled part**. **Single parts** include *HeadTop* and *HeadBottom*. **Coupled parts** include *Shoulder*, *Elbow*, *Hand*, *Hip*, *Knee* and *Ankle*. Note that, for coupled parts, we use one part to represent two symmetric real body parts, for instance *Ankle* is employed to represent the abstract part, which is actually the combination of the *left* and *right* ankles. The design of abstract body parts is to remove simple cycles for the body part relational graph, but keep the ability to model the symmetric body parts. For example, in Fig.5.1(b), in each frame, the green and blue edges are designed to model the body part relationships, but there are many simple cycles in a single frame. After introducing the abstract body parts in Fig.5.1(c), the symmetric parts are combined, thus there is no simple cycle anymore in each frame. There are still simple cycles between frames, which will be handled by the abstract body part tracklets in Section 5.4 and 5.5.

5.3 Body Part Hypotheses in a Single Frame

There are several ways to generate body part hypotheses in each video frame. In [136], N-Best method [78] is used to generate the hypotheses. In the proposed method, although N-Best method [78] is still an essential step, we also explore generation of body part hypotheses by deep learning method (to be specific, we are using deep convolutional neural networks with graphical model (D-CNNGM) [12]), in our experiments. For the sake of completeness, we briefly explain hypotheses generation using N-Best method [78] and DCNNGM method [12] in this Section.

5.3.1 N-Best Hypotheses

N-Best human pose estimation approach [78] combined appearance and structural constraints to generate basic pose estimation. And it can generate a group of pose hypotheses for a single image, which can be used to infer better pose configurations in videos. N-Best method is applied to each video frame to generate N best full body pose hypotheses. N is usually a large number (normally $N > 300$). And for each real body part, the body part hypotheses are body part locations extracted from the N-best poses. The body part hypotheses are sampled by an iterative non-maximum suppression (NMS) scheme based on the detection score map. Detection score is a combination of max-marginal [78] and foreground score,

$$\Phi_s(p) = \alpha\Phi_M(p) + (1-\alpha)\Phi_F(p), \tag{5.1}$$

where Φ_s is the detection score, Φ_M is the max-marginal derived from [78], Φ_F is the foreground score obtained by the background subtraction [20], and p is the location of the body part.

5.3.2 DCNNGM Hypotheses

Compared to N-Best method [78], deep learning methods for human pose estimation [111, 75, 12, 110] have shown better performance in single image pose estimation, and we want to take advantage of this. In this Chapter, we employ the deep convolutional neural networks with graphical model approach (DCNNGM) [12] to generate pose hypotheses. We chose this approach because its code is available and it is one of the state-of-the-art deep learning methods for pose estimation, however, we believe other state-of-the-art methods [111, 75, 110] can also get comparable, or even better results. There are two problems to solve before we can employ DCNNGM in our framework: **1)** Although the method can process a small image in a reasonable amount of time, it is still prohibitive to handle videos which have many frames and the frame sizes are relatively large (e.g. 1280×720). **2)** The method does not output multiple hypotheses for each frame.

We solve the first problem by a pre-processing step: N-Best method [78] is employed to estimate the poses in each video frame, and the top estimations are used to infer the rough bounding boxes of the people in each frame. The bounding boxes are cropped and input to DCNNGM to estimate the poses and results are mapped back to the original video frames. N-Best method [78] is an essential step here, but not the only option, and any other pose estimation method (e.g. [130]) or human detection method (e.g. [24]) may be employed here with proper adjustments.

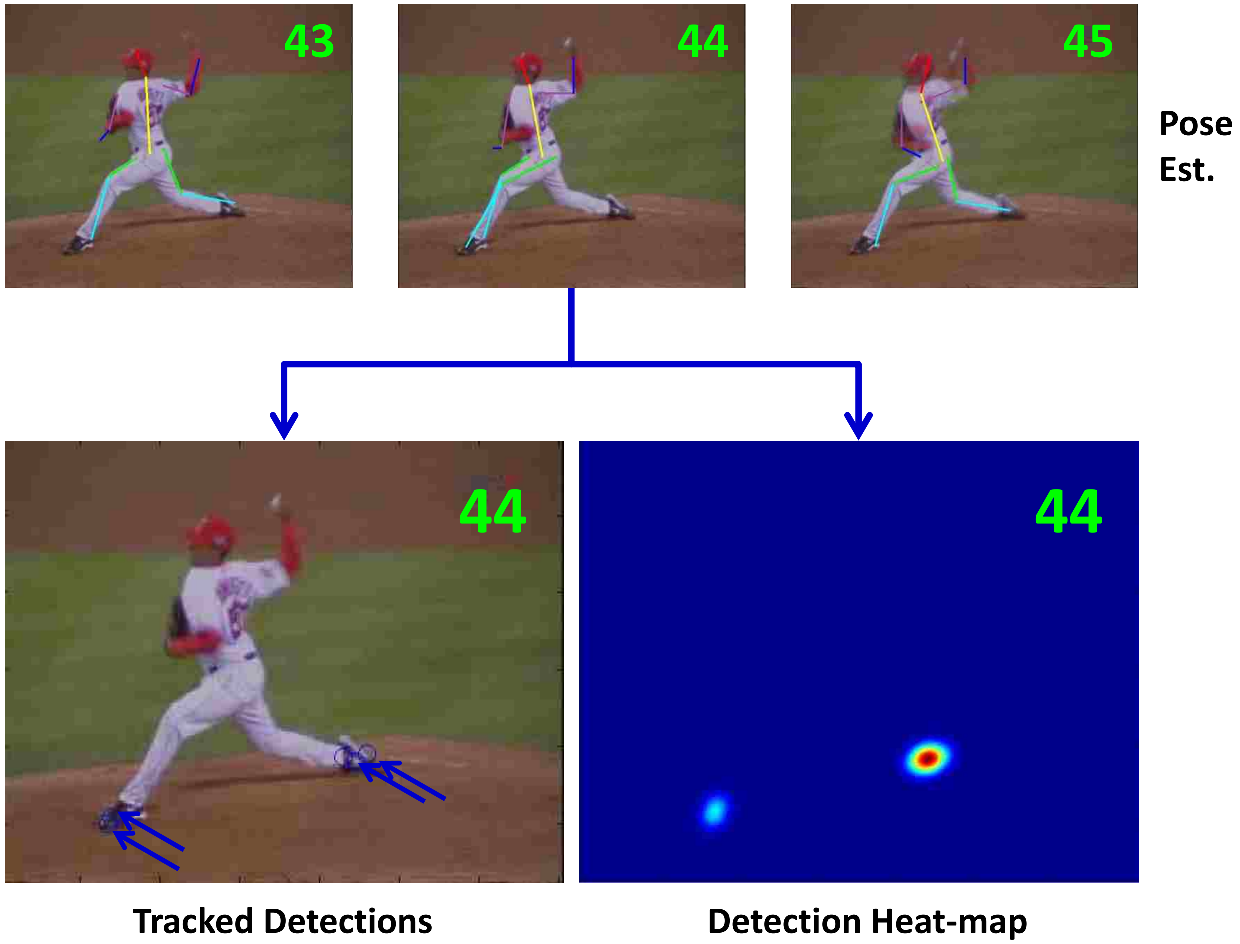


Figure 5.4: An Illustration for DCNNGM Pose Hypotheses Generation. The top row shows the pose estimation results using DCNNGM [12], and the bottom row shows the tracked “right ankle” for the neighboring frames and the combined detection heat-map. In this example, in frame 44, the right leg was not correctly localized; however, in some of its neighboring frames, the right leg was localized correctly. Therefore, by tracking using neighboring frames, we can generate good detection hypotheses for ’right ankle’ and thus a better detection map.

We solve the second problem by tracking the pose estimation from neighboring frames. This process not only generates more pose hypotheses in each video frame, but also makes the estimation more robust and consistent (please refer to Fig. 5.4 for an illustration). In every frame, each body part is tracked backward and forward to neighboring frames by a simple NCC (normalized cross correlation) tracker. NCC tracker is employed here due to its simplicity and efficiency, and since the parts are only tracked for a few frames. We decided not to employ heavy-weight trackers not to overkill the problem, although it is possible that the results may be improved slightly by employing some fancier tracker (e.g. [51, 120]).

In each frame and for every body part, by tracking poses from neighboring frames, the outcome will be several candidate locations (one from the detection in the current frame, and others are from the tracking results from other frames). A 2D Gaussian is applied to each location and a detection heat-map is obtained (please refer to Fig.5.4). By sampling many locations from the detection heat-map, several body part hypotheses are generated. The values of the detection heat-map are used as $\Phi_M(p)$ in Eqn.5.1.

5.3.3 Abstract Body Part Hypotheses

After generating the body part hypotheses, either using the method in Section 5.3.1 or Section 5.3.2, the abstract body part hypotheses are obtained: the abstract body part hypotheses for a single part are the same as its corresponding real body part hypotheses. And the abstract body part hypotheses for a coupled part are the permutation of its corresponding left and right body part hypotheses.

5.4 Single Part Tracklets

Based on the abstract body part hypotheses generated in Section 5.3, we want to obtain several best single part and coupled part tracklets through the video frames. The problem is how to select one hypothesis from each frame ensuring that they have high detection scores and are consistent throughout the frames. The relational graph for this problem is shown in Fig.5.5(a), and the hypothesis graphs for single parts and coupled parts are shown in Fig.5.5 (b) and (c) respectively.

With the single part hypotheses, a single part tracklet hypothesis graph is built (Fig.5.5(b)) for each single part (*headTop* and *headBottom*). Each node represents a single part hypothesis and the detection score $\Phi_s(p)$ in Eqn.5.1 is used to assign the node an unary weight. Edges are added between every pair of nodes from the adjacent frames. Binary weights are assigned to the edges which represent similarities between hypotheses in adjacent frames. The binary weight is defined as a combination of optical flow predicted spatial distance and Chi-square distance of HOG features as follows:

$$\Psi_s(p^f, p^{f+1}) = \exp(-\frac{\chi^2(\Upsilon(p^f), \Upsilon(p^{f+1}) \cdot \|\hat{p}^f - p^{f+1}\|_2^2)}{\sigma^2}), \tag{5.2}$$

where p^f and p^{f+1} are two arbitrary hypotheses from frames f and $f+1$, $\Upsilon(p)$ is the HOG feature vector centered at location p, $\hat{p}^f$ is the optical flow predicted location for p^f in frame $f+1$, and σ is a parameter. The goal is to select one node from each frame to maximize the overall combined unary and binary weights. Given an arbitrary selection of nodes from the graph $s_s = \{s_s^i|_{i=1}^F\}$ (F is the number of frames), the objective function is given by

$$\mathcal{M}_s(s_s) = \sum_{i=1}^{F} \Phi_s(s_s^i) + \lambda_s \cdot \sum_{i=1}^{F-1} \Psi_s(s_s^i, s_s^{i+1}), \tag{5.3}$$

where λ_s is the parameter for adjusting the binary and unary weights, and $s_s^* = \arg\max_{s_s}(\mathcal{M}(s_s))$ gives the optimal solution. It is clear that the relational graph of this problem is a degenerated tree (i.e. single branch tree, please see Fig.5.5(a)), and the problem can be solved using dynamic programming efficiently. After the optimal solution is obtained, the selected nodes are removed from the graph and the next optimal solution is obtained. This process is iterated for multiple times in order to get several tracklets from the graph.

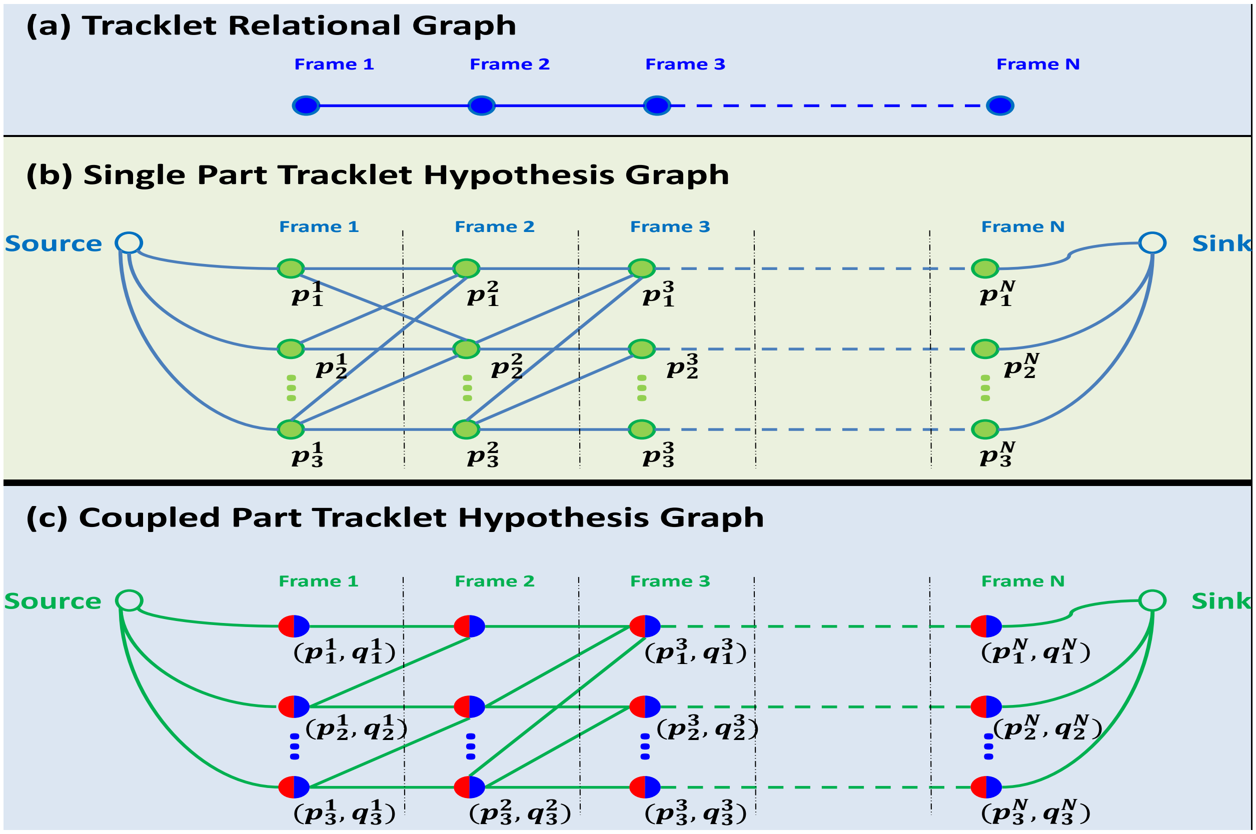


Figure 5.5: **Spatiotemporal Graph for Tracklet Generation**. (a) Shows the relational graph for the abstract body part tracklet generation. (b) Shows the tracklet hypothesis graph for single body parts. Each node represents one hypothesis location of the body part in a specific frame, and edges show the similarity between the connected body part hypotheses in adjacent frames. (c) Shows the tracklet hypothesis graph for coupled parts. Each node represents a coupled body part hypothesis, which is the combination of the corresponding symmetric body parts (that is why each node is colored into two halves). The edges represent the similarities between connected coupled body parts in adjacent frames. Note that, (b) and (c) are only illustrations, and for simplicity, not all edges are shown.

5.5 Coupled Part Tracklets

The relational graph for the coupled part tracklets generation is the same as for the single part; however, the nodes and edges are defined differently. In this case, each hypothesis node is composed of the locations of a pair of symmetric parts (e.g. left and right ankles). Fig.5.5(c) shows an illustration of the graph. Such design aims to model the symmetric relationship between coupled parts, including mutual location exclusions and appearance similarity in order to reduce double counting. As discovered in previous research [85], double counting is a key issue which severely hinders the pose estimation. Theoretically, tree based model [130] lacks the ability to model spatial relationship of the coupled parts (e.g. left and right ankles). Furthermore, as discussed in Section 5.1, attempting to model such spatial relationship would inevitably induce simple cycles in the graph which would severely increase the computational complexity. By introducing the coupled parts, this dilemma could be effectively solved. In the coupled part tracklet hypothesis graph, each node $r = (p, q)$ represents a composition of a pair of symmetric parts p and q. Unary weights are assigned to the nodes, which represent the detection confidence and the compatibility between the two symmetric parts, is defined as:

$$\Phi_c(r) = \frac{(\Phi_s(r.p) + \Phi_s(r.q)) \cdot (\Lambda(r.p)^T \cdot \Lambda(r.q)))}{1 + e^{-|r.p - r.q|/\theta}}, \tag{5.4}$$

where Φ_s is from Eqn.5.1, $r.p$ and $r.q$ respectively represent the left and right components of the coupled part r, $\Lambda(p)$ is the normalized color histogram of a local patch around p, the denominator is the inverse of a sigmoid function which penalizes the overlap of symmetric parts, and θ is the parameter that controls the penalty. The binary weights of the edges are computed as

$$\Psi_c(r^f, r^{f+1}) = \Psi_s(r.p^f, r.p^{f+1}) + \Psi_s(r.q^f, r.q^{f+1}), \tag{5.5}$$

where Ψ_s is from Eqn. 5.2.

Here, the goal is to select one node (which is a composition of a pair of symmetric parts) from each frame to maximize the overall combined unary and binary weights. Given an arbitrary selection of nodes from the graph $s_c = \{s_c^i|_{i=1}^F\}$ (where F is the number of frames), the objective function is

$$\mathcal{M}_c(s_c) = \sum_{i=1}^{F} \Phi_c(s_c^i) + \lambda_c \cdot \sum_{i=1}^{F-1} \Psi_c(s_c^i, s_c^{i+1}), \tag{5.6}$$

in which λ_c is the parameter to adjust the binary and unary weights, and $s_c^* = \arg\max_{s_c}(\mathcal{M}(s_c))$ gives the optimal solution. The problem can also be solved by dynamic programming efficiently, and iterated for multiple times to get several tracklets.

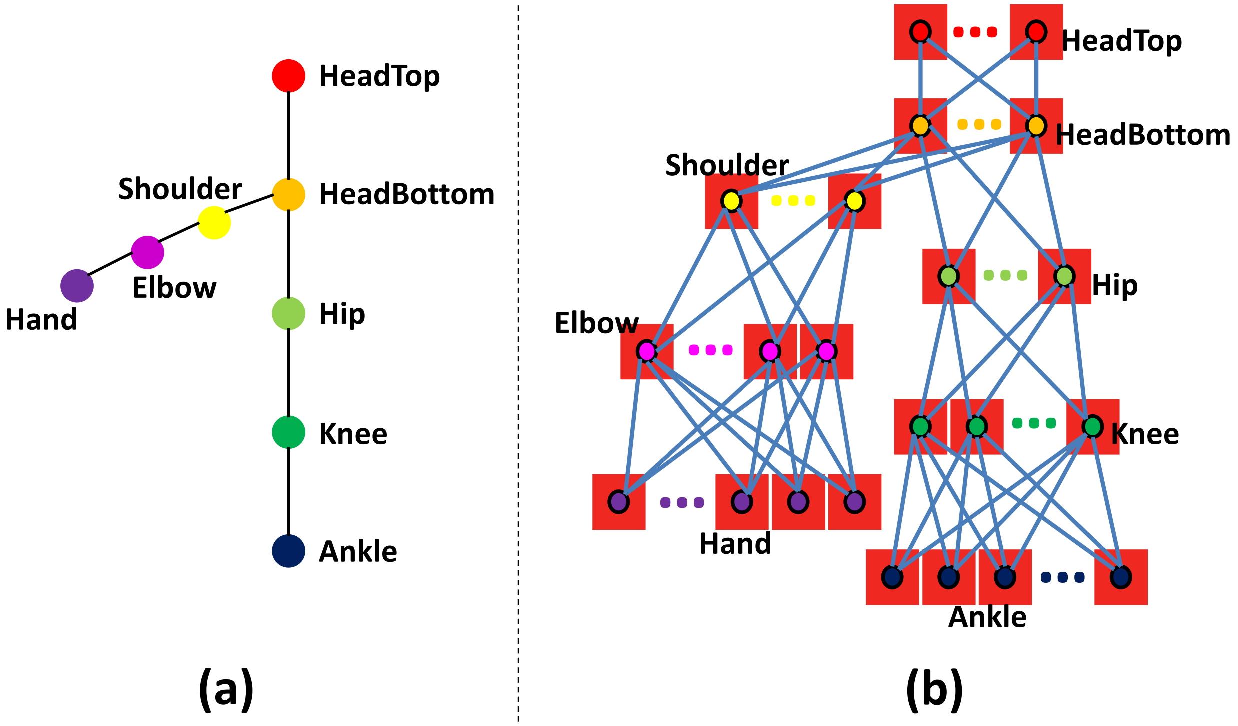


Figure 5.6: **Spatiotemporal Graph for Pose Estimation in Videos**. (a) is the pose relational graph. Each node represents one abstract body part and edges represent the relationship between the connected body parts. (b) is the pose hypothesis graph. Each node is a tracklet for the part, and edges represent the spatial compatibility of connected nodes.

5.6 Optimal Pose Estimation using Part Tracklets

Since the best tracklets for each abstract body parts are obtained by the methods introduced in Section 5.4 and 5.5, the next step is to select the best ones which are compatible. The relational graph, $G_T = (V_T, E_T)$, for this final tracklet based optimal pose estimation is shown in Fig.5.6(a). Each node represents an abstract body part, and the edges model the spatial relationships between them.

Following the definitions of the abstract body parts and using the part tracklets generated for these abstract body parts, a pose hypothesis graph can be built to get the optimal pose (as shown in Fig.5.6(b)). In this graph, each node represents an abstract body part tracklet and edges represent the spatial constraints. For each hypothesis tracklet node, s, depending on if it corresponds to a single part or a coupled part, $E_s(s)$ from Eqn.5.3, or $E_c(s)$ from Eqn.5.6 is used as its unary weight $\Phi_T(s)$. Let $\Psi_d(p_i, q_j) = \omega_{i,j} \cdot \psi(p_i - q_j)$ be the relative location score in [130] ($\omega_{i,j}$ and ψ are defined the same as in [130]), the binary weight between a pair of adjacent single part tracklet nodes $s_s = \{s_s^i|_{i=1}^F\}$ and $t_s = \{t_s^i|_{i=1}^F\}$ is

$$\Psi_T(s_s, t_s) = \sum_{i=1}^{F} \Psi_d(s_s^i, t_s^i), \tag{5.7}$$

the binary weight between a single part tracklet node $s_s = \{s_s^i|_{i=1}^F\}$ and an adjacent coupled part tracklet node $t_c = \{t_c^i|_{i=1}^F\}$ is

$$\Psi_T(s_s, t_c) = \sum_{i=1}^{F} (\Psi_d(s_s^i, t_c^i.p) + \Psi_d(s_s^i, t_c^i.q)), \tag{5.8}$$

and the binary weight between a pair of adjacent coupled tracklet part nodes $s_c = \{s_c^i|_{i=1}^F\}$ and

$t_c = \{t_c^i|_{i=1}^F\}$ is

$$\Psi_T(s_c, t_c) = \sum_{i=1}^{F} (\Psi_d(s_c^i.p, t_c^i.p) + \Psi_d(s_c^i.q, t_c^i.q)). \tag{5.9}$$

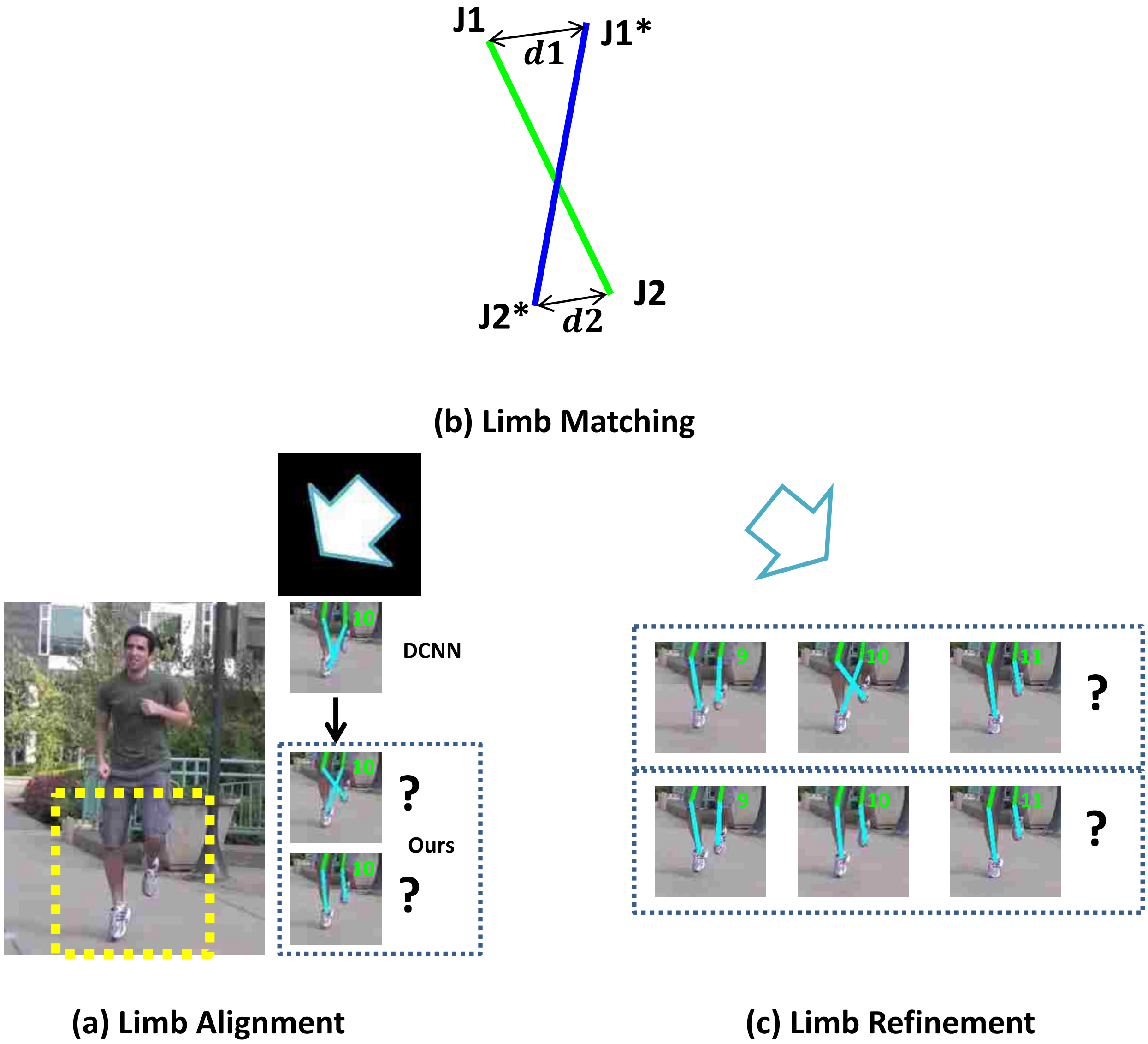
The problem is to select only one tracklet for each abstract body part in order to maximize the combined unary (detection score) and binary (compatible score) weights. Given an arbitrary tree selected from the hypothesis graph $s_T = \{s_T^i|_{i=1}^{|V_T|}\}$, the objective function is

$$\mathcal{M}_T(s_T) = \sum_{v_T^i \in V_T} \Phi_T(s_T^i) + \lambda_T \cdot \sum_{(v_T^i, v_T^j) \in E_T} \Psi_T(s_T^i, s_T^j), \tag{5.10}$$

where λ_T is a parameter for adjusting the binary and unary weights, and the optimal solution $s_T^* = \arg\max_{s_T}(\mathcal{M}(s_T))$ can also be obtained by dynamic programming efficiently. The body part locations in each frame are extracted from this final optimal solution.

5.7 Limb Alignment and Refinement

Most of the human pose estimation methods (e.g. [130, 78, 12]) do not infer the anatomical left/right body parts (some exceptions: [42, 110]), which means that they only infer whether the parts are "visually" left or right (we call this "visual left/right" body parts). Of course, this relies on the training data annotations and theoretically there is nothing wrong with it; however, sometimes it is problematic. For instance, in Fig.5.8(C5), the anatomical left and right knees appear "visually" right and left respectively in the frame, but the anatomical left and right ankles appear "visually" left and right respectively in the frame. In this case, failing to infer which knees correspond to which ankles will result in incorrect poses (e.g. Fig.5.8(A3),(B3) and (C3)). The proposed framework, so far (until Section 5.6), does not have an effective mechanism to handle this issue, therefore we deal with it in this section.



(b) Limb Matching

(a) Limb Alignment

(c) Limb Refinement

Figure 5.7: An Illustration for Limb Alignment and Refinement. (b) is an illustration for matching a single limb with its reference. $J1, J2$ are the two joints corresponding to the limb detection, and $J1^*, J2^*$ are the two joints for the reference limb. (a) shows the limb alignment process: by matching with the limb part configurations from DCNNGM results, the limb part configurations (lower legs) can be inferred correctly. (c) shows the limb refinement process: by matching with the limb configurations from neighboring frames, the wrong limb configuration (please see frame 10, lower legs) can be corrected.

Here, we define **limb inference problem** as the problem to determine the left-right relationship between adjacent coupled part pairs (e.g. to decide which knee matches with which ankle). Here "limb" has a specific definition, which refers to "upper arms", "lower legs", and so on, not for "full arms" or "full legs". So far, we do not have an effective scheme for limb inference, thus although each body part's localization has improved, there are still some mismatches (please see Fig.5.8). This is because during the previous steps, we have not exploited limb inference results from image based pose estimation. We would like to utilize this information from DCNNGM [12] results. Although the body part localization results from DCNNGM are not as good as obtained by the proposed method, the limb inferences are much better and in most of the cases DCNNGM method can infer them correctly.

5.7.1 Limb Alignment

Based on the assumption that the limb inferences of the DCNNGM are correct in most of the video frames, we align our results with them. As illustrated in Fig.5.7 (b), a limb mismatching function is defined based on the joint locations of the reference limb and estimated limb. The mismatch score is defined as:

$$\mathcal{M}_s(L, \mathcal{L}) = (\|L(1) - \mathcal{L}(1)\|_2^2 + \|L(2) - \mathcal{L}(2)\|_2^2), \tag{5.11}$$

where L and $\mathcal{L}$ denote the detected limb and reference limb respectively (in this situation, $\mathcal{L}$ is the results from DCNNGM [12]). $L(1), L(2), \mathcal{L}(1), \mathcal{L}(2)$ are the two joint locations of the limbs.

For a pair of limbs (i.e. the left and right limbs), the mismatching score is defined as

$$\mathcal{M}_p(L^l, L^r | \mathcal{L}^l, \mathcal{L}^r) \quad = \quad \mathcal{C}(L^l, L^r | \mathcal{L}^l, \mathcal{L}^r) \quad \cdot \quad (\mathcal{M}_s(L^l, \mathcal{L}^l) \quad + \quad \mathcal{M}_s(L^r, \mathcal{L}^r)), \tag{5.12}$$

where $\mathcal{C}(L^l, L^r|\mathcal{L}^l, \mathcal{L}^r)$ is a limb intersection penalty function which is defined as

$$\mathcal{C}(L^l, L^r|\mathcal{L}^l, \mathcal{L}^r) = \begin{cases} c & L^l, L^r \text{ are compatible with } \mathcal{L}^l, \mathcal{L}^r \\ 1-c & \text{otherwise} \end{cases} \tag{5.13}$$

where "compatible" means both L^l, L^r and $\mathcal{L}^l, \mathcal{L}^r$ intersect each other, or both of them do not intersect each other.

Using the mismatching function for each pair of limbs in each frame, the configuration which has lower mismatching score is selected to be the correct limb inference. For instance in Fig.5.7(a), compared to the DCNNGM reference, the bottom configuration of the lower legs has lower mismatching score, thus it is considered to be the correct limb inference for lower legs. It can be seen from this example that the proposed method gives better joint localization results (i.e. for feet), however potentially wrong limb inferences. Using DCNNGM limb inference results, the aligned results are now better in both the joint localization and limb inferences.

5.7.2 Limb Refinement

The limb alignment can correct most of the mismatched limbs; however, sometimes DCNNGM also makes wrong limb inference, therefore only relying on the limb alignment does not solve the limb inference problem completely. Consequently, the limb inferences in neighboring frames is utilized to refine the results. Similar to Eqn.5.12, a mismatching score is introduced for two neighboring frames (frame f and $f+1$):

$$\begin{aligned} &\mathcal{M}_n(L^l_f, L^l_{f+1}, L^r_f, L^r_{f+1}|\mathcal{L}^l_f, \mathcal{L}^l_{f+1}, \mathcal{L}^r_f, \mathcal{L}^r_{f+1}) \\ &\quad = \mathcal{C}(L^l_f, L^r_f|\mathcal{L}^l_f, \mathcal{L}^r_f) \cdot \mathcal{C}(L^l_{f+1}, L^r_{f+1}|\mathcal{L}^l_{f+1}, \mathcal{L}^r_{f+1}) \cdot (\mathcal{M}_s(L^l_f, L^l_{f+1}) + \mathcal{M}_s(L^r_f, L^r_{f+1})), \end{aligned} \tag{5.14}$$

where $\mathcal{M}_s$ and $\mathcal{C}$ are defined similarly as in Eqn.5.11 and Eqn.5.13. The best configurations of the limbs through all the video frames are obtained simultaneously by minimizing the sum of mismatching scores of all consecutive frames in the video using simple dynamic programming.

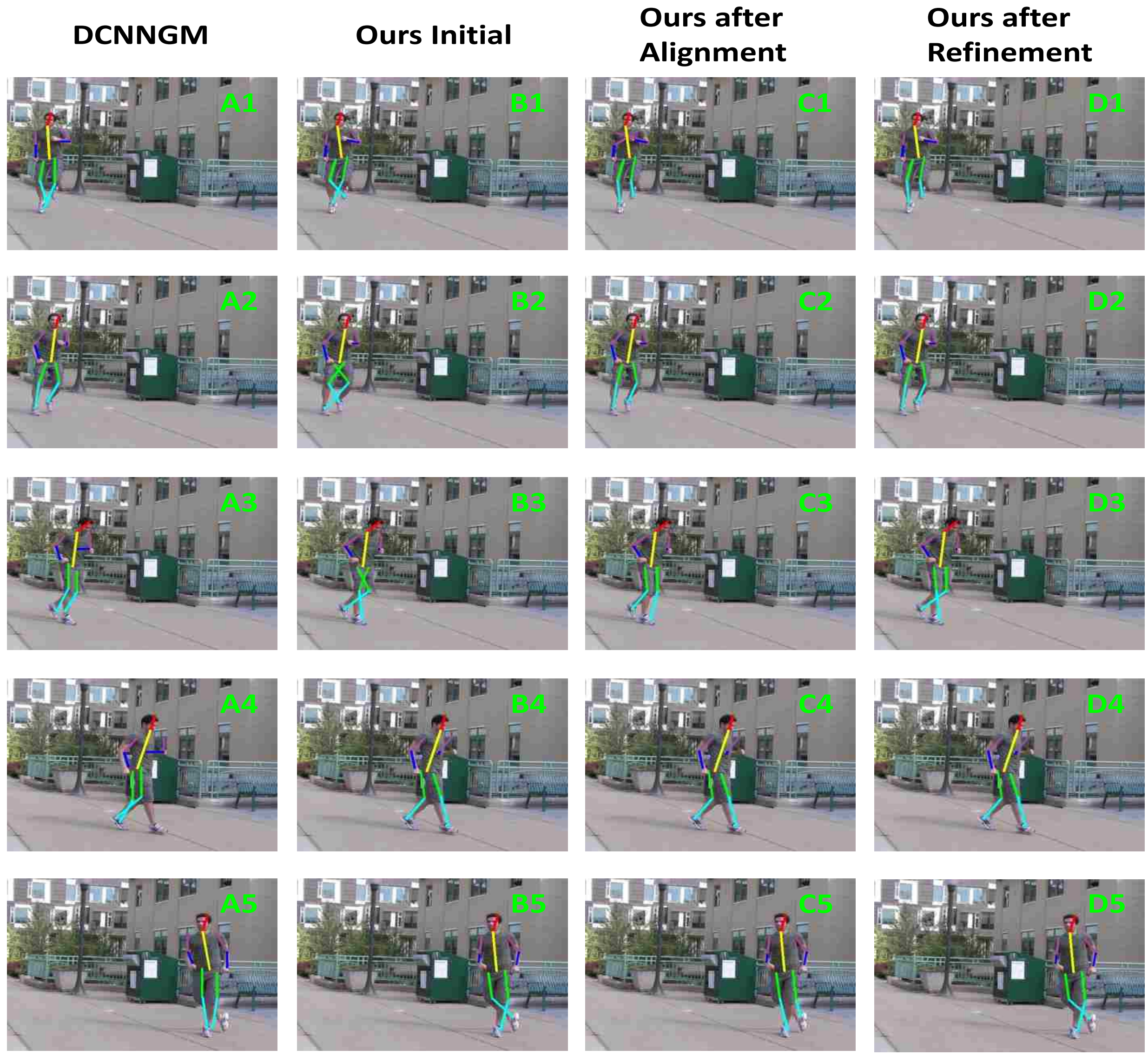


Figure 5.8: An Illustration for Limb Alignments and Refinements. “DCNNGM” shows the initial pose estimation; “Ours Initial” shows the results after “Abstraction” and “Association”; “Ours after Alignment” and “Ours after Refinement” show the results after these two steps respectively.

Fig. 5.8 shows some results of the proposed limb alignment and refinement schemes. As can be seen from the results, joints are much better localized (please see the feet in Fig.5.8(B4) and (B5) compared to (A4) and (A5)); however, the limb inferences are not correct in some frames (e.g. the lower legs in Fig.5.8(B1) and (B2)). After the limb alignment, most of the limb inferences are corrected (e.g. the lower legs in Fig.5.8(B1) and (B2)), however some of them are still not perfect (e.g. Fig.5.8(C3) and (C5)). After the limb refinement, these errors are corrected (e.g. Fig.5.8(C3) and (C5)). Please note that the "limb alignment" sometimes deteriorate the results (e.g. please compare Fig.5.8(C5) to (B5)), and this is because the DCNNGM made a wrong limb inference (e.g. in Fig.5.8(A5)), which failed the limb alignment process. Nevertheless, the "limb refinement" process corrected these errors and we achieve better performance (e.g. Fig.5.8(D3) and (D5)).

5.8 Experiments

5.8.1 Datasets

We evaluated our method on three publicly available datasets:

Outdoor Pose Dataset: this dataset was collected by the authors of [85], which contains six video sequences from outdoor scenes. There are a lot of self-occlusions of the body parts in this dataset. Annotations of more than 1,000 frames are provided by the authors.

Human Eva-I: this dataset [100] contains human activities in indoor controlled conditions. The activities are synchronized with a ground truth of 3D motion capture data, which can be converted into 2D joint locations. In order to have a fair comparison with [85], we use 250 frames from the sequences: *S1_Walking*, *S1_Jog*, *S2_Jog* captured by camera 1.

N-Best Dataset, this dataset was collected by the authors of [78] and has four sequences in total. As a fair comparison to [85], we also report results on sequences *walkstraight* and *baseball*.

5.8.2 Evaluation Metrics

Similar to [85], we use PCP and KLE to evaluate our results. Probability of a Correct Pose (PCP) [26] is a standard evaluation metric, which measures the percentage of correctly localized body parts within a threshold. Keypoint Localization Error (KLE) [85] measures the average Euclidean distance from the ground truth to the estimated keypoints, normalized by the size of the head in each frame.

5.8.3 Comparison of Results

We compare the proposed method with three state-of-the-art video based human pose estimation methods: N-Best method [78], Symmetric Tracking method [85], Mixing Body-part method [15], and a deep learning baseline method (i.e. DCNNGM [12]); we did not compare with some upper body pose estimation/tracking methods [95, 108], since they focus on the modeling of hands/elbows using motion and appearance features but do not handle other body parts. Since [15] was designed for upper-body pose estimation, we re-implemented its algorithm by reusing most of their implementation but extended it to a full-body detection model. Quantitative results are shown in Table 5.1, 5.2 and 5.3, and qualitative results are shown in Fig.5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15 and Fig.5.16. Note that the figures for Symmetric Tracking method are re-produced from figures in [85] since the code is not publicly available.

Table 5.1: Comparisons with the state-of-the-art methods and intermediate stages of the proposed method, on Outdoor Dataset [85] . Note that PCP is an accuracy measure, so the larger the better, with a max of 1; and KLE is an error measure, so the smaller the better. Please refer to Section 5.8.3 for a detailed discussion.

Outdoor Dataset [85]								
Metric	**Method**	**Head**	**Torso**	**U.L**	**L.L**	**U.A.**	**L.A.**	**Average**
PCP	Sym-Trk [85]	0.99	0.86	0.95	0.96	0.86	0.52	0.86
	N-Best [78]	0.99	0.83	0.92	0.86	0.79	0.52	0.82
	Mix-Part [15]	0.87	0.97	0.68	0.89	0.78	0.52	0.79
	DCNNGM [12]	1.00	1.00	0.98	0.94	0.94	0.85	0.95
	Ours HPEV [136] (Baseline)	0.92	1.00	0.84	0.73	0.68	0.47	0.77
	Ours HPEV [136] (Abt. Only)	0.99	1.00	0.89	0.77	0.72	0.53	0.82
	Ours HPEV [136] (Asc. Only)	0.99	1.00	0.87	0.76	0.79	0.56	0.83
	Ours HPEV [136]	0.99	1.00	**1.00**	0.97	0.91	0.66	0.92
	Ours (HPEV [136] + DCNNGM)	1.00	1.00	0.96	0.93	0.81	0.77	0.91
	Ours Final (HPEV [136] + DCNNGM + A.R.)	1.00	1.00	0.97	**0.98**	**0.95**	**0.88**	**0.96**
KLE	Sym-Trk[85]	0.39	0.58	0.48	0.48	0.88	1.42	0.71
	N-Best [78]	0.44	0.58	0.55	0.69	1.03	1.65	0.82
	Mix-Part [15]	0.31	0.72	0.91	0.36	0.44	0.72	0.58
	DCNNGM [12]	0.18	0.15	**0.28**	0.32	0.27	0.34	0.26
	Ours HPEV [136] (Baseline)	0.58	0.45	0.61	0.78	0.75	1.11	0.71
	Ours HPEV [136] (Abt. Only)	**0.16**	0.23	0.48	0.69	0.55	0.78	0.48
	Ours HPEV [136] (Asc. Only)	**0.16**	0.20	0.47	0.64	0.44	0.71	0.44
	Ours HPEV [136]	0.19	0.22	0.35	0.37	0.41	0.61	0.36
	Ours (HPEV [136] + DCNNGM)	0.18	0.15	0.38	0.36	0.38	0.38	0.30
	Ours Final (HPEV [136] + DCNNGM + A.R.)	0.18	0.15	**0.28**	**0.26**	**0.25**	**0.30**	**0.24**

Table 5.2: Comparisons with the state-of-the-art methods and intermediate stages of the proposed method, on Human Eva-I Dataset [100]. Note that PCP is an accuracy measure, so the larger the better, with a max of 1; and KLE is an error measure, so the smaller the better. Please refer to Section 5.8.3 for a detailed discussion.

Human Eva-I Dataset [100]								
Metric	**Method**	**Head**	**Torso**	**U.L**	**L.L**	**U.A.**	**L.A.**	**Average**
PCP	Sym-Trk[85]	0.99	1.00	0.99	**0.98**	**0.99**	0.53	0.91
	N-Best [78]	0.97	0.97	0.97	0.90	0.83	0.48	0.85
	Mix-Part [15]	0.99	1.00	0.90	0.89	0.96	0.62	0.89
	DCNNGM [12]	1.00	1.00	1.00	0.93	0.98	0.74	0.94
	Ours HPEV [136] (Baseline)	1.00	1.00	0.93	0.62	0.44	0.24	0.71
	Ours HPEV [136] (Abt. Only)	1.00	1.00	0.98	0.66	0.43	0.30	0.73
	Ours HPEV [136] (Asc. Only)	1.00	1.00	0.94	0.62	0.45	0.27	0.71
	Ours HPEV [136]	1.00	1.00	1.00	0.94	0.93	0.67	0.92
	Ours (HPEV [136] + DCNNGM)	1.00	1.00	1.00	0.96	0.63	0.67	0.88
	Ours Final (HPEV [136] + DCNNGM + A.R.)	1.00	1.00	1.00	0.96	0.97	**0.78**	**0.95**
KLE	Sym-Trk [85]	0.27	0.48	**0.13**	0.22	1.14	1.07	0.55
	N-Best [78]	0.23	0.52	0.24	0.35	1.10	1.18	0.60
	Mix-Part [15]	**0.13**	0.40	0.23	0.16	**0.14**	0.24	0.22
	DCNNGM [12]	0.19	0.42	0.15	0.17	0.17	0.22	0.22
	Ours HPEV [136] (Baseline)	0.17	0.40	0.34	0.45	0.66	0.84	0.48
	Ours HPEV [136] (Abt. Only)	0.17	0.41	0.29	0.41	0.66	0.75	0.45
	Ours HPEV [136] (Asc. Only)	0.17	0.39	0.33	0.42	0.63	0.74	0.45
	Ours HPEV [136]	0.16	0.42	**0.13**	0.15	0.20	0.24	0.22
	Ours (HPEV [136] + DCNNGM)	0.19	0.42	0.17	0.17	0.29	0.26	0.25
	Ours Final (HPEV [136] + DCNNGM + A.R.)	0.19	0.42	0.14	**0.14**	0.17	**0.20**	**0.21**

Table 5.3: Comparisons with the state-of-the-art methods and intermediate stages of the proposed method, on N-Best Dataset [78]. Note that PCP is an accuracy measure, so the larger the better, with a max of 1; and KLE is an error measure, so the smaller the better. Please refer to Section 5.8.3 for a detailed discussion.

N-Best Dataset [78]								
Metric	**Method**	**Head**	**Torso**	**U.L**	**L.L**	**U.A.**	**L.A.**	**Average**
PCP	Sym-Trk [85]	1.00	0.69	0.91	0.89	0.85	0.42	0.80
	N-Best [78]	1.00	0.61	0.86	0.84	0.66	0.41	0.73
	Mix-Part [15]	1.00	1.00	0.91	0.90	0.69	0.39	0.82
	DCNNGM [12]	1.00	1.00	0.91	0.91	0.96	0.77	0.92
	Ours HPEV [136] (Baseline)	1.00	1.00	0.92	0.87	0.87	0.52	0.86
	Ours HPEV [136] (Abt. Only)	1.00	1.00	0.91	0.89	0.87	0.65	0.89
	Ours HPEV [136] (Asc. Only)	1.00	1.00	**0.93**	0.91	0.87	0.55	0.88
	Ours HPEV [136]	1.00	1.00	0.92	**0.94**	0.93	0.65	0.91
	Ours (HPEV [136] + DCNNGM)	1.00	1.00	0.89	0.87	0.80	0.78	0.89
	Ours Final (HPEV [136] + DCNNGM + A.R.)	1.00	1.00	0.92	0.92	**0.97**	**0.87**	**0.95**
KLE	Sym-Trk [85]	0.53	0.88	0.67	1.01	1.70	2.68	1.25
	N-Best [78]	0.54	0.74	0.80	1.39	2.39	4.08	1.66
	Mix-Part [15]	0.15	0.23	0.31	0.37	0.46	1.18	0.45
	DCNNGM [12]	0.14	0.15	0.29	0.40	0.23	0.43	0.27
	Ours HPEV [136] (Baseline)	0.15	0.19	0.36	0.49	0.32	0.84	0.39
	Ours HPEV [136] (Abt. Only)	0.15	0.19	0.31	0.43	0.34	0.60	0.34
	Ours HPEV [136] (Asc. Only)	0.15	0.17	0.27	0.42	0.29	0.68	0.33
	Ours HPEV [136]	0.15	0.17	**0.24**	0.37	0.30	0.60	0.31
	Ours (HPEV [136] + DCNNGM)	0.14	0.15	0.34	0.36	0.36	0.46	0.30
	Ours Final (HPEV [136] + DCNNGM + A.R.)	0.14	0.15	0.26	**0.29**	**0.23**	**0.34**	**0.24**

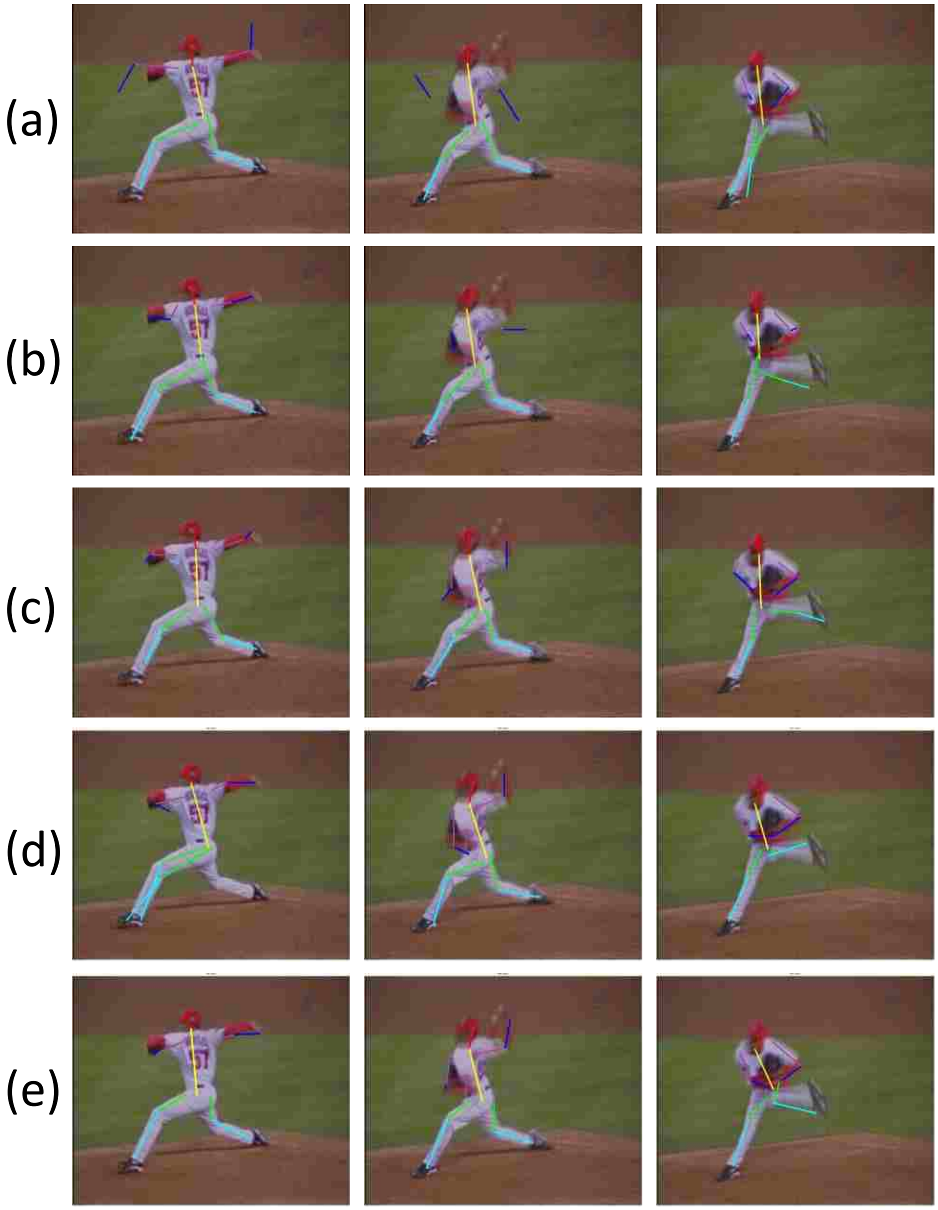


Figure 5.9: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from N-Best Dataset. Body parts are shown in different colors.

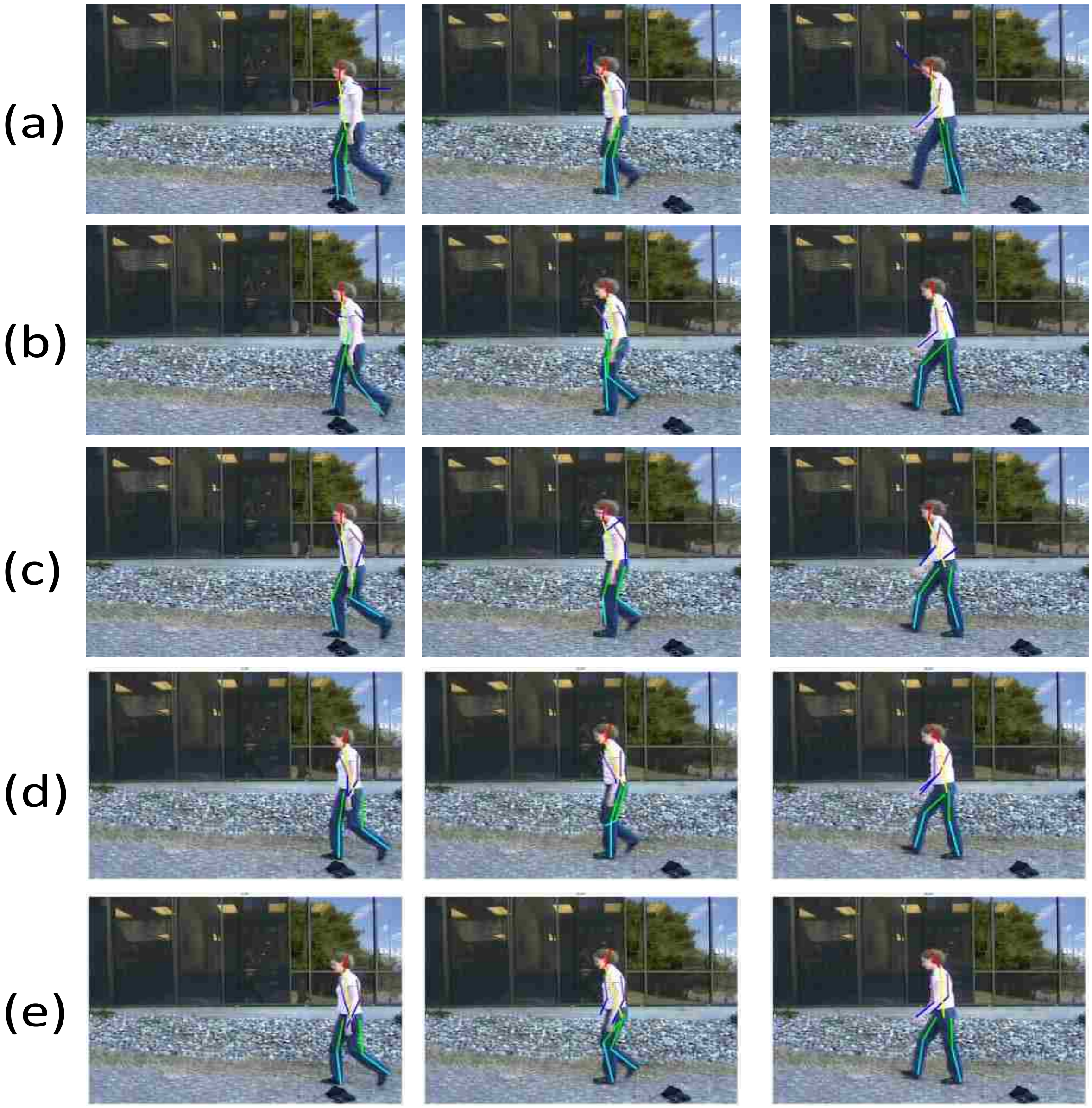


Figure 5.10: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from N-Best Dataset. Body parts are shown in different colors.

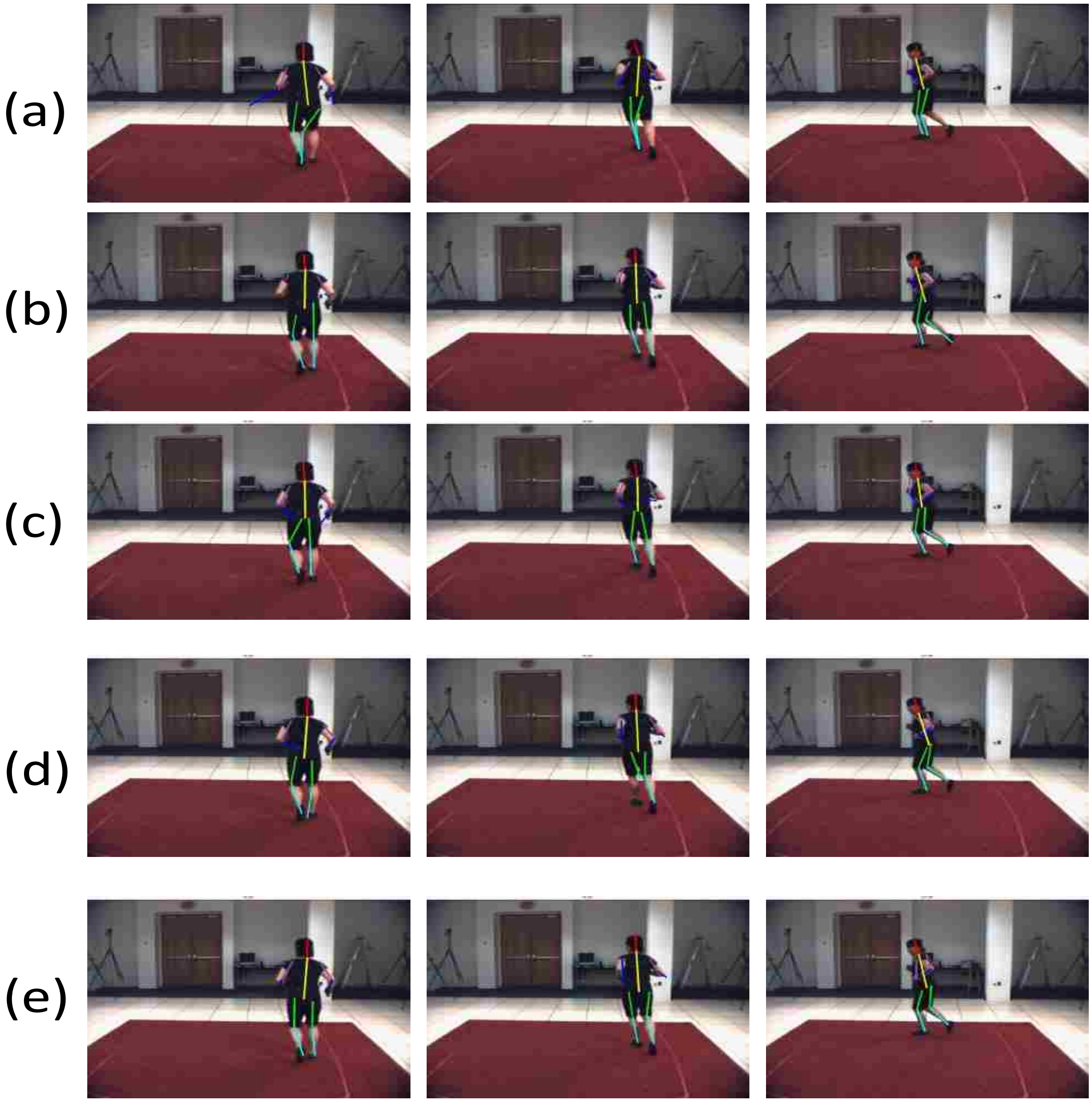


Figure 5.11: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from HumanEva I Dataset. Body parts are shown in different colors.

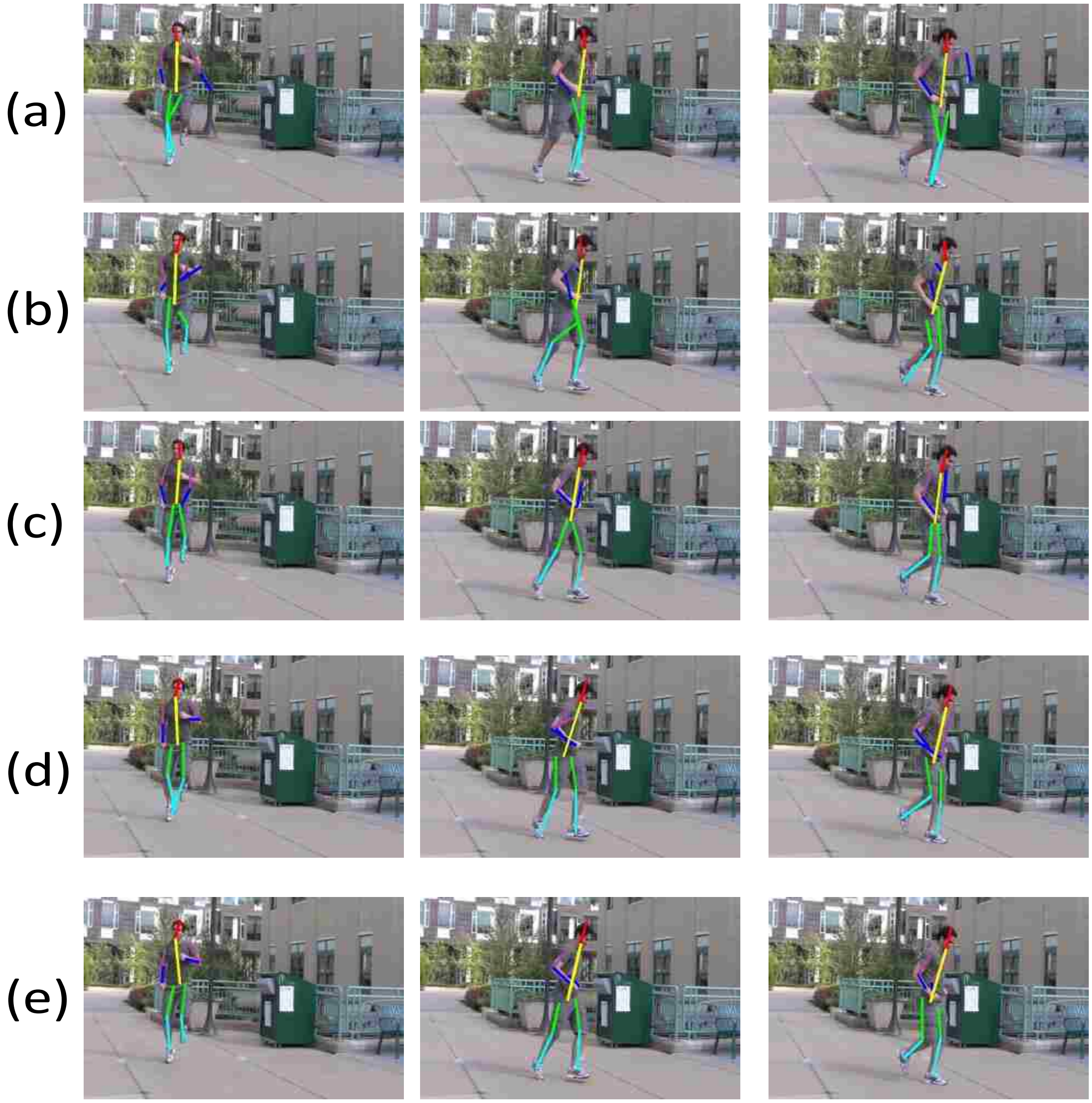


Figure 5.12: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors.

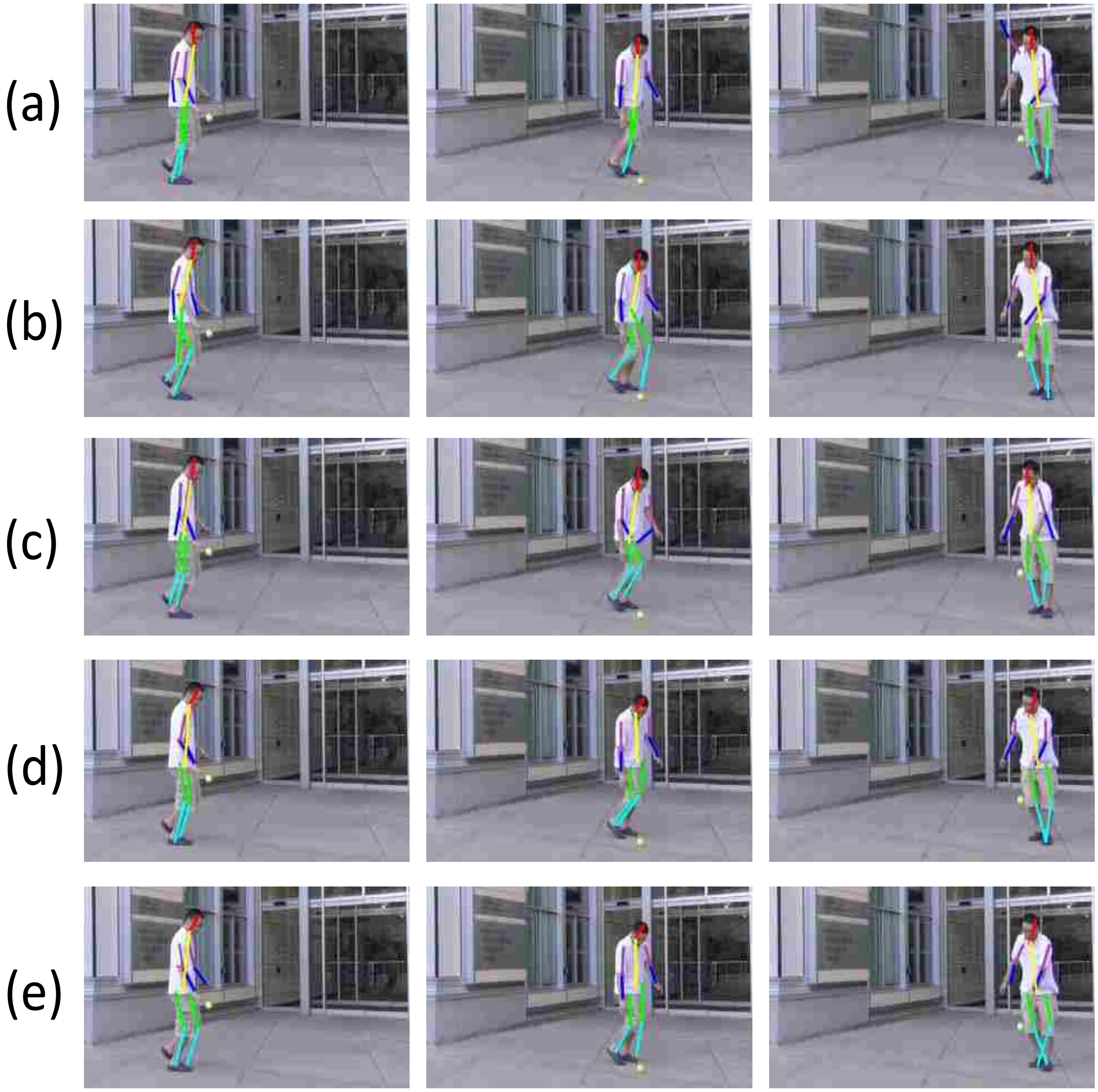


Figure 5.13: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors.

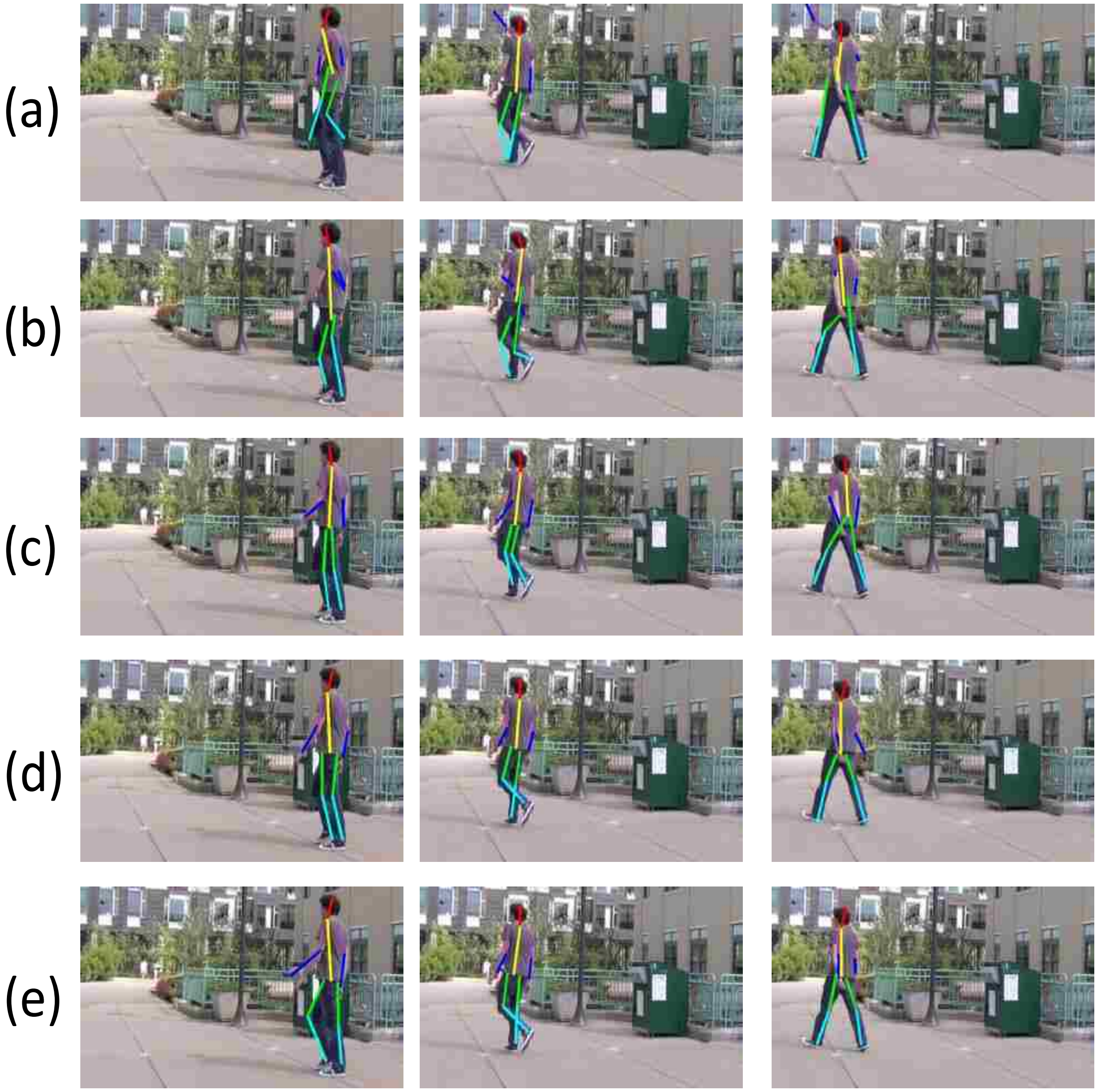


Figure 5.14: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors.

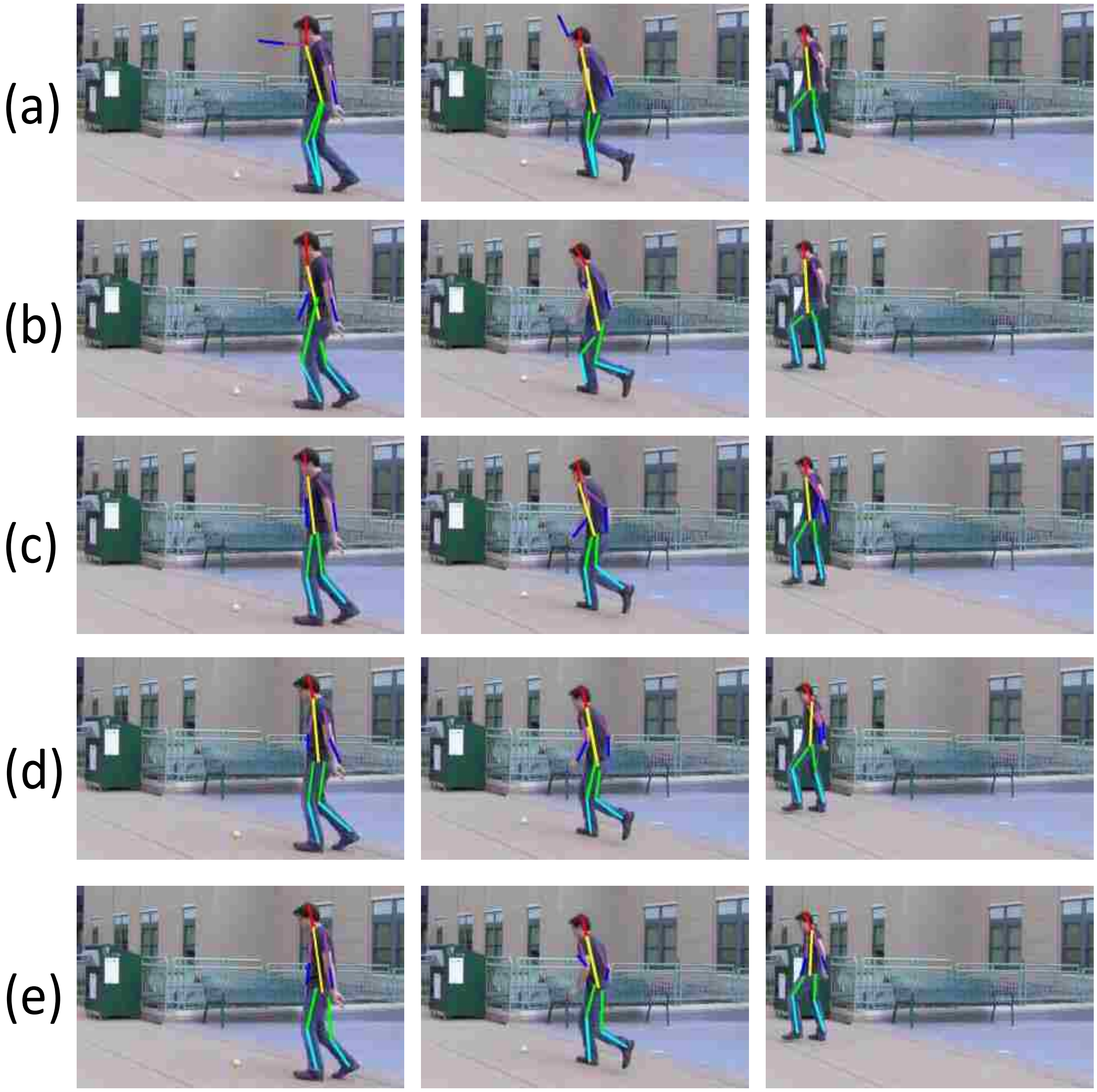


Figure 5.15: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors.

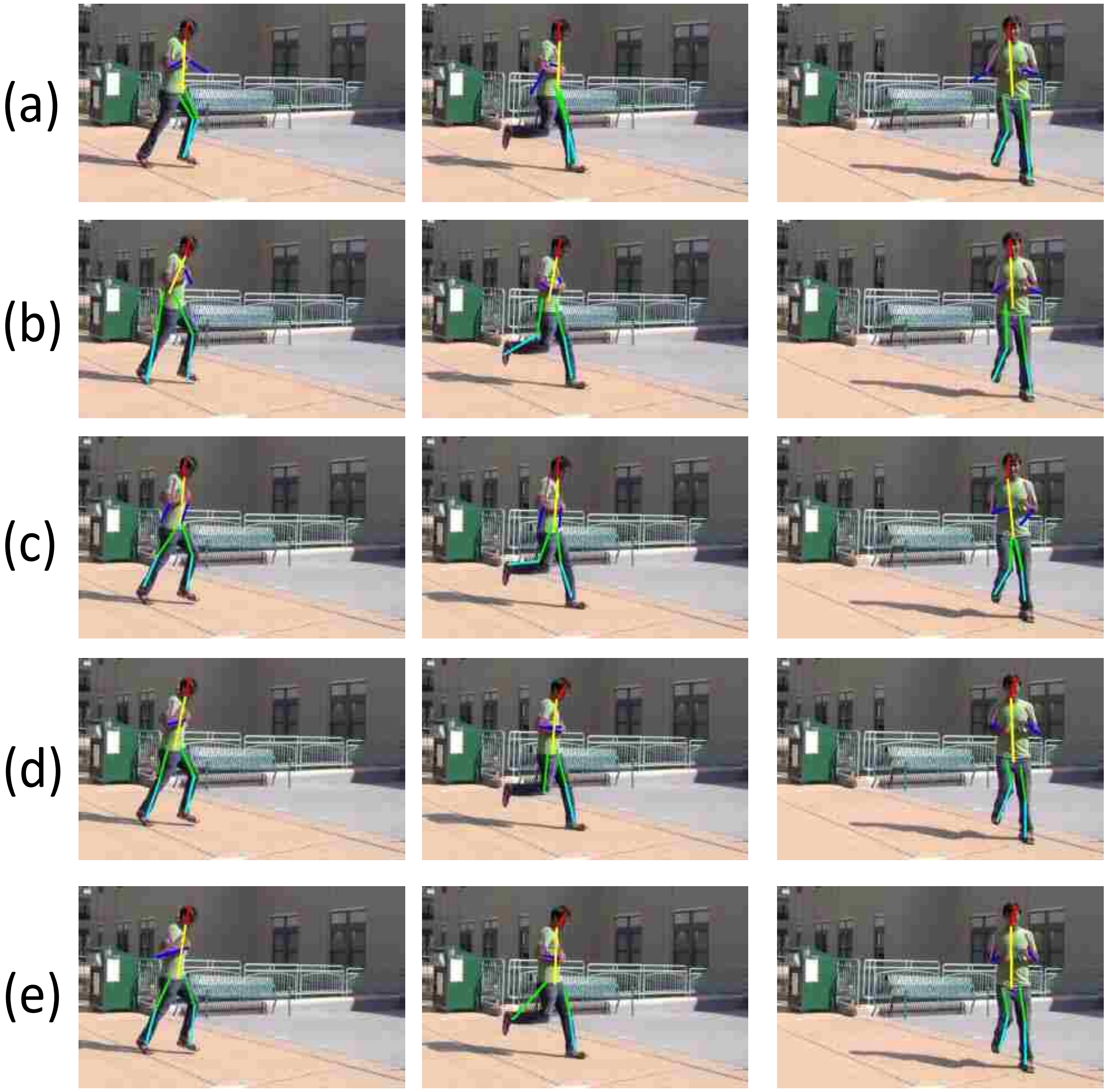


Figure 5.16: Examples and Comparisons with the State-of-the-art Methods: (a) N-Best [78], (b) Symmetric Tracking (Sym. Trk.) [85] , (c) HPEV [136], (d) DCNNGM [12] and (e) Ours. These frames are from Outdoor Dataset. Body parts are shown in different colors.

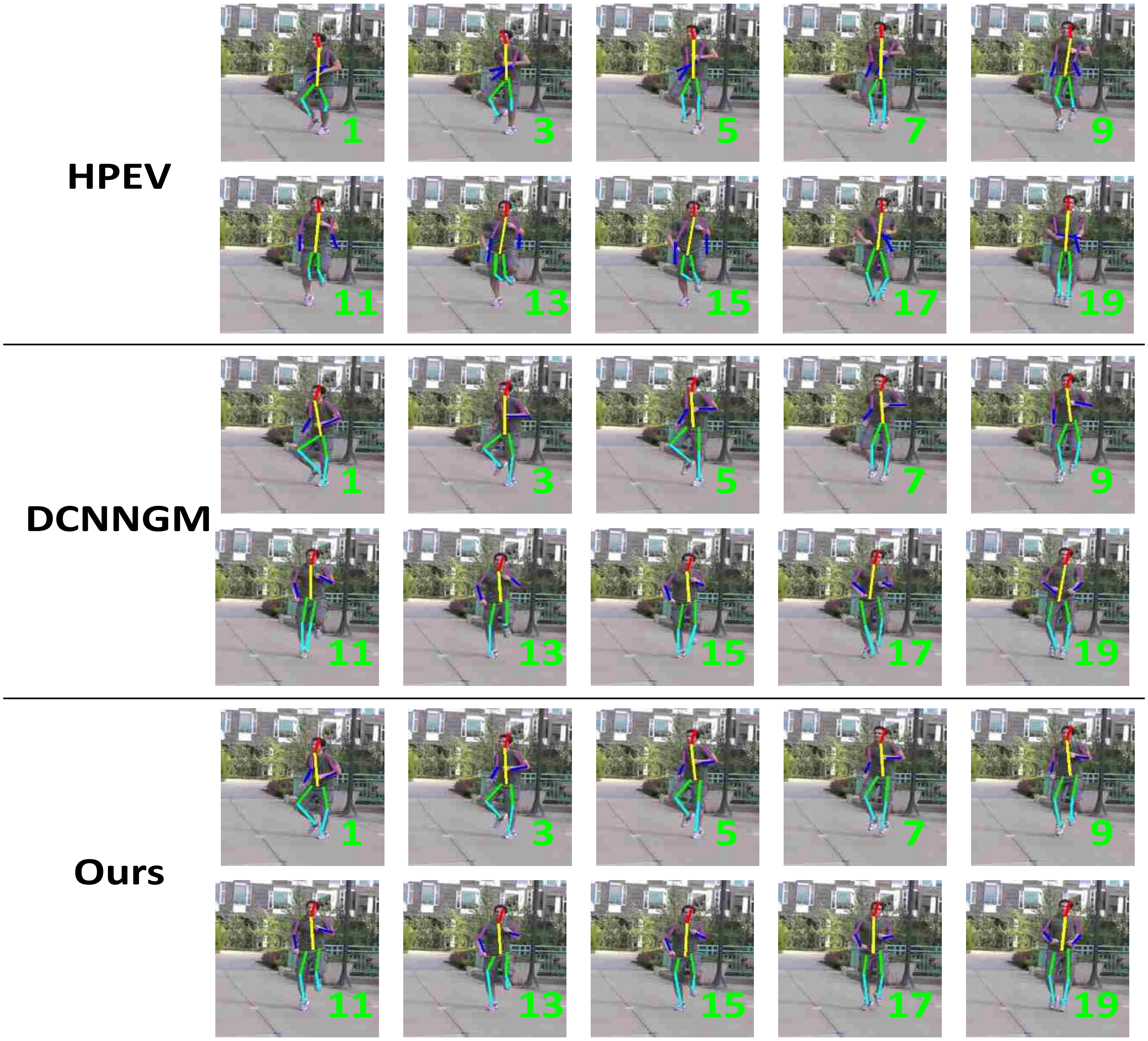


Figure 5.17: Detailed Comparisons of Results from a Video Sequence. Here we want to show the consistency of the results for the proposed method. Please note the pose estimations for the legs: the proposed method can localize the two ankles accurately and smoothly, while "HPEV" results are also smooth but cannot localize the feet correctly in several frames, and "DCNNGM" results are quite noisy and the performance vary a lot through the frames.

We also show detailed results to analyze the contributions of each step of the proposed method. In Table 5.1, 5.2 and 5.3, “Ours HPEV (Baseline)” is the proposed method without using “Abstraction” or “Association”; “Ours HPEV (Abt. Only)” shows the results for only applying the “Abstraction” step of our method; “Ours HPEV (Ass. Only)” shows the results for only using the “Association” step of the proposed method; and “Ours HPEV” is the proposed method with “Abstraction” and “Association”. From these results we found that “Abstraction” is more important than “Association” in the proposed method, due to the fact that it contributes more to the quantitative improvement. Furthermore, “Ours (HPEV + DCNNGM)” is the proposed method with DCNNGM [12] detection + “Abstraction” + “Association”; and “Ours Final (HPEV + DCNNGM + A.R.)” shows the results with “Limb Alignment and Refinement” steps added to the system. As we can see from the results, combining DCNNGM with our method does not necessarily improve the results (actually results deteriorate slightly). This is due to the fact that our original method [136] can not infer the limb configurations accurately (please see Fig.5.8); however, the proposed method has better joint localization results (please see Fig.5.8). Although PCP and KLE do not measure the joint localization accuracy, we can see the joint localization improvement from the figure and also from the final results. Our method boosts the average PCP accuracy above 0.95 for all the datasets. Compared to the state-of-the-art methods,the average improvement in KLE is more than 10%, for PCP we reduced the error by about 15%, which means the proposed method can localize the body parts much more accurately.

From these results, a good question to ask is why there is more improvement in KLE compared to PCP. We believe PCP is a relatively loose measure, and the state-of-the-art methods can already get about 0.8, which means 80 percent of the parts could be localized correctly following the PCP criteria. The possible improvement margin is only 0.2 and our improvement is 0.15. Another reason is that, based on the definition of PCP [26], the parts would be considered ‘correctly’ localized if the overlap ratio is below a threshold; however, KLE measures the distance between the body part and the ground truth. Therefore, sometimes it happens that a body part is considered ‘correctly’ localized by PCP but still it is relatively far from the ground truth. As shown in Fig.5.18, both body parts shown in pink circles are ‘correctly’ localized following the PCP criteria, however it is clear that our part estimations are much closer to the real locations of the body parts.

Sometimes the results are not fully reflected by the numbers. We show detailed results for some consecutive frames (sampled every two frames) in Fig.5.17, and here we only compare our method with two best performing methods. As we can see clearly that the proposed method performs much better (please note the legs, in particular), and the poses are much consistent and all the body parts are located quite precisely. However, when we look at the numbers, the PCP results are exactly the same, and KLE numbers do not reflect the big improvement.

Implementation Details: We process 15 consecutive frames each time. For Eqn.5.3 and 5.6, we normalized the unary and binary weights in each frame between 0 and 1. We use $\alpha = 0.5$ in Eqn.5.1, and $\lambda_c = \lambda_s = \lambda_T = 1$ for Eqn.5.3,5.6 and 5.10. For σ in Eqn.5.2 and θ in Eqn.5.4, we use 10% of the median height (normally 15-30 pixels) of N-Best poses [78] obtained from the step in Section 5.3. For each real body part (Section 5.3), we generate 20 hypotheses, and for each abstract body part we select the top 10 tracklets (Section 5.4 and 5.5). In Eqn.5.13, we use $c = 1/3$

5.8.4 Computation Time

We performed experiments on a desktop computer with Intel Core i7-3960X CPU at 3.3GHz and 16GB RAM. On average, to process one frame (typical frame size: 600×800, we resize the larger frames), the Matlab implementation took $0.5s$ to generate the body part hypotheses and weights, $0.5s$ to build the graph and compute the tracklets, and it took $0.1s$ to build the pose hypothesis graph (Section 5.6) and get the optimal solution. The limb alignment and refinement took less than $2s$ for 100 frames.

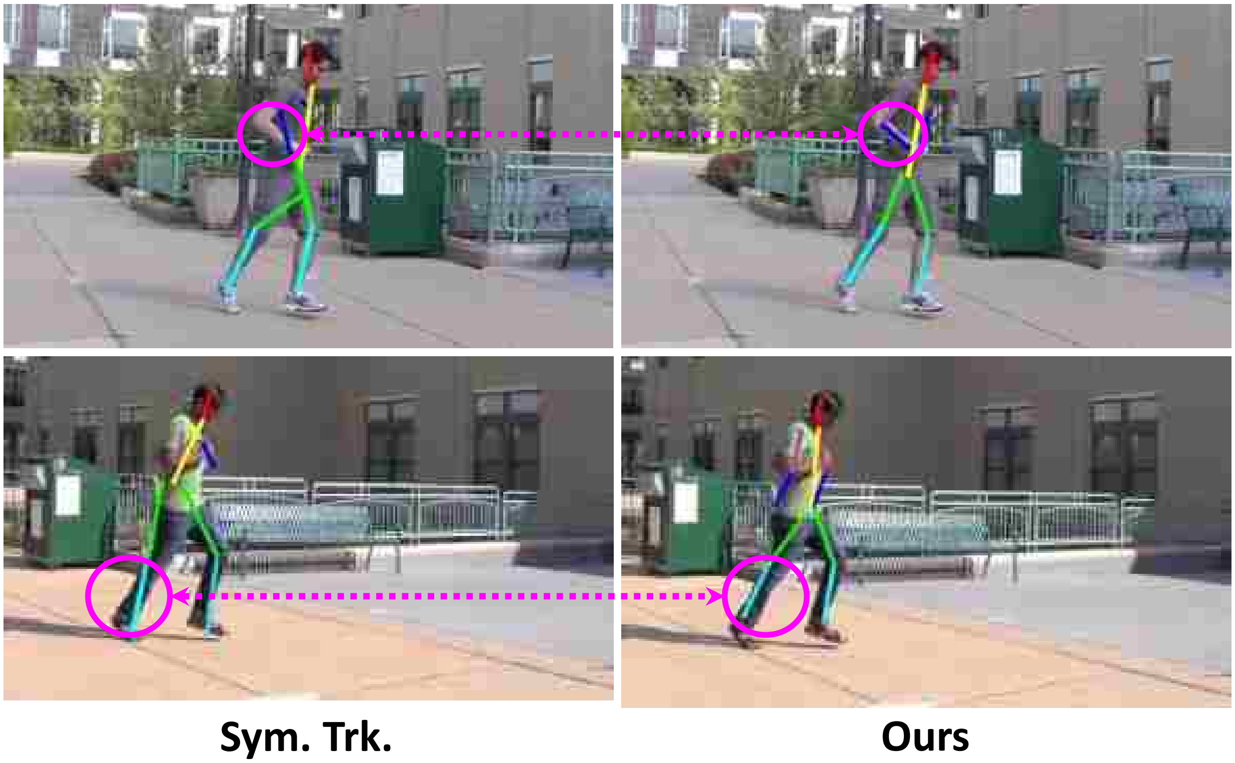


Figure 5.18: The Detailed Comparisons of the Results Obtained by Symmetric Tracking (Sym. Trk. [85]) and the Proposed Method. This example shows that the body part location estimation could be considered ‘correctly’ by PCP [26], but would induce larger error using KLE [85]. For example, for the arms in the top figures and legs in the bottom figures, results for both methods satisfy the PCP criteria; however, our method would have much smaller KLE since the estimated body parts are much closer to the ground truth.

5.8.5 Limitations and Failure Cases

Although, the proposed method works quite well on the datasets, it still has some limitations and possible failure cases (please refer to Figure 5.19). The main reason for the failure cases is that the proposed method relies on the pose hypothesis generation (i.e. using N-Best method [78] or DCNNGM [12]) to generate multiple hypotheses for each body parts in each frame. And if the hypothesis generation method can not generate feasible body part hypotheses, the proposed method will fail. Figure 5.19 shows some failure cases (even though they may not be totally ’failed’ but have larger errors) and they can be classified into three cases. 1) Occlusions (the left three images): if the body parts are occluded, image based pose estimation methods [78, 12] can not predict the exact locations for both the body parts accurately. 2) One or more body parts are far away from the body center (the right three images): if the body parts are far away from the body center, they will be penalized by the spatial constraints from the graphical model in [78, 12]. 3) The human pose in the frame is very different from the poses in the training set (image 1). In the training set of pose estimation model, most of the frames are captured from frontal views.

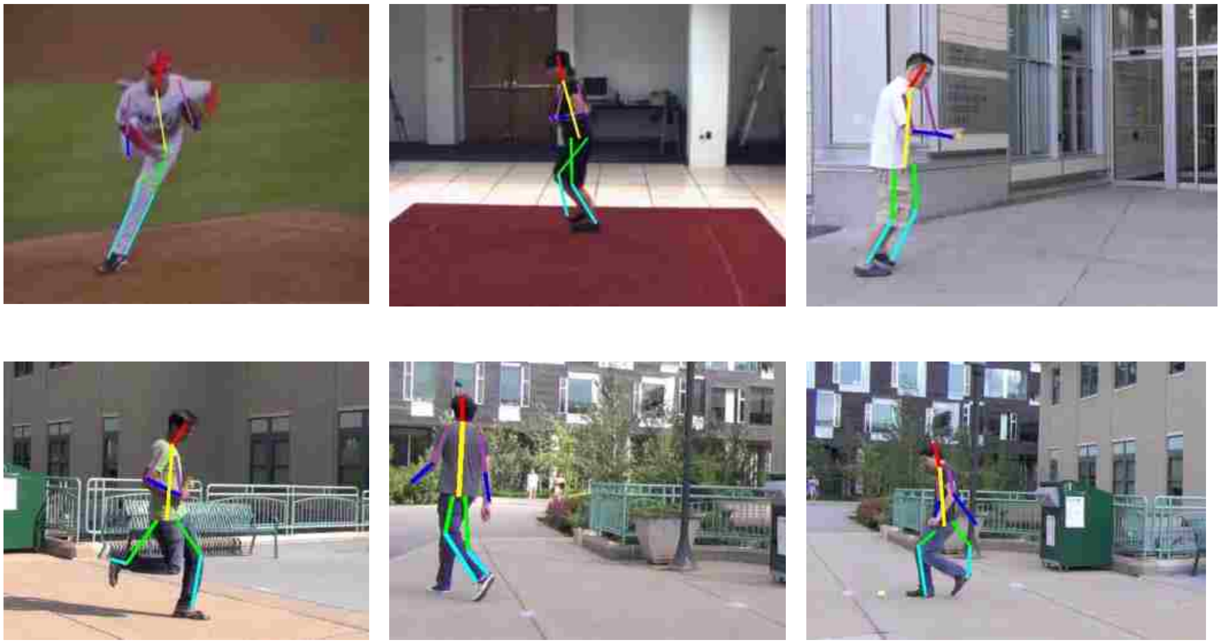


Figure 5.19: Failure Cases. Some of the failure cases are caused by occlusion (1-3), and some are caused by failed detection of joints (4-6).

5.9 Summary

We have proposed a tree-based optimization method for human pose estimation in videos. We have demonstrated that, by using the temporal information within the frames of a video, the performance of human pose estimation in videos can be significantly improved over the the image based pose estimation methods (even for deep learning methods). Our main contribution is mostly focused on reformulating the problem to remove the simple cycles from the graph at the same time maintaining the useful connections at the greatest possible extent, in order to transform the original NP-hard problem into a simpler tree based optimization problem, for which the exact solution exists and can be solved efficiently. It is clear from the experiments that the proposed approach not only elegantly formulates the problem, but also dramatically improves the human pose estimation results in videos. The proposed formulation is general, it has a potential to be employed in solving some other problems in computer vision.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

We have demonstrated in this dissertation that by properly formulating the problems into spatiotemporal graphs, three important video based problems can be solved efficiently and accurately. Two of the algorithms (video object segmentation and pose estimation) introduced in this dissertation can obtain exact solutions, which means that they do not get stuck into local maxima/minima. Furthermore, we have demonstrated that the same dynamic programming solution can be applied to above two problems.

In Chapter 3, we have proposed a novel and efficient directed acyclic graph (DAG) based approach to segment the primary moving object in videos. The major contribution is that we formulate the problem into a DAG, for which the inference can be efficiently performed by dynamic programming. The results from the graph inference are consistent object proposals for all the video frames. This approach also uses innovative mechanisms to compute the 'objectness' of a region and similarity between object proposals across frames. A final step with GMM and MRF ensures an accurate pixel-level video object segmentation. The proposed approach outperforms the existing methods on the publicly available datasets.

In order to handle multiple videos, in Chapter 4, we have formulated the video object discovery and co-segmentation problem into a Regulated Maximum Weight Clique (RMWC) Problem and solved it using a modified version of Bron-Kerbosch Algorithm. The success of the proposed method relies on: 1) use of the objectness measure to obtain spatially coherent region proposals, 2) tracking of object proposals, which selects proposals with consistent appearance and smooth motion over time, and 3) use of different weighting functions for inter-video and intra-video matching for graph construction, which results in improved grouping. Experimental results have shown that the method outperforms the state-of-the-art video co-segmentation methods.

Chapters 3 and 4 aim to segment generic objects from videos. However, human, as the most important object class in videos, is deformable and which makes it very difficult to segment using video object segmentation/co-segmentation methods. Therefore, in Chapter 5, we have proposed a tree-based optimization method for human pose estimation in videos, which can be employed to improve the human segmentation in videos. We have demonstrated that, by using the temporal information within the video frames, the performance of human pose estimation in videos can be significantly improved over the the image based pose estimation methods. Our main contribution is mostly focused on reformulating the problem to remove the simple cycles from the graph at the same time maintaining the useful connections at the greatest possible extent. By doing this, we effectively transform the original NP-hard problem into a simpler tree based optimization problem, for which the exact solution exists and can be solved efficiently. It is clear from the experiments that the proposed approach not only elegantly formulates the problem, but also dramatically improves the human pose estimation results in videos.

In summary, in this dissertation, we have investigated three problems in computer vision and employed spatiotemporal graphs to solve them. For each problem we have different formulations and solutions, but all of them fall into the domain of spatiotemporal graphs. With the spatiotemporal graph formulation, most of the algorithms employed in the dissertation can give us exact solutions. We have compared our results with the state-of-the-art methods on publicly available datasets, and for all of the three problems, we have dramatically improved the results.

6.2 Future Work

In this dissertation, we have shown the success of applications of spatiotemporal graphs in computer vision problems and improved performances on publicly available datasets. However, this is not the end of the research, and further improvements of the methods are needed. Also, inspired by the work of this dissertation, more problems can be explored in a similar way by spatiotemporal graph formulations, such as multi-object tracking, action co-localization, feature point matching, etc. Since we have different formulations and solutions for each of the three problems we explored in the dissertation, we will introduce the future works for each of the problems as follows.

For video object segmentation, the proposed method is highly dependant on the object proposal methods, and cannot handle occlusions very well. Further research is needed to fuse the object proposal methodology with the graph formulation. Also, more robust graph formulation is needed to handle occlusions. Similar formulations and solutions can be extended to online multi-object tracking. For this tracking problem, the objects can be represented by nodes and the spatial constraints between each pair of he objects can be represented by edges. The graph can be simplified properly into a tree, and exact solutions can be obtained efficiently. This can also be applied to crowd tracking in which many objects are moving and spatial constraints are crucial.

For video object co-segmentation, the performance still needs to be improved. Although the method proposed in this dissertation outperforms the state-of-the-art methods, the accuracies are still low. A major issue for the existing methods is that the features are not robust enough to capture the similarity for the objects in different videos, and deep learning methods can be employed to solve this issue. The formulations and solutions can be extended to action co-localization, for which the inputs are multiple videos which contain the same actions, and we want to localize these actions simultaneously. Different features have to be used in this problem, but the graph formulations can be very similar to those in video object co-segmentation.

For human pose estimation in videos, a major drawback of the existing methods is that they rely on the image-based pose estimation results. Motion information is important in videos to infer the locations of body parts. Therefore, an important future research is how to combine the motion information within the deep learning framework to further improve the joint localization accuracy. Similarly, spatiotemporal graphs can be employed to solve the problem of feature point matching and image registration in videos. Traditional methods handle each feature point separately; however, in videos, the spatiotemporal information is also important for feature point matching, since nearby feature points usually have their matches within a small neighborhood. A matching score can be assigned to each feature point. The spatiotemporal constraints can be applied to the neighboring points. And the objective is to match the feature points while keeping their spatiotemporal relationship.

Of course the future works are not limited to these mentioned above, and both theoretical and practical aspects of the spatiotemporal graphs and their applications need further investigation. Also, the applications of spatiotemporal graphs are not limited to computer vision, they can be employed in other disciplines, such as civil engineering, weather forecasting, etc.

LIST OF REFERENCES

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010.

[2] M. Andriluka, S. Roth, and B. Schiele. People tracking-by-detection and people detection-by-tracking. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008.

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2009.

[4] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2010.

[5] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Transactions on Graphics*, 28(3):70, 2009.

[6] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[7] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295, 2010.

[8] X. Cao, F. Wang, B. Zhang, H. Fu, and C. Li. Unsupervised pixel-level video foreground object segmentation via shortest path algorithm. *Neurocomputing*, 172:235–243, 2016.

[9] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, pages 3241–3248, 2010.

[10] D.-J. Chen, H.-T. Chen, and L.-W. Chang. Video object cosegmentation. In *ACM MM*, pages 805–808, 2012.

[11] L. Chen, J. Shen, W. Wang, and B. Ni. Video object segmentation via dense trajectories. *Multimedia, IEEE Transactions on*, 17(12):2225–2234, 2015.

[12] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2014.

[13] H.-T. Cheng and N. Ahuja. Exploiting nonlocal spatiotemporal structure for video segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 741–748. IEEE, 2012.

[14] X. Cheng, W. Lv, H. Liu, X. You, B. Li, and X. Yuan. Improving video foreground segmentation with an object-like pool. *Journal of Electronic Imaging*, 24(2):023034–023034, 2015.

[15] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing body-part sequences for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2014.

[16] W.-C. Chiu and M. Fritz. Multi-class video co-segmentation with a generative multi-video model. In *CVPR*, 2013.

[17] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, pages 1530–1537, 2009.

[18] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.

[19] D. DeMenthon and R. Megret. *Spatio-temporal segmentation of video by hierarchical mean shift analysis*. Computer Vision Laboratory, Center for Automation Research, University of Maryland, 2002.

[20] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision (ECCV)*. 2000.

[21] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, pages 575–588, 2010.

[22] M. D. Fairchild. *Color appearance models*. John Wiley & Sons, 2013.

[23] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014.

[24] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[25] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[26] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008.

[27] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4083–4090. IEEE, 2015.

[28] K. Fragkiadaki, H. Hu, and J. Shi. Pose from flow and flow from pose. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.

[29] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1846–1853. IEEE, 2012.

[30] H. Fu, D. Xu, B. Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173, 2014.

[31] H. Fu, D. Xu, B. Zhang, S. Lin, and R. K. Ward. Object-based multiple foreground video co-segmentation via multi-state selection graph. *Image Processing, IEEE Transactions on*, 24(11):3415–3424, 2015.

[32] Z. Fu, B. Wang, and H. Xiong. Transductive video co-segmentation on the temporal trees. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4471–4475. IEEE, 2015.

[33] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, pages 670–677, 2009.

[34] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *Computer Vision–ACCV 2012*, pages 760–774. Springer, 2012.

[35] F. Galasso, M. Iwasaki, K. Nobori, and R. Cipolla. Spatio-temporal clustering of probabilistic region trajectories. In *ICCV*, pages 1738–1745. IEEE, 2011.

[36] F. Galasso, M. Keuper, T. Brox, and B. Schiele. Spectral graph reduction for efficient image and streaming video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–56, 2014.

[37] F. Galasso, N. Nagaraja, T. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3527–3534, 2013.

[38] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato. Superpixel-based video object segmentation using perceptual organization and location prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4814–4822, 2015.

[39] H. Greenspan, J. Goldberger, and A. Mayer. A probabilistic framework for spatio-temporal video representation & indexing. In *Computer VisionECCV 2002*, pages 461–475. Springer, 2002.

[40] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010.

[41] J. Guo, L.-F. Cheong, R. T. Tan, and S. Z. Zhou. Consistent foreground co-segmentation. In *Computer Vision–ACCV 2014*, pages 241–257. Springer, 2014.

[42] A. Jain, J. Tompson, Y. LeCun, and C. Bregler. Modeep: A deep learning framework using motion features for human pose estimation. In *Computer Vision–ACCV 2014*, pages 302–315. Springer, 2014.

[43] B. Jain and K. Obermayer. Extending bron kerbosch for solving the maximum weight clique problem. *arXiv preprint arXiv:1101.1266*, 2011.

[44] G. JIAMING. *Content Extraction Based on Video Co-Segmentation*. PhD thesis, 2014.

[45] H. Jiang. Human pose estimation using consistent max covering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1911–1918, 2011.

[46] H. Jiang and D. R. Martin. Global pose estimation using non-tree models. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008.

[47] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference (BMVC)*, 2010.

[48] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2011.

[49] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, pages 1943–1950. IEEE, 2010.

[50] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, pages 542–549. IEEE, 2012.

[51] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, 2012.

[52] L. Karlinsky and S. Ullman. Using linking features in learning non-parametric part models. In *European Conference on Computer Vision (ECCV)*. 2012.

[53] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3271–3279, 2015.

[54] A. Khoreva, F. Galasso, M. Hein, and B. Schiele. Learning must-link constraints for video segmentation based on spectral clustering. In *Pattern Recognition*, pages 701–712. Springer, 2014.

[55] A. Khoreva, F. Galasso, M. Hein, and B. Schiele. Classifier based graph construction for video segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 951–960. IEEE, 2015.

[56] J. Kleinberg and E. Tardos. *Algorithm design*. Pearson Education and Addison Wesley, 2006.

[57] M. P. Kumar, P. H. Torr, and A. Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76(3):301–319, 2008.

[58] D. Kumlander. A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search. In *Proc. 5th Int. Conf. on Modelling, Computation and Optimization in Information Systems and Management Sciences*, pages 202–208, 2004.

[59] X. Lan and D. P. Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *Computer Vision (ICCV), IEEE International Conference on*, 2005.

[60] Y. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011.

[61] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Spatiotemporal closure. In *Computer Vision–ACCV 2010*, pages 369–382. Springer, 2010.

[62] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011.

[63] C. Li, L. Lin, W. Zuo, W. Wang, and J. Tang. An approach to streaming video segmentation with sub-optimal low-rank decomposition. *IEEE Transactions on Image Processing*, 25(5):1947–1960, 2016.

[64] C. Li, L. Lin, W. Zuo, S. Yan, and J. Tang. Sold: sub-optimal low-rank decomposition for efficient video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5519–5527, 2015.

[65] K. Li, J. Zhang, and W. Tao. Unsupervised co-segmentation for indefinite number of common foreground objects. 2016.

[66] Z. Lou and T. Gevers. Extracting primary objects by video co-segmentation. *Multimedia, IEEE Transactions on*, 16(8):2110–2117, 2014.

[67] B. Luo, H. Li, T. Song, and C. Huang. Object segmentation from long video sequences. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, pages 1187–1190. ACM, 2015.

[68] T. Ma and L. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012.

[69] T. Ma and L. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677. IEEE, 2012.

[70] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2004.

[71] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *CVPR*, pages 1881–1888. IEEE, 2011.

[72] B. M. Nair, K. D. Kendricks, V. K. Asari, and R. F. Tuttle. Body joint tracking in low resolution video using region-based filtering. In *Advances in Visual Computing*, pages 619–628. Springer, 2014.

[73] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, pages 1583–1590. IEEE, 2011.

[74] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *Computer Vision–ECCV 2014*, pages 737–752. Springer, 2014.

[75] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2337–2344, 2014.

[76] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

[77] S. Paris. Edge-preserving smoothing and mean-shift segmentation of video streams. In *Computer Vision–ECCV 2008*, pages 460–473. Springer, 2008.

[78] D. Park and D. Ramanan. N-best maximal decoders for part models. In *Computer Vision (ICCV), IEEE International Conference on*, 2011.

[79] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.

[80] S. Poullot and S. Satoh. Vabcut: a video extension of grabcut for unsupervised video foreground object segmentation. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 2, pages 362–371. IEEE, 2014.

[81] V. S. Prasath, R. Pelapur, K. Palaniappan, and G. Seetharaman. Feature fusion and label propagation for textured object video segmentation. In *SPIE Defense+ Security*, pages 908904–908904. International Society for Optics and Photonics, 2014.

[82] B. Price, B. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, pages 779–786, 2009.

[83] S. A. Ramakanth and R. V. Babu. Seamseg: Video object segmentation using patch seams. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 376–383. IEEE, 2014.

[84] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *European Conference on Computer Vision (ECCV)*. 2012.

[85] V. Ramakrishna, T. Kanade, and Y. Sheikh. Tracking human pose by tracking symmetric parts. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.

[86] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision (ECCV)*. 2014.

[87] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136, 2006.

[88] D. Ramanan, D. A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2005.

[89] M. Ramanathan, W.-Y. Yau, and E. K. Teoh. Human body part detection using likelihood score computations. In *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2014 IEEE Symposium on*, pages 160–166. IEEE, 2014.

[90] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, pages 1–8, 2007.

[91] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2012.

[92] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics*, volume 23, pages 309–314, 2004.

[93] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *CVPR*, pages 993–1000. IEEE, 2006.

[94] J. C. Rubio, J. Serrat, and A. López. Video co-segmentation. In *ACCV*, pages 13–24. Springer, 2013.

[95] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2011.

[96] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *ICCV*, pages 1219–1225, 2009.

[97] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998.

[98] Q. Shi, H. Di, Y. Lu, and F. Lv. Human pose estimation with global motion cues. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 442–446. IEEE, 2015.

[99] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56:116–124, 2013.

[100] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision (IJCV)*, 87:4–27, 2010.

[101] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2006.

[102] E. Simo-Serra, A. Ramisa, G. Alenyà, C. Torras, and F. Moreno-Noguer. Single image 3d human pose estimation from noisy observations. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2012.

[103] M. Sun, M. Telaprolu, H. Lee, and S. Savarese. An efficient branch-and-bound algorithm for optimal human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2012.

[104] B. Taylor, V. Karasev, and S. Soattoc. Causal video object segmentation from persistence of occlusions. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4268–4276. IEEE, 2015.

[105] D. Teney, M. Brown, D. Kit, and P. Hall. Learning similarity metrics for dynamic scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2084–2093, 2015.

[106] J. Tian, L. Li, and W. Liu. A robust framework for 2d human pose tracking with spatial and temporal constraints. In *Digital lmage Computing: Techniques and Applications (DlCTA), 2014 International Conference on*, pages 1–8. IEEE, 2014.

[107] Y. Tian, C. L. Zitnick, and S. G. Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *European Conference on Computer Vision (ECCV)*. 2012.

[108] R. Tokola, W. Choi, and S. Savarese. Breaking the chain: liberation from the temporal markov assumption for tracking human poses. In *Computer Vision (ICCV), IEEE International Conference on*, 2013.

[109] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42, 2006.

[110] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014.

[111] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2014.

[112] D. Tsai, M. Flagg, and J. Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, page 1, 2010.

[113] D. Varas, M. Alfaro, and F. Marques. Multiresolution hierarchy co-clustering for semantic segmentation in sequences with small variations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4579–4587, 2015.

[114] D. Varas and F. Marques. Region-based particle filter for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3470–3477, 2014.

[115] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, pages 268–281, 2010.

[116] C. Wang, Y. Guo, J. Zhu, L. Wang, and W. Wang. Video object co-segmentation via subspace clustering and quadratic pseudo-boolean optimization in an mrf framework. *Multimedia, IEEE Transactions on*, 16(4):903–916, 2014.

[117] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.

[118] H. Wang and T. Wang. Primary object discovery and segmentation in videos via graph-based transductive inference. *Computer Vision and Image Understanding*, 143:159–172, 2016.

[119] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng. Video object discovery and cosegmentation with extremely weak supervision. In *Computer Vision–ECCV 2014*, pages 640–655. Springer, 2014.

[120] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015.

[121] T. Wang and H. Wang. Graph transduction learning of object proposals for video object segmentation. In *Computer Vision–ACCV 2014*, pages 553–568. Springer, 2014.

[122] W. Wang, J. Shen, X. Li, and F. Porikli. Robust video object cosegmentation. *Image Processing, IEEE Transactions on*, 24(10):3137–3148, 2015.

[123] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3395–3402, 2015.

[124] Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *European Conference on Computer Vision (ECCV)*. 2008.

[125] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. Jots: Joint online tracking and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2226–2234, 2015.

[126] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, pages 626–639. 2012.

[127] Y. Xu, D. Song, and A. Hoogs. An efficient online hierarchical supervoxel segmentation algorithm for time-critical applications. In *BMVC*, 2014.

[128] J. Yang, G. Zhao, J. Yuan, X. Shen, Z. Lin, B. Price, and J. Brandt. Discovering primary objects in videos by saliency fusion and iterative appearance estimation. 2015.

[129] M. Y. Yang, M. Reso, J. Tang, W. Liao, and B. Rosenhahn. Temporally object-based video co-segmentation. In *Advances in Visual Computing*, pages 198–209. Springer, 2015.

[130] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2011.

[131] S. Yi and V. Pavlovic. Multi-cue structure preserving mrf for unconstrained video segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3262–3270, 2015.

[132] T.-H. Yu, T.-K. Kim, and R. Cipolla. Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.

[133] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *ICCV*, pages 1451–1458, 2009.

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

[135] D. Zhang, O. Javed, and M. Shah. Video object co-segmentation by regulated maximum weight cliques. In *Computer Vision–ECCV 2014*, pages 551–566. Springer, 2014.

[136] D. Zhang and M. Shah. Human pose estimation in videos. In *Computer Vision (ICCV), IEEE International Conference on*, 2015.

[137] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia. Semantic object segmentation via detection in weakly labeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3641–3649, 2015.

[138] L. Zhao, X. Gao, D. Tao, and X. Li. Tracking human pose using max-margin markov models. *Image Processing, IEEE Transactions on*, 24(12):5274–5287, 2015.

[139] S. Zuffi, J. Romero, C. Schmid, and M. J. Black. Estimating human pose with flowing puppets. In *Computer Vision (ICCV), IEEE International Conference on*, 2013.