1-1-2013

Multigranularity Representations for Human Inter-Actions: Pose, Motion and Intention

Aikaterini Fragkiadaki
University of Pennsylvania, katef1789@gmail.com

Multigranularity Representations for Human Inter-Actions: Pose, Motion and Intention

Abstract

Tracking people and their body pose in videos is a central problem in computer vision. Standard tracking representations reason about temporal coherence of detected people and body parts. They have difficulty tracking targets under partial occlusions or rare body poses, where detectors often fail, since the number of training examples is often too small to deal with the exponential variability of such configurations.

We propose tracking representations that track and segment people and their body pose in videos by exploiting information at multiple detection and segmentation granularities when available, whole body, parts or point trajectories.

Detections and motion estimates provide contradictory information in case of false alarm detections or leaking motion affinities. We consolidate contradictory information via graph steering, an algorithm for simultaneous detection and co-clustering in a two-granularity graph of motion trajectories and detections, that corrects motion leakage between correctly detected objects, while being robust to false alarms or spatially inaccurate detections.

We first present a motion segmentation framework that exploits long range motion of point trajectories and large spatial support of image regions.

We show resulting video segments adapt to targets under partial occlusions and deformations.

Second, we augment motion-based representations with object detection for dealing with motion leakage. We demonstrate how to combine dense optical flow trajectory affinities with repulsions from confident detections to reach a global consensus of detection and tracking in crowded scenes.

Third, we study human motion and pose estimation.

We segment hard to detect, fast moving body limbs from their surrounding clutter and match them against pose exemplars to detect body pose under fast motion. We employ on-the-fly human body kinematics to improve tracking of body joints under wide deformations.

We use motion segmentability of body parts for re-ranking a set of body joint candidate trajectories and jointly infer multi-frame body pose and video segmentation.

We show empirically that such multi-granularity tracking representation is worthwhile, obtaining significantly more accurate multi-object tracking and detailed body pose estimation in popular datasets.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Computer and Information Science

First Advisor
Jianbo Shi

Keywords
perceptual organization, pose estimation, tracking, video segmentation

Subject Categories
Computer Sciences | Robotics

MULTI-GRANULARITY REPRESENTATIONS FOR HUMAN INTER-ACTIONS: POSE, MOTION AND INTENTION

Aikaterini Ioannou Fragkiadaki

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2013

____
Jianbo Shi, Associate Professor
Computer and Information Science
Supervisor of Dissertation

____
Val Tannen, Professor
Computer and Information Science
Graduate Group Chairperson

Dissertation Committee

Kostas Daniilidis, Professor,
Computer and Information Science

Camillo J. Taylor, Professor,
Computer and Information Science

Ben Taskar, Boeing Associate Professor,
Computer Science and Engineering
University of Washington

Martial Hebert, Professor,
School of Computer Science
Carnegie Mellon University

MULTI-GRANULARITY REPRESENTATIONS FOR HUMAN INTER-ACTIONS: POSE, MOTION AND INTENTION

To my family:

Mom and Dad, for giving me a lifelong place to call home;

my sister Valia for lifelong friendship.

Acknowledgements

I would like to thank my advisor Jianbo Shi whose excitement and devotion to Computer Vision has been more inspiring than anything else. His advising has shaped my research skills equally well as my character towards discipline, patience, persistency, humbleness. The education he has given me will always be part of my research work.

I would like to thank George Pappas for accepting my application at Penn and for allowing me to change topics according to my interests. I would like to thank Kostas Daniilidis for his support and encouragement, throughout my Ph.D. and as chair of my committee. I would like to thank the rest of my committee members, CJ Taylor for his thorough comments and positive energy and Martial Hebert for his directions towards wide and detailed experimentation. I especially thank Ben Taskar for making the FLIC dataset available to me before its public release and wish him a happy and most successful career in UoW.

I would like to thank my co-authors Geng Zhang, Weiyu Zhang, Elena Bernardis, Han Hu for good and fruitful collaboration. We have spent many agonizing and celebrating moments together which I shall never forget, especially the agonizing ones. I would like to thank Ben Sapp for the great discussions we had inside and outside the lab, his great examples on MATLAB hacking and the mutual support during deadlines. I would like to thank Kosta Derpanis for proofreading our papers and providing great suggestions for improvements. I would like to thank Mike Felker for his great secretarial work, taking care of all of us throughout our Ph.Ds.

I would like to thank Anton for making me happy even on days of no results. Thank you for your effort on maintaining our laptops and server, that ensured I could work with comfort from any place outside the lab, our home in Philly or D.C., Cretan seaside and Bulgarian countryside.

ABSTRACT

MULTI-GRANULARITY REPRESENTATIONS FOR HUMAN INTER-ACTIONS: POSE, MOTION AND INTENTION

Aikaterini Ioannou Fragkiadaki

Jianbo Shi

Tracking people and their body pose in videos is a central problem in computer vision. Standard tracking representations reason about temporal coherence of detected people and body parts. They have difficulty tracking targets under partial occlusions or rare body poses, where detectors often fail, since the number of training examples is often too small to deal with the exponential variability of such configurations.

We propose tracking representations that track and segment people and their body pose in videos by exploiting information at multiple detection and segmentation granularities when available, whole body, parts or point trajectories. Detections and motion estimates provide contradictory information in case of false alarm detections or leaking motion affinities. We consolidate contradictory information via graph steering, an algorithm for simultaneous detection and co-clustering in a two-granularity graph of motion trajectories and detections, that corrects motion leakage between correctly detected objects, while being robust to false alarms or spatially inaccurate detections.

We first present a motion segmentation framework that exploits long range motion of point trajectories and large spatial support of image regions. We show resulting video segments adapt to targets under partial occlusions and deformations. Second, we augment motion-based representations with object detection for dealing with motion leakage. We demonstrate how to combine dense optical flow trajectory affinities with repulsions from confident detections to reach a global consensus of detection and tracking in crowded scenes. Third, we study human motion and pose estimation. We segment hard to detect, fast moving body limbs from their surrounding clutter and match them against pose exemplars to detect body pose under fast motion. We employ on-the-fly human body kinematics to improve tracking of body joints under wide deformations. We use motion segmentability of body parts for re-ranking a set of body joint candidate trajectories and jointly infer multi-frame body pose and video segmentation.

We show empirically that such multi-granularity tracking representation is worthwhile, obtaining significantly more accurate multi-object tracking and detailed body pose estimation in popular datasets.

Contents

Acknowledgements **vi**

1 Introduction **1**

1.1 Previous literature 3

1.2 Remaining challenges 5

1.3 Contributions of this thesis 7

1.3.1 Models for spatio-temporal perceptual organization (Chapter 3) . 9

1.3.2 Two-granularity tracking for concurrent multi-object tracking and segmentation (Chapter 4) 9

1.3.3 Graph steering for inferring classification-clustering in two - granularity graphs (Section 4.4) 10

1.3.4 Motion for body pose detection (Chapters 5,6) 10

1.4 Published work supporting this thesis 11

2 Preliminaries **12**

2.1 Spectral Partitioning 13

2.2 Clustering with Bias 16

3 Spatio-temporal Perceptual Organization **19**

3.1 Introduction 20

3.2 Related work 25

3.3 Motion segmentation from trajectories and regions 28

3.3.1 Trajectory spectral discontinuities 28

3.3.2 Trajectory to pixel partitioning via random walkers 35

3.3.3 Experiments 38

3.4 Topological Tracking 46

3.4.1 Experiments 50

3.4.2 Discussion 52

4 Two-Granularity Tracking 53

4.1 Introduction 54

4.2 Related work 58

4.3 Tracking units 60

4.3.1 Fine-grained point trajectories 61

4.3.2 Coarse-grained detectlets 62

4.3.3 Trajectory to detectlet associations 64

4.4 Two-view steering cut 65

4.4.1 Graph steering 65

4.4.2 Steering cuts 68

4.4.3 Graph steerability 73

4.5 Experiments 77

4.6 Discussion 90

5 Two-Granularity Body Pose Tracking 91

5.1 Introduction 91

5.2 Related work 93

5.3 Pose and segmentation model 96

5.3.1 Space-time body pose graph 96

5.3.2 Detection-by-tracking: State binding for space-time pose estimation 98

5.3.3 Trajectory motion graph 101

5.3.4 Body part to trajectory cross-associations 102

5.3.5 Joint pose and motion figure-ground segmentation 103

5.3.6 Body pose estimation under interactions 105

5.4 Experiments 106

5.5 Discussion 112

6 Articulated Optical Flow 113

6.1 Introduction 114

6.2 Related work 116

6.3 From flow to pose 117

6.3.1 Region motion affinities 118

6.3.2 Region detection-driven repulsions 119

6.3.3 Steering cut 120

6.3.4 Matching pose segmentations to exemplars 122

6.4 From pose to flow 123

6.4.1 Articulated optical flow 123

6.5 Experiments 127

6.6 Discussion 129

7 Conclusion 130

List of Tables

3.1 Quantitative segmentation results in the Berkeley motion segmentation benchmark. 40

3.2 Quantitative segmentation results of topological tracking in Berkeley motion segmentation dataset 50

3.3 Quantitative segmentation results of topological tracking in Figment . . . 51

4.1 Two-granularity tracking results 83

List of Figures

1.1 Multi-granularity representations for person and body pose tracking. . . . 3

1.2 Previous tracking literature. 4

1.3 Human body pose manifold. 6

3.1 Gains and ambiguities in motion estimation. 23

3.2 Trajectory spectral discontinuities 30

3.3 Multiscale trajectory partitioning 33

3.4 Robustness of discontinuities to the number of eigenvectors 34

3.5 Random walkers on spatio-temporal region graphs 37

3.6 Qualitative segmentation results in the Berkeley motion segmentation benchmark. 39

3.7 Video segmentation results in the dataset of Chen and Corso (2010) . . . 42

3.8 Results in the CMU boundary detection benchmark 43

3.9 Pixel probability of occlusion boundary map. 44

3.10 Precision-recall curve in the CMU occlusion boundary benchmark 44

3.11 Comparison of spectral embedding discretizations. 46

3.12 Motion versus topology 47

3.13 Trajectory motion saliency 49

3.14 Topology-driven trajectory repulsions 49

3.15 Qualitative segmentation results in Figment 51

4.1 Two-granularity trackers 55

4.2 Two-granularity tracking overview. 57

4.3 Complementarity of motion segmentation and detection. 60
4.4 Detection linking into detectlets 63
4.5 Trajectory to detectlet cross-associations 64
4.6 Graph steering 66
4.7 Steering and graph connectivity 67
4.8 Comparison of motion, disparity and detection driven trajectory affinities 70
4.9 Graph steering versus co-clustering 72
4.10 Feasibility analysis of steering cut 74
4.11 Two-granularity tracking results in UrbanStreet 78
4.12 Two-granularity tracking versus standard tracking-by-detection 79
4.13 Groundtruth segmentations in UrbanStreet 80
4.14 Drifting in two-granularity tracking 81
4.15 Comparison of 2D and 3D two-granularity tracking in UrbanStreet 82
4.16 Tracklet fragmentations 83
4.17 Two-granularity baseline systems 84
4.18 Two-granularity tracking results in PETS S2.L1 85
4.19 Two-granularity tracking results in TUD crossing 86
4.20 Two-granularity tracking results in ETH sunny day 87
4.21 Two-granularity tracking results in iLids Hard dataset 88
4.22 Two-granularity tracking for multi-class object tracking in crowded scenes 89
5.1 Body pose estimation under occlusions. 94
5.2 Markov Random Field for space-time human body pose. 97
5.3 Body joint state binding. 99
5.4 Tree decomposition. 101
5.5 Trajectory motion graph. 102
5.6 Body part segmentability. 104
5.7 Pose detection under occlusions 105
5.8 Assigning body joint trajectories to tracklets. 107

5.9 Quantitative pose estimation results in FLIC dataset. 108
5.10 Qualitative pose estimation results in FLIC dataset. 110
5.11 Trajectory motion graph. 111
6.1 Long trails of human body pose distribution. 115
6.2 Pose from flow. 117
6.3 Articulated optical flow 124
6.4 Comparison of articulated flow and standard optical flow. 125
6.5 Qualitative pose detection results in the Friends dataset 127
6.6 Quantitative pose detection results in the Friends dataset 128

Chapter 1

Introduction

> The only thing worse than being blind is having sight but no vision.
>
> — Hellen Keller

Visual sensors on cellphones, tablets, glasses, clothes, computers, produce large amounts of video footage daily. Progress on robotic automation and human-computer interaction depends on how well we can understand and predict the visual world captured from these sensors. Understanding what the people are doing, how they interact with their environment, and what they want to do next in the video footage is important for applications in entertainment, safety, health care, education. This thesis is about estimating people's body motion in unconstrained videos from various sensors, as a step towards understanding their activities.

The fact that human body motion aids recognition of human activity dates back to the experiments of Johansson (1973): Johansson showed that image sequences of point-lights attached to the limbs of a moving actor could be identified as depicting actions, although they did not define a form when stationary. In other words, a set of point light trajectories were enough to create the perception of complex human actions, such as ballet pirouettes.

Obtaining automatically long temporal correspondences of moving body joints, similar to those perceived by the human subjects in Johansson's experiments, is challenging. Body motion estimation is a difficult problem on its own and has received a lot of attention in Computer Vision community. In fact, estimating pixel motion is often ambiguous without appropriate object specific or object independent, smoothness driven, priors. We will explore representations that combine imperfect motion estimates with detectors for segmenting and tracking people and their body pose in videos.

We propose multi-granularity representations for detection, segmentation and temporal association in videos.

- Detectors reliably recognize image parts well captured in their training sets. Usually such image parts come at a coarse granularity since too small templates are ambiguous to recognize. This is the case also for part based representations, where a set of parts are scored together to disambiguate the small image support of each one of them.

- Video segmentors partition fine-grain entities such as pixels or point trajectories into a potentially exponential number of groups, guided by object independent grouping principles, such as motion coherence. These groups may correspond to very small or very large image parts, that can or cannot be reliably recognized by detectors.

- Trackers associate image parts from one frame to another, by learning a model for the image part against its surroundings, or by exploiting smoothness of correspondence and global optimization, or both. They may incorporate object specific kinematic constraints.

Our goal is to obtain detections at any granularity needed in the video (e.g., body parts of a half occluded person) and segmentation at the right semantic granularity (e.g., segments do not fragment torso from legs or leak across similarly moving people). We will show that jointly reasoning over segmentation and detection in videos can achieve this goal. Specifically, we propose segmentation of fine-grain point trajectories guided by both

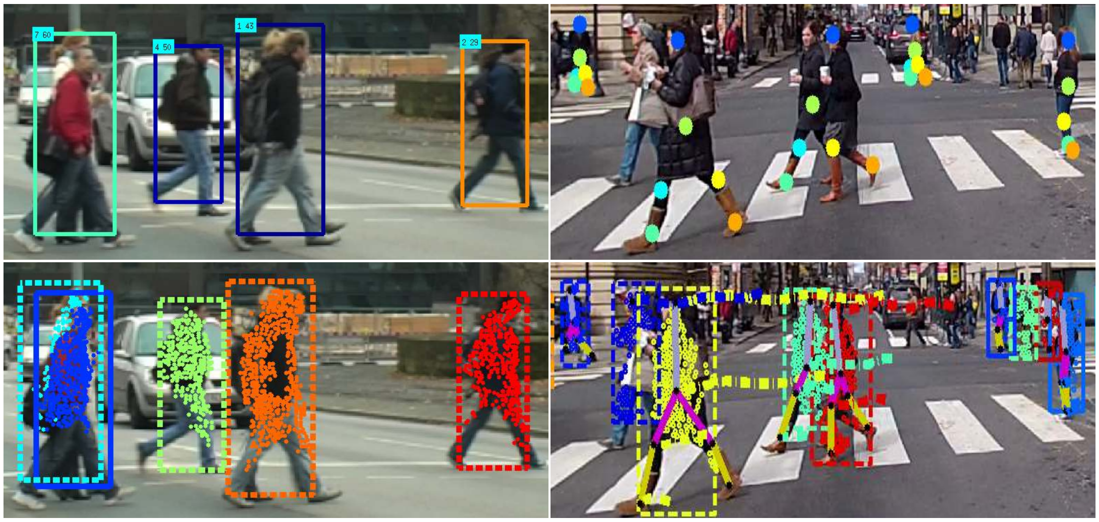


Figure 1.1: Multi-granularity representations for people and body pose tracking. 1st Row: Detection responses of Bourdev et al. (2010) (left) and body pose estimates of Yang and Ramanan (2011), retrained on pedestrian poses of Andriluka et al. (2008) (right). Detectors reliably recognize configurations captured in their training sets, such as the distinctive wide leg walking pose. They often fail at partial occlusions where detection responses span across closeby people. 2nd Row: Our proposed two-granularity tracking for whole object (left) and detailed body pose tracking (right). We track heavily occluded people and recognize partially visible body poses by combining detections with spatio-temporal grouping of trajectories.

object independent, motion based grouping relationships and object specific, detection based repulsive ones. We will use the term "tracking" to refer to parsing of people and/or their body pose in videos, although the term may be overloaded.

1.1 Previous literature

There is a tremendous previous Computer Vision literature on tracking people and their body pose in monocular or multi-view videos. To understand the connections between previous methods and their contributions, we decompose the tracking problem into two

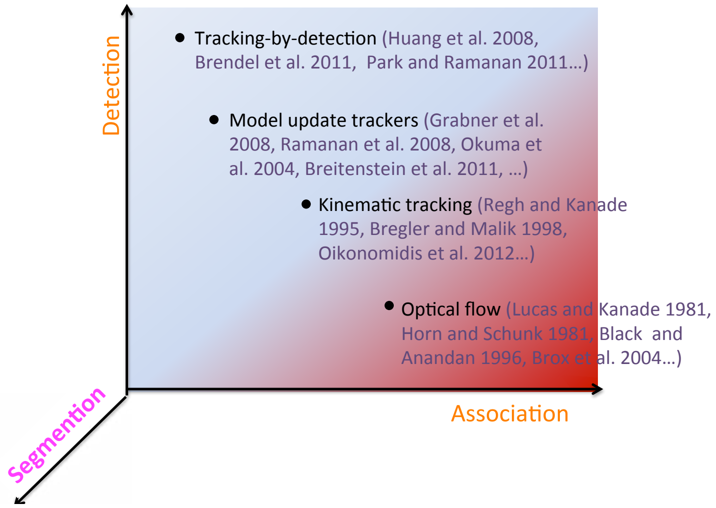

Detection
Tracking-by-detection (Huang et al. 2008, Brendel et al. 2011, Park and Ramanan 2011…)
Model update trackers (Grabner et al. 2008, Ramanan et al. 2008, Okuma et al. 2004, Breitenstein et al. 2011, …)
Kinematic tracking (Regh and Kanade 1995, Bregler and Malik 1998, Oikonomidis et al. 2012…)
Optical flow (Lucas and Kanade 1981, Horn and Schunk 1981, Black and Anandan 1996, Brox et al. 2004…)
Association
Segmention

Figure 1.2: We explore the interplay between association, motion segmentation and detection for tracking under partial occlusions and body deformations, where standard detectors and data association often fail.

tasks: detection and association. Determining the state of a target at a given frame can be done both by corresponding to a training exemplar (detection) or by corresponding to the target state in the previous (or next) frames (association). Detection is by definition object specific. Association can be object specific using object tailored kinematic constraints, or object independent, using general smoothness driven assumptions in matching pixel appearance from frame to frame.

We place previous works in a 2D diagram that has detection and association as its main axes, shown in Figure 1.2. In the bottom right part of the diagram we have optical flow methods that estimate pixel motion in an object independent way, without using any information about the content of the video scene. Kinematic tracking methods track an initialized template in time using object specific kinematic constraints. Recent progress of object detectors has shifted attention towards tracking-by-detection approaches that track by linking detections across consecutive frames, such as works of Brendel et al. (2011); Huang et al. (2008b); Park and Ramanan (2011). Tracking works of Breitenstein et al. (2009); Grabner et al. (2008); Okuma et al. (2004); Yang and Nevatia (2012) use along with pre-trained object models target specific models ones, which they update online.

1.2 Remaining challenges

Partial occlusions, unusual object configurations and fast body motion are the three challenges we identify, still hard to deal with state-of-the-art tracking methods.

Partial occlusions People interactions result in partial occlusions, a challenge for both detection and data association:

1. Detectors often fail under partial occlusions. Object like features are erroneously aggregated from both the occluder and the occludee, resulting in either a spatially inaccurate detection, spanning across the two closeby objects, or a miss detection due to mismatched or missing template information, as depicted in Figure 1.1. In essence, partial occlusions ask for a large number of different detection models to be checked against the visual input for matching, each one representing a different occlusion scenario. Learning and calibrating different occlusion models is not easy from current size of training sets.

2. Tracking objects under partial occlusions is difficult due to the continuous change of object visibility masks. Committing to a fixed bounding box shaped object state often misses objects under occlusions and cannot keep good track of target's visibility.

Unusual object configurations Unusual object configurations are captured by few training examples and are hard to detect reliably with current detectors. The distribution of visual data is often characterized by few, frequently re-occurring templates and a large collection of rare ones, forming the long tails of the distribution. This is particularly the case for the human body. People take a wide range of body poses, with few re-occurring ones, depicted at the upper right part of Figure 1.3, and lots of rare ones, depicted at the lower left. While each one of them is rare, collectively they comprise a big chunk of the body pose distribution. This fact is not a result of dataset bias but rather reflects the unbalanced frequency of body poses in actors' pose repertoires.

Unusual object configurations are hard to harvest training exemplars for. One solution is to use graphics simulations and rendering techniques to collect training exemplars for the desired rare configurations, as proposed in Shotton et al. (2013). Though this may be a promising direction, we expect the realism of such exemplars to be low for simulating human motion.

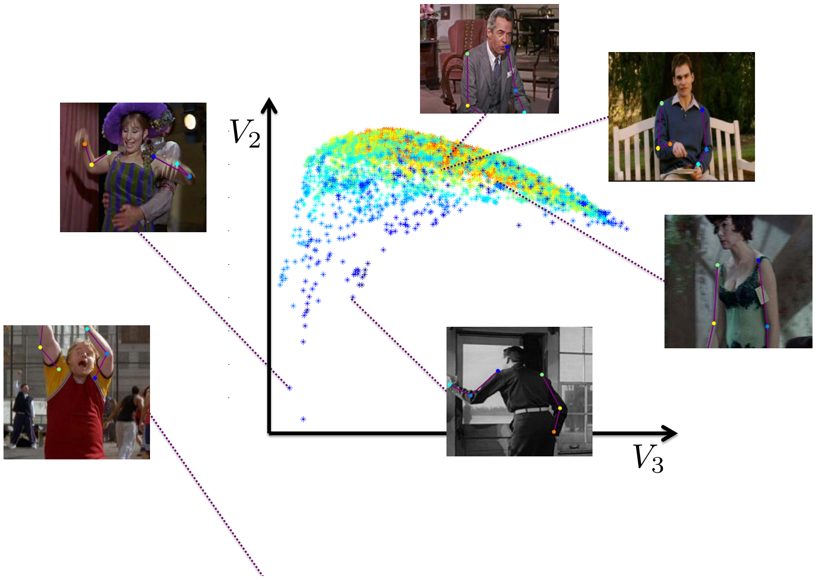


Figure 1.3: Human body pose manifold computed from the pose training exemplars of Sapp and Taskar (2013). We visualize body pose training exemplars using the top two spectral eigenvectors of their affinities, that depend on body pose similarity. Point color indicates number of close neighbors, red corresponds to large number of neighbors. Widely deformed poses are outliers in the dataset. Notice the large number of dark blue points. They correspond to rare body poses. Each one of them is rare but collectively they comprise a large chunk of the body pose distribution.

Fast body motion Fast body motion is hard to track with state-of-the-art coarse-to-fine gradient-based motion estimation schemes such as Brox et al. (2004); Lucas and Kanade (1981): body parts are small and thus are often lost in the coarse levels of the image pyramid. Their large displacements are hard to recover from the finer pyramid levels. On the other hand, descriptor matches often slide along body part axes due to aperture problems.

As a result, descriptor augmented optical flow methods, such as Brox and Malik (2010a), often fail to track fast body motion. Kinematically constrained tracking introduced in Bregler and Malik (1998) exploits human body connectivity along articulated chains, and may outperform bottom-up motion estimation methods, such as optical flow or point tracking, when the human skeleton is initialized. However, it still fails under frequent self occlusions from large body deformations, as noted in Datta et al. (2008).

1.3 Contributions of this thesis

This thesis contributes to the tracking literature with representations and inference frameworks that employ multi-granularity detection and association for tracking people and their body pose under partial occlusions and unusual pose configurations.

We start by presenting a spatio-temporal perceptual organization method in Chapter 3 that segments a video using long range motion of trajectories and large spatial support of image regions. Pixel trajectories are computed *bottom-up*, independent to the video content. They encode strong grouping relationships in their long range motion (and disparity in case of multi-view video) based similarities.

The success of the video segmentation framework in tracking entangled objects in monocular videos shows there is rich information in the video signal and its motion even without any model matching. Resulting segments accurately capture partially occluded objects and objects under unusual configuration, which are challenges for object detectors. However, we identify two problems with our video segmentation method, which characterize any object independent motion segmentation method in general:

1. Model selection. For general non-rigid object motion, grouping cost functions have multiple optima corresponding to coarser of finer partitioning of the objects and the scene. Since in general object motion is not uniform, such partitionings may capture the whole object or object parts. Object independent segmentation is an ill

posed problem in that sense since all partitionings are equally good. The one corresponding to the desired semantic segmentation is not easy to pick without additional information.

2. Leakage under lack of distinct motion across objects. In case of no motion or accidental long range motion similarity across objects, motion affinities and resulting clusters leak across objects.

To recover from those problems, in Chapters 4 and 5 we combine video segmentation with detector responses. We ask for alignment between segments and detections, which selects the right segmentation granularity and addresses the model selection problem. Further, we introduce graph steering where motion affinities are changed (steered) from repulsions induced between trajectories associated with incompatible detections. We show robustness of our framework against spatially inaccurate or false alarm detections and their wrongly induced repulsions. Graph steering addresses the problem of motion leakage in case of accidental motion similarity across objects. Trajectories propagate detections across miss-detection gaps, where detections are whole objects in Chapter 4 and body joints in Chapter 5.

Object independent point trajectories though have limitations: they fail to track fast body motion. Their frequent fragmentations under body self occlusions and deformations, limit their usefulness. In Chapter 6 we explore ways of using on-the-fly body pose detection to inject kinematic constraints in computing motion trajectories. Such constraints help deal with the untexturedness and aperture problems of human body limbs and their fast motion.

We summarize our technical contributions below.

1.3.1 Models for spatio-temporal perceptual organization (Chapter 3)

We cast video segmentation as partitioning of point trajectories and image regions. We have the following contributions:

- **Multiscale trajectory partitioning** for video segmentation. We introduce a discontinuity detector that estimates a probability of boundary between spatially adjacent trajectories. It detects sudden drops or peaks of spectral embedding affinities between neighboring trajectories. Thresholding such probability of boundary in various cutoff values provides trajectory partitions in different levels of granularity. Each trajectory cluster corresponds to an object hypothesis in space and time.

- **Random walkers on multiscale region graphs** for mapping point trajectory clusters to image regions. Regions that overlap well with trajectory clusters are the designated seeds. Seed labels are propagated efficiently to regions that fall on trajectory gaps using multiscale appearance based region affinities.

- **Object connectedness constraints from foreground topology** as repulsive weights between point trajectories. Motion is insufficient for segmenting articulated bodies: articulated parts move distinctly while distinct objects may move similarly. We show object connectedness constraints can help video segmentation under motion ambiguities.

1.3.2 Two-granularity tracking for concurrent multi-object tracking and segmentation (Chapter 4)

Tracking objects in crowds is a joint detection and segmentation problem, due to the continuously changing spatial support of the objects, while interacting with each other. Motivated by this observation, we propose two-granularity tracking, a graph theoretic framework that classifies and clusters object detections and point trajectories. Two-granularity tracklets are comprised of detection responses, under good object visibility, and trajectory clusters, under object partial occlusions or deformations. Miss-detection gaps are "bridged" by trajectory clusters instead of utilizing motion smoothness assumptions or lowering the detection confidence threshold. Our goal is to improve tracking robustness, diminishing drifts caused by interpolation across miss detection gaps, rather than the segmentation itself. Point trajectories are independent of object categories. Thus, two-granularity tracking can easily be used for tracking any object category with the appropriate replacement of the object detector.

1.3.3 Graph steering for inferring classification-clustering in two - granularity graphs (Section 4.4)

Graph steering is our tool for inference in two-granularity representations. It is a clustering with bias algorithm that computes spectral clustering in a graph of motion/appearance based node affinities and detection-driven repulsions. The detection input may contain false alarms. Clustering in the steered graph is robust to wrong (false alarm) or spatially inaccurate detections, as analyzed in Section 4.4.3. Previous clustering with bias algorithms such as random walkers of Grady (2006) or pixel labeling works of Boykov et al. (2001); Komodakis et al. (2011) assume a label set known. Graph steering does not assume a known label set, but rather uses soft not-group constraints from pairs of detections, for discovering the grouping of graph nodes into objects. Graph nodes may be trajectories or image regions.

1.3.4 Motion for body pose detection (Chapters 5,6)

Motion is a strong cue for body pose detection. We have the following contributions:

- **Body joint temporal binding** for body pose inference under temporal correspondences. (Section 5.3.2). We bind body joint candidates on multiframe trajectories and estimate their unary scores from motion voting. Approximate inference on loopy space-time Markov Random Fields (MRFs) for human body pose often omits temporal edges for efficiency of inference in a simplified tree structured graph. Inference under temporal binding of the state candidates is less vulnerable to such MRF graph decompositions.

- **Pose specific motion segmentability** classifiers for re-ranking body joint candidate based on the agreement with the underlying video segmentation (Section 5.3.5). Previous approaches use optical flow boundaries as a feature for body part detection. They suffer from contradictions between poses under large or no motion, where flow boundaries are strong or nonexistent respectively. Pose specific motion segmentability can recover from such contradictions.

- **Articulated optical flow for tracking fast body part motion** (Section 6.4) We use detected articulated joints to impose kinematic constraints in optical flow estimation, for tracking arm kinematic chains through deformations. Standard optical flow estimates without kinematic constraints frequently drift to surroundings under fast body part motion.

1.4 Published work supporting this thesis

Part of our video segmentation work in Chapter 3 first appeared in Fragkiadaki and Shi (2011); Fragkiadaki et al. (2012a). The two-granularity tracking framework discussed in Chapter 4 was introduced in Fragkiadaki et al. (2012c). The articulated flow work and pose segmentation in Chapter 6 appears in Fragkiadaki et al. (2013). Work of Chapter 5 is under review. While not discussed in this thesis, the video segmentation framework of Chapter 3 found application in 3D cell segmentation presented in Fragkiadaki et al. (2012b).

Chapter 2

Preliminaries

Graph partitioning algorithms can be categorized according to their apriori information on the number and properties of a label set for the graph nodes. Apriori label information often allows to establish per node label probabilities, independent of inter-node relationships. In cases a label set is given apriori, graph partitioning is often referred to as node labeling. We call this a ”closed world“ partitioning framework.

In this chapter we review “open” and “closed” world graph partitioning or graph labeling formulations. We will use the term open world to refer to graph partitioning frameworks that do not assume apriori information on the number or properties of node labels, e.g. bottom-up segmentation methods of Cheng (1995); Shi and Malik (2000). We will use the term closed world to refer to graph partitioning frameworks that assume apriori information on the number and properties of different labels (or classes), e.g. in the form of seed nodes or training exemplars. Notably, such apriori information often allows to establish per node label probabilities, independent of inter-node relationships (unary label scores). The more closed the assumption about the world, the more emphasis the algorithms put on per node label probabilities as opposed to node cross-associations.

Section 2.1 reviews spectral clustering, a popular open world clustering framework with many variants in image and video segmentation. Section 2.2 reviews some closed or semi-closed world clustering formulations, with main stress on graph cuts of Boykov et al. (2001) and random walkers of Grady (2006).

2.1 Spectral Partitioning

Let $\mathbb{G}(\mathbb{V}, \mathbf{W})$ be a weighted graph over a node set $\mathbb{V}$. Let n denote the cardinality of $\mathbb{V}$. Matrix $\mathbf{W}$ is assumed symmetric and non-negative. Partitioning the node set $\mathbb{V}$ in K clusters is finding disjoint sets $\mathbb{V}_1 \cdots \mathbb{V}_K$ so that $\cup_{i=1}^{K} \mathbb{V}_k = \mathbb{V}$. Let Γ_{VF}^{K} denote the partitioning.

Normalized Cut Partitioning Criterion Given node sets $\mathbb{A}$, $\mathbb{B}$, we define $\mathrm{links}(\mathbb{A}, \mathbb{B})$ to be the total weighted connections from $\mathbb{A}$ to set $\mathbb{B}$:

$$\mathrm{links}(\mathbb{A}, \mathbb{B}) = \sum_{i \in \mathbb{A}, j \in \mathbb{B}} \mathbf{W}(i, j). \tag{2.1}$$

The degree of a set is defined as the total links of its nodes to all the nodes in $\mathbb{V}$:

$$\mathrm{degree}(\mathbb{A}) = \mathrm{links}(\mathbb{A}, \mathbb{V}). \tag{2.2}$$

The cut of a node set $\mathbb{A}$ is defined as the total links from $\mathbb{A}$ to its complement:

$$\mathrm{cut}(\mathbb{A}) = \mathrm{links}(\mathbb{A}, \mathbb{V}/\mathbb{A}). \tag{2.3}$$

The *normalized cut* of a node set $\mathbb{A}$ is defined as the fraction of its cut and its degree:

$$\mathrm{ncut}(\mathbb{A}) = \frac{\mathrm{cut}(\mathbb{A})}{\mathrm{degree}(\mathbb{A})}. \tag{2.4}$$

The criterion for K-way graph partitioning proposed in Shi and Malik (2000); Yu and Shi (2003) minimizes the normalized cuts of the clusters in the partitioning $\Gamma_{\mathbb{V}}^{K}$:

$$\mathrm{kncuts}(\Gamma_{\mathbb{V}}^{K}) = \sum_{k=1}^{K} \frac{\mathrm{cut}(\mathbb{V}_k)}{\mathrm{degree}(\mathbb{V}_k)}. \tag{2.5}$$

Since

$$\mathrm{links}(\mathbb{V}_k, \mathbb{V}/\mathbb{V}_k) = \mathrm{degree}(\mathbb{V}_k) - \mathrm{links}(\mathbb{V}_k, \mathbb{V}_k), \tag{2.6}$$

minimizing ncuts is equivalent to maximizing intra-cluster normalized associations:

$$\text{knassoc}(\Gamma_{\mathbb{V}}^{K}) = \sum_{k=1}^{K} \frac{\text{links}(\mathbb{V}_k, \mathbb{V}_k)}{\text{degree}(\mathbb{V}_k)}. \tag{2.7}$$

We consider the partition matrix $X = [X_1 \cdots X_K] \in \{0,1\}^{n \times K}$. Let $\mathbf{D_W} \in \mathbb{R}^{n \times n}$ denote the diagonal degree matrix of the affinity matrix $\mathbf{W}$:

$$\mathbf{D_W} = \text{Diag}(\mathbf{W}\mathbf{1}_n). \tag{2.8}$$

Then the K-way normalized association criterion can be re-written as:

$$\text{knassoc}(X) = \sum_{k=1}^{K} \frac{X_k^T \mathbf{W} X_k}{X_k^T \mathbf{D_W} X_k} \tag{2.9}$$

and results in the following maximization problem:

$$\begin{array}{ll} \max\limits_{X}. & \epsilon(X) = \sum_{k=1}^{K} \frac{X_k^T \mathbf{W} X_k}{X_k^T \mathbf{D_W} X_k} \\ \text{subject to} & X \in \{0,1\}^{n \times K}, \quad \sum_{k=1}^{K} X_K = \mathbf{1}_n. \end{array} \tag{2.10}$$

The problem above has been shown in Shi and Malik (2000) to be NP-complete even for $K = 2$. Below we will show the spectral relaxation that is typically used to obtain a near global optimum.

Spectral relaxation We relax the problem in Eq. 2.10 by ignoring its constraints. We do the following change of variables: we divide each indicator vector X_k with the square root of the degree $(X_K^T \mathbf{D_W} X_k)$ of the corresponding cluster:

$$Z_k = X_k (X_k^T \mathbf{D_W} X_k)^{-\frac{1}{2}}, \quad Z = X (X^T \mathbf{D_W} X)^{-\frac{1}{2}}. \tag{2.11}$$

Then, it is easy to see that

$$Z^T \mathbf{D_W} Z = (X^T \mathbf{D_W} X)^{-\frac{1}{2}} X^T \mathbf{D_W} X (X^T \mathbf{D_W} X)^{-\frac{1}{2}} = I_K, \tag{2.12}$$

and we obtain the following maximization problem:

$$\begin{array}{ll} \max\limits_{Z}. & \epsilon(Z) = \text{tr}(Z^T \mathbf{W} Z) \\ \text{subject to} & Z^T \mathbf{D_W} Z = I_K. \end{array} \tag{2.13}$$

If we do a last change of variables:

$$Y = \mathbf{D}_{\mathbf{W}}^{\frac{1}{2}} Z, \quad Z = \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} Y \tag{2.14}$$

we obtain:

$$\begin{aligned} \max_{Y}. \qquad & \epsilon(Y) = \operatorname{tr}(Y^T \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} Y) \\ \text{subject to} \qquad & Y^T Y = I_K. \end{aligned} \tag{2.15}$$

Above we recognize a Rayleigh quotient optimization problem. We consider the Lagrangian relaxation:

$$\operatorname{tr}(Y^T \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} Y) - \lambda(Y^T Y - I_K). \tag{2.16}$$

We differentiate with respect to Y and obtain:

$$\mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} Y + (\mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}})^T X - 2\lambda Y = 0 \Leftrightarrow \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} Y = \lambda Y, \tag{2.17}$$

which shows that columns of Y are eigenvectors of the symmetric normalized affinity matrix $\mathbf{W}_{nsym} = \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}}$. We obtain the Y that maximizes $\operatorname{tr}(Y^T \mathbf{W}_{nsym} Y)$ by the top K eigenvectors $\bar{V}$ of $\mathbf{W}_{nsym}$. We convert back to Z using Eq. 2.14 and obtain $V = \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} \bar{V}$. Given that each node has a positive degree and matrix $\mathbf{D}_{\mathbf{W}}$ is invertible, (x, λ) is an eigenvector, eigenvalue pair of $\mathbf{W}_{nsym}$ if and only if $(\mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} x, \lambda)$ is an eigenvector and eigenvalue pair of the random walk matrix $\mathbf{D}_{\mathbf{W}}^{-1} \mathbf{W}$. Thus, we can obtain Z directly by computing the rank K eigen decomposition of $\mathbf{D}_{\mathbf{W}}^{-1} \mathbf{W}$, as noted in Meila and Shi (2001).

Discretizing the spectral embedding We obtain the node partitioning by discretizing the solution of the spectral relaxation. There are two popular ways of discretizing the continuous eigenvectors V:

1. K-means on the node embedding coordinates, proposed in Ng et al. (2001).

2. eigenvector rotation, proposed in Yu and Shi (2003). The authors exploit the fact that the maximizer of Eq. 2.13 is not unique, but rather is arbitrary up to an orthogonal transformation $R, R^T R = I$. Their algorithm alternates between finding a

discrete solution by assigning each node to the cluster is has the largest value in the corresponding row of the continuous matrix V, and finding a rotation matrix R that brings closer V to the discretized solution.

Both discretization algorithms often break coherent regions. Regularizations are needed to find a stable discrete partitioning. In Section 3.3.1 we present our discretization algorithm where discontinuities of embedding affinities are used to detect and fix artificial fragmentations.

2.2 Clustering with Bias

Many works have addressed clustering with some type of *bias*, in the form of seed nodes in Duchenne et al. (2008); Grady (2006); Joachims (2003), multi-node relationships in Maire et al. (2011); Yu et al. (2002), not-group (repulsion) constraints in Yu and Shi (2001), training exemplars for different labels in Munoz et al. (2010) or any general per node label bias in Boykov et al. (2001) and general Markov Random Fields. The stronger the per node label bias (or label potentials) the less critical are the cross-node associations.

We are going to take a closer look into random walkers of Grady (2006) as a representative of seeded graph partitioning methods. Furthermore, we will use the random walker framework for mapping trajectory clusters to image regions for video segmentation in Section 3.3.2.

Seeded segmentation with random walkers Let $\mathbb{G}(\mathbb{V}, \mathbf{W})$ be a weighted graph over a node set $\mathbb{V}$ of cardinality n, same as in the previous section. Let $\mathcal{L} = \{1 \cdots K\}$ denote a label set. Let the label be known in a set of nodes $\mathbb{V}_M$ (marked or *seeds*). We want to find the node labels in $\mathbb{V}_U = \mathbb{V}/\mathbb{V}_M$. For now we will assume $K = 2$. For ease of notation let $\mathbb{V}_F$, $\mathbb{V}_B$ denote the corresponding two seed node sets (F, B stand for foreground and background).

It has been established in Kakutani (1945) that the probability a random walker initialized at a graph node first reaches a seed node exactly equals the solution to the Dirichlet problem with boundary conditions at the location of the seed nodes and the seed node in question fixed to unity while the others set to 0. Random walkers introduced in Grady (2006) consider the combinatorial Dirichlet problem on an arbitrary graph and compute analytically the probability x_a^k that a random walker starting at node v_a first reaches a seed node with label k.

Let $x \in \mathbb{R}^{n \times 1}$ denote node potentials in our graph $\mathbb{G}$. Then a combinatorial formulation of the Dirichlet integral, using the discrete Laplace operator, is:

$$D(x) = \frac{1}{2} \sum_{i,j} \mathbf{W}_{ij} (x_i - x_j)^2. \tag{2.18}$$

Observe that $D(x) = \frac{1}{2} x^T \mathbf{L} x$, where $\mathbf{L} = \mathbf{D_W} - \mathbf{W}$ is the unnormalized Laplacian of $\mathbf{W}$. We have the following optimization problem:

$$\begin{aligned} \min_{x}. \qquad & D(x) = \tfrac{1}{2} x^T \mathbf{L} x \\ \text{subject to} \qquad & x_B = 0, \quad x_F = 1. \end{aligned} \tag{2.19}$$

Assuming without loss of generality that nodes are ordered into marked (seeds) and unmarked (non seeds), the previous equation can be decomposed as:

$$D(x_U) = \frac{1}{2} \begin{bmatrix} x_M^T & x_U^T \end{bmatrix} \begin{bmatrix} \mathbf{L}_M & \mathbf{L}_{MU} \\ \mathbf{L}_{MU}^T & \mathbf{L}_U \end{bmatrix} \begin{bmatrix} x_M \\ x_U \end{bmatrix} = \frac{1}{2} \left(x_M^T \mathbf{L}_M x_M + 2 x_U^T \mathbf{L}_{MU}^T x_M + x_U^T \mathbf{L}_U x_U \right), \tag{2.20}$$

where $x_M = \begin{bmatrix} x_F \\ x_B \end{bmatrix}$, x_U correspond to potentials of marked and unmarked nodes respectively. Since $\mathbf{L}$ is positive semi-definite, critical points of Eq. 2.20 will be minima. Differentiating $D(x_U)$ with respect to x_U yields:

$$\mathbf{L}_U x_U = -\mathbf{L}_{MU}^T x_M. \tag{2.21}$$

Solving the system of linear equations results in potentials for each unlabeled node. We solve Eq. 2.21 for each label $l \in \mathcal{L}$ and assign each unmarked node to the label for which it has the highest potential.

Note that random walkers use the unnormalized graph Laplacian $\mathbf{L}$ while spectral clustering uses the spectrum of the normalized Laplacian $\mathbf{L}_{nsym} = \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}}$ (it is easy to show that the top K eigenvectors of the symmetric normalized affinity matrix $\mathbf{W}_{nsym}$ correspond to the bottom K eigenvectors of $\mathbf{L}_{nsym}$). Under seed guidance, normalization with respect to cluster size is not needed.

Min-cuts Another popular graph partitioning with bias criteria is min-cut. Under the same graph setup, min-cut minimizes the following objective:

$$\begin{aligned} \min_{x}. \quad & \tfrac{1}{2}\textstyle\sum_{i,j} \mathbf{W}_{ij}|x_i - x_j| \\ \text{subject to} \quad & x_B = 0, \quad x_F = 1. \end{aligned} \tag{2.22}$$

It can be shown that the dual of the problem in Eq. 2.22 is a max-flow problem (Kleinberg and Tardos (2005)). As such, min-cut can be computed efficiently and has found wide applicability in any image or video labeling problem in which reliable per node class probabilities can be obtained, e.g., in Boykov and Funka-Lea (2006); Xiao and Shah (2005). Note that such per node probabilities (unary label scores) can be incorporated in Eq. 2.22 by adding edges from each node to phantom foreground or background nodes. Work of Boykov et al. (2001) extends the binary labeling formulation to multiple labels.

Chapter 3

Spatio-temporal Perceptual Organization

We see in order to move, we move in order to see.

— William Gibson *The Perception of the Visual World*

Perceptual organization is the process of structuring visual information into *coherent* units. Coherence may be measured in terms of uniformity in color, texture, motion, or contour curvature continuity. Gestalts philosophy advocates a set of principles underlying perceptual organization in animals' visual perception, such as the principle of proximity, similarity, closure, good continuation, figure-ground, as described in Wertheimer (1938). Gestaltic principles have inspired numerous computer vision algorithms that aim at segmenting images and videos into coherent groups and delineating their boundaries, without the use of any model matching.

In the era of model-driven image parsing, the question is what perceptual organization has to offer to visual recognition. In contrast to model-driven approaches, perceptual organization methods can process pixels/voxels occupied by *rare or unseen* concepts. The visual distribution is known to have few re-occurring configurations (frequent objects under canonical pose or viewpoint) and many rare ones, it has long tails as noted in Hoiem et al. (2012). Thus, there will always be cases where model-driven visual parsing is unreliable.

Spatio-temporal percepts (segments) are useful for visual recognition because they allow opportunistic parsing in time. Thanks to their temporal extent, matching them against object models can be done sparsely rather than densely in time, in the frames when the object pose or viewpoint is easy to recognize. This parsing then propagates along the temporal dimension of the percepts. Opportunistic parsing in time is a big advantage we see in spatio-temporal perceptual organization over static image segmentation.

In this chapter, we present a spatio-temporal organization framework that partitions a video sequence into moving objects and the world scene, exploiting long range motion and appearance coherence. In the following chapters, we will combine the video representation developed here with information from object detectors, in order to track objects and their pose through occlusions and wide deformations.

3.1 Introduction

Motion as a grouping cue for perceptual organization has long occupied scientists and philosophers of animal vision and perception. "We see in order to move, we move in order to see", writes Gibson in his work on motion perception Gibson (1951). Grouping by motion similarity is expressed in the Gestaltic principle of "common fate" in Johansson (1973). Motion segregation as a perceptual cue, aside of motion similarity, is explored in the psychophysics experiments of Nothdurft (1992): bar-link visual concepts are perceived as a group, when viewed in background of similar concepts that dier from them in motion or orientation, despite their low intra-coherence.

Numerous computer vision algorithms have been proposed that exploit motion similarity and motion segregation to segment a video and/or identify occlusion boundaries, without matching to object models. Most of them critically depend on the accuracy of pixels' apparent 2D displacements from frame to frame. Dense pixel displacement fields are referred as *optical flow* in the computer vision literature, a term borrowed from Gibson (1951). In fact, progress in optical flow estimation may have been the most critical determinant behind high accuracy video segmentation and occlusion boundary detection, as reported in Garg et al. (2013); Sundberg et al. (2011).

Optical flow estimation has a large impact in all aspects of video analysis and has always been an active area of Computer Vision research. Most motion estimation works assume constancy of some pixels' properties under motion, such as image brightness in Lucas and Kanade (1981), image gradient in Brox et al. (2004), and, more recently, aggregated gradient histograms (HOG descriptors) in Brox and Malik (2010a); Xu et al. (2012b). The seminal work of Lucas and Kanade (1981) assumes that pixel motion represented by a translation or affine transformation is constant within a small image window. Their method expresses the brightness of the displaced patch as a function of the pixel displacement using first order Taylor expansion, under small displacement assumptions. Window-based motion estimation methods are referred to as 'local' because they do not couple motion estimates across different image windows. The seminal work of Horn and Schunck Horn and Schunck (1981) introduced a variational model for optical flow estimation which minimizes pixel brightness differences (linearized with respect to the pixel displacement as in Lucas and Kanade (1981)) and a quadratic penalizer of displacement gradient magnitude, enforcing smoothness of the estimated motion field. The original Horn and Schunck model has been modified and extended in two main directions: 1) Tackling motion discontinuities and occlusions by employing non-quadratic penalizers in the smoothness and data terms Black and Anandan (1996); Memin and Perez (1998). 2) Relaxing small displacement assumptions by employing either coarse-to-fine warping schemes as in Brox et al. (2004) (similar to those used in Lucas and Kanade (1981)), or discrete-continuous optimization as in the works of Lempitsky et al. (2008); Xu et al. (2012b).

Many video segmentation approaches in order to take advantage of longer time horizon use point trajectories instead of per frame flow fields. Given a video sequence, point trajectories are computed on pixels with reliable frame-to-frame correspondence. Early work of Shi and Tomasi (1994) computes trajectories on corner-like features and employs long range affine matching to determine drift and trajectory termination. Recent work of Sundaram et al. (2010) computes trajectories on a dense pixel grid and employs a per frame forward-backward consistency check of optical flow estimates to determine trajectory termination. Such check fails under occlusion or dis-occlusion of pixels as well as at very low textured image regions, where correspondence is ambiguous, as shown in Figure 3.7. Experiments of Sundaram et al. (2010) quantify that trajectories computed from linking state-of-the-art optical flow fields of Brox and Malik (2010a) are more accurate than the long standing KLT trajectories of Lucas and Kanade (1981); Shi and Tomasi (1994), while being denser.

Multi-body factorization methods cluster trajectories by reasoning about relationships between the corresponding motion subspaces Costeira and Kanade (1995); Rao et al. (2008); Yan and Pollefeys (2006). Each trajectory cluster ideally corresponds to an object hypothesis in space and time. These works extend the factorization framework introduced in Tomasi and Kanade (1991), under low rank assumptions on per frame 3D shape deformations in Costeira and Kanade (1995); Yan and Pollefeys (2006) or multi-frame trajectory motion in Akhter et al. (2011). Most factorization methods require trajectories to have the same (large enough) length. This is often an infeasible requirement under articulated motion where frequent self occlusions and deformations of the objects cause frequent trajectory terminations. Works of Brostow and Cipolla (2006); Brox and Malik (2010b); Fradet et al. (2009) and ours Fragkiadaki and Shi (2011) cluster trajectories directly from similarities of their 2D motion profiles, without modeling the camera projection process or attempt 3D reconstruction. These works do not require trajectories to have the same length. In fact, trajectory spectral clustering computed from 2D motion information has shown to outperform factorization methods in Brox and Malik (2010b).

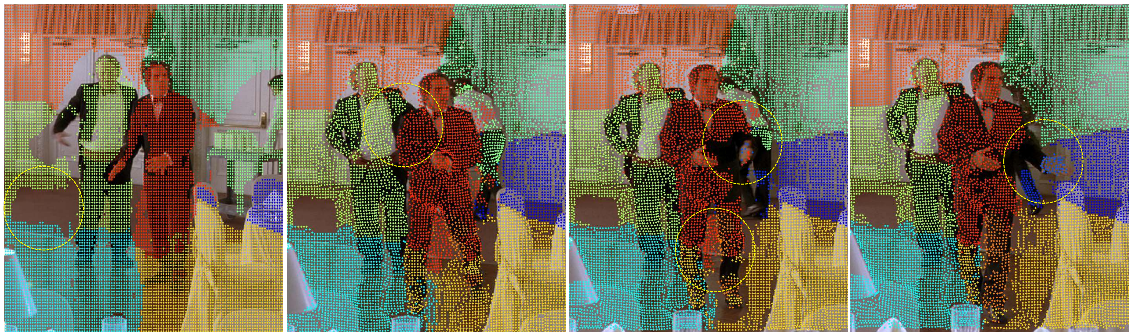


Figure 3.1: Gains and ambiguities in motion estimation. Column 1: Trajectory sparsity in untextured regions due to unreliable frame to frame correspondence. Column 2 : Flow estimates do not drift under accidental appearance similarity across objects. The resulting trajectory partitioning digs out the faint boundaries boundary between the two men wearing the same black suit and bridges fake shirt-suit contour. Column 3: Optical flow estimates bleed to the untextured background between the legs of the actor. Out of plane rotations of limbs cause trajectories to terminate suddenly. Column 4: Trajectory sparsity in one dimensional limbs due to aperture problems.

The clustering is obtained by discretizing the top eigenvectors of a normalized trajectory affinity matrix; affinities reflect motion similarity between the corresponding point trajectories. In essence, trajectory spectral partitioning extends the per frame motion profile partitioning work of Shi and Malik (1998) to large temporal horizon, crucial for dealing with per frame motion ambiguities.

Despite the progress in optical flow and trajectory computation, there remain intrinsic ambiguities in motion estimation of low textured image regions and articulated structures. Untextured backgrounds cause optical flow of the foreground region to "bleed" across, as shown in Figure 3.1 and described in Thompson (1998). Wide deformations, self occlusions, out-of-plane rotations of human body limbs are hard to track with bottom-up motion estimation methods, due to aperture problems.

To deal with limitations of motion estimation, many segmentation methods employ both motion and appearance coherence for spatio-temporal grouping. Works of Fragkiadaki et al. (2012a); Lezama et al. (2011); Ochs and Brox (2011) combine point trajectories with image regions to produce a pixel-wise video segmentation from sparse trajectory clusters. Work of Xu et al. (2012a) computes a hierarchical voxel segmentation based on pixel proximity and color similarity, bypassing optical flow computation. While fast, it fails under color similarity across different objects. This is precisely the strength of motion based frameworks: they can separate objects with similar appearance and distinct motion, digging out faint contours while bridging fake, interior ones, e.g., due to colorful clothing. This is depicted in Figure 3.1 column 2.

Given a video sequence that contains object or camera motion, we want to compute a hierarchical segmentation of the objects and the background. We exploit information in dense point trajectories (of large temporal and small spatial support) and static image regions (of large spatial and small temporal support) in textured and untextured areas of the video selectively. We establish a point trajectory adjacency graph whose edge weights convey boundary probability, the probability of two adjacent trajectories belonging to different objects. We compute link boundary probabilities using the spectral embedding of trajectory motion affinities. Thresholding link boundary probabilities at different cutoff values provides trajectory clusterings of different granularities. Given a trajectory clustering, we map trajectory clusters to image regions using random walkers on a multiscale space-time region graph. Such mapping effectively recovers from optical flow bleeding effects and trajectory sparsity under low image texturedness.

We present quantitative and qualitative results of our method that outperform previous approaches on established segmentation as well as motion boundary detection datasets. Our code is available at `www.seas.upenn.edu/∼katef/videoseg.html`. We explore generality and limitations of our approach with varying spatial resolution, object deformation, articulation and scale.

3.2 Related work

Numerous works have addressed simultaneous segmentation and motion estimation in layer video representations. Works of Ayer and Sawhney (1995); Jepson et al. (2002); Pawan Kumar et al. (2008); Soatto (2005); Wang and Adelson (1994); Xiao and Shah (2005) consider affine layer motion models, which assume that image layers are projections of planar 3D patches. Parametric models have been shown to be too restrictive to capture the often diverse layer motion, as discussed in Sun et al. (2010). Work of Weiss (1997) imposes a smoothness constraint on non parametric layer motion fields and work of Sun et al. (2010) assumes affine layer motions with a robust penaliser of deviations from it. Inferred segmentation boundaries on motion discontinuities block diffusion of the smoothness motion coupling in Xiao et al. (2006). While earlier works mostly consider pair of frames, works of Sun et al. (2012); Xiao and Shah (2005) infer motion and segmentation across multiple frames simultaneously, exploiting structure consistency over time to disambiguate layer depth ordering.

Numerous works have attempted to extend the notion of segments or superpixels from the static image domain to videos. Approaches that rely on pixel appearance similarity for spatio-temporal grouping, such as Brendel and Todorovic (2009); Vazquez-Reina et al. (2010); Wang et al. (2011); Xu et al. (2012a), aim at temporal consistency of superpixel labels, whether they capture moving or stationary objects. Approaches that rely on motion similarity, such as Shi and Malik (1998), aim at segmenting moving objects from the static world scene, and neglect groups of non distinct apparent motion. Approaches of Gao et al. (2008); Mahadevan and Vasconcelos (2010); Rahtu et al. (2010a) compute motion saliency via a center-surround motion dissimilarity computation in a sliding window fashion across multiple scales. They focus on fast segmentation of the moving ensemble from the background, without trying to dis-entangle adjacent moving objects. Early work of Shi and Malik (1998) segments a video frame into moving objects by spectral clustering of pixel motion profiles, i.e., probability distributions over possible pixel displacements. It is dependent on choosing the pair of frames with large motion difference between the objects. Work of Grundmann et al. (2010) builds a hierarchical super-voxel graph, using dense optical flow and color similarity for establishing super-voxel affinities. Work of Xu et al. (2012a) bypasses optical flow computation, to produce a streaming hierarchical video segmentation by reasoning about pixel intra-frame and cross-frame color similarities. While fast, it fails under accidental color similarity across different objects. Segmentation ambiguities from cross-object appearance similarities can be resolved with long range trajectory motion, as shown in Figure 3.1.

In order to take advantage of longer time horizon many approaches use point trajectories. Multi-body factorization methods of Costeira and Kanade (1995); Elhamifar and Vidal (2009); Rao et al. (2008); Yan and Pollefeys (2006) segment rigid object motion relying on properties of an affine camera model. These works extend the low rank constraint on a trajectory matrix proposed in Tomasi and Kanade (1991), under assumptions about 3D object deformation and camera projection. Recent work of Akhter et al. (2011) uses trajectory rather than shape decomposition of the 2D trajectory matrix and can reconstruct point trajectories without the need to pre-infer their segmentation. It is the first work to reconstruct ensemble of articulated objects. Reconstruction allows segmentation to take place in the estimated 3D point cloud rather than in 2D trajectories. Factorization methods thought generally require all trajectories to have the same length, and quality of reconstruction depends on long trajectories that track the object across rotations. However, tracking is a hard problem on its own and factorization methods often assume the trajectory input to be given, which makes them impractical. Works of Elhamifar and Vidal (2009); Rao et al. (2008) have tried to recover from the requirement of trajectory length equality to a certain extent. However, deformable or articulated motion still poses challenges to the factorization literature. Authors of Yaser Sheikh and Kanade (2009) obtain a figure ground classification of trajectories under a projective camera model by estimating the basis for trajectories of the static rigid world scene using RANSAC.

Works of Brostow and Cipolla (2006); Brox and Malik (2010b); Fradet et al. (2009); P.Ochs and T.Brox (2012) and ours Fragkiadaki and Shi (2011) cluster trajectories directly from similarities of their 2D motion profiles, without modeling the camera projection process. While most approaches employ pairwise trajectory similarities, work of P.Ochs and T.Brox (2012) considers trajectory hyper-graphs with affinities on trajectory triplets using in plane rotation models. The authors back-project to pairwise trajectory affinities for spectral clustering.

Numerous works have addressed occlusion boundary detection, manifested as motion boundaries under object motion or camera motion, due to parallax. Works of Ayvaci and Soatto (2012); Ravichandran et al. (2012) use motion discontinuities and pixel occlusions, output of the occlusion-aware optical flow of Ayvaci et al. (2012) for video segmentation. Work of Derpanis and Wildes (2010) uses spatio-temporal filter responses for texture and structure boundary detection. Work of He and Yuille (2010) assumes a rigid video scene and scaled-orthographic projection, and estimates camera projection matrices and pixel pseudo-depths, so that the difference of their camera projections in consecutive frames matches the optical flow estimates. The estimated pseudo-depths and appearance cues are used to classify superpixel boundaries as occlusion or not. The motion boundary detector of Stein et al. (2007) estimates translational motion for pair of regions adjacent to a superpixel boundary, down-weighting the contribution of pixels close to the boundary to avoid contamination of the estimated motion models. The local boundary strength measurements from motion disagreements are incorporated into an MRF for inferring globally consistency boundaries, which are further used for object segmentation in Stein et al..

Recently, work of Sundberg et al. (2011) showed that aggregating state-of-the-art optical flow estimates of Brox et al. (2004) in image regions and comparing the fitted affine estimates along the shared region boundary works better than estimating flow and regions together: information from the shared region contour is too important to be neglected in the motion estimation process, despite lack of information regarding local segmentation and possibility of bleeding. Interestingly, increasing spatial resolution alleviates from untexturedness of low resolution videos, improving optical flow estimates and diminishing their bleeding effects. However, when the regions are too small for the computed cues to be reliable, spurious boundaries are detected. Also, body deformations may give rise to many interior boundaries, not corresponding to objects.

3.3 Motion segmentation from trajectories and regions

Point trajectories We define a point trajectory tr_i to be a sequence of video pixels that correspond to 2D projections of the same 3D physical point in time:

$$\mathrm{tr}_i = \{(x_i^t, y_i^t), \quad t \in T_i\}, \tag{3.1}$$

where T_i is the frame span of tr_i.

We compute point trajectories by linking optical flow fields, as proposed in Sundaram et al. (2010): a trajectory is produced by following the optical flow vectors. A trajectory terminates at the frame when forward-backward flow consistency check fails, indicating ambiguity in correspondence. This is usually the case under occlusion or dis-occlusions of the reference pixel, as well as under low image texturedness. Given a video sequence I, we consider the trajectory set $\mathcal{T} = \{\mathrm{tr}_i, \ i = 1 \cdots n_T\}$, where n_T is the number of trajectories.

3.3.1 Trajectory spectral discontinuities

We seek a motion discontinuity measure that given a pair of spatially neighboring trajectories reflects the probability they belong to different objects. We measure trajectory spatial neighborness using Delaunay triangulations on trajectory points of each frame, as shown in Figure 3.2. By definition of the Delaunay triangulation, three trajectory points are connected if no other point is contained in the circumcircle of their triangle. Each triangulation is a planar graph on trajectory points, with Delaunay edges incident to spatially neighboring trajectory points (we denote spatial neighborness with symbol $\sim$). We consider the trajectory adjacency graph $\mathrm{G}(\mathcal{T}, E^D)$ that aggregates per frame triangulations in time, from trajectory points to trajectories:

$$(i,j) \in E^D \quad \text{iff} \quad \exists\, t \quad \text{s. t.} \quad (x_i^t, y_i^t) \sim (x_j^t, y_j^t), \quad i,j = 1 \cdots n_T \tag{3.2}$$

Two trajectories are adjacent in G if they are adjacent in any frame during their time overlap.

We seek a discontinuity measure $\mathbf{d}$ that reflects the probability that two trajectories adjacent to an edge in G belong to different objects:

$$\text{Trajectory Spectral Discontinuities} \quad \mathbf{d} : E^D \rightarrow [0,1]. \tag{3.3}$$

Discontinuities depend on trajectory motion (dis)similarities, which we describe right below.

Trajectory motion affinities We compute trajectory motion affinities $\mathbf{A}_T \in [0,1]^{n_T \times n_T}$, where $\mathbf{A}_T(i,j)$ measures motion similarity between trajectories tr_i and tr_j. Our motion affinities are a function of the maximum velocity difference between the corresponding trajectories, as proposed in Brox and Malik (2010b):

$$\mathbf{A}_T(i,j) = \exp\left(-\frac{d_{ij}\Delta u_{ij}}{\sigma}\right) \cdot \delta(T_i \cap T_j \neq \emptyset), \tag{3.4}$$

where δ is the Dirac function being one if its argument is true and 0 otherwise, d_{ij} is the maximum Euclidean distance between tr_i and tr_j. Δu_{ij} is the largest velocity difference between tr_i and tr_j during their time overlap:

$$\Delta u_{ij} = \max_{t \in T_i \cap T_j} \frac{|\vec{u}_t^i - \vec{u}_t^j|_2^2}{t_f}, \tag{3.5}$$

where $\vec{u}_t^i = (x_{t+t_f}^i - x_t^i, y_{t+t_f}^i - y_t^i)$ is the velocity of tr_i at time t. The largest velocity difference Δu between two trajectories in Eq. 3.4 is the most informative measurement regarding their association. It avoids periods of accidental motion similarity between objects, e.g., when objects are stationary with respect to each other. We use $\sigma = 100$ and $t_f = 5$. If trajectories are shorter than 5 frames, we use $t_f = \min(T_i, T_j)$.

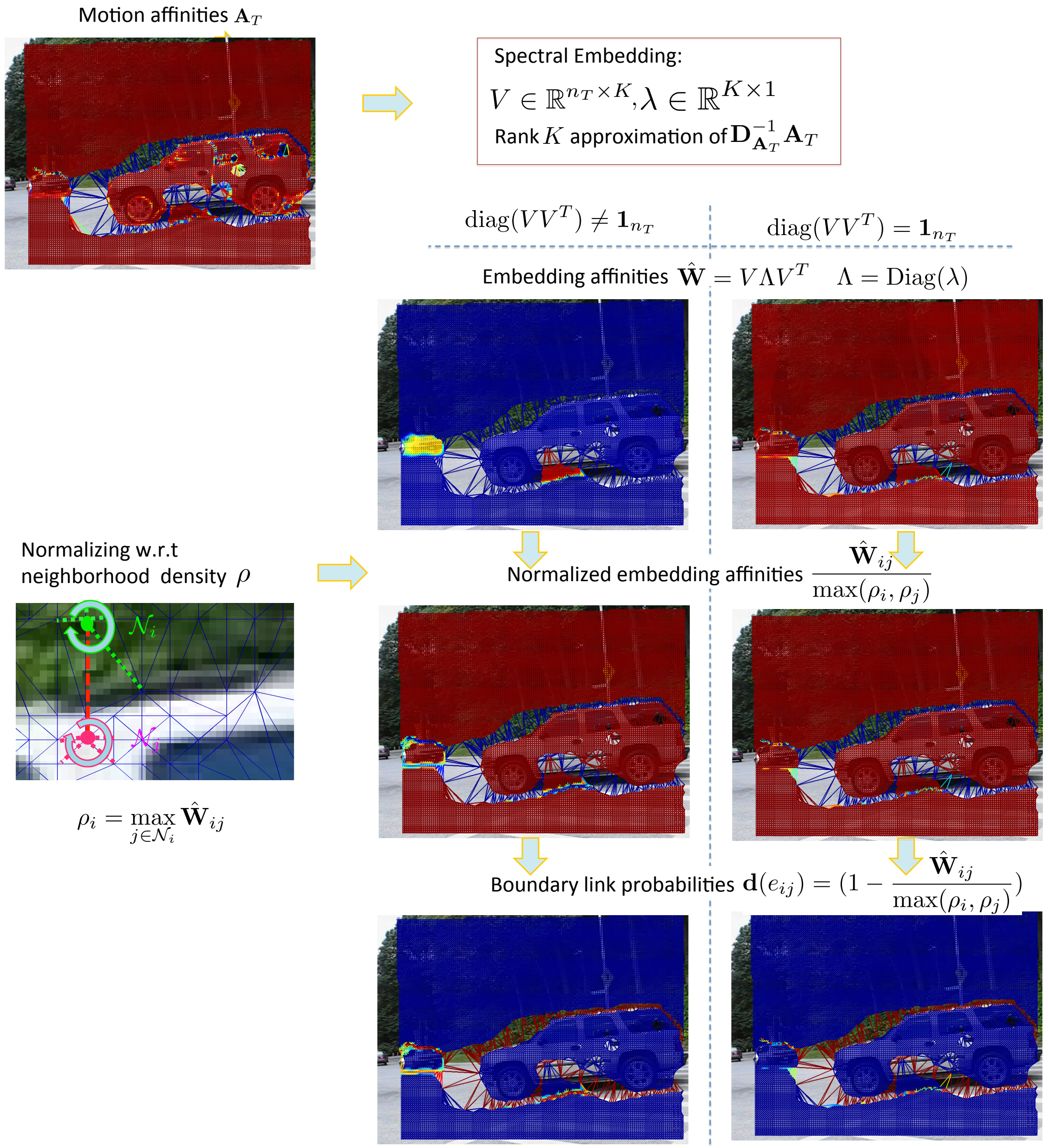

Motion affinities $\mathbf{A}_T$
Spectral Embedding:
$V \in \mathbb{R}^{n_T \times K}, \lambda \in \mathbb{R}^{K \times 1}$
Rank K approximation of $\mathbf{D}^{-1}_{\mathbf{A}_T}\mathbf{A}_T$
$\mathrm{diag}(VV^T) \neq \mathbf{1}_{n_T}$
$\mathrm{diag}(VV^T) = \mathbf{1}_{n_T}$
Embedding affinities $\hat{\mathbf{W}} = V \Lambda V^T \quad \Lambda = \mathrm{Diag}(\lambda)$
Normalizing w.r.t neighborhood density ρ
$\mathcal{N}_i$
$\mathcal{N}_j$
$\rho_i = \max_{j \in \mathcal{N}_i} \hat{\mathbf{W}}_{ij}$
Normalized embedding affinities $\frac{\hat{\mathbf{W}}_{ij}}{\max(\rho_i, \rho_j)}$
Boundary link probabilities $\mathbf{d}(e_{ij}) = (1 - \frac{\hat{\mathbf{W}}_{ij}}{\max(\rho_i, \rho_j)})$

Figure 3.2: Trajectory spectral discontinuities. We show trajectory motion affinities $\mathbf{A}_T$, embedding affinities $\hat{\mathbf{W}}$ and discontinuities $\mathbf{d}$ on per frame Delaunay edges. Discontinuities lies in $[0, 1]$ and are calibrated against scales and kinematic nature of the various objects in the scene. We have links with large discontinuities between trajectories capturing different objects.

Trajectory spectral embedding Motion affinities in $\mathbf{A}_T$ are not calibrated across different objects in the video scene. Some objects have more distinct apparent motion with respect to their surroundings than others, e.g., in Figure 3.2 the small car appears to move slower than the larger one, partly because it is further away from the camera. Consequently, the motion affinities on edges E^D spanning the larger car and its background are smaller than the affinities on edges spanning the small car and its background. Furthermore, deformable objects may have lower interior motion affinities than rigid moving ones, due to the non-uniform deformation field. Summarizing, the varying kinematic nature, scale and deformability of the objects in the video scene cause motion affinities $\mathbf{A}_T$ to have very different values on the different trajectory boundary edges, which makes them unsuitable for indicating probability of object boundaries.

The observation that different parts of a visual scene have asymmetric, hard to compare affinity profiles, with very tight or loose closest neighbor affinity values, is in fact quite old. The seminal work of Shi and Malik (2000) proposes normalization of the affinity matrix as a simple way of calibrating pixel affinities for image segmentation. We perform the same (row) normalization for our trajectory affinity matrix $\mathbf{A}_T$ and obtain:

$$\mathbf{W} = \mathbf{D}_{\mathbf{A}_T}^{-1} \mathbf{A}_T, \tag{3.6}$$

where $\mathbf{D}_{\mathbf{A}_T} = \mathrm{Diag}(\mathbf{A}_T \mathbf{1}_{n_T})$ is the diagonal degree matrix of $\mathbf{A}_T$. Let $V \in \mathbb{R}^{n_T \times K}$, $\lambda \in [0,1]^K \times 1$ denote the top K eigenvectors and eigenvalues of $\mathbf{W}$. We obtain the K rank approximation of $\mathbf{W}$ by:

$$\hat{\mathbf{W}} = V \Lambda V^T, \tag{3.7}$$

which is a smooth version of $\mathbf{W}$. Here $\Lambda = \mathrm{Diag}(\lambda)$. We have shown in Chapter 2 that node partitioning by discretizing the top K eigenvectors of a normalized affinity matrix minimizes approximately the K-way normalized cut criterion of Shi and Malik (2000). We select K by thresholding the eigenvalues λ at spectral threshold eig.

Rows of V represent trajectory embedding coordinates and can have different norms. It has been shown in Gallier (2013) that columns of V correspond to cluster indicator vectors which in the most general form take the form $\{0, \frac{1}{\sqrt{\alpha_k}}\}$, where α_k the degree of the kth cluster. This means that small clusters are embedded further from the origin than larger ones. The resulting intra affinities $\hat{\mathbf{W}}$ are high for small degree clusters and low for large degree clusters, as shown in Figure 3.2 2nd row left: the two small clusters have higher $\hat{\mathbf{W}}$ than the clusters on the large car or the background. Previous works normalize the rows of V so that all trajectory embedded coordinates to have unit norm, as shown in Figure 3.2 2nd row right. Normalization with respect to neighborhood density in Eq. 3.8 does not require rows of V to have equal norm.

We are now ready to introduce our trajectory spectral discontinuities, measuring sudden drops or peaks of $\hat{\mathbf{W}}$ between spatially adjacent trajectories. Let $\mathcal{N}_i$ denote the set of trajectories being spatial neighbors of tr_i, $\mathcal{N}_i = \{j, (i,j) \in E^D\}$. For each trajectory tr_i, we define the density ρ_i to be the maximum embedding affinity to its Delaunay neighbors $\mathcal{N}_i$. Then the spectral discontinuities take the form:

$$\rho_i = \max_{j \in \mathcal{N}_i} \hat{\mathbf{W}}_{ij}, \tag{3.8}$$

$$\mathbf{d}(e_{ij}) = (1 - \frac{\hat{\mathbf{W}}_{ij}}{\max(\rho_i, \rho_j)}),\ \forall\, e_{ij} \in E^D. \tag{3.9}$$

Discontinuities $\mathbf{d}$ lie on motion boundaries and provide a strong indication of object boundaries, as shown in Figure 3.2 4th row. We show $\mathbf{d}$ with and without norm normalization of rows in V. By skipping norm normalization resulting discontinuities can be more informative, as shown in Figure 3.2 3rd row left: the small car is more clearly delineated from its surroundings. This is the case because the norm of the embedding coordinates carries information with regard to cluster degree, which is useful for clustering.

Multiscale trajectory clustering We want to compute a multiscale trajectory partitioning that provides trajectory clusters in different levels of granularity, depending on a cutoff probability of boundary threshold. Currently, the standard way of obtaining a multiscale partitioning is by discretizing varying number of spectral eigenvectors K. The larger the

K, the finer the partitioning. However, the two typical discretization methods used in the literature, namely K-means of Ng et al. (2001) and eigenvector rotation of Yu and Shi (2003), often break coherent regions into chunks, as noted it Arbelaez et al. (2009). Resulting cluster boundaries are often not correlated to true object boundaries, as shown in Figure 3.4 1st row.

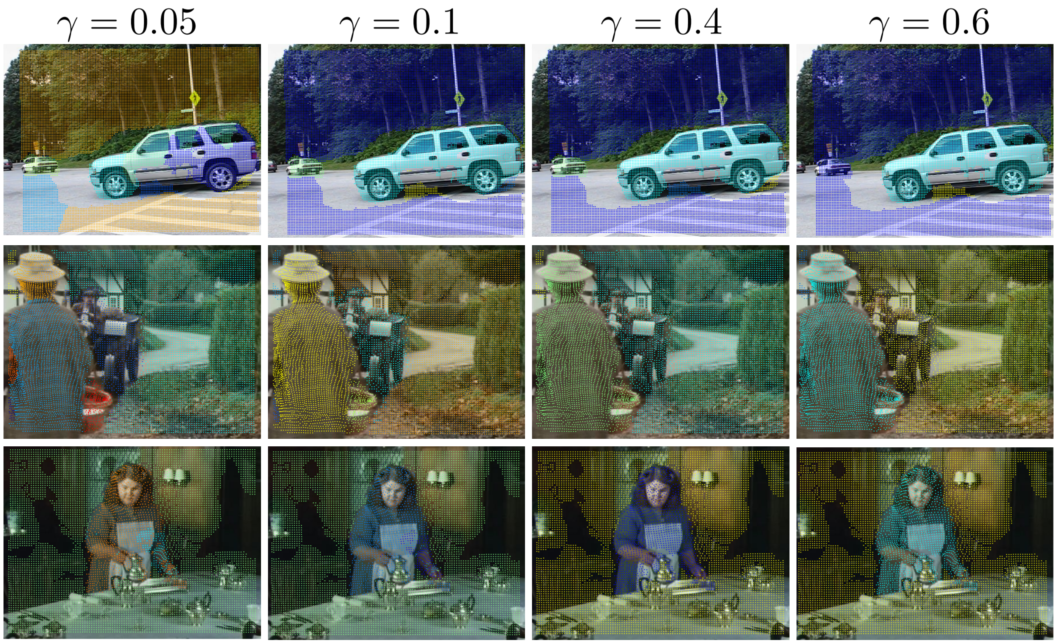


Figure 3.3: Multiscale trajectory partitioning. By varying our discontinuity cutoff threshold γ we obtain finer or coarser granularity clusters.

In static image segmentation, the spectral rounding work of Tolliver (2006) attempts to recover from discretization artifacts by using an iterative rounding procedure for discretizing the eigenvectors. Work of Maire et al. (2008) instead of discretizing the eigenvectors, computes a probability of boundary map by measuring difference in eigenvector values in adjacent half discs. It obtains closed regions using oriented watershed transform on the resulting probability of boundary map.

In video segmentation, we propose a trajectory multiscale partitioning by merging

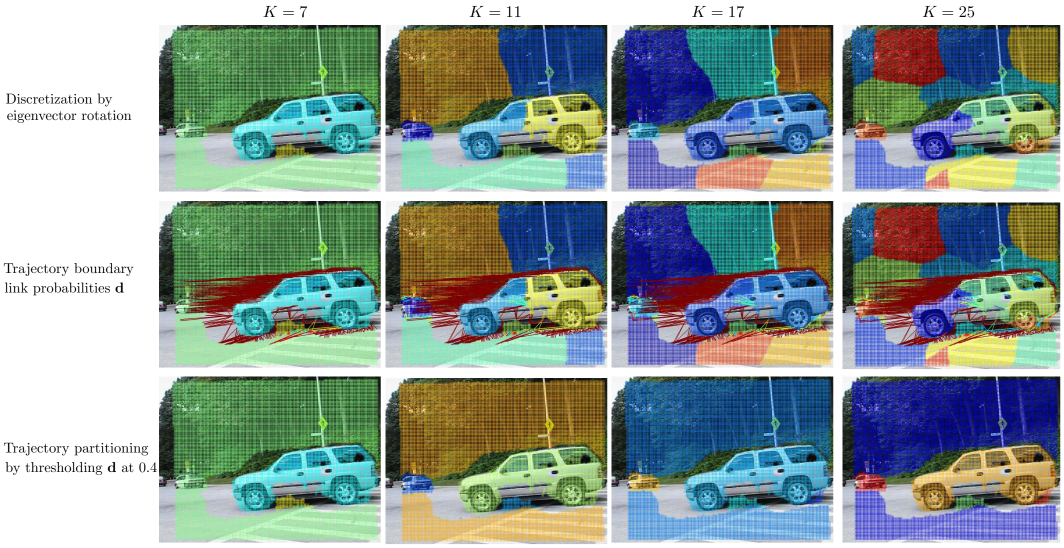

$K = 7$
$K = 11$
$K = 17$
$K = 25$
Discretization by eigenvector rotation
Trajectory boundary link probabilities $\mathbf{d}$
Trajectory partitioning by thresholding $\mathbf{d}$ at 0.4

Figure 3.4: Robustness of discontinuities to the number of eigenvectors. We compare our trajectory multiscale partitioning with previous discretization methods with varying number of eigenvectors K. The resulting trajectory partitionings of our method shown in 3rd row are very similar for different number of eigenvectors K. This shows that Trajectory spectral discontinuities $\mathbf{d}$, shown in 2nd row, are robust to K. In contrast, the clustering of previous discretizations varies a lot with K, as shown in 1st row. The small trajectory cluster below the large car is comprised of trajectories that erroneously slide along the low textured car door.

clusters whose inter-cluster boundaries have discontinuities below a designated threshold. We choose eigenvector rotation as the discretization method for obtaining an initial trajectory partitioning due to its deterministic nature. In contrast, K-means is sensitive to cluster center initialization. For each pair of spatially neighboring trajectory clusters $\mathcal{T}_a, \mathcal{T}_b$, we define their inter-cluster discontinuity $\bar{\mathbf{d}}_{ab}$ to be:

$$\bar{\mathbf{d}}_{ab} = \frac{\sum\limits_{e_{ij} \in E^D, \mathrm{tr}_i \in \mathcal{T}_a,\ \mathrm{tr}_j \in \mathcal{T}_b} \mathbf{d}(e_{ij})}{|\{e_{ij} \in E^D, \mathrm{tr}_i \in \mathcal{T}_a,\ \mathrm{tr}_j \in \mathcal{T}_b\}|}.$$

Given a discontinuity threshold $\gamma \in [0,\ 1]$, we merge clusters (a, b) with inter-cluster discontinuities $\bar{\mathbf{d}}_{a,b} < \gamma$.

In Figure 3.4, 3rd row we show our trajectory clustering for different number of eigenvectors K and $\gamma = 0.4$. It captures true object boundaries and does not suffer from artificial fragmentations of previous methods. While for $K = 10$ the small car is merged with its surrounding, after $K = 11$ the resulting segmentation is the same with increasing number of eigenvectors. By varying γ we obtain finer or coarser clusterings. In scenes with rigid motion, such as the one depicted in Figure 3.4, clusterings from different values γ will be very similar. This is not the case for scenes with deforming or articulated motion, where motion clustering is more ambiguous and less well defined. We show trajectory clusterings while varying γ threshold in Figure 3.7.

3.3.2 Trajectory to pixel partitioning via random walkers

Point trajectories are sparse on low textured regions, e.g., the road in Figure 3.2. Lack of texture results in ambiguous motion estimates. Furthermore, trajectories on untextured backgrounds are often "dragged" by nearby occluding boundaries, a phenomenon referred as optical flow "bleeding" in Thompson (1998).

While lack of texture causes ambiguity in optical flow estimation, at the same time, untextured regions usually have salient boundaries, easy to detect from appearance cues. We cast mapping of trajectory clusters to pixel regions as a seeded superpixel partitioning problem. Seeds are provided from superpixels that well overlap with trajectory clusters. We propagate seed labels to non-seed superpixels via random walkers on multiscale superpixel affinity graphs. We show such mapping is efficient and robust to low image texturedness and optical flow bleeding. Details are presented right below.

Spatio-temporal multiscale region graphs

Given a video sequence I, we compute a set of superpixels by thresholding the output of globalPb of Arbelaez et al. (2009) at value β_{min} at each frame. Let $\mathcal{R} = \{\mathrm{r}_p, p = 1 \cdots n_R\}$ denote the set of superpixels, where n_R is the number of superpixels. We will use the notation r_p to denote both the pth superpixel as well as its pixel mask. Let t_p denote the frame of superpixel r_p.

While superpixels rarely leak across object boundaries, their spatial support is often too small to compute a reliable mapping with trajectory clusters: many of the superpixels overlap with trajectory gaps, as shown in Figure 3.5. We will use random walkers on region affinity graphs to propagate labels of well regions well overlapping with trajectory clusters (seeds) to ambiguous ones.

We establish intra-frame and cross-frame superpixel affinities $\mathbf{A}_R \in \mathbb{R}^{n_R \times n_R}$. In each frame, we compute multi-scale superpixel affinities from ultra-contour maps of Arbelaez et al. (2009). The ultra-contour map provides a different superpixel labeling $\mathbf{s}^{t,\beta} \in \mathbb{N}^{n_R^t \times 1}$ for each probability of boundary threshold $\beta \in [\beta_{\min},\ 1]$, where n_R^t the number of superpixels at frame t. The intra-frame superpixel affinities are as follows:

$$\mathbf{A}_R(\mathrm{r}_p, \mathrm{r}_q) = \max_{\beta,\ \mathbf{s}_p^{t_p,\beta} = \mathbf{s}_q^{t_q,\beta}} \exp(-\frac{\beta^3}{0.1^2}) \cdot \delta(t_p = t_q). \tag{3.10}$$

intuitively, the affinity between two superpixels of the same frame t depends on the threshold β for which they have the same label in $\mathrm{s}^{t,\beta}$, the higher β the lower the affinity. In this way, intra-frame superpixel affinities is $\mathbf{A}_R$ have large spatial connection radius. In each frame, they do not form a planar graph as is often the case in the literature, where each superpixel is connected only to its spatial neighbors. Such long range connectivity between superpixels is lost once globalPb is thresholded at a single scale and resulting regions are treated as independent.

We compute cross-frame superpixel affinities from optical flow. Let r_p^+ denote the pixel mask after translating pixels in r_p with their optical flow displacements: $\mathrm{r}_p^+ = \{(x_\mathrm{p} + u_\mathrm{p}, y_\mathrm{p} + v_\mathrm{p}), \mathrm{p} \in \mathrm{r}_p\}$. The cross-frame superpixel affinities are as follows:

$$\mathbf{A}_R(\mathrm{r}_p, \mathrm{r}_q) = \frac{|\mathrm{r}_p^+ \cap \mathrm{r}_q|}{|\mathrm{r}_p^+ \cup r_q|} \cdot \delta(t_p = t_q + 1). \tag{3.11}$$

Cross-frame region affinities are established only between regions of consecutive video frames. Affinities between regions of non-adjacent frames can be considered using point trajectory overlap, as in Galasso et al. (2012).

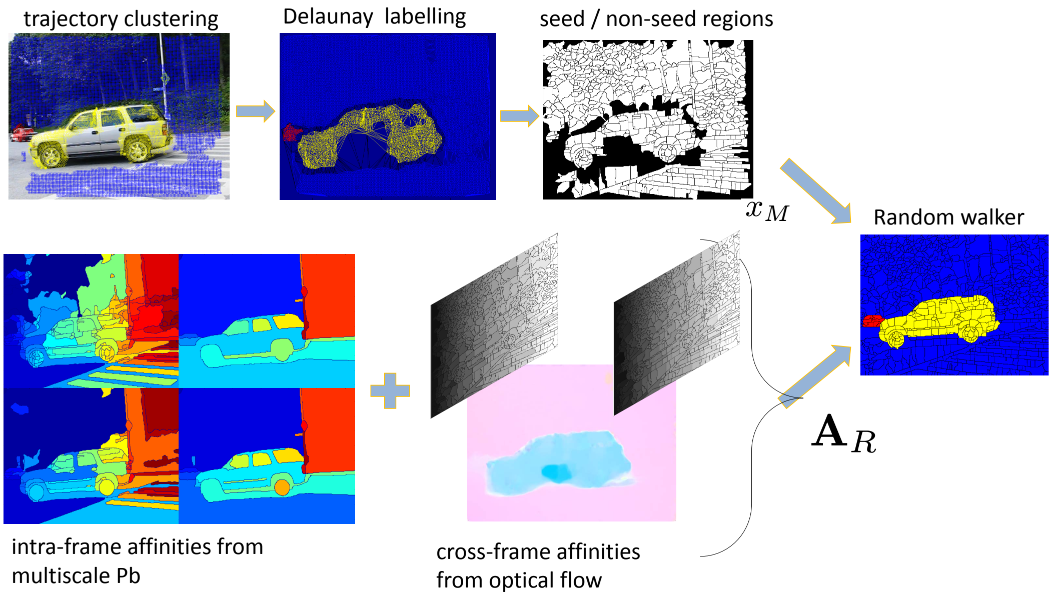


Figure 3.5: Random walkers on spatio-temporal region graphs. The region graph $\mathbf{A}_R$ extends across multiple frames.

Trajectory seeded superpixel labeling

Let $l \in \mathcal{L}^{n_T \times 1}$ denote a trajectory labeling, where $\mathcal{L} = \{1 \cdots L\}$ is the trajectory label set. We want to estimate a corresponding superpixel labeling.

We partition superpixels into seeds (marked) and non seeds (unmarked) depending on their overlap with labeled trajectory Delaunay triangles, as shown in Figure 3.5. We assign to each Delaunay triangle the label shared by its vertices or leave it unlabeled if its vertices do not have the same label. We then compute intersection of each superpixel r with the colored Delaunay triangulation. Seed regions are those that have more that 50% overlap with a trajectory label. We want to estimate the superpixel labels of the rest of the superpixels.

For each label $l \in \mathcal{L}$, let $x \in [0,1]^{n_R \times 1}$ denote the corresponding region potentials. Potential x_a corresponds to the probability of superpixel x_a to be assigned label l. We denote F the seed superpixels of label l, and B the seed superpixels of any other label.

We minimize the following criterion for our superpixel potentials:

$$\begin{aligned} \min_{x}. \qquad & D(x) = \tfrac{1}{2} \textstyle\sum_{a,b} \mathbf{A}_R(\mathrm{r}_a, \mathrm{r}_b)(x_a - x_b)^2 = \tfrac{1}{2} x^T \mathbf{L} x, \\ \text{subject to} \qquad & x_B = 0, \quad x_F = 1, \end{aligned} \tag{3.12}$$

where $\mathbf{L} = \mathbf{D}_{\mathbf{A}_R} - \mathbf{A}_R$ is the unnormalized Laplacian of $\mathbf{A}_R$.

We seek the potential function x that minimizes Eq. 3.12. We assume without loss of generality that superpixel are ordered into marked (seeds) and unmarked (non seeds), and x_M, x_U correspond to potentials of seeded and unseeded superpixel nodes respectively. Then, x_U that minimizes Eq. 3.12 is given by taking the gradient of our cost function and setting it to zero, which gives:

$$\mathbf{L}_U x_U = -\mathbf{L}_{MU}^T x_L, \tag{3.13}$$

as already discussed in the Chapter 2. We solve one linear system for each trajectory label $l \in \mathcal{L}$ and assign each unmarked superpixel to the label it has the highest potential for.

3.3.3 Experiments

The benchmarks available in the literature for scoring performance of video segmentation algorithms mostly focus on one of the following two tasks: 1) object segmentation, where the extracted object masks are scored against ground-truth labeled objects as in Brox and Malik (2010b); Chen and Corso (2010), and 2) occlusion boundary detection, where extracted boundaries are scored against human labeled boundaries without scoring grouping of boundaries into objects, as in Stein et al. (2007); Sundberg et al. (2011). We evaluate our algorithm on both segmentation and boundary detection tasks, quantifying its generality and limitations with varying spatial and temporal resolution, object scale, object deformability and articulation.

We first test our method on the Berkeley motion segmentation benchmark introduced in Brox and Malik (2010b). It is comprised of 26 video sequences of 19 to 700 frames long and extends the Hopkins segmentation dataset of Tron and Vidal (2007). It contains

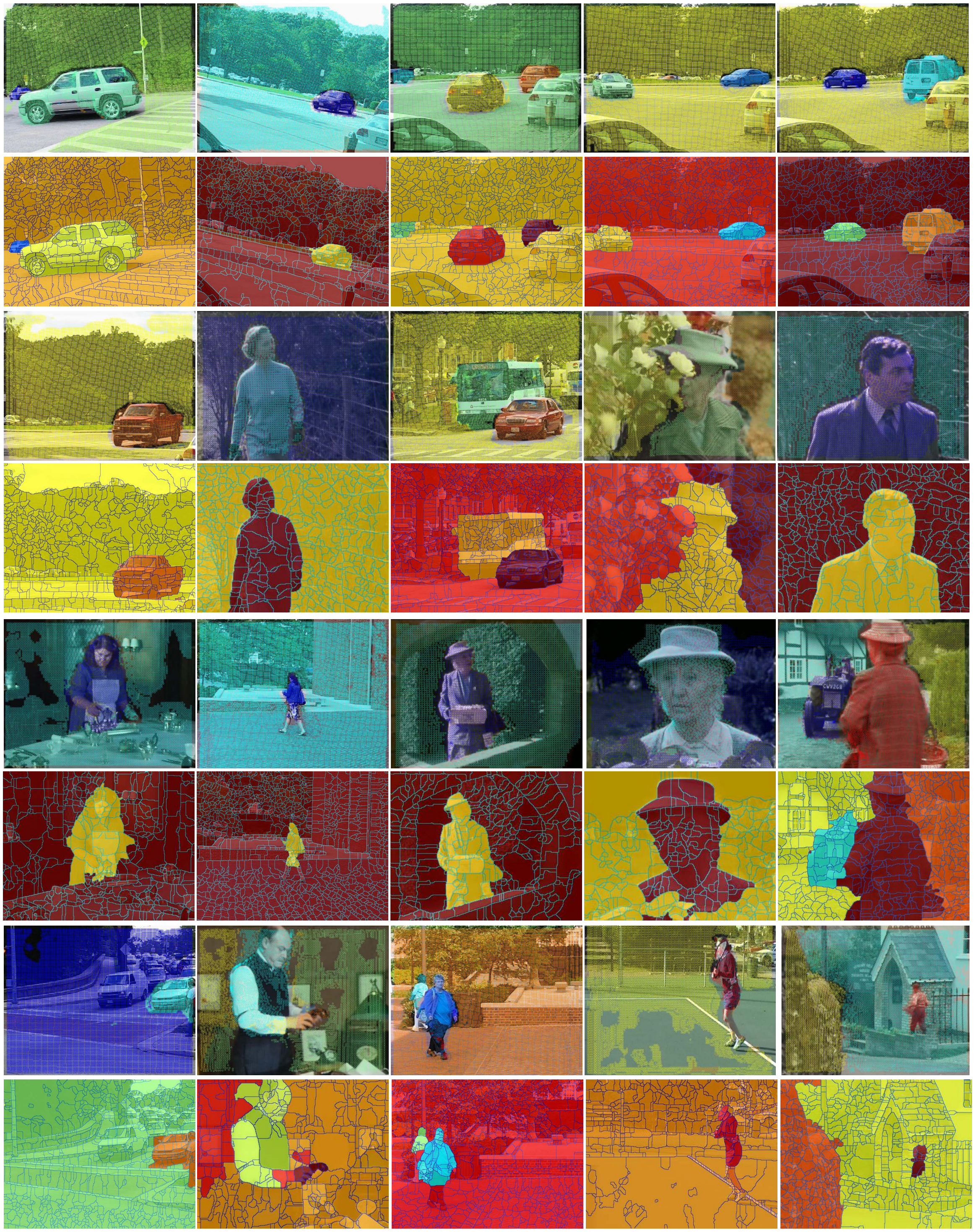


Figure 3.6: Qualitative segmentation results in the Berkeley motion segmentation benchmark. The odd rows show trajectory labelings and the even ones show corresponding pixel labellings.

objects of various scales that exhibit mostly rigid motions. The benchmark scores a single spatio-temporal segmentation map against human labeled objects. We threshold our trajectory link boundary probability $\mathbf{d}$ at $\gamma = 0.3$ and map resulting trajectory clusters to pixel regions as presented in Section 3.3.2. We used spectral threshold $\text{eig} = 0.85$ and globalPb threshold $\beta = 0.05$.

Dataset	density(%)	overall error(%)	region error(%)	over-segment	detections
Moseg10 our method (traj)	5.06	**3.84**	26.09	**0.1**	23
Moseg10 our method (regions)	**87.24**	4.66	29.72	0.15	19
Moseg10 Brox and Malik (2010b)	3.32	4.29	**23.7**	0.35	**24**
Moseg50 our method (traj)	4.97	**3.37**	**22.1**	0.75	**27**
Moseg50 our method (regions)	**87.32**	4.24	27.79	0.75	20
Moseg50 Brox and Malik (2010b)	3.32	3.50	27.09	**0.45**	26
Moseg200 our method (traj)	4.94	4.38	**19.3**	2.3	**30**
Moseg200 our method (regions)	**87.49**	5.06	23.95	2.05	25
Moseg200 Brox and Malik (2010b)	3.31	**3.74**	24.66	**1.05**	29

Table 3.1: Quantitative segmentation results in the Berkeley motion segmentation benchmark. Detections are missed in the pixel labelings when wrong mapping increases the region error above 10%.

Quantitative results for both our trajectory and pixel labeling are shown in Table 3.1. We test on the first 10, 50 and 200 frames in each video sequence. When the sequence has less frames, we use the whole sequence. The benchmark evaluation code optimally assigns extracted video segments to ground-truth objects and background. *Clustering error* measures percent of wrongly labeled pixels. *Region clustering error* computes percent of correctly labeled pixels in each object rather than the whole scene. This metric is important as in videos where the background occupies a very large part of the scene an algorithm that labels all pixels with one (background) label achieves low clustering error. Objects with region error below 10% are assumed correct *detections*. *Over-segmentation* measures how many extracted video segments are assigned to the same ground-truth object. We use trim mean to average clustering error and region clustering error across the video sequences and their objects respectively and we reject the top and bottom 10% of the measurements. Qualitative results of our approach are shown in Figure 3.6.

Our method performs well and segments the objects correctly. It occasionally over-segments the objects as we see from the increased over-segmentation error for sequence length 200, but such over-segmentations occur mostly in the background. Wrong cluster to region mappings, that assign parts of objects to the background rather than to the right cluster label cause the number of detections to drop for our region partitionings.

The qualitative results in Figure 3.6 show limitations of our framework under articulated motion. Freely moving body parts may be disconnected from the main body and merged to the background. And indeed, our trajectory affinities in Eq. 3.4 penalize motion discontinuities of different articulated parts. More importantly though, trajectories on body limbs are often too short to be informative, due to the frequent self occlusions. Topological tracking presented in Section 3.4 attempts to recover from such limitations by employing information of video figure-ground topology along with trajectory motion.

Next, we test our method on the video segmentation benchmark of Chen and Corso (2010). It contains 8 video sequence of average length 85 frames. The sequences have a wide range of motions but have low spatial resolution. Spatial resolution is often a determinant parameter of success or failure of trajectory based algorithms, since quality of optical flow decreases with decrease of spatial or temporal resolution. We show qualitative results of our approach on all eight sequences of the dataset in Figure 3.7. We used spectral threshold $\mathrm{eig} = 0.85$ and globalPb threshold $\beta = 0.2$. In the first row, the reflectance on the bus confuses motion estimation. Notice in the 4th row the optical flow bleeding on the untextured soccer field, next to the legs of the players, and how mapping to regions alleviates from this problem.

Next, we test our algorithm in the CMU occlusion boundary detection benchmark, introduced in Stein et al. (2007). The benchmark contains 30 short video sequences and focuses on occlusion boundary extraction without scoring boundary grouping into objects. Each video sequence may contain object or camera motion. We used spectral threshold

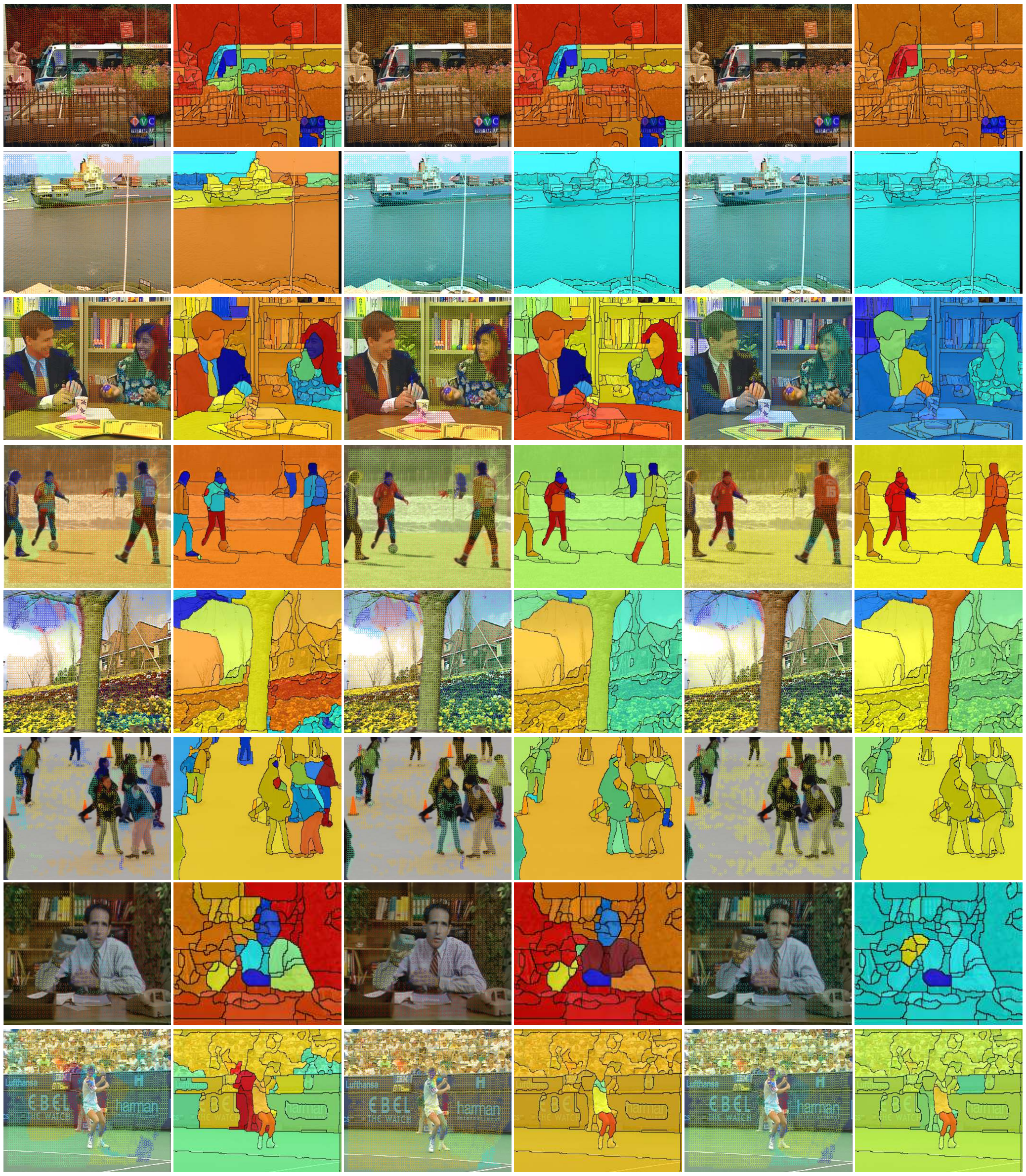


Figure 3.7: Video segmentation results in the dataset of Chen and Corso (2010). The dataset contains eight video sequences, shown along the eight rows. Columns 1, 3, 5: trajectory clustering with probability of boundary link thresholded at 0.1, 0.3 and 0.6 respectively. Columns 2, 4, 6: pixel segmentation by mapping corresponding trajectory clusters to regions.

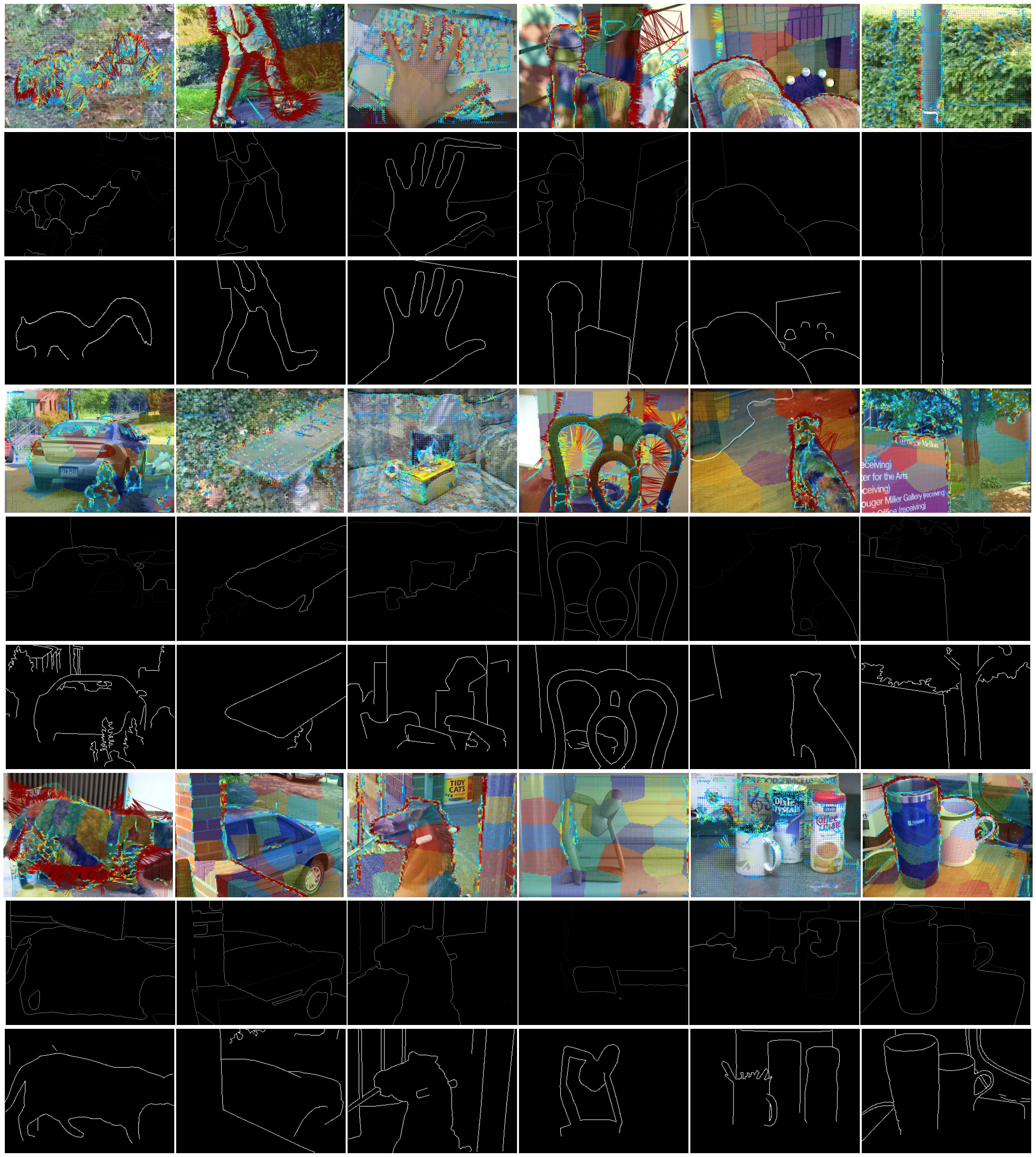


Figure 3.8: Results in the CMU boundary detection benchmark of Stein et al. (2007). Rows 1, 4, 7: trajectory link boundary probability $\mathbf{d}$. We show only the edges e_{ij} with nonzero $\mathbf{d}(e_{ij})$. Point color indicates trajectory cluster labels of the finest trajectory partitioning. Rows 2, 5, 8: the resulting pixel probability boundary maps from our multiscale trajectory partitioning and region mapping. Rows 3, 6, 9: ground-truth occlusion boundaries.

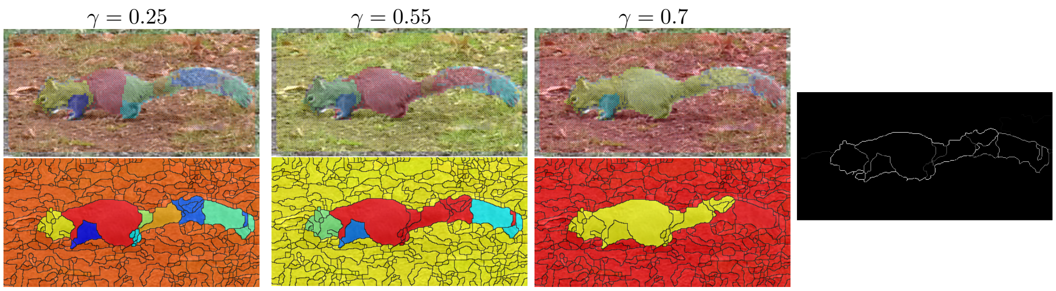


Figure 3.9: Pixelwise probability of occlusion boundary map. Each ucm contour fragment has occlusion probability equal to the maximum γ threshold for which it is a boundary contour in the corresponding superpixel labeling map.

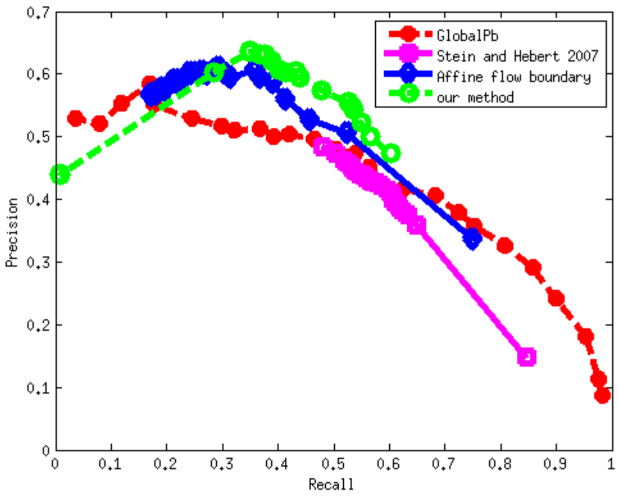


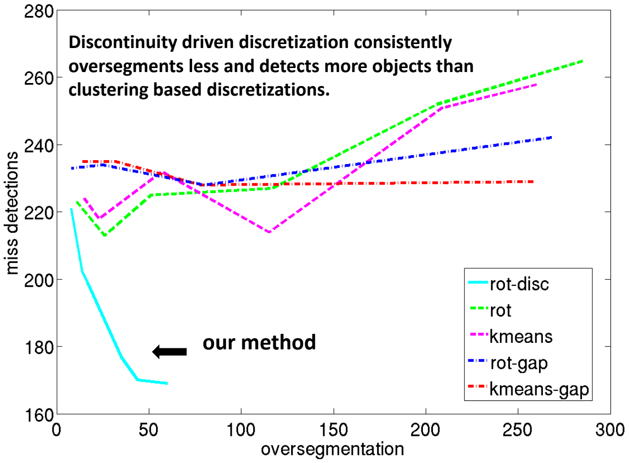
Figure 3.10: Precision-recall curve in the CMU occlusion boundary benchmark.

eig $= 0.4$ and minimum globalPb threshold $\beta = 0.2$. We compute a pixelwise occlusion probability of boundary map as follows: we threshold our trajectory link probability $\mathbf{d}$ in various cutoff values γ in the interval $[0,\ 1]$ and map the resulting trajectory clusters to image regions. Each ucm boundary fragment (shared by a pair of adjacent superpixels) takes the value of the highest cutoff value γ for which the boundary fragment exists. We show our computed occlusion boundary maps in Figure 3.8. We also visualize the trajectory link boundary probabilities $\mathbf{d}$.

We show precision-recall curves of our method and baselines in Figure 3.10. We compute the precision-recall curves by mapping extracted boundaries to groundtruth boundaries under various cut-off values, using the assignment code of Martin et al. (2001). We accept an extracted edgel as groundtruth if the Euclidean distance is below 0.01 of the maximum of width and height of the image, as used in the Berkley static boundary detection benchmark. Along with our algorithm we evaluate the globalPb detector of Arbelaez et al. (2009) for which code is publicly available and the occlusion boundary detector of Stein et al. (2007) for which authors supply their extracted boundary probability maps. We also score our own implementation of a baseline motion boundary detector that for each ucm boundary fragment fits affine models to the optical flow vectors of the left and right adjacent regions and compares the affine motion estimates on the common boundary. The larger the disagreement between the two affine estimates, the larger the probability that the fragment corresponds to an occlusion boundary. This computation is in the heart of the occlusion boundary detector of Sundberg et al. (2011), for which code or results are not available. The authors do not specify which Pb threshold they use to obtain the initial segmentation. We used $\beta = 0.2$, same for our algorithm.

Finally, we evaluate our discontinuity detector as a general way of discretizing the spectral embedding. We denote our method as *rot-disc* (rotation+discontinuity based cluster merging) and compare with four other discretization algorithms: 1) K-means and 2) eigenvector rotation (*rot*), with number of eigenvectors K selected by thresholding eigenvalues, 3) K-means and 4) eigenvector rotation, with K selected by thresholding consecutive eigenvalue difference (denoted by *K-means-gap* and *rot-gap* respectively). In Figure 3.11 we plot the average over-segmentation error (i.e., the number of interior fragmentations not corresponding to object boundaries) against the average miss detection error (i.e., the number of groundtruth objects or world scene that were not matched to a cluster with intersection over union score above 70%), as we vary the thresholds that determine K of the various algorithms. We average across the 26 video sequences of the Berkeley motion segmentation benchmark. Our method outperforms standard discretizations, it has considerably smaller over-segmentation error for the same miss-detection error, which shows that the local discontinuity values are a simple yet effective fix to discretization artifacts of previous approaches.

Figure 3.11: Comparison of spectral embedding discretizations. Curves are computed by varying the number of spectral eigenvectors K while keeping our discontinuity threshold fixed at $\gamma = 0.3$.



Running time Our method for video sequences of 50 frames in Moseg dataset took on average 5.5 minutes on a 2.6 GHz processor, excluding the optical flow and globalPb computation that can be parallelized for the different frames in the video. Memory requirements do not scale well with increasing number of trajectories, which means that a long video sequence would need to be chopped into subsequences of smaller length. Topological tracking presented in the next section, alleviates to a certain extent from intense memory requirements, by computing early a figure-ground classification and clustering only on the foreground trajectories, without sacrificing performance.

3.4 Topological Tracking

The results of the previous section show that long range trajectory motion is very effective in segmenting rigidly moving objects. At the same time, we saw limitations of our method under articulated motion, since aperture problems and frequent self-occlusions and deformations of the human body cause trajectories to be sparse and short. Articulated limbs were often merged to the background. For general deforming and articulated motion, grouping by motion similarity may be insufficient, causing over-fragmentations of objects into distinctly moving parts. In general, model selection, i.e., choosing the right level of granularity from the segmentation hierarchy, is a hard, and maybe ill-posed problem, in absence of model information. For the resulting segmentation to have a semantic interpretation, the grouping cues need to go beyond appearance or motion similarity.

In this section, we will explore motion saliency for segmenting interacting articulated bodies in monocular videos. Motion saliency approaches, such as Gao et al. (2008); Mahadevan and Vasconcelos (2010); Rahtu et al. (2010a), employ center-surround filters for extracting the moving ensemble - which may not have coherent motion - from the otherwise static world scene (not necessarily static camera). In this way, they compute a pixel figure-ground segmentation in each frame. They do not dis-entangle the different objects in the moving ensemble though. Motion saliency works are often inspired by psychophysics experiments of Nothdurft (1992) on human motion perception: entities with dissimilar motion are perceived as a group when viewed against smoothly moving entities.

We present an approach that couples saliency information with trajectories to correctly delineate the interacting objects of the foreground. Specifically, we will use object connectedness constraints from video foreground to establish repulsions (not-group relationships) between trajectories.

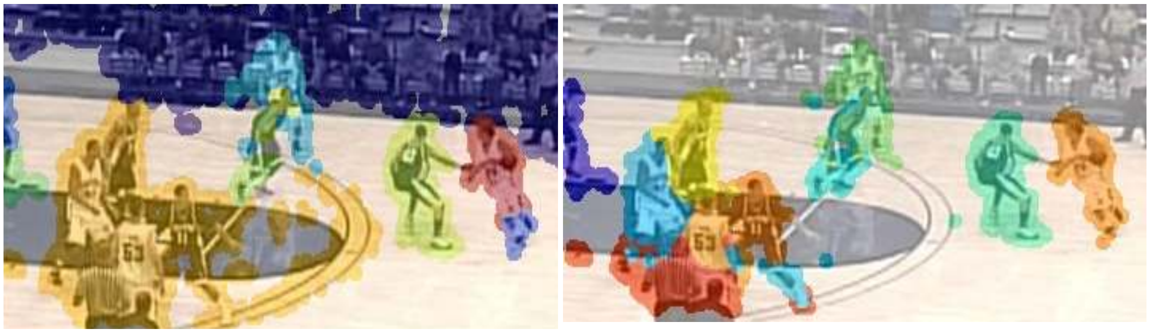


Figure 3.12: Left: Segmentation using only motion affinities. Right: Segmentation using motion affinities and topology-driven repulsions.

Topology-driven trajectory repulsions

In each frame I_t, we compute a saliency map using the multi-scale center-surround filter of Rahtu et al. (2010b) on optical flow magnitude. We classify trajectories into foreground and background depending on their intersection with the motion salient pixels. Trajectories that intersect salient foreground for more than 10% of their lifespan are classified as salient and are used to compute per frame foreground maps $F_t \in \{0,1\}^{w \times h}, t = 1 \cdots T$, where w, h the width and height of I_t. Computing saliency on trajectories rather than pixels allows to assign salient even objects in frames when they are stationary, as shown in Figure 3.13.

We compute connected components in the per frame foreground maps. Let $C_t : \mathcal{T} \rightarrow \mathbb{N}$ denote the function that assigns to each trajectory the connected component index in the foreground frame map F_t. We use the term foreground topology to describe the assignment of trajectories to connected components.

- Foreground topology *cannot* indicate when two trajectories should be grouped together: a connected component in a foreground map may contain a single agent or a group of agents.
- Foreground topology *can* indicate when two trajectories *cannot* be grouped together if assigned to different connected components, since they would violate object connectedness.

Nevertheless, indicating separation is as useful as indicating attraction. We establish trajectory repulsions between trajectories that at any frame of they time overlap they belong to different connected components, depicted also in Figure 3.14:

$$\mathbf{R}_T(i,j) = \delta(\exists\, t \in T_i \cap T_j, C_t(\mathrm{tr}_i) \neq C_t(\mathrm{tr}_j)). \tag{3.14}$$

We cancel trajectory affinities on repulsive links:

$$\mathbf{A}'_T = \mathbf{A}_T \bullet (1 - \mathbf{R}_T), \tag{3.15}$$

where $\bullet$ denotes Hadamard product. We have found affinity cancellation to be more robust to wrong repulsions in $\mathbf{R}_T$ than spectral clustering with attraction and repulsion of Yu and Shi (2001).

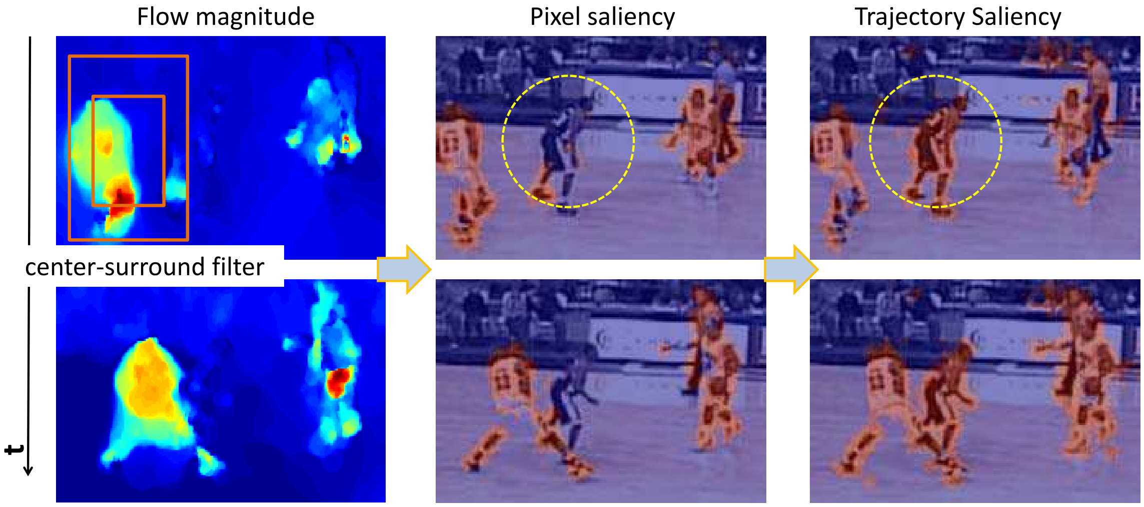


Figure 3.13: Trajectory motion saliency. Although the basketball player is not moving in the current frame, he is assigned as salient in the trajectory saliency maps. Computing saliency on trajectories rather than pixels propagates information from frames with object distinct motion to frames with no motion.

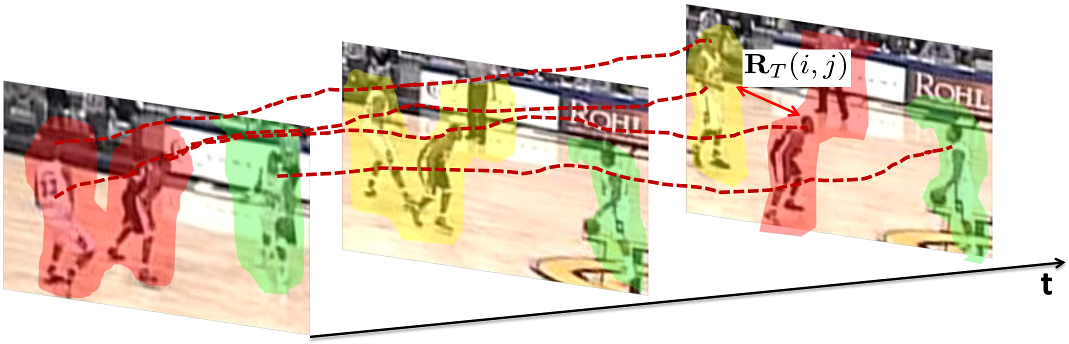


Figure 3.14: Topology-driven trajectory repulsions. The connected components of the foreground maps are shown in different colors. Repulsive weights are set between trajectories that belong to different connected components at any frame of their time overlap.

We compute spectral clustering in $\mathbf{A}'_T$ matrix. We use different number of eigenvectors for discretizing the embedding and keep only clusters $x \in \{0,1\}^{n_T \times 1}$ that do not have interior repulsions $x^T \mathbf{R}_T x$.

3.4.1 Experiments

We test topological tracking in the Berkeley motion segmentation benchmark and in Figment segmentation dataset which we introduce. Figment (*Fig*ure untangle*ment*) dataset contains 18 video sequences of 50-80 frames each, with scenes from a basketball game collected by Vondrick et al. (2010). In each sequence, we supply groundtruth labels for all players and the background scene every seven frames. For evaluation, each trajectory cluster is optimally assigned to one groundtruth object based on maximum intersection. The metrics are familiar from Section 3.3.3. The new metric *leakage* measures the percentage of leaking trajectory clusters, i.e., clusters that have high intersection over union score with more than one groundtruth labeled masks (more than 50 % of the one with their assigned mask). Quantitative results are shown in Tables 3.2, 3.3 and qualitative results are shown in Figure 3.15.

Dataset	density(%)	overall error(%)	region error(%)	over-segment	detections
Moseg50 motion tracking	**4.97**	**3.37**	22.1	0.75	**27**
Moseg50 topological tracking	3.22	3.76	**22.06**	1.15	25
Moseg50 Brox and Malik (2010b)	3.32%	3.50%	27.09%	**0.45**	26

Table 3.2: Segmentation results of topological tracking in Berkeley motion segmentation dataset.

In contrast to the Berkeley motion segmentation dataset, where there is no gain from the use of foreground topological information, under articulation and object deformation, connectedness constraints improve performance by a large margin. This is due to the fact that articulated motion is not always informative to provide a semantic video segmentation. However, in case the objects in the scene do not separate, i.e., they belong to the

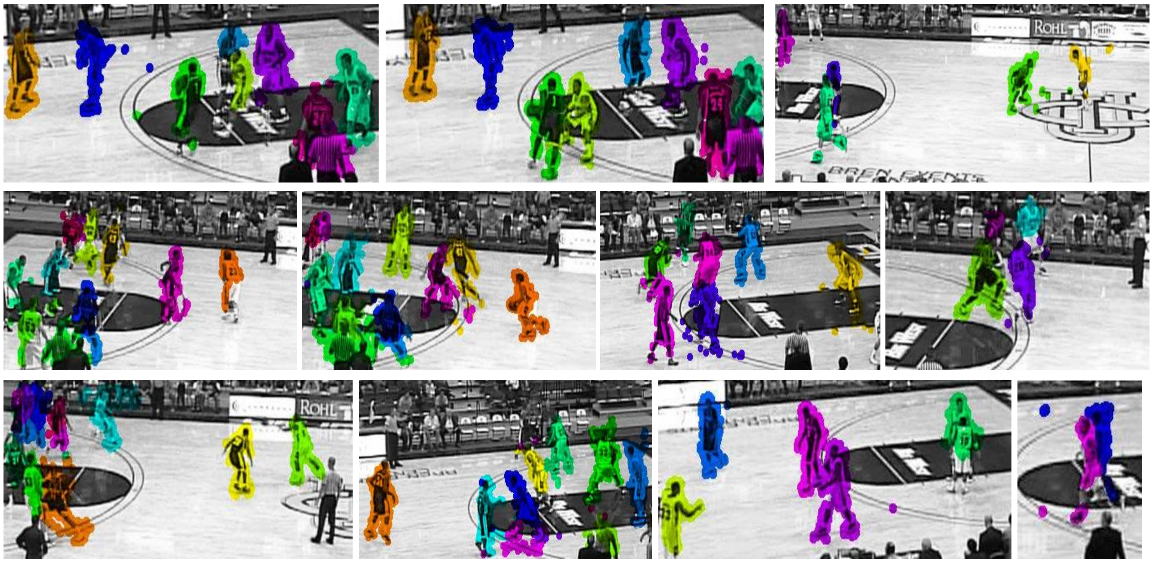


Figure 3.15: Segmentation results in Figment dataset. We show dilated trajectory points in each frame. The basketball players are correctly delineated in most cases.

Dataset	density(%)	overall error(%)	region error(%)	over-segment	leakage(%)
Figment motion tracking	4.90	17.49	41.06	3.21	44.96
Figment topological tracking	**5.21**	**4.73**	**20.32**	1.57	**16.52**
Figment Brox and Malik (2010b)	0.57	20.74	86.43	**0**	81.55

Table 3.3: Quantitative segmentation results of topological tracking in Figment dataset. Under articulated motion and close agent interactions the gain from topological information is substantial.

same connected component throughout the whole video sequence, then motion is still the only cue for segmentation. In practise, we found it hard to recover from isolated trajectory groups on the background, that were found salient due to noisy motion estimates, as shown also in the high oversegmentation error of topological tracking in Berkeley motion segmentation dataset.

3.4.2 Discussion

We presented object connectedness constraints from trajectory saliency as a way to obtain repulsive (not-group) relationships between trajectories, that depend on target topological separations rather than motion dissimilarity. In the next chapter, we will use detection responses for inducing repulsive constraints between trajectories, necessary for resolving motion leakage under lack of distinct motion. Detector-driven repulsions are less dependent on “lucky” target separations, that topological tracking relies upon.

Chapter 4

Two-Granularity Tracking

> What we see depends mainly on what we look for.
>
> — John Lubbock

Our goal is tracking people in crowded scenes. People moving in crowds often occlude each other. We present a two-granularity tracking framework that exploits model information from object detectors and long term motion information of point trajectories to track objects through partial occlusions. Detectors alone are often insufficient for accurately parsing cross-object occlusions. Motion alone is ambiguous under lack of distinct object motion or low texture, as already discussed in Chapter 3. We propose a grouping framework that combines trajectory motion affinities with detection-driven repulsions to correct motion leakages and select the right segments that correspond to the different people in the scene. These segments, in contrast to bounding boxes, accurately capture the targets as they undergo heavy occlusions while navigating in the crowded scene.

We will use a whole object representation as opposed to objects parts. This will not allow to extract detailed object pose of the targets. In the next Chapter, we will build upon our two-granularity tracking to present a body pose estimation framework in crowded scenes.

4.1 Introduction

Frameworks combining perceptual grouping information and object detection have a long history in segmentation and recognition of static scenes, such as the works of Amir and Lindenbaum (1998); Borenstein and Ullman; Hariharan et al. (2011); Ionescu et al. (2011b); Levin and Weiss (2006); Mori et al. (2004). In the video domain, most recognition frameworks rely on frame-by-frame detection. Perceptual motion based grouping has not been exploited in current tracking-by-detection systems. The large data throughput of videos - in comparison to still images - requests fast, time efficient processing, as noted in Xu et al. (2012a). Temporal demands are especially prominent in real time applications. Timely motion estimation is possible in hardware based optical flow implementations, which are not widely available yet. For this reason, two lines of work, namely 1) tracking-by-detection, and 2) motion/appearance based video segmentation, have developed independently, targeting different applications and time requirements.

Current state-of-the-art tracking algorithms Breitenstein et al. (2009); Brendel et al. (2011); Leibe et al. (2007) link detections over time. Object detection under persistent partial occlusions is challenging since features extracted from a window around an object may be corrupted by surrounding occluders. A box tracker cannot adapt to the changing visibility mask of a partially occluded object. As a result, detection responses come as loose-fit / under-fit boxes around a target, or as hallucinated detections spanning over or in the gap of two objects, which causes difficulties to data association during tracking. Apart from occlusions, object deformation poses additional challenges, resulting in a difficult trade-off between precision and recall for deformable object detection.

Motion based video segments can adapt to the changing visibility masks of moving targets under occlusions, as shown in Figure 4.1, 2nd column. However, they fail under similar motion across different targets, which is often the case when people move in

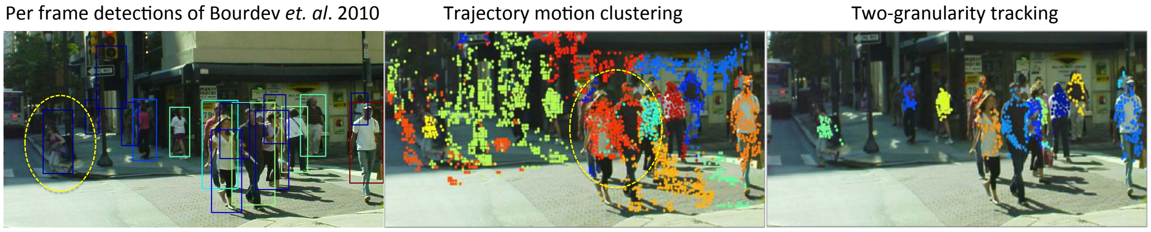

Per frame detections of Bourdev *et. al*. 2010
Trajectory motion clustering
Two-granularity tracking

Figure 4.1: Left: Detections of poselet detector of Bourdev et al. (2010). Many detector responses come as a loose fit or under-fit around the targets, especially under partial occlusions (the crowd in the center) or deformations (the lady running on the left). Center: Trajectory spectral clustering. Deforming targets are captured and moving people under partial occlusions are correctly delineated. However, clusters leak across objects that move similarly (the couple in the center). Right: Two granularity tracking. Trajectory spectral clustering in the steered graph of motion attractions and detection-driven repulsions. Motion leakages are fixed and trajectory clusters clusters adapt to the visibility mask of the targets under partial occlusions.

crowds.

We propose a tracking framework that exploits cues in two levels of tracking granularity:

1. tracking-by-detection, and
2. dense point trajectories.

We cast mutli-object tracking as a joint detection and trajectory partitioning problem. We establish trajectory affinities using both long range motion similarity and associations to detections. Specifically, incompatible detections induce repulsive weights between trajectories associated with them, as shown in Figure 4.2. In this way, motion leakage is corrected across similarly moving objects captured by confident detections. At the same time, our partitioning framework can generate potentially an exponential number of trajectory clusters to fit the changing visibility masks of targets under partial occlusions. Resulting trajectory clusters link detections across miss detection gaps. Our goal is to improve robustness in tracking, minimize drifts due to interpolation due to lack of reliable detections, rather than the segmentation itself. We call our framework “graph steering” since detection information is incorporated in the form of link cancellation, that steer (change) corresponding motion affinities.

We show that graph steering is resistant to noisy dis-associations of false alarm or spatially inaccurate detections. Resulting trajectory clusters in the steered affinity graph provide feedback to detection classification by rejecting detections misaligned with them. Each resulting trajectory-detection co-cluster corresponds to one object hypothesis in space and time, as shown in Figure 4.1 3rd column. We analyze the graph connectivity and resulting spectral clustering as we vary the rate of false alarm detections.

A byproduct of the two-granularity representation is the relative depth ordering of the resulting object tracks, by analyzing the lifespans of point trajectories in the corresponding detection and trajectory co-clusters. In contrast, previous tracking-by-detection frameworks cannot easily differentiate a miss detection gap from an occlusion gap. To that respect, two-granularity representations better help the analysis of target behavior by grounding each bounding box to the relevant trajectory content and inferring its occlusions and dis-occlusions.

We test our algorithm in a variety of tracking benchmarks available in the literature and show its capability to track people under persistent partial occlusions. We also introduce a new tracking dataset, we call *UrbanStreet*, captured from a stereo rig mounted on a car driving in the streets of Philadelphia, PA. We provide segmentation masks rather than bounding boxes as groundtruth pedestrian labels, since often times the targets are partially occluded while navigating in traffic. We show qualitative and quantitative results of our system under CLEAR MOT tracking metrics and quantify its performance under both monocular and binocular input. The UrbanStreet dataset and the code of our algorithm is available at `www.seas.upenn.edu / ∼ katef/steer.html/`.

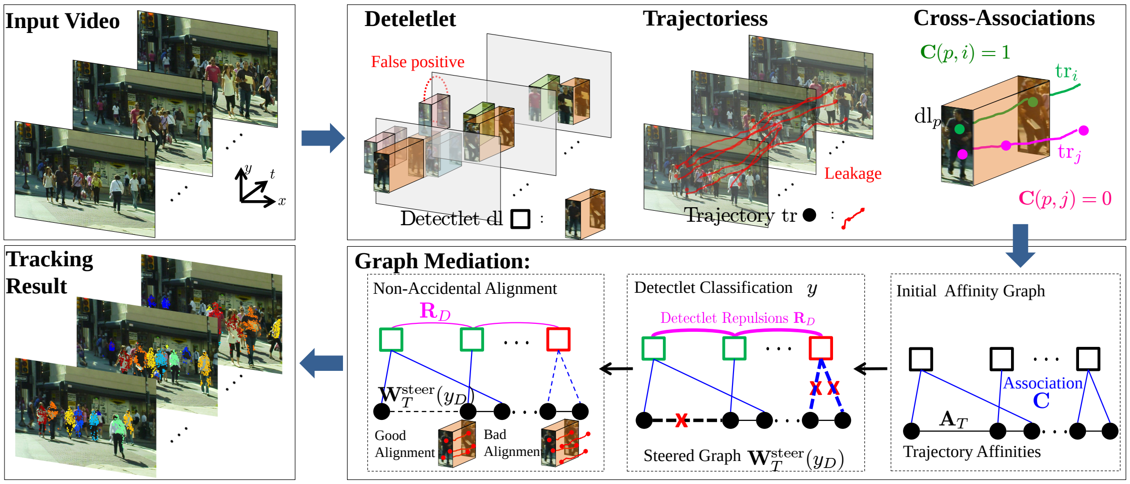


Figure 4.2: Two-granularity tracking overview. We establish one *detectlet* (detection tracklet) and one trajectory graph with repulsive and attractive weights $\mathbf{R}_D$ and $\mathbf{A}_T$ respectively and cross-associations $\mathbf{C}$. We jointly optimize over detectlet classification y and co-clustering X, Y via graph steering: Selected detectlets induce dis-associations between their associated point trajectories. Clustering in the modified graph $\mathbf{W}_T^{\mathrm{steer}}(y)$ verifies or rejects detectlet hypotheses depending on their alignment with trajectory clusters, changing accordingly their classification y. Here, the green detectlets are accepted while the red one is misaligned and thus rejected. Each detectlet/trajectory co-cluster corresponds to an object hypothesis in space and time.

4.2 Related work

Most tracking algorithms estimate the states of targets over time using two types of pattern matching: 1) matching image patches to pre-trained detector templates 2) matching image patches to on-the-fly built detection models for each target. Both matching problems are ambiguous under 1) target deformations and 2) target interactions and occlusions, which cause the appearance of a target to divert from 1) the common patterns in the training set of the pre-trained detector, 2) the learnt pattern of the on-the-fly detection model. This results in an important trade-off between precision and recall for detection and data association in tracking.

Improved object detectors, using expressive mixture models learnt from larger training sets, have led to higher accuracies in multi-object tracking-by-detection Brendel et al. (2011). Cross-object occlusions still pose challenges to current detectors due to the combinatorial number of resulting object configurations. Researchers seek ways to attack configuration explosion under object interactions in different ways: 1) learning more templates, following visual similarity rather than categorical description of such configurations Malisiewicz and Efros (2009); Sadeghi and Farhadi (2011), 2) searching for part-based representations that would effectively share parts between "rare" and "common" configurations Desai and Ramanan (2012b). Some multi-object configurations are stable, repeatable in the datasets, and thus easy to detect as a template, e.g., man on horse, hand holding cup, cars parked in a row Pepik et al. (2013) etc. General cross-object occlusions though may not always be same as repeatable or stable. Instead of trying to improve the pre-trained detection model, in this work, we explore motion segmentation for untangling interacting objects as a way of dealing with limitations of standard object detectors.

Researchers have explored ways of linking sparse confident detections in time in numerous ways. Works of Bibby and Reid (2008); Mitzel et al. (2010); Ren and Malik (2007) use figure-ground segmentation and level-set segmentors. Works of Shu et al. (2012); Wu and Nevatia (2007) use body parts tracklets in the place of whole body tracklets for tracking partially occluded objects. Numerous works delay the data association process and use future information to decide assignment of detections to targets Brendel et al. (2011); Huang et al. (2008a). Work of Huang et al. (2008a) proposes a hierarchical association of detection responses, updating the cross-tracklet affinities from information of increasing tracklet length. Numerous works focus on goal planning for targets as they navigate in their environment, considering scene exits/entries to estimate termination of tracklets Huang et al. (2008a), sidewalk information and pedestrian walking preferences Kitani et al. (2012), cross-target collision prediction and resolution Gong et al. (2011); Pellegrini et al. (2009), periodic walking cycles Andriluka et al. (2008), or simple motion smoothness priors Huang et al. (2008a).

Many approaches learn on-the-fly appearance models with the aim to adapt to the appearance distribution of the target at hand, which may differ significantly from the distribution under which the pre-trained detector is learnt Gall et al. (2011); Okuma et al. (2004). Under stationary cameras, background subtraction is used to help detection of targets Berclaz et al. (2011). Work of Breitenstein et al. (2009) employs on the-fly-learnt target appearance classifiers with a probabilistic gating function and continuous detection maps, instead of discretized, non-maxima suppressed detections, for guiding a particle filter in a causal, online tracking system.

Our works focuses on estimating concurrent tracking and segmentation of targets, with the aim of linking sparse detections of a pre-trained model via motion trajectory clusters. Instead of on-the-fly learning target appearance classifiers, we employ a robust variational coarse-to-fine optical flow computation for frame-to-frame pixel matching. We link such pixel matches into trajectories and compute long-range motion trajectories similarities. Trajectory units are more powerful than level set segmentors or per frame appearance models thanks to their large temporal support: they can distinguish targets with different motions despite accidental appearance similarity, where target appearance models would fail. While right now our system is not causal (we compute two-granularity tracking using information in all frames that are available), one could consider only trajectories spanning past frames for developping a causal tracking system.

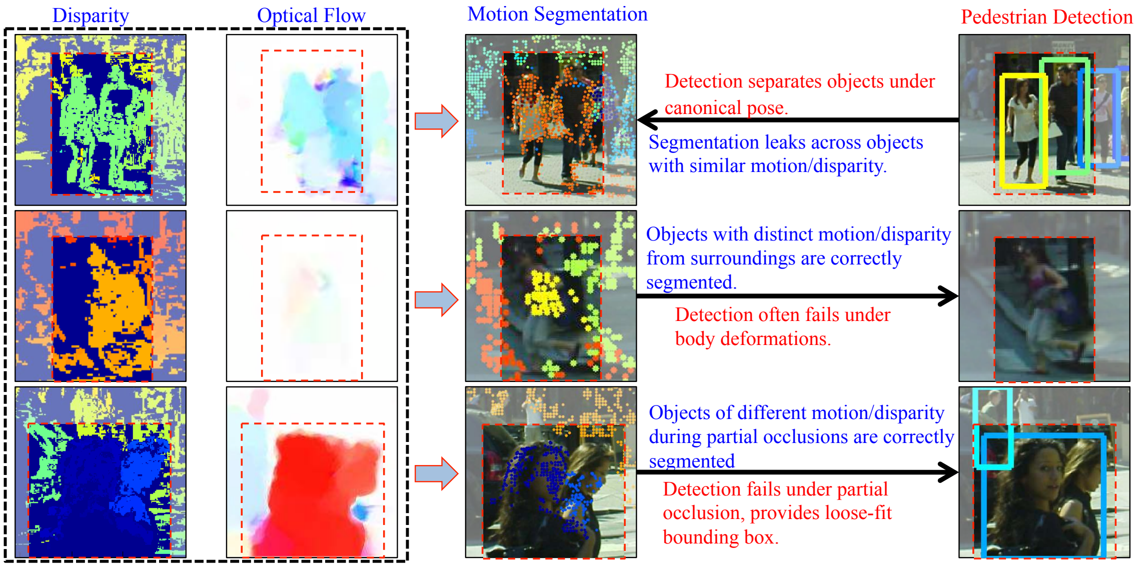

Disparity
Optical Flow
Motion Segmentation
Pedestrian Detection
Detection separates objects under canonical pose.
Segmentation leaks across objects with similar motion/disparity.
Objects with distinct motion/disparity from surroundings are correctly segmented.
Detection often fails under body deformations.
Objects of different motion/disparity during partial occlusions are correctly segmented
Detection fails under partial occlusion, provides loose-fit bounding box.

Figure 4.3: Complementarity of detectlets and point trajectories. 1st Row: Similarly moving objects. 2nd Row: Body deformations. 3rd Row: Partial occlusions. Motion segmentation is computed by spectral clustering on motion and disparity based trajectory affinities.

4.3 Tracking units

Detection tracklets (we will call them *detectlets* for short) and point trajectories provide complementary information for tracking in different points in space and time:

1. Detectlets may be sparse in time. They often miss objects under severe occlusions or extreme deformations. In contrast, point trajectories are dense in space and time.

2. Detection bounding boxes are often spatially inaccurate. In contrast, point trajectories have small spatial support, hence trajectory clusters can adapt to the changing visibility mask of occluded pedestrians.

3. Detectlets can separate objects under canonical pose, despite their motion or disparity being similar to surroundings. In contrast, trajectory affinities leak across objects moving in groups with (persistently) similar motion and disparity.

A summary of advantages and disadvantages of detection and motion/disparity segmentation is presented in Figure 4.3. In Sections 4.3.1 and 4.3.2 we present our trajectory and detectlet units and their pairwise affinities $\mathbf{A}_T$, and repulsions $\mathbf{R}_D$, respectively, and in Section 4.3.3 we present their cross-associations $\mathbf{C}$.

4.3.1 Fine-grained point trajectories

We define a trajectory tr_i to be a sequence of space-time points: $\mathrm{tr}_i = \{(x_i^t, y_i^t), t \in T_i\}$ where T_i is the frame span of tr_i. In case of stereo or multi-view input, each trajectory is augmented with a disparity or depth value per frame, depending on whether camera calibration information is available: $\mathrm{tr}_i = \{(x_i^t, y_i^t, z_i^t), t \in T_i\}$. In cases of calibrated cameras, we still prefer to use the pixel locations x_i^t, y_i^t instead of true 3D coordinates X_i^t, Y_i^t to avoid errors during disparity computation and triangulation.

We obtain point trajectories by tracking pixels across frames following the per frame optical flow fields. Point trajectories are dense in space and can have various lengths depending on the occlusion frequency of the scene part they capture. Trajectory computation is bottom-up, oblivious to any object knowledge.

Trajectory affinities Point trajectories encode rich grouping information in their motion and depth differences. The depth channel helps differentiate targets that have the same apparent 2D motion but reside in different depths, e.g., targets that move perpendicular to the image plane of the camera.

We compute motion based trajectory affinities similar to the previous chapter and obtain:

$$\mathbf{A}_T^{\mathcal{M}}(i,j) = \exp\left(-\frac{d_{ij}\Delta u_{ij}}{\sigma}\right) \cdot \delta(T_i \cap T_j \neq \emptyset), \tag{4.1}$$

where d_{ij} is the maximum Euclidean distance between tr_i and tr_j and Δu_{ij} is the largest velocity difference between tr_i and tr_j during their time overlap:

$$\Delta u_{ij} = \max_{t \in T_i \cap T_j} \frac{|\vec{u}_t^i - \vec{u}_t^j|_2^2}{t_f}, \tag{4.2}$$

where $\vec{u}_t^i = (x_{t+t_f}^i - x_t^i, y_{t+t_f}^i - y_t^i)$ is the velocity of tr_i at time t. Similarly, we compute depth based trajectory affinities as:

$$\mathbf{A}_T^{\mathcal{D}}(i,j) = \exp\big(-\frac{z_{ij}\Delta w_{ij}}{\sigma_z}\big) \cdot \delta(T_i \cap T_j \neq \emptyset), \tag{4.3}$$

where z_{ij} is the maximum depth and Δw_{ij} is the largest Z velocity difference between tr_i and tr_j during their time overlap:

$$\Delta w_{ij} = \max_{t \in T_i \cap T_j} \frac{|\vec{w}_t^i - \vec{w}_t^j|_2^2}{t_f}, \tag{4.4}$$

where $\vec{w}_t^i = (z_{t+t_f}^i - z_t^i)$ is the Z component of the velocity of tr_i at time t. Notably, trajectory affinities in 4.3 can differentiate targets that have the same depth for a number of frames, as long as they reside in different depths for at least one frame during their time overlap. The final combined motion and disparity trajectory affinities take the form:

$$\mathbf{A}_T = \mathbf{A}_T^{\mathcal{M}} \bullet \mathbf{A}_T^{\mathcal{D}}, \tag{4.5}$$

where $\bullet$ denotes Hadamard product. In Figure 4.8 we show video segmentations using motion only, disparity only and motion plus disparity in trajectory affinities.

4.3.2 Coarse-grained detectlets

We define a detectlet dl_p to be a sequence of detector responses $\mathrm{dl}_p = \{(\mathrm{box}_p^t, c_p^t), t \in T_p\}$, where box_p^t is the detection bounding box at frame t, c_p^t is the corresponding detection score and T_p is the frame span of the detectlet. We define the confidence of detectlet dl_p to be the sum of confidences of its detection responses: $\mathbf{c}_p = \sum_{t \in T_p} c_p^t$.

We obtain detectlets by conservatively linking detections using trajectory anchoring, as shown in Figure 4.4. Let $\mathrm{box}_a = [x_a^{ul}\ y_a^{ul}\ x_a^{br}\ y_a^{br}]$, $\mathrm{box}_b = [x_b^{ul}\ y_b^{ul}\ x_b^{br}\ y_b^{br}]$ denote two detection responses in frames t_1, t_2 with $t_2 - t_1 = g \in \mathbb{N}^+$. Let $\mathcal{T}_a, \mathcal{T}_b$ denote the trajectory sets overlapping with each bounding box. We define the compatibility score between $\mathrm{box}_a, \mathrm{box}_b$ according to the similarity of the relative positions of the common trajectories inside the two boxes:

$$P^g(a,b) = \exp(-\frac{1}{\sigma^2} \operatorname*{median}_{\mathrm{tr}_i \in \mathcal{T}_a \cap \mathcal{T}_b} |(x_i^t - x_a^{ul}) - (x_i^t - x_b^{ul})|^2 + |(y_i^t - y_a^{ul}) - (y_i^t - y_b^{ul})|^2)\delta(\frac{|\mathcal{T}_a \cap \mathcal{T}_b|}{|\mathcal{T}_a \cup \mathcal{T}_b|} > 0.3). \tag{4.6}$$

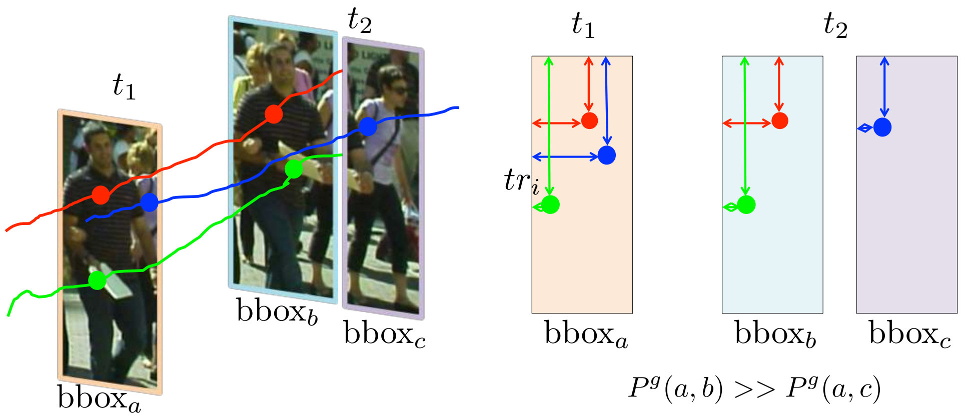


Figure 4.4: Detection linking into detectlets. Compatibility score of detection boxes box_a, box_b measures the stability of the relative position of their common point trajectories w.r.t. the upper left box corners.

We use only the upper left corner to measure trajectory relative positions, as shown in Figure 4.4. The upper boundary of a bounding box response is better anchored with respect to the target than the lower boundary, because head and torso is more stable configuration than legs.

For each $g \in \{1 \cdots 1.5\text{fps}\}$ we compute matrix P^g, encoding compatibilities between detection responses that reside g frames apart (fps denotes frames per second of the video sequence). For each g, the matrix summation $P^{g+} = P^1 + \cdots + P^g$ encodes compatibilities between detection responses that reside at most g frames apart. We apply a double threshold to each of the matrices $P^{g+}, g \in \{1 \cdots 1.5\text{fps}\}$, discarding detection pairs with compatibility score below a threshold or whose compatibility score is not 0.3 times larger that the second best competing detection pair, denoting ambiguity in association. Detection pairs that survive the double thresholding are linked into detectlets.

Detectlet repulsions We establish repulsions $\mathbf{R}_D \in \{0,1\}^{n_D \times n_D}$ between detectlets overlapping in time, expressing their inability to span the same object. Such incompatibilities are implicit in most previous approaches which never link detectlets that overlap in time. We have:

$$\mathbf{R}_D(p,q) = \delta(|T_p \cap T_q| > 0). \tag{4.7}$$

4.3.3 Trajectory to detectlet associations

We set associations $\mathbf{C} \in \{0,1\}^{n_D \times n_T}$ between detectlets and trajectories according to spatio-temporal overlap:

$$\mathbf{C}(p,i) = \delta(\forall t \in T_i \cap T_p,\ \ (x_i^t, y_i^t) \in \text{box}_p^t). \tag{4.8}$$

Computing associations between point trajectories and detectlets rather than between pixels and detections benefits from large trajectory horizon: It saves from erroneous associations between a detectlet and background trajectories or trajectories of nearby targets due to accidental per frame overlaps of detection bounding boxes. This is depicted in Figure 4.5 Right.

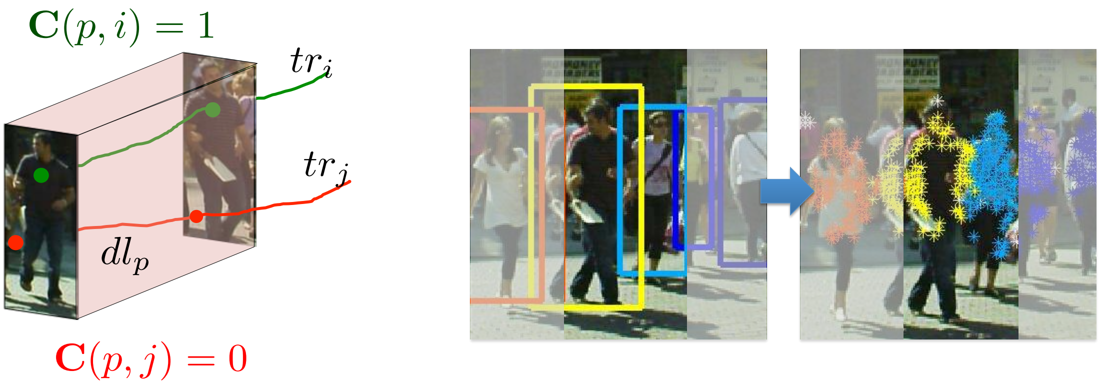


Figure 4.5: Trajectory to detectlet associations $\mathbf{C}$. Left: A trajectory is associated to a detectlet if it resides inside its bounding box for all their common frames. Right: We color trajectories according to their box color of the detectlets they are associated with. Trajectories associated with more that on detectlets or with none are shown in white. Thanks to the large trajectory lifespans, when people come close, spatial overlaps of the detectlet bounding boxes do not confuse associations in $\mathbf{C}$.

4.4 Two-view steering cut

We formulate multi-object tracking as classification-clustering in the joint detectlet and trajectory space. Each resulting co-cluster of detectlets and trajectories corresponds to one object hypothesis in space and time.

We establish what we call a "steered" trajectory graph by canceling motion/disparity affinities between trajectories associated with incompatible detectlets (Section 4.4.1). We show such link cancellation policy is robust to false alarms or spatially inaccurate detectlets: we vary false alarm detectlet rate and show steering corrects leaking affinities without disconnecting object interiors from trajectory associated to false alarms.

Different detectlets result in different graph steers. We propose a steering cut criterion that seeks for detectlet classification whose steering minimizes normalized cuts while maximizing alignment of clusters with detectlets (Section 4.4.2).

We will use the following notation:

$$\begin{aligned}
y &\in \{0,1\}^{n_D \times 1} && : \text{detectlet classification} \\
Y &\in \{0,1\}^{n_D \times K} && : \text{detectlet cluster indicator} \\
X &\in \{0,1\}^{n_T \times K} && : \text{trajectory cluster indicator,}
\end{aligned}$$

where K is the total number of clusters. The number of objects K is unknown (there is no one-to-one correspondence between detectlets and objects in the scene). It is part of our optimization variables.

4.4.1 Graph steering

We want to compute a trajectory graph that benefits from detector responses in order to correct leaking motion and disparity based affinities across objects moving similarly. Below we present construction of our steered graph as a function of detectlet classification y.

Repulsions between selected detectlets in $\mathcal{Y} = \{p, y_p = 1\}$ induce repulsions between their associated trajectories. Induced trajectory repulsions $\mathbf{R}_T(y) : y \to \{0,1\}^{n_T \times n_T}$ take the form:

$$\mathbf{R}_T(y) \quad = \delta\, (\exists (p,q) \text{ s.t. } y_p y_q \mathbf{R}_D(p,q)\mathbf{C}(p,i)\mathbf{C}(q,j)(\neg(\mathbf{C}(p,j)+\mathbf{C}(q,i)) = 1.) \tag{4.9}$$

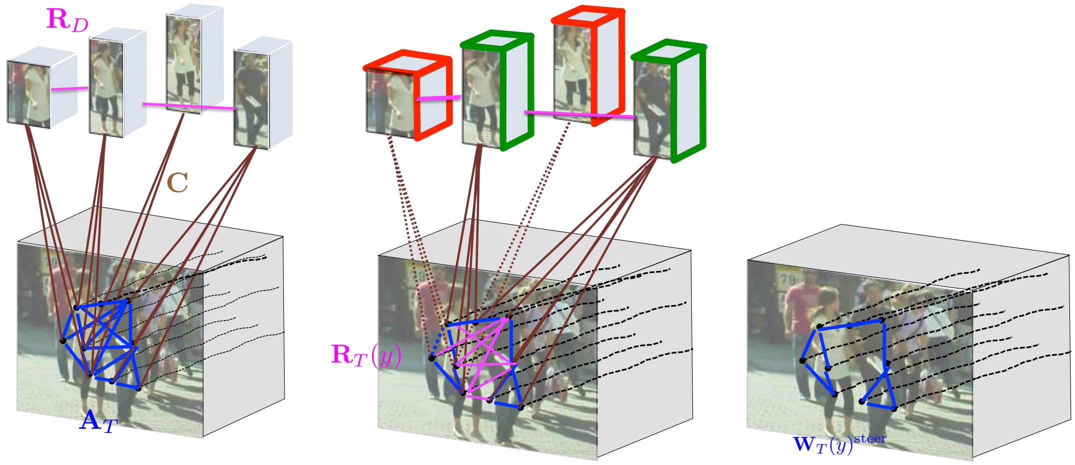
Intuitively, two trajectories have a repulsive weight if there is a pair of selected incompatible detectlets to which they are exclusively associated to, i.e., none of the two trajectories is associated with both detectlets. We visualize such affinity cancellation in Figure 4.6.

We define the steered affinity graph $\mathbf{W}_T^{\text{steer}}(y)$ to be the graph resulting from canceling motion affinities on repulsive trajectory links:

$$\mathbf{W}_T^{\text{steer}}(y) = (\mathbf{1}_{n_T \times n_T} - \mathbf{R}_T(y)) \bullet \mathbf{A}_T, \tag{4.10}$$

where $\bullet$ denotes Hadamard product.

The goal of graph steering is to improve connectivity in the trajectory graph by alleviating from leakages across objects captured by detections. However, if false alarms



Initial affinity graph

Induced trajectory repulsions

Steered graph

Figure 4.6: Graph steering. We show in green and red the selected and dis-selected detectlets in y. Repulsions are induced between trajectories associated with selected detectlets. Affinities are canceled on repulsive links. Notice that the trajectory claimed by both detectlets has unchanged affinities.

are erroneously classified as true positives in y, steering can potentially disconnect object interiors, harming graph connectivity.

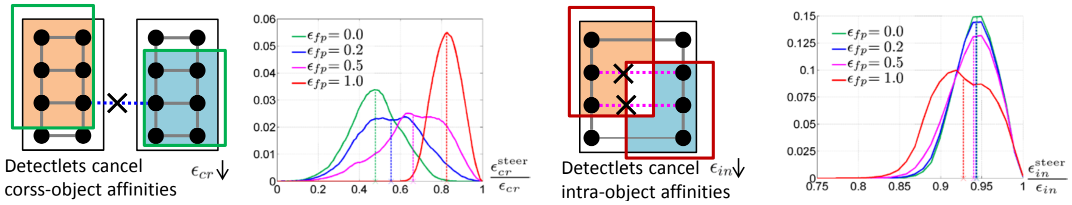


Figure 4.7: Steering and graph connectivity. Left: We assume a densely connected affinity graph between trajectories (shown in black dots) of two objects (shown in black boxes). Given a false alarm rate ϵ_{fp}, we sample two detectlets assuming true positive detectlets have intersection over union score at least 50% with the object they capture while false alarms at most 50%. Given the pair of sampled detectlets, we cancel trajectory affinities according to Eq. 4.9,4.10 and compute resulting $\epsilon_{cr}^{\text{steer}}$. Multiple detectlet samplings result in the distribution curve of $\frac{\epsilon_{cr}^{\text{steer}}}{\epsilon_{cr}}$. Right: Given one object we assume a densely connected affinity graph between trajectories. We sample pairs of detectlets and compute the steered affinities as before, for different false alarm rate ϵ_{fp}. We compute resulting $\epsilon_{in}^{\text{steer}}$. For each ϵ_{fp}, the decrease of cross-object leaking affinity rate ϵ_{cr} is much larger than the decrease of intra-object affinity ϵ_{in}. This is the case because false detectlets do not align well with object boundaries and cause fewer link cancellations.

We will analyze how the leaking affinity rate and the intra-object affinity rate change before and after steering of our affinity graph, while varying the false alarm detectlet rate in y. We assume for simplicity that the affinity graph $\mathbf{A}_T$ is binary. We will use the following notation:

ϵ_{in} : the probability of an intra-object link

ϵ_{cr} : the probability of a cross-object link

ϵ_{fp} : the probability an object is captured by a false alarm detectlet in y.

In Figure 4.7 we show the empirical distribution of $\frac{\epsilon_{cr}^{\text{steer}}}{\epsilon_{cr}}$, and $\frac{\epsilon_{in}^{\text{steer}}}{\epsilon_{in}}$ for different rates

ϵ_{fp}. The distributions are computed using simulations for detectlet generation assuming true positive detectlets have intersection over union score at least 50% with the object they capture while false alarms at most 50%. Ideally, we want $\frac{\epsilon_{cr}^{\text{steer}}}{\epsilon_{cr}} = 0$ (correcting all leakages) and $\frac{\epsilon_{in}^{\text{steer}}}{\epsilon_{in}} = 1$ (not harming any true, intra-object links). For each ϵ_{fp}, the decrease of cross-object leaking affinity rate ϵ_{cr} is much larger than the decrease of intra-object affinity ϵ_{in}. This means that steering improves affinity accuracy overall.

4.4.2 Steering cuts

Given a set of detectlets and point trajectories, we want to compute a joint partitioning (X, Y) so that resulting co-clusters correspond to the objects in our video scene. Columns of matrices (X, Y) correspond to trajectory and detectlet cluster indicators. We have the following constraint between detectlet clustering and detectlet classification:

$$\sum_{k=1}^{K} Y_k = y, \tag{4.11}$$

where Y_k denotes the kth column of Y. This means only detectlets selected in y (true positives) participate in the clustering. False alarm detectlets do not belong to any clusters.

We propose the following steering-cut criterion over detectlet classification y and co-clustering (X, Y):

Two-view Steering Cut:

$$\max_{y,X,Y,K}. \quad \sum_{k=1}^{K} \left(\underbrace{\frac{X_k^T \mathbf{W}_T^{\text{steer}}(y) X_k}{X_k^T \mathbf{D}_{\mathbf{W}_T^{\text{steer}}(y)} X_k}}_{coherence} \cdot \underbrace{\frac{Y_k^T \mathbf{C} X_k}{Y_k^T \mathbf{D}_{\mathbf{C}} Y_k}}_{alignment} \cdot \underbrace{Y_k^T \mathbf{c}}_{confidence} \right)$$

$$\text{s.t.} \quad \forall k,\ \frac{Y_k^T \mathbf{C} X_k}{Y_k^T \mathbf{D}_{\mathbf{C}} Y_k} > \mathrm{h}, \quad \forall k\ Y_k^T \mathbf{R}_D Y_k = 0, \quad \sum_{k=1}^{K} X_k \leq \mathbf{1}_{n_T}, \quad \sum_{k=1}^{K} \quad Y_k = y,$$

$$X \in \{0,1\}^{n_T \times K}, \quad Y \in \{0,1\}^{n_D \times K} \quad , y \in \{0,1\}^{n_D \times 1},$$

where $\mathbf{D_C} = \mathrm{Diag}(\mathbf{C}\mathbf{1}_{n_T})$. Cluster coherence is measured using the intra-cluster normalized affinities, same as in spectral clustering. Alignment between detectlets and trajectories in each co-cluster (X_k, Y_k) is measured by normalized intra-cluster associations. Cluster confidence is measured by the sum of detectlet confidence scores. The first constraint ensures that each co-cluster has a minimum alignment score h. The second constraint ensures that detectlets assigned to the same co-cluster are not repulsive. The third constraint ensures that each each trajectory is assigned to at most one cluster, since we do not want background trajectories to participate in the clustering. The forth constraint ensures only detectlets with positive values in y participate in the clustering.

In bottom-up segmentation methods there is an inherent model selection problem: coarser or finer partitionings minimize equally well the cut criterion. In our steering cut, alignment of trajectory clusters with detectlet clusters allows to pick the right segmentation granularity and reject over-fragmentations or leakages. The first constraint makes detectlets to compete in claiming point trajectories. This causes co-clusters with false alarm detectlets to be rejected by competing with better aligned and more confident ones. It also gives feedback from segmentation to detectlet classification: detectlets not aligning well with trajectory clusters are discarded (classified as false alarms).

We approximately optimize our steering cut cost function with multiple (steered) segmentations. We sample y according to detectlet confidence $\mathbf{c}$ and compute normalized cut clustering in the steered graph $\mathbf{W}_T^{\text{steer}}(y)$:

$$\begin{aligned} \max_{X}. \quad & \sum_{k=1}^{K} \left(\frac{X_k^T \mathbf{W}_T^{\text{steer}}(y) X_k}{X_k^T \mathbf{D}_{\mathbf{W}_T^{\text{steer}}(y)} X_k} \right) \\ \text{s.t.} \quad & \sum_{k=1}^{K} X_k = \mathbf{1}_{n_T}, \quad X \in \{0,1\}^{n_T \times K}. \end{aligned} \tag{4.12}$$

This results in a large pool of trajectory clusters. For each trajectory cluster X_l, we greedily select detectlets with highest to lowest alignment scores and reject detectlets incompatible with already selected ones. For each resulting co-cluster (X_l, Y_l), we measure

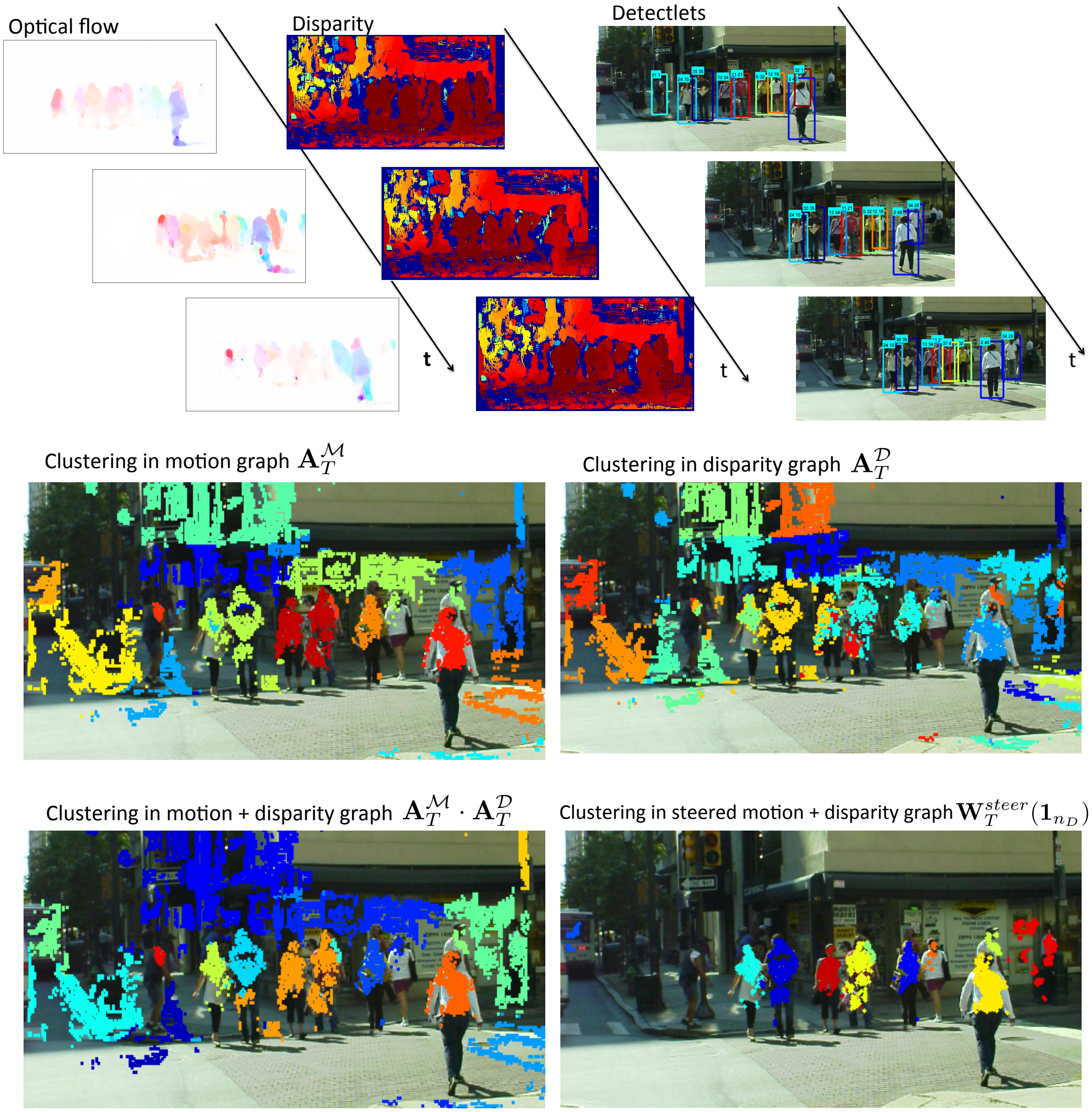

Optical flow
Disparity
Detectlets
t
t
t
Clustering in motion graph $\mathbf{A}_T^{\mathcal{M}}$
Clustering in disparity graph $\mathbf{A}_T^{\mathcal{D}}$
Clustering in motion + disparity graph $\mathbf{A}_T^{\mathcal{M}} \cdot \mathbf{A}_T^{\mathcal{D}}$
Clustering in steered motion + disparity graph $\mathbf{W}_T^{steer}(\mathbf{1}_{n_D})$

Figure 4.8: Disparity based affinities are often not informative for targets residing far from the camera. Longer trajectories can better differentiate targets than shorter ones. In Bottom Left we show steering cut only on trajectories captured by some detectlet, as explained in Section 4.4.3.

alignment score $\frac{Y_l^T \mathbf{C} X_l}{Y_l^T \mathbf{D_C} Y_l}$ and confidence score $\mathbf{c}^T Y_l$. We prune co-clusters whose alignment score is below a threshold h. Co-cluster score is measured by $\frac{Y_l^T \mathbf{C} X_l}{Y_l^T \mathbf{D_C} Y_l} \cdot Y_l^T \mathbf{c}$. We obtain the tracking solution by sequentially choosing highest to lowest scoring co-clusters (X_l, Y_l), rejecting those that have non zero trajectory intersection with already chosen ones.

Our resulting co-clusters (X_l, Y_l) terminate at full occlusions because trajectories before and after full occlusions do not overlap in time and thus have zero affinities in both $\mathbf{A}_T$ and $\mathbf{W}_T^{\text{steer}}$. To link co-clusters through full occlusions we compute compatibility scores between all pairs of co-clusters depending on first order motion smoothness. We link co-clusters whose compatibility score survives a double thresholding policy, described in Section 4.3.2.

Multiple steered segmentations versus multiple segmentations The proposed graph steering framework incorporates detection information early, in the segmentation graph $\mathbf{W}_T^{\text{steer}}$. In contrast, multiple segmentation approaches such as Russell et al. (2006), sample a number of segmentation proposals to be post processed with an object model. This does not allow to recover from mistakes of bottom-up grouping affinities.

Graph steering versus co-clustering

Clustering in the joint space of detections and image pixels has been considered in Yu et al. (2002) for simultaneous detection and segmentation in static images. The authors bypass explicit object hypotheses classification by assigning false alarms to a background cluster and compute a clustering in the joint matrix:

$$\mathbf{A} = \left[\mathbf{A}_T \;\; \mathbf{C}^T \, ; \, \mathbf{C} \;\; \mathbf{A}_D\right], \tag{4.13}$$

where $\mathbf{A}_D$ stands for detectlet affinities. Problems of spectral clustering in this augmented graph come from false alarm detectlets and affinity contradictions between trajectories and detections:

1. *False associations.* Assigning a false detectlet to the background cluster needs to cut association edges between the false detectlet and its associated in $\mathbf{C}$ point trajectories (overlapping with it). Such cut cost may be prohibitively large and can confuse the solution, as shown in Figure 4.9 *Left*.

2. *Affinity contradictions*. In places detectlet and trajectory graphs disagree, incorrect affinities confuse the solution, as shown in Figure 4.9 *Right*.

The above problems stem from the fact that co-clustering is computed in the initial affinity graph $\mathbf{A}$ and are eliminated in the proposed graph steering framework where the affinity graph changes according to detectlet classification.

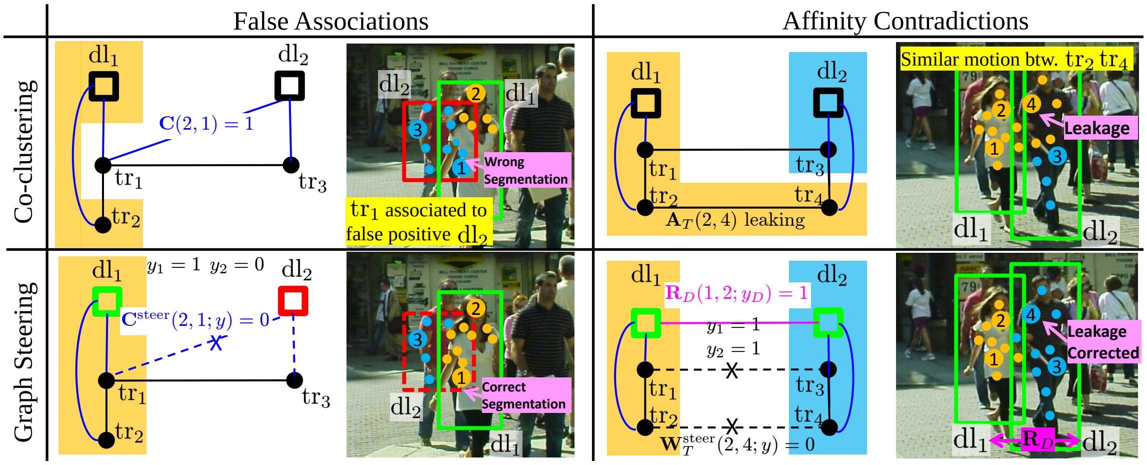


Figure 4.9: Graph steering versus co-clustering. Left: Only selected in y detectlets, here dl_1, can claim trajectories. In this way, we avoid

contaminating the spectral partitioning solution with false detectlets (here dl_2). Right: We cancel trajectory affinities ($\mathbf{A}_T(2,4)$, $\mathbf{A}_T(1,3)$) between incompatible detectlets ($\mathrm{dl}_1, \mathrm{dl}_2$). Spectral partitioning in the steered graph does not leak across similarly moving targets.

4.4.3 Graph steerability

We assume again for simplicity that the affinity graph $\mathbf{A}_T$ is binary. We will use the following notation:

ϵ_{in} : the probability of an intra-object link

ϵ_{cr} : the probability of a cross-object link

ϵ_{fg} : the probability of a figure-ground link

ϵ_{gg} : the probability of a ground-ground link

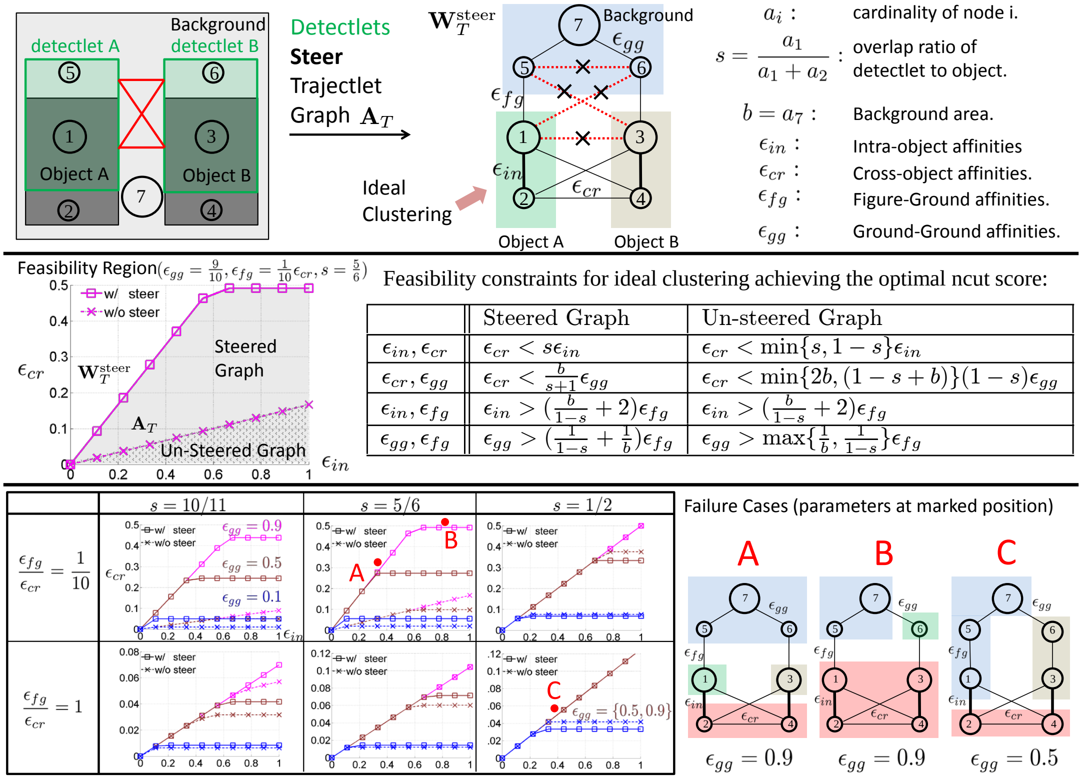
We have already discussed how graph steering impacts the connectivity of $\mathbf{W}_T^{\text{steer}}$. In this section, we analyze how graph steering impacts the normalized cut of the resulting graph $\mathbf{W}_T^{\text{steer}}(y)$. There are three challenging cases:

1. Case A: Intra-object connections are only marginally larger than cross-object ones. This can happen due to similar motion/disparity across objects, and large deformation (articulated motion) within the objects. If the repulsion induced by detectlets is too weak (due to large misalignment with the object they capture), the segments will still leak across the two objects. If the induced repulsion is strong but the detectlets do not align well with the objects they capture then the objects will be fragmented into two parts, one for each repulsive region, while the parts outside the detectlets (but on the objects) will still leak across.

2. Case B: Misalignment of detectlets with underlying objects induces wrong repulsions between background regions and over-fragments the background if ground-ground affinity rate ϵ_{gg} is low.

3. Case C: Figure-ground affinity rate ϵ_{fg} is large and foreground objects leak to the background. Steering cannot correct figure-ground leakage since repulsions are designed to cancel only cross-object affinities.

To quantify feasibility of Steering Cut under these challenges, we analyze the simplified case of a 7 node graph shown in Figure 4.10 Top. Each node corresponds to a trajectory set. The desired clustering X^{ideal} assigns $\{1,2\}$ and $\{3,4\}$ in two foreground objects, and $\{5,6,7\}$ in the background group. Nodes $\{5,6\}$ are background nodes mistakenly declared as foreground by detectlet, due to detectlets' misalignments. Detectlets A



Feasibility constraints for ideal clustering achieving the optimal ncut score:

	Steered Graph	Un-steered Graph
$\epsilon_{in}, \epsilon_{cr}$	$\epsilon_{cr} < s\epsilon_{in}$	$\epsilon_{cr} < \min\{s, 1-s\}\epsilon_{in}$
$\epsilon_{cr}, \epsilon_{gg}$	$\epsilon_{cr} < \frac{b}{s+1}\epsilon_{gg}$	$\epsilon_{cr} < \min\{2b, (1-s+b)\}(1-s)\epsilon_{gg}$
$\epsilon_{in}, \epsilon_{fg}$	$\epsilon_{in} > (\frac{b}{1-s}+2)\epsilon_{fg}$	$\epsilon_{in} > (\frac{b}{1-s}+2)\epsilon_{fg}$
$\epsilon_{gg}, \epsilon_{fg}$	$\epsilon_{gg} > (\frac{1}{1-s}+\frac{1}{b})\epsilon_{fg}$	$\epsilon_{gg} > \max\{\frac{1}{b}, \frac{1}{1-s}\}\epsilon_{fg}$

Figure 4.10: Feasibility analysis of steering cut. Top: A simplified graph for 7 tracklets nodes. Nodes $\{1,2\}$, $\{3,4\}$ are two foreground objects, $5, 6$ are background imposters defined by detectlets A, B, 7 is a background distractor. A, B cancel links between nodes $1, 5$ and $3, 6$ in the steered graph. Middle Right: Conditions on $\epsilon_{in}, \epsilon_{cr}, \epsilon_{fg}, \epsilon_{gg}$ for the desired clustering achieving the optimal ncut score in the steered (shaded area) and non-steered (textured area) graph. Middle Left: Plots of the feasibility region of ϵ_{in}, ϵ_{cr} for the steered comparing with the initial graphs. Bottom Left: Feasibility regions relating cross-object and intra-object affinity rates ϵ_{in},ϵ_{cr} for different detectlet / object overlap s and different affinity ratios $\frac{\epsilon_{fg}}{\epsilon_{cr}}$. Bottom Right: Failure cases outside of feasibility regions corresponding to the three challenges outlined. The size of the nodes indicates their cardinality.

and B cancel graph links between nodes $1, 5$ and $3, 6$ (shown in red) in the steered graph, $\mathbf{W}_T^{\text{steer}}$, in Figure 4.10 Top.

Let $a_1 ... a_7$ denote the cardinality of the trajectory sets corresponding to the 7 nodes. Let $s = \frac{a_1}{a_1+a_2}$ denote detectlet to object overlapping ratio and let b denote area of background node a_7. We analytically derive the conditions for the ideal segmentation X^{ideal} achieving the optimal value of $\text{ncut}(\mathbf{W}_T^{\text{steer}}, X^{\text{ideal}})$ as a function or affinity link rates $\epsilon_{cr}, \epsilon_{in}, \epsilon_{fg}, \epsilon_{gg}$, object-detectlet alignment s and background area b. There are number of these conditions. A simplified set of feasibility conditions are stated in the table shown in Figure 4.10 Middle. We plot a subset of the feasibility regions for ϵ_{in} and ϵ_{cr} for different values of s ground-ground connections ϵ_{gg} and different ratios of figure-ground and cross-object affinity rates $\frac{\epsilon_{fg}}{\epsilon_{cr}}$ in Figure 4.10 Bottom. The feasibility conditions of Steering Cut lead to three conclusions:

- **Steering improves feasibility margin between ϵ_{cr} and ϵ_{in} affinity rates,** as shown by the steeper slope of the linear part of the curves for the steered segmentation case compared to non-steered one, in Figure 4.10 Bottom Left. Ratio of cross-object over intra-object affinity rates $\frac{\epsilon_{cr}}{\epsilon_{in}}$ is upper-bounded by detectlet object overlap:

  $$\frac{\epsilon_{cr}}{\epsilon_{in}} < s. \tag{4.14}$$

  The larger the overlap s, the stronger the steering between objects and the larger the feasibility region. In the typical case where $\frac{\epsilon_{fg}}{\epsilon_{cr}} = \frac{1}{10}$, $\epsilon_{gg} = 0.9$, $s = \frac{5}{6}$ the feasible ratio $\frac{\epsilon_{cr}}{\epsilon_{in}}$ in the steered graph is 5 times larger over the unsteered one whose slope is $1 - s$, as shown in Figure 4.10 Middle Left. The assumption of small figure-ground over cross-object affinity ratio $\frac{\epsilon_{fg}}{\epsilon_{cr}}$ is justified since motion/depth difference between objects and background is larger than motion/depth difference across objects, that often move in coherent groups. Also, coherence of background motion and depth justifies large ground-ground affinity rate $\epsilon_{gg} = 0.9$.

- **Steering is limited by background incoherence**. Cross-object affinity rate is upper bounded by total background connectivity $b\epsilon_{gg}$:

$$\epsilon_{cr} < \frac{b}{s+1}\epsilon_{gg}. \tag{4.15}$$

This causes the saturated part of the feasibility curves for both the steered and unsteered segmentation case. In case of graph steering, saturation is caused by *wrong repulsions* introduced between background nodes $5, 6$ due to imperfect alignment of the detectlets with the corresponding objects, causing background over-fragmentation. The larger the background connectivity $b\epsilon_{gg}$ over the overlap ratio s, the later we reach the saturation point. Moreover, if $\frac{b}{s+1}\epsilon_{gg} < s$ we never reach the saturation point, as is the case for the right most curves in magenta in Figure 4.10 Bottom.

- **Steering is limited by average figure-ground leakage** ϵ_{fg}. Large figure-ground leaking rate ϵ_{fg} dramatically decreases the ability of both steered and unsteered graph to generate the right segmentations. In that case, graph steering improves by a smaller margin over segmentation in the unsteered graph as shown in Figure 4.10 bottom row. This is expected since steering is designed to deal with object-object and not figure-ground entanglement.

We revisit the challenges outlined, relate them to points on the feasibility plots and discuss corresponding failure cases A, B and C depicted in Figure 4.10 Bottom. Cases A and B have the same overlap ratio s and average ground-ground connectivity ϵ_{gg}. Case A resides close to the linear part of the curve while B is at its saturation point. In A objects are over-fragmented because intra-object affinities are weak relatively to cross-object affinities ϵ_{cr} and fail to piece the objects as a whole. Increasing intra-object affinity ϵ_{in} can recover the right clustering. In case B, background over-fragments and foreground under-segments. This is due to background incoherence and imposters 5 and 6 create more salient clusters. Background fragmentation cannot be corrected by increasing intra-object affinity ϵ_{in}. In case C, the object leaks to the background, independently of the amount of steering between object nodes. This happens because of large figure-ground affinity rate ϵ_{fg}, equal to the cross-object affinity rate ϵ_{cr}.

Findings of our theoretical analysis have been verified empirically. High figure-ground affinity rate limits the effectiveness of our steering framework. We encounter high figure-ground affinities in cases of standing, not moving pedestrians. In our implementation we recover from this problem by computing steered clustering only for trajectories that are associated to some detectlet, and discard (background) trajectories not associated to any of them. We show the resulting clustering for this set of trajectories in Figure 4.8 Bottom Left. The few trajectories associated to false alarms on the background do not create a problem. We further compute one bottom-up clustering (using unsteered affinities in $\mathbf{A}_T$) for the full set of trajectories, for capturing targets with very few detections on them but who move saliently. Resulting clusters from both partitionings populate our cluster pool and are matched against detectlets to produce resulting co-clusters.

4.5 Experiments

We test two-granularity tracking in the following established datasets for multi-object tracking: 1) TUD crossing of Andriluka et al. (2008), 2) PETS 2009 dataset, scenario S2.L1 , 3) ETH sunny-day dataset used in Ess et al. (2007), 4) AVSS AB Hard, part of i-Lids dataset for AVSS 2007 (International Conference on Advanced Video and Signal based Surveillance), and 5) our own UrbanStreet dataset. The datasets vary with respect to camera motion (static, translating, zoom in-zoom out), object scale, degree of target entanglement, objects' speed.

In all datasets apart from UrbanStreet we use a single camera view. We discard stereo or multiview information available in ETH sunny day and PETS respectively, to be in accordance with most previous works tested on these datasets. For PETS, ilids and ETH sunny day datasets we use the pre-trained detector of Felzenszwalb et al. (2010). For TUD crossing and UrbanStreet we use the pre-trained poselet detector of Bourdev et al. (2010),

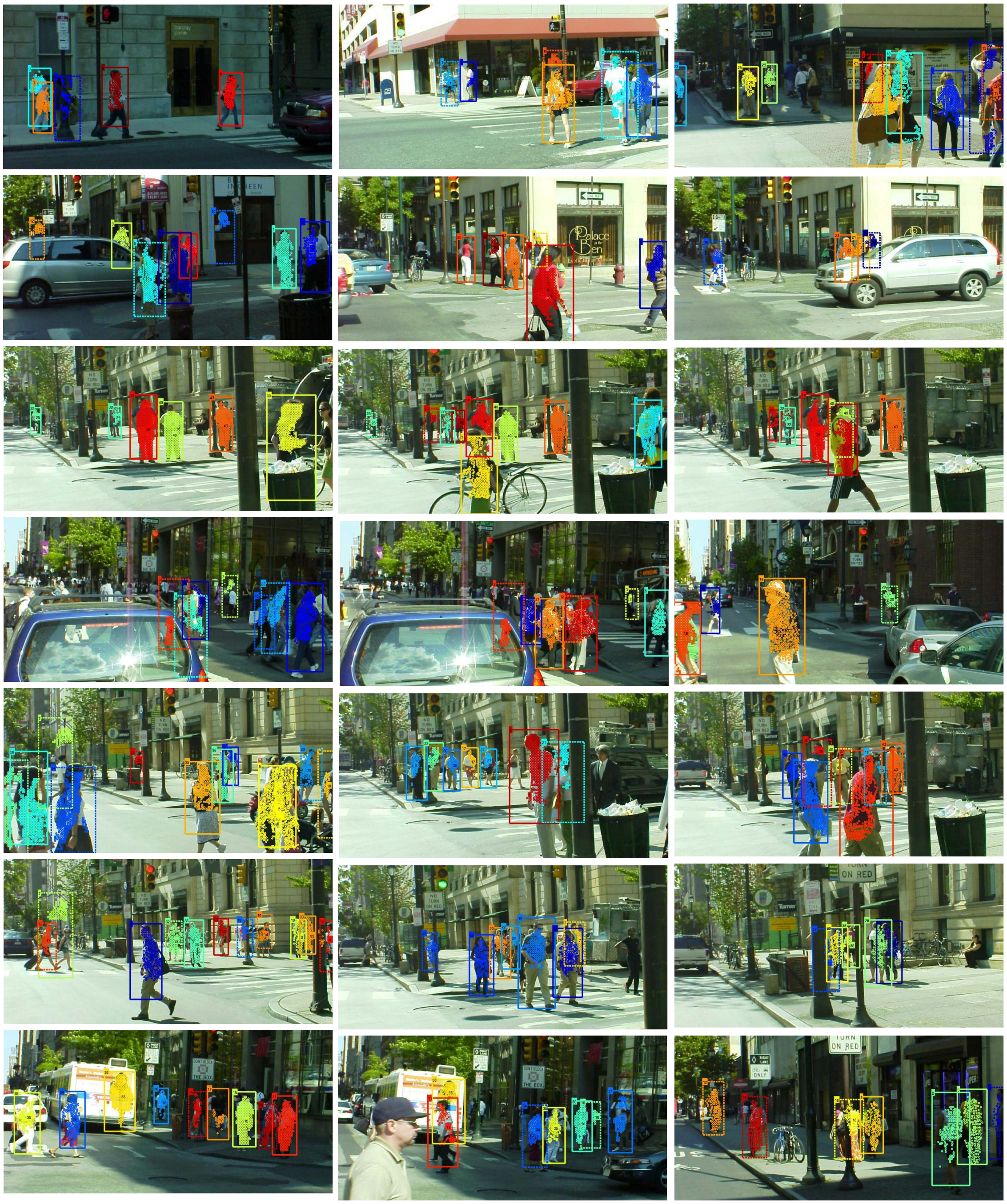


Figure 4.11: Two-granularity tracking results in UrbanStreet. Point trajectory clusters adapt to the changing target visibility masks during partial occlusions.

mainly because people that appear close to the camera are often only half visible. The camera is static in PETS, TUD-crossing and iLids and moving in ETH sunny-day and UrbanStreet. For TUD-crossing, PETS S2.L1 and ETH sunny-day we used groundtruth provided by the authors of Yang and Nevatia (2012). Groundtruth is not available for AVSS AB Hard and we show only qualitative results.

UrbanStreet dataset contains 17 video sequences taken from a stereo rig mounted on a car traversing the central streets of Philadelphia during rush hour. We supply segmentation masks for all visible targets every four frames (0.6 seconds) in each sequence, with a total of 2500 pedestrian masks labeled. Groundtruth samples are shown in Figure 4.13. The sequences contain a wide variability of target size, motion and entanglement, and the camera may be stationary (when the car is stopped on a cross-road) or moving. We evaluate performance both with and without the disparity channel. We use the disparity channel in two ways: 1) we prune out-of-perspective detections, assuming the mean human height to be 1.7m, 2) we compute trajectory affinities $\mathbf{A}_T$ from maximum disparity difference between point trajectories, as discussed in Section 4.3.1. We compute disparity by a 4-way dynamic programming on SIFT descriptor matching scores. Due to our large baseline, the resulting disparity fields are coarse and mostly informative for targets close to the camera.

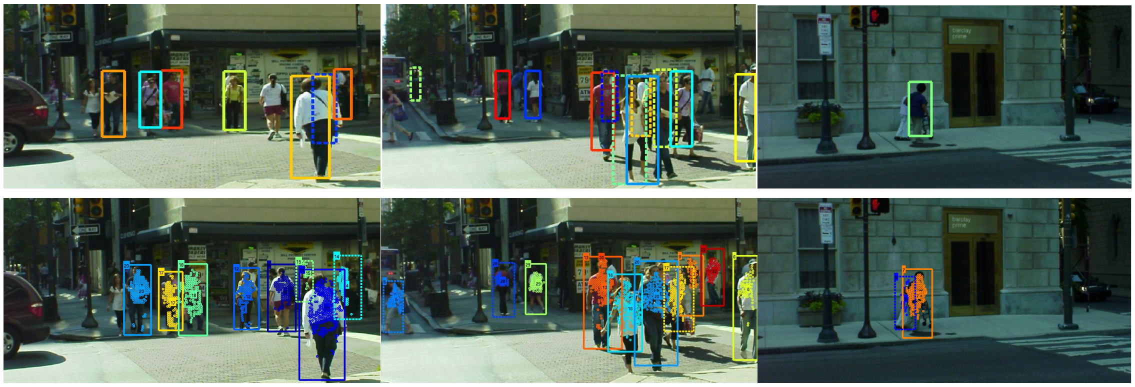


Figure 4.12: Top: tracking-by-detection of Gong et al. (2011). It interpolates across occlusion and miss detection gaps. The interpolated boxes are shown with dashed line. Bottom: two granularity tracking. The detectlet-trajectory co-clusters have accurate grounding of the targets during partial occlusions.

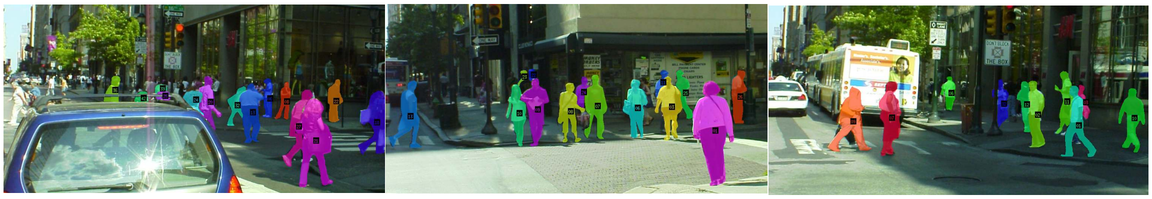


Figure 4.13: Groundtruth segmentations in UrbanStreet.

We measure tracking performance using the CLEAR MOT metrics described in Bernardin and Stiefelhagen (2008). In each frame, we compute intersection over union score between the box hypotheses of our two-granularity tracker and groundtruth boxes. We compute a greedy one-to-one assignment between box hypotheses and groundtruth boxes whose intersection over union score is above 50%. Groundtruth boxes not assigned to any hypotheses are counted as miss detections (false negatives) and box hypotheses not assigned to any groundtruth are counted as false positives. A hypothesized box can be a false positive either because it has less than 50% intersection over union score with all ground-truth boxes of its frame or because another hypothesis is assigned to the ground-truth box with which it overlaps well. We report true positive, false positive and false negative rates, which are the number of true positives, false positives and false negatives over the total number of groundtruth boxes. An identity switch is reported whenever a tracklet is assigned to different ground-truth tracklets in consecutive frames. A fragmentation is reported whenever a groundtruth tracklet is assigned to different tracklet hypotheses in consecutive frames. Finally, CLEARMOT precision is defined as the average intersection over union score of the true positive hypotheses. We did not find a script publicly available for evaluating tracking performance. Specifically, the publicly available script supplied from authors of Bagdanov et al. (2012) that implements the CLEARMOT evaluation, 1) misses identity switches that occur when a tracker drifts from one object to another during termination of a groundtruth track and 2) does not differentiate between fragmentations and identity switches, which is important for analyzing tracking performance. We use our own evaluation code by adapting the script of Bagdanov et al. (2012) accordingly. We add groundtruth detections for heavily occluded people or people close to the image borders that are often missing from

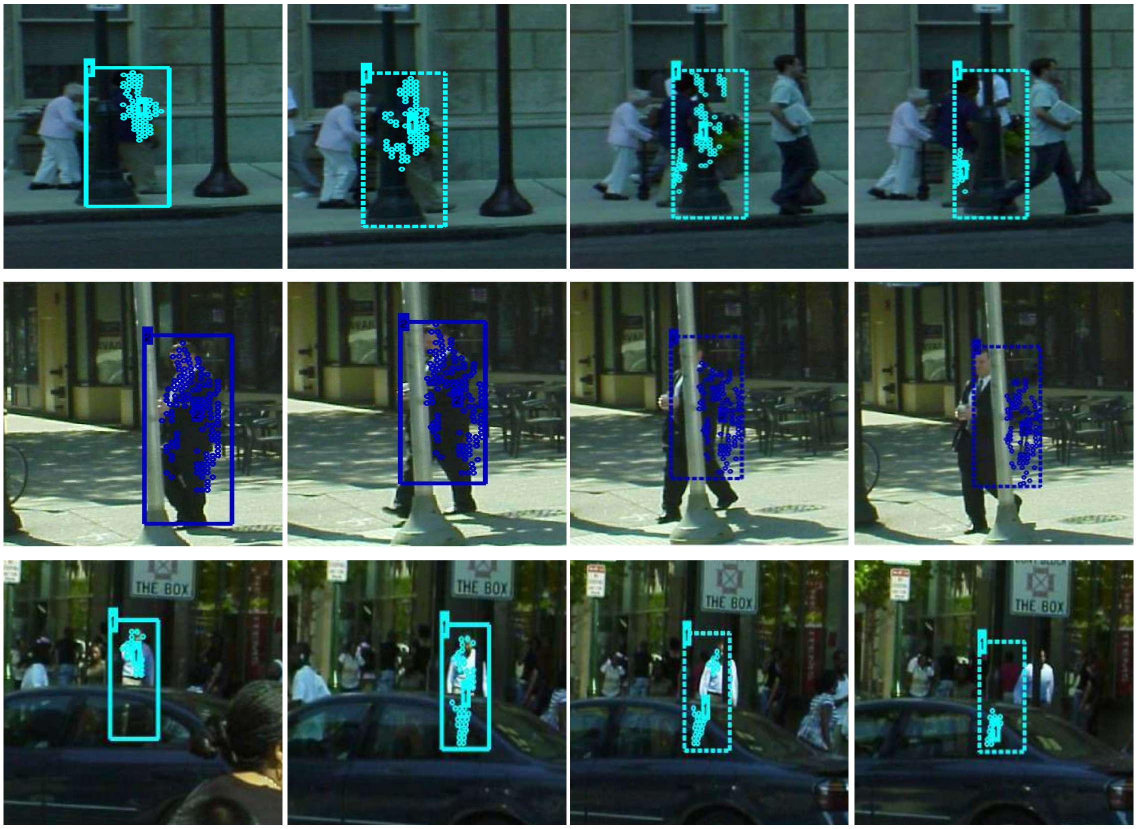


Figure 4.14: Drifting in two-granularity tracking. 1st, 2nd Rows: The optical flow trajectories drift from the lady and the man to the street lamp. Low temporal resolution of UrbanStreet (6fps) sometimes causes optical flow drifting during target occlusions. Trajectory drifting is decreased by making the forward-backward flow consistency check stricter, which though results in sparser trajectory coverage, especially for fast moving objects that are close to the camera. 3rd Row: The tracklet drifts from the man in white to part of the car. This leaking trajectory cluster is proposed during our multiple segmentations in $\mathbf{W}_T^{\text{steer}}(y)$. It happens to align well with the detection responses shown in solid line. Finer clustering separates the two sets of point trajectories. Establishing detectlet affinities between non overlapping in time detectlets can reject those obvious cases of bad trajectory clusters.

the ground truth of Yang and Nevatia (2012). Finally, we visually inspect the identity switches and fragmentations (as advised also by Bagdanov et al. (2012)) since the greedy

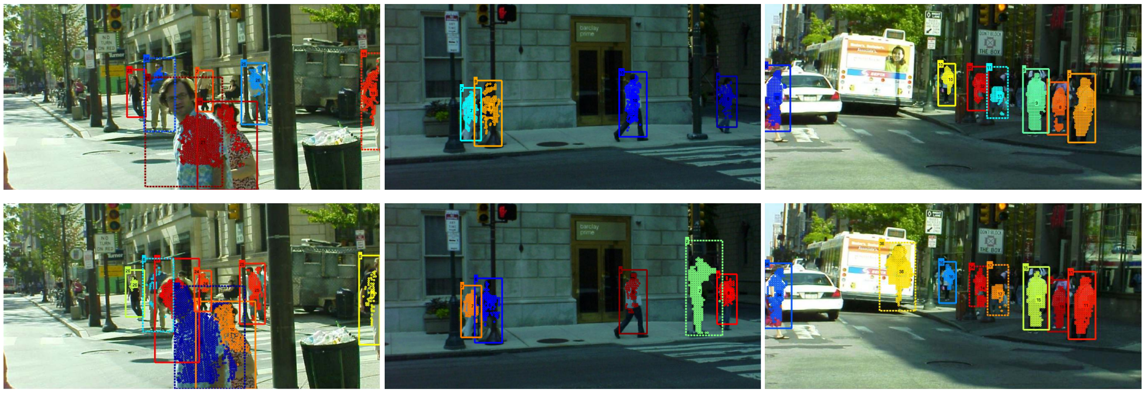


Figure 4.15: Comparison of 3D (top row) and 2D (bottom row) two-granularity tracking in UrbanStreet. Depth channel enables pruning out-of-perspective detections and discard the false alarms shown in green and yellow in the 2nd and 3rd columns of bottom row. In case of similarly moving targets under occlusions, such as the couple in the 1st column, both motion based affinities and detection driven repulsions are weak. Disparity based affinities can differentiate the targets due to depth difference between the occluder and occludee. Due to the coarseness of our disparity fields, disparity based trajectory affinities are informative mostly for targets close to the camera.

assignment often makes mistakes under closely interacting targets. For UrbanSteet we fit tight boxes to segmentation masks of each target and use 30% (instead of (50%)) as the cutoff threshold of intersection over union score. We show quantitative results in Table 4.1 and qualitative results in Figures 4.11, 4.18, 4.19, 4.20, and 4.21.

We compare with the following baseline systems:

1. Two-granularity tracking with the initial affinity graph $\mathbf{A}_T$ instead of $\mathbf{W}_T^{\text{steer}}$. We call this baseline *nosteering*.

2. Two granularity tracking using the following set of coclusters: $(X_k, Y_k), Y_k = \epsilon_k, X_k = C(k,:)^T$, where e_k is an n_D long vector of zeros with one at kth position. That is, we pair each detectlet with its associated with it trajectories. We call this baseline *detectlets*.

3. We pair each detectlet with its associated with it trajectories and propagate the detections using trajectory anchoring. We call this baseline *trajectory classification*.

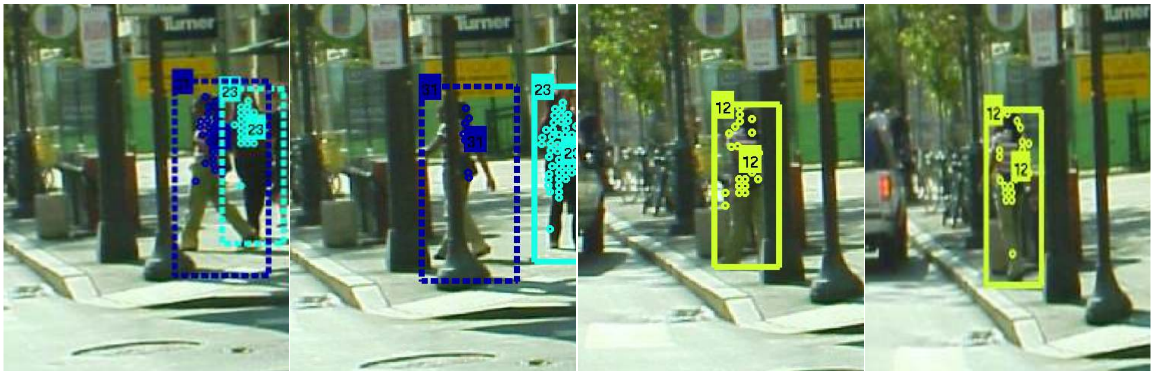


Figure 4.16: Tracklet fragmentations. The trajectory/detectlet co-clusters terminate at full occlusions since there are no affinities between trajectories that do not overlap in time. We link co-clusters conservatively through full occlusions based on motion smoothness. In cases the target's motion before and after the occlusion is not similar enough, corresponding co-clusters are not linked which results in fragmentations. Here, the lady's motion changes after the occlusion: she stops to wait for the car to pass by.

Dataset	T.Pos.(%)	F. Pos. (%)	ID-switch	Fragment.	Precision
TUD-crossing	90.75	1.09	2	2	74.00
PETS S2.L1	94.56	47.00	1	21	73.80
Sunny day (ETH)	70.90	4.26	0	6	77.18
UrbanStreet2D	63.50	14.80	2	40	67.60
UrbanStreet3D	64.34	11.55	1	44	67.40
UrbanStreet3D baseline-detectlets	48.62	5.03	1	69	71.84
UrbanSteet3D baseline-nosteering	30.70	1.46	5	15	63.37
UrbanSteet3D baseline-trclassification	66.12	55.06	17	56	65.02

Table 4.1: Two-granularity tracking results in CLEAR MOT metrics.

Baseline nosteering has the lowest false alarm rate. This indicates the non-accidentalness of alignment between trajectory clusters with detectlets. False alarm detectlets are rejected due to mis-alignment with the underlying trajectory motion/disparity organization. Such alignment resembles the non-additive grouping based verification advocated in the early work of Amir and Lindenbaum (1998). On the other hand, nosteering baseline persistently

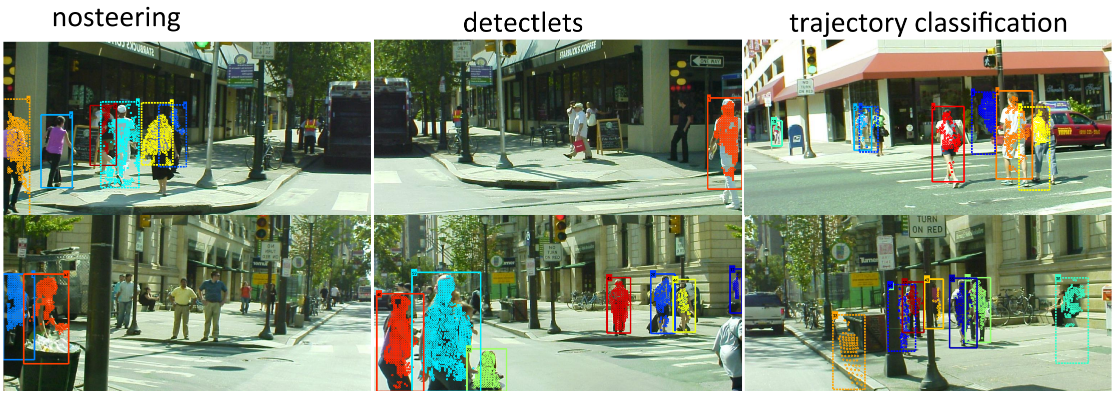


Figure 4.17: Two-granularity baseline systems. Nosteering persistently misses stationary pedestrians and reliably captures distinctly moving ones. Detectlets misses people under deformation or partial occlusions. Trajectory classification suffers from propagation based on wrongly associated with detectlets background trajectories.

misses stationary pedestrians far away from the camera, whose disparity difference with the surroundings is non distinct. This results in very low true positive rate.

Detectlets baseline has very few identity switches thanks to our conservative detection linking policy. The lower true positive rate in comparison to our full system is due to miss detections on deformed or occluded pedestrians. Notice that our detectlets already do a trajectory based interpolation across miss detection gaps that are shorter than 1 second. The large number of fragmentations shows that detectlets are too short for our motion smoothness based long range co-cluster linking to bridge the miss detection gaps. Comparing our false alarm rate with the detectlet baseline shows that our two-granularity co-clusters often amplify in time a false alarm that would otherwise have short temporal framespan.

Trajectory classification baselines suffers from large number of drifting co-clusters. This shows that 1) our detectlet-trajectory cross-associations **C** often associate detectlets with spurious background trajectories, and 2) trajectory affinities clustering can isolate drifting from non drifting trajectories. Qualitative results for our baselines systems are shown in Figure 4.17.

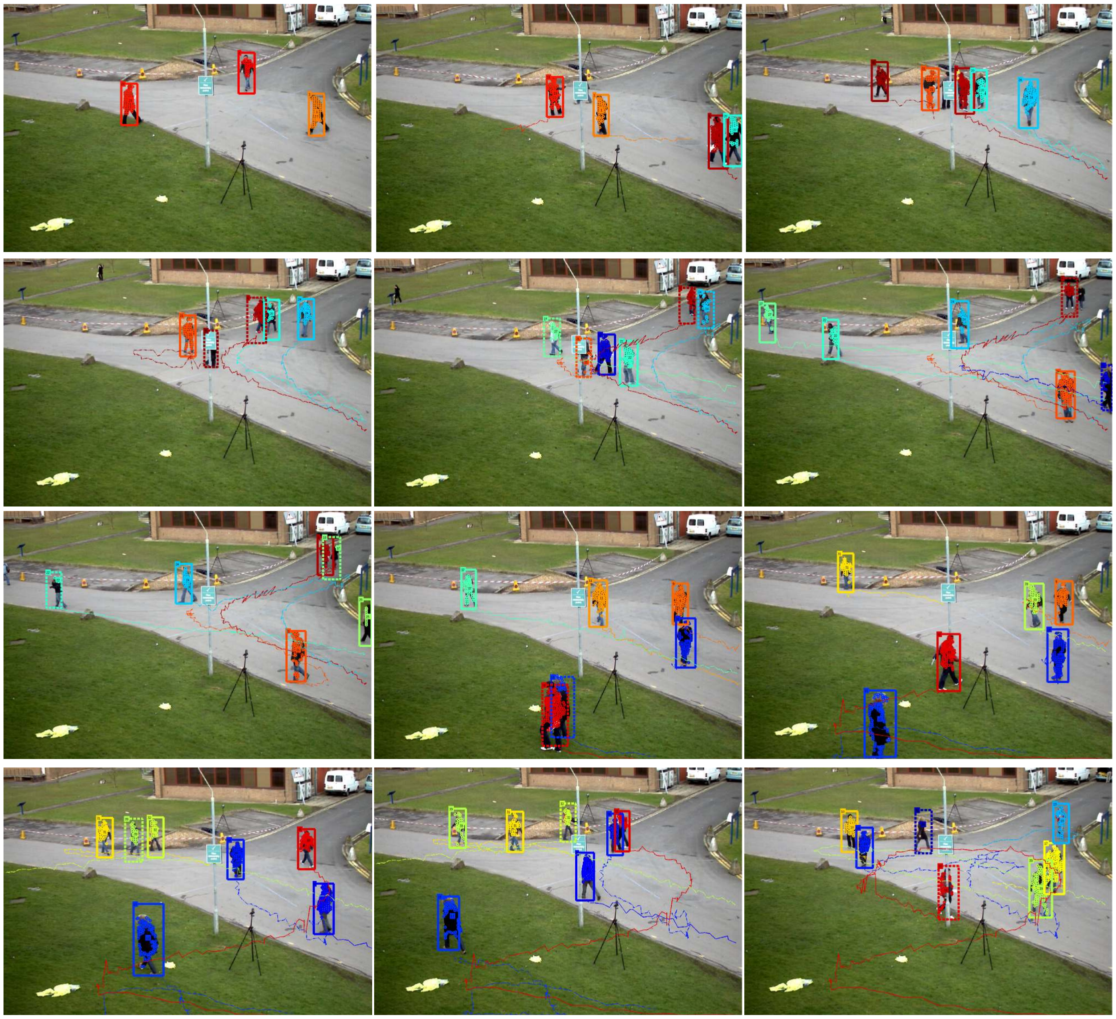


Figure 4.18: Two-granularity tracking results in PETS S2.L1 dataset. We use only the first camera view (from the 8 available). Point trajectories terminate at cross-target occlusions and when targets walk behind the lamp. The trajectory-detectlet co-clusters are linked though full occlusions using motion smoothness. Incorporating target appearance models during linking across full occlusions can help in cases of non smooth motion and result in less tracklet fragmentations. In PETS S2.L1 the camera is static and background subtraction is used to discard background trajectories and decrease computational times. Figure-ground trajectory classification can be used in cases of static camera and long footage.

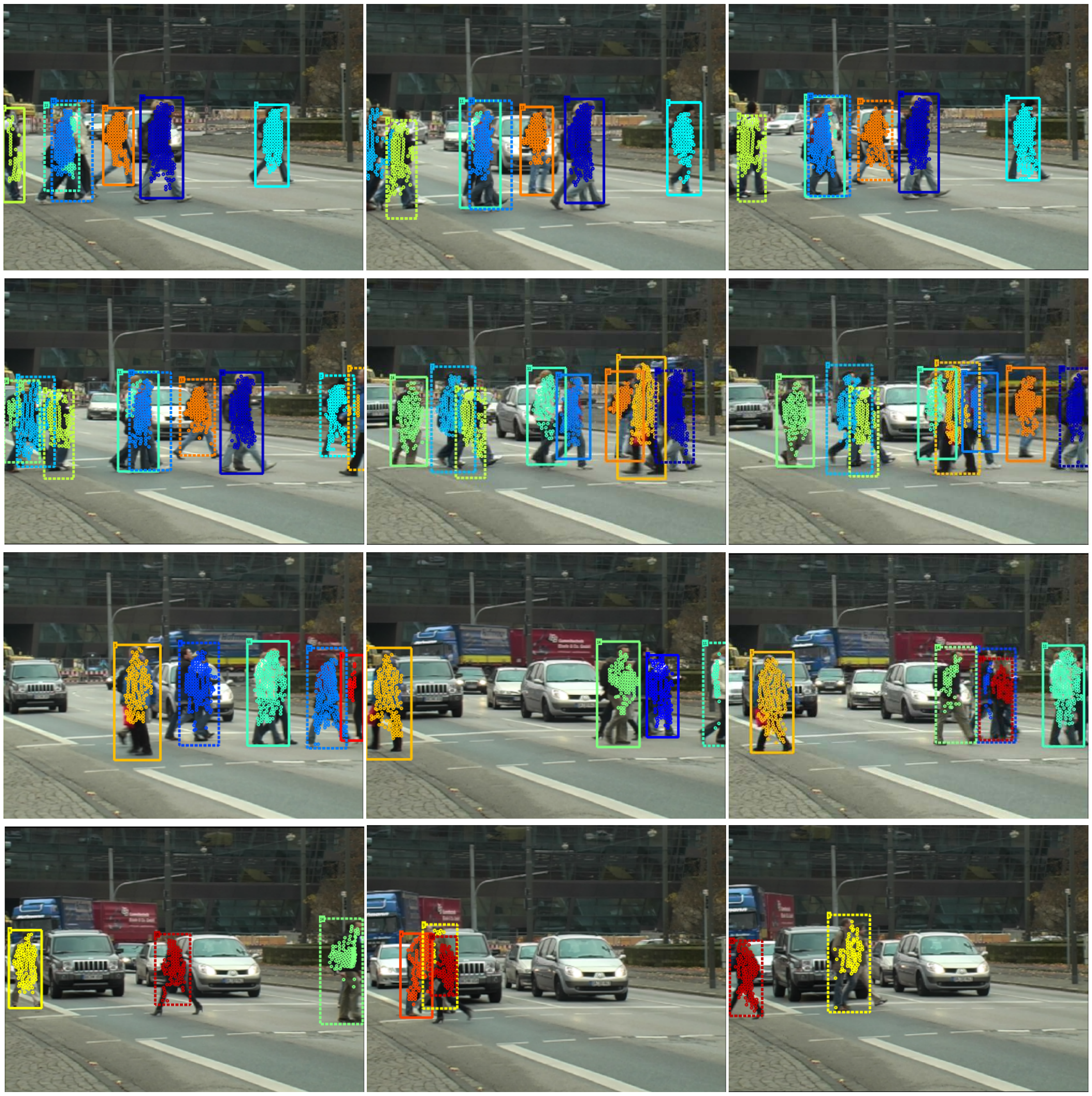


Figure 4.19: Two-granularity tracking results in TUD crossing. In the last row, although the two men are tracked as separate entities before their full occlusion, lack of detections after, cause them to be considered one entity. Information about scene functionality that indicates enties/exits where targets may appear and disappear, in combination with target appearance models, can resolve these failure tracking cases.

Our tracklets adapt to the visibility mask of the targets under occlusions. In datasets TUD, PETS, and ETH sunny day, the two-granularity tracklets rarely drift once targets come close or stay close. The few drifts (id-switches) reported in TUD dataset are due to

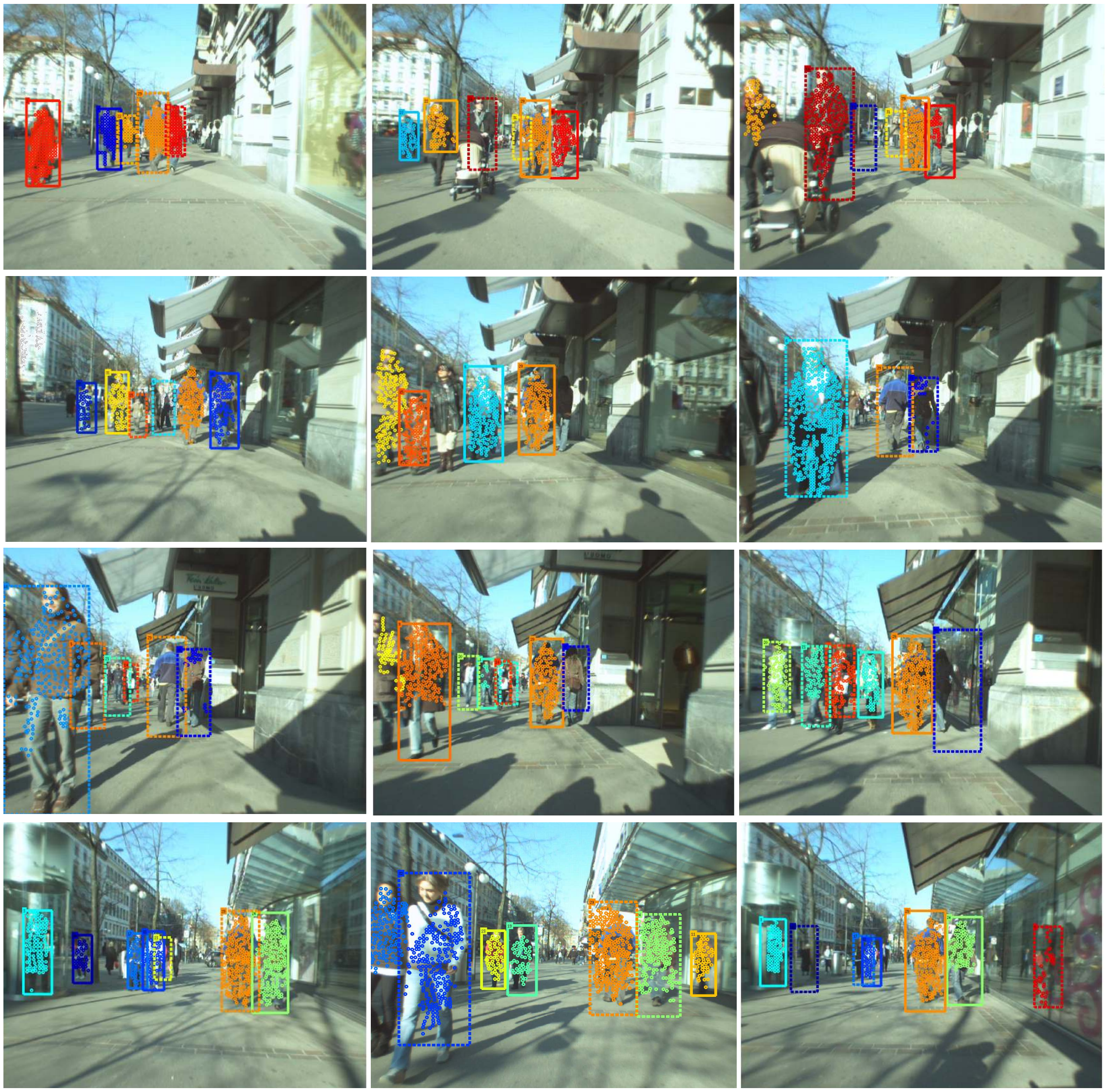


Figure 4.20: Two-granularity tracking results in ETH sunny day. We use only the left view. Many targets have out-of-plane plane motion (they move roughly perpendicularly to the camera image plane). Their apparent 2D motion is less distinct in this case in comparison to in plane motion.

the loose fit of the bounding box, as shown in Figure 4.19 Row 1, rather than drifting of the point trajectories. In UrbanStreet, due to the low frame rate (6fps), we have more drifting point trajectories and resulting drifting tracklets. An analysis of drifting co-clusters is

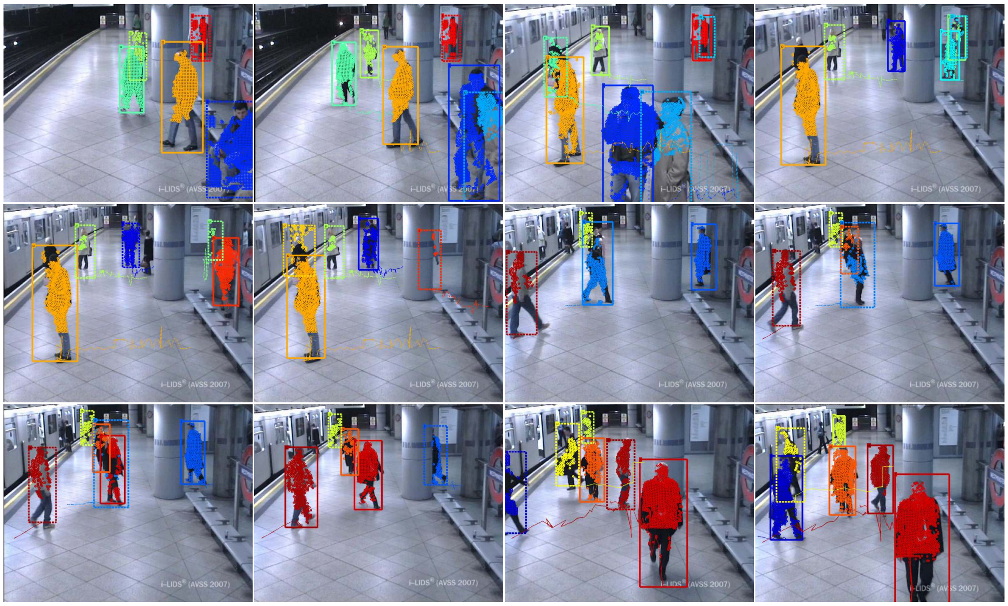


Figure 4.21: Two-granularity tracking results in iLids Hard.

shown in Figure 4.14. The higher the temporal and spatial resolution, the more robust the optical flow estimation and the lower the drifting rate of point trajectories.

The false negatives are often due to overlapping tracklets capturing (accurately) the same target. In ETH sunny day there are few false alarms on windows. The use of 3D information for pruning out-of-perspective detections can minimize this type of false alarms. The large number of fragmentations in PETS dataset is due to our weak linking model of tracklets through occlusion gaps. Building target appearance models as in Gall et al. (2011) and/or using information of scene functionality and goal planning as proposed in Gong et al. (2011); Kitani et al. (2012) can improve linking of tracklets through occlusion gaps. In Urbanstreet, scene functionality is less useful for long range tracklet linking, since targets often disappear in the crowd, rather than at the designated exits of the scene.

Two-granularity tracking can be easily used for tracking objects of any class with the appropriate replacement of the object detetcor. We show in Figure 4.22 results of tracking people and cars in crowded urban scenes.

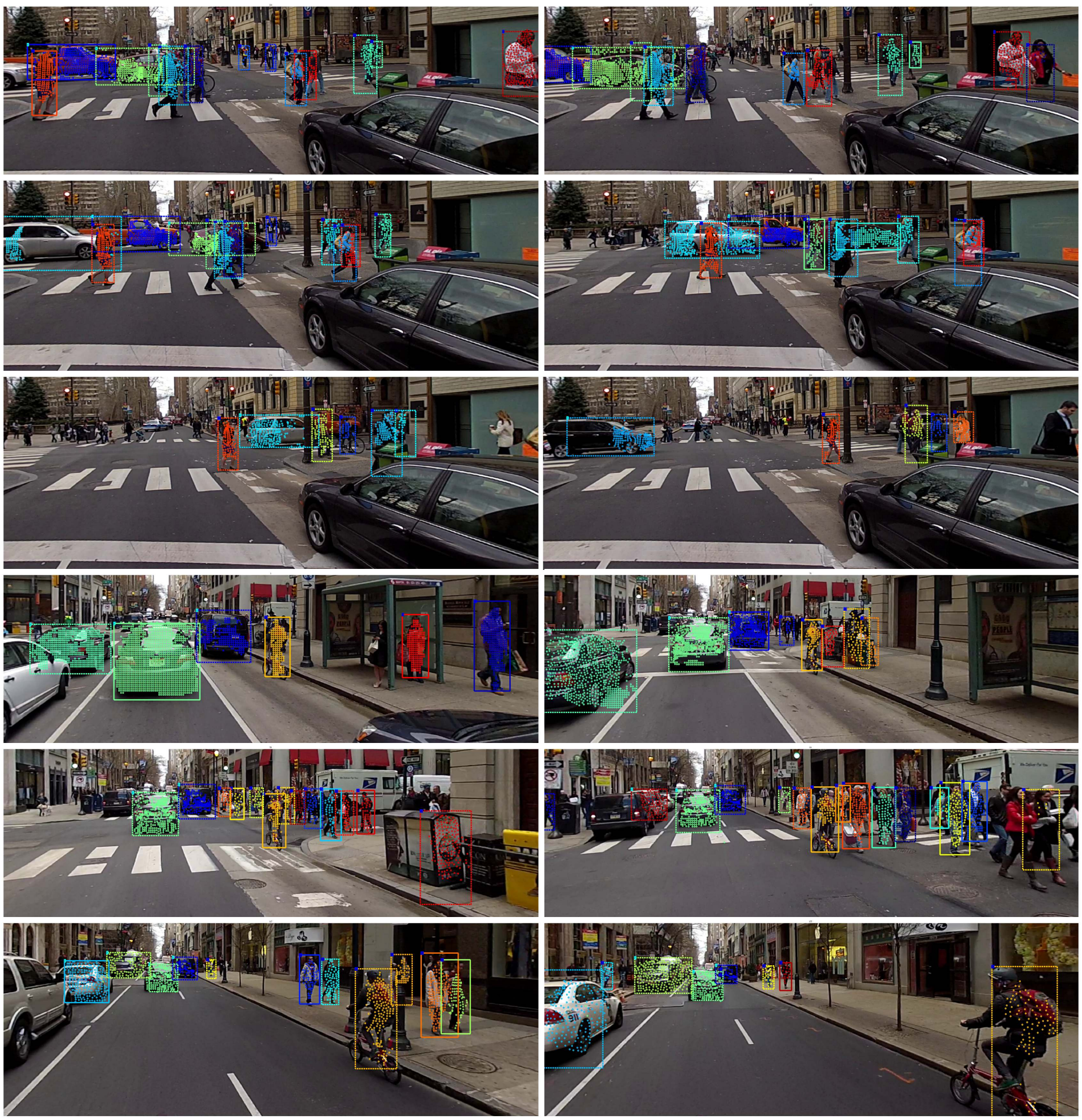


Figure 4.22: Two-granularity tracking for multi-class object tracking in crowded scenes. We show tracking of people and cars in the urban streets of Philadelphia.

Running times The computational bottleneck in our algorithm is the steering cut computation that involves computing eigenvectors of the normalized steered affinity matrix of the point trajectories in the video. For iLids and PETS datasets, that are 750 and 1045 frames long respectively we use background subtraction to remove background trajectories for saving computation. This can be done whenever the camera is static and a background median image can be built accurately. The running times for PETS dataset are 13 mins, iLids 18 mins, TUD crossing (200 frames without background subtraction) 14mins on a 2.6 Ghz processor. The running times exclude optical flow computation and per frame object detection that can be parallelized. Memory restrictions would require our algorithm to process a long video at frame intervals, although the datasets used here did not need this.

4.6 Discussion

We presented a two-granularity tracking framework for tracking and segmenting objects in crowded scenes. Our method mediates information between detectlets and point trajectories via graph steering by repulsion, where classification of detectlets changes the trajectory graph, canceling affinities between point trajectories associated to incompatible detectlets. We showed that the proposed two-view steering cut can effectively handle contradictions in detectlet and trajectory graphs as well as false alarm detectlets in contrast to standard co-clustering. Two-granularity tracking can greatly benefit tracking-by-detection approaches, for better handling detection gaps and tolerating detection sparsity while providing a target accurate mask.

Chapter 5

Two-Granularity Body Pose Tracking

> If the doors of perception were cleansed everything would appear to man as it is, infinite.
>
> — William Blake

Our goal is to estimate 2D human body pose from monocular videos "in the wild": arbitrary clothing, intra-body and background clutter, camera motion, lighting variations, scene and self occlusions. Under partial occlusions, pose detectors often fail and output a pose estimate spanning across two close-by targets. We build upon our two-granularity tracking framework to estimate rough segmentation masks for the people in the scene. We compute an asignment of body joint trajectories to targets' segmentation masks and infer space-time body pose separately for each target in the scene.

5.1 Introduction

Pose specific part templates of Bourdev et al. (2010); Johnson and Everingham (2011); Yang and Ramanan (2011) and pose specific geometric potentials Sapp and Taskar (2013) have recently contributed great performance boosts in body pose detection from static images. Current mixture of parts or mixture of trees representations can better adapt to the multi-modality of appearance of the human body. It is expected that larger number of training examples will boost this performance even further Johnson and Everingham (2011).

Despite the progress, pose detectors still cannot effectively handle self or scene body occlusions. Low part unary potentials are often not discriminative enough to indicate occlusion of a part. Work of Desai and Ramanan (2012a) infers part occlusions by learning instead the typical appearances of the occluder. Though occluder's appearance may indeed be informative for self occlusions (e.g. straight body contour in side view), the appearance of a general scene occluder is widely unconstrained, and the corresponding HOG template will simply learn to set zero weight on this position. So, essentially, learning occluder's appearance results in canceling part template support.

We propose using the spatio-temporal organization of video pixels to constrain the task of body pose estimation in videos. The way objects move, establishes segregations and attractions between video pixels and their corresponding temporal trajectories. In static images, fake boundaries in body interiors or faint boundaries across objects with accidental similarity in appearance create ambiguities in segmentation, despite the progress of boundary detectors Arbelaez et al. (2009). In videos, motion segmentation can dig out faint contours between objects with distinct motion and does not over-fragment object interiors since motion is smooth on textured torsos. As such, motion segmentation holds great potentials in assisting pose estimation in videos.

We demonstrate the usefulness of motion segmentation for pose estimation in videos by mediating information between a motion trajectory graph and a space-time graph over random variables representing body joints in each frame. Our contributions are two-fold:

1. A detection-by-tracking approach for populating the state space of the body joint random variables. We represent body joints candidates as trajectories, rather than per frame instances, exploiting pixel temporal correspondences in video: we track

the MAP body pose estimate in time using optical flow, each trajectory point becomes a body joint candidate in the corresponding frame. Previous approaches such as Batra et al. (2012) sample multiple modes from a static pose detector to populate the state space of each body joint variable. We quantitatively show that such pose tracking can fill in mis-detections gaps more effectively than standard per frame pose sampling, and produces a better pose oracle than per frame pose sampling methods. Given body joint candidates binded on trajectories, we estimate unary potentials for each hypothesized body joint using motion based trajectory voting. We show that motion voting of body joint trajectories can isolate spatially inaccurate body joint candidates that reside on trajectories that drift to surroundings. Furthermore, the long temporal lifespan of body joint trajectories helps their association with whole body tracklets for pose estimation under close object interactions.

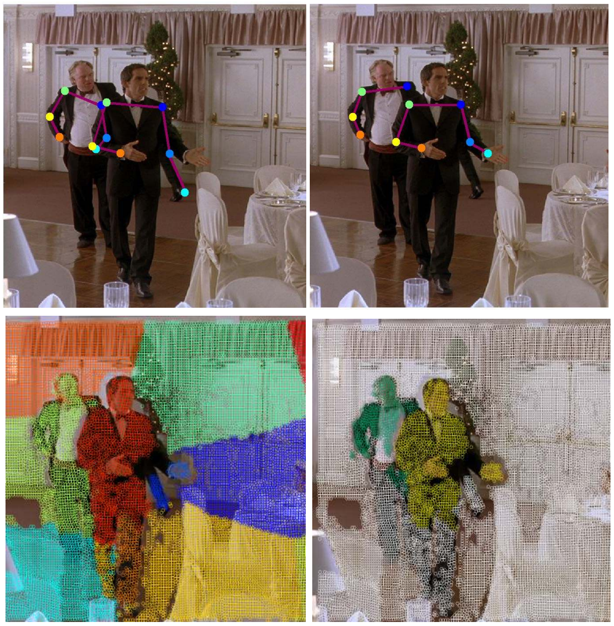
2. An inference framework that maximizes goodness of figure-ground segmentation and pose fit. We propose the use of consensus between part assignments and motion grouping for evaluating confidence of our pose estimates. For crowded scenes, we build upon our two-granularity tracking work and compute associations of body joint trajectories to two-granularity tracklets, filtering in this way body joints on closeby targets.

We have tested our algorithm on the FLIC movie dataset introduced in Sapp and Taskar (2013) and on video sequences of pedestrians in crowded urban scenes. We compare against popular sample-and-link approaches of Park and Ramanan (2011). We show our method outperforms by a large margin baselines that ignore spatio-temporal organization of video pixels.

5.2 Related work

Pose tracking-by-detection Early approaches for human pose estimation in videos, track a manually initialized body pose using kinematic constraints in 3D, as in Bregler

Figure 5.1: Top Left: The result of the state-of-the art pose detector of Sapp and Taskar (2013). Cross-object occlusions confuse the pose estimation algorithm. Bottom left: Motion based trajectory partitioning. Right: The pose and segmentation result of our method. Motion dissimilarity between occluder and occludee is an important cue for scene occlusions, whether the occluder corresponds to another person (as is the case here) or to a background object.



and Malik (1998), or 2D, as in Ju et al. (1996). Pose tracking-by-detection approaches such as Batra et al. (2012); Park and Ramanan (2011) sample a set of poses in each frame and link them in time according to temporal coherence. Recent progress in static pose detection reported in Bourdev et al. (2010); Johnson and Everingham (2011); Yang and Ramanan (2011) make pose tracking-by-detection approaches increasingly popular. They allow automatic recovery from pose drifts, in contrast to manually initialized pose trackers. Pairwise temporal dependencies may be represented at the level of whole pose samples, as in Park and Ramanan (2011), where inference is carried out by dynamic programming, or at the level of individual body parts, as in Ferrari et al. (2009b); Sapp et al. (2011); Sigal et al. (2012), which creates loops in the spatio-temporal graphical model of the human body. Activity specific temporal dynamics have been explored in Lan and Huttenlocher (2004); Sminchisescu et al. (2005), where the parts in consecutive frames are coupled through a latent variable, that controls the evolution of the activity, e.g., state of the walking cycle. Works of Ferrari et al. (2009b); Ramanan et al. (2005b) use ”lucky“ frames of confident detections to learn instance specific part appearance models tailored to the video sequence in hand.

Pose under Occlusions Multiple works have extended the basic pictorial structure formulation with occlusion part states Sigal and Black (2006), where occlusion unary potentials depends on low scoring of the image evidence. This is often not discriminative enough to determine absence or presence of a body part Desai and Ramanan (2012a). Authors of Wang and Mori (2008) have proposed mixtures of pictorial structures, each mixture corresponds to a different occlusion case for the human body. Work of Eichner and Ferrari (2010) fits multiple pictorial structure models simultaneously in (family) pictures of multiple people, estimating occlusions in case of pose overlaps. Finally, work of Gammeter et al. (2008) uses a multi-object detection tracker to estimate multiple pedestrian trajectories in urban scenes, and for each pedestrian trajectory estimates a segmentation prior on which it regresses towards body joint estimates. It can successfully deal with upright poses for which an informative prior can be extracted from the pedestrian trajectory.

Joint detection and segmentation Numerous approaches have proposed co-inference of pose and segmentation in static images. However, the complexity of interactions in this multi-level joint model of parts and pixels, has led to relaxations that essentially use multiple samples of poses to be evaluated against segmentation cut energy Bray et al. (2006) or multiple figure-ground segmentation samples to be evaluated against pose fit energy Ionescu et al. (2011a), in a discriminative manner. Wang and Koller Wang and Koller (2011) compute a foreground appearance model from *all* the part detections (true or false alarms) and ask for the total number of parts to explain the image foreground.

5.3 Pose and segmentation model

5.3.1 Space-time body pose graph

Given a video sequence $I = \{I_t, t = 1 \cdots T\}$, we represent the upper body pose of an actor with a pairwise Markov Random Field $G(V, (E^A, E^T))$ that extends in space and time, depicted in Figure 5.2. Each node $v_s \in V$ corresponds to a random variable Y_s that represents the pixel location of one of the 6 body joints in one of the T video frames of the sequence. We have $n = 6T$ random variables in total. For simplicity we will assume that all random variables have state space of equal cardinality m, that is $Y_s \in \{1 \cdots m\}, s = 1 \cdots n$.

Potentials functions $\phi_i(Y_i)$ encode scores of each body joint location. Potential functions $\Psi^A_{i,j}(Y_i, Y_j)$ on intra-frame edges E^A encode articulation (geometric) compatibilities between assignments of pairs of random variables that are neighbors in the per frame articulation tree. Potential functions $\Psi^T_{i,j}(Y_i, Y_j)$ on cross-frame edges E^T encode temporal compatibilities between random variables of the same body joint on consecutive frames. The joint distribution represented by this MRF is:

$$P(Y) = \frac{1}{Z} \prod_i \phi_i(Y_i) \prod_{i,j \in E^A} \Psi^A_{i,j}(Y_i, Y_j) \prod_{i,j \in E^T} \Psi^T_{i,j}(Y_i, Y_j). \tag{5.1}$$

The Maximum Aposteriori (MAP) inference problem is to maximize $P(Y)$ over all possible joint assignments $Y \in \{1, ..., m\}^n$.

Let $y_{ia} \in \{0, 1\}$ be a binary random variable with $y_{ia} = 1$ iff $Y_i = a$. We concatenate each $y_{i,a}$ in a vector $y \in \{0, 1\}^{nm \times 1}$. Since each random variable can take exactly one value we have the constraint $\sum_a y_{ia} = 1$, which can be written as a linear constraint $Cy = 1$ for some matrix C. We consider matrix $U \in \mathbb{R}^{mn \times 1}$, with $U_{ia} = \log(\Phi_i(a))$ and matrices $P^A, P^T \in \mathbb{R}^{nm \times nm}$, with $P^A_{ia,jb} = \log(\Psi^A_{i,j}(a, b))$ and $P^T_{ia,jb} = \log(\Psi^T_{i,j}(a, b))$ (if $ij \notin E^A, E^T$ then $P^A(ia, jb) = 0, P^T(ia, jb) = 0$). With these notations the MAP problem becomes:

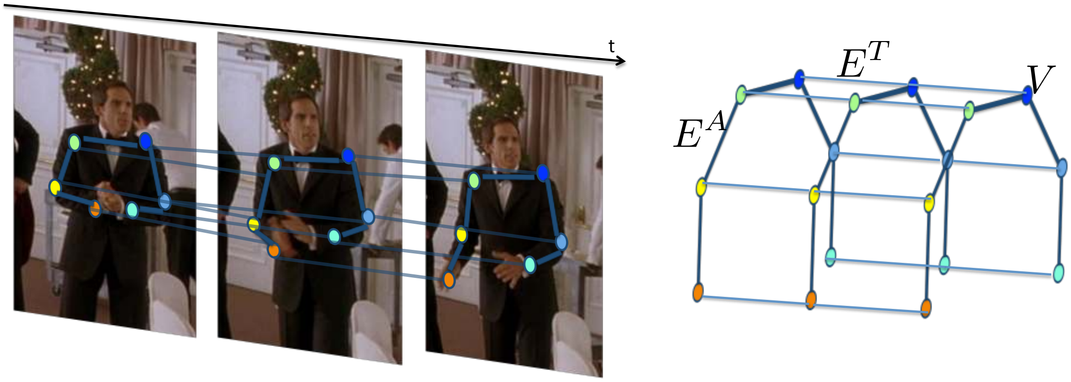


Figure 5.2: Space-time Markov Random Field $G(V,(E^A,E^T))$ for human pose pose. Temporal edges create loops in the graph. Color indicates body joint labels.

$$\max_{y}. \quad \theta(y) = y^T(P^A + P^T)y + U^T y, \quad \text{s.t.} \quad Cy = 1, \quad y \in \{0,1\}^{nm\times 1}. \tag{5.2}$$

For succinctness, we rewrite $\theta(y) = y^T\mathbf{P}y$, where $\mathbf{P} = P^A + P^E + \text{Diag}(U)$.

In practice, balancing the per frame log potentials U, P^A and temporal log potentials P^T in Eq. 5.2 may be difficult: often the right pose is not temporally smooth, e.g., a fast moving arm, while a false alarm detection may be very temporally stable, residing either at the (non deforming) torso interior, or at the background. Second, for partially occluded body poses, unary log potentials U are not always informative. Template matching of the lower body limbs is often not sufficient to indicate presence versus absence of a body part.

In this work, we try to address these issues with two ideas: a) Exploiting *long range* temporal pixel correspondences for constraining long temporal associations of body joints candidates (Section 5.3.2). b) Using pose and motion segmentation consensus for jointly inferring body pose and motion figure-ground segmentation (Section 5.3.3).

5.3.2 Detection-by-tracking: State binding for space-time pose estimation

In each frame I_t, we compute the MAP pose estimates of a static pose detector. We consider the set of body joint detections from all frames $\mathcal{D} = \{\mathrm{d}_p, p = 1 \cdots n_D\}$, where $\mathrm{d}_p = (x_p, y_p, t_p, l_p, c_p)$ and (x_p, y_p, t_p) denote the space-time location of the body joint, $l_p \in \{1 \cdots 6\}$ denotes the body joint label and c_p the score of the detection. Body joint hypotheses that result from the same pose detection share the same score.

We use a detection-by-tracking method for populating the state space of the random variables in our graphical model. We track each detection d_p forward and backward in time using optical flow as proposed in Sundaram et al. (2010) and obtain body joint trajectories $\mathcal{T}^D = \{\mathrm{tr}_p^D, p = 1 \cdots n_D\}$. Trajectory points $\{(x_p^t, y_p^t), t = 1 \cdots T_P, p = 1 \cdots n_D\}$ populate the state space of the random variables of corresponding body joint label l_p and frame t. Body joint detection-by-tracking can jump missed detection gaps by propagating detected body joints from ”lucky” frames to unlucky ones, where deformation prevents a reliable pose detection. The resulting state space provides a better pose oracle in comparison to pose sampling per frame of Batra et al. (2012); Park and Ramanan (2011), as depicted also in Figure 5.9. This means the resulting state space has higher probability of containing the groundtruth body joint location.

We compute motion affinities $\mathbf{A}_D \in \mathbb{R}^{n_D \times n_D}$ between body joint trajectories of the same label according to motion similarity:

$$\mathbf{A}_D(p,q) = \exp\big(-\frac{d_{pq}\Delta u_{pq}}{\sigma}\big) \cdot \delta(T_p \cap T_q \neq \emptyset) \cdot \delta_{(l_p = l_q)}, \tag{5.3}$$

where we use the same computation and notation as in previous chapters.

We compute unary and pairwise intra-frame potentials $u \in \mathbb{R}^{n_D \times 1}$, $p^A \in \mathbb{R}^{n_D \times n_D}$ of body joint trajectories with motion similarity driven voting: the higher the motion similarity between two body joint candidates, the higher the vote:

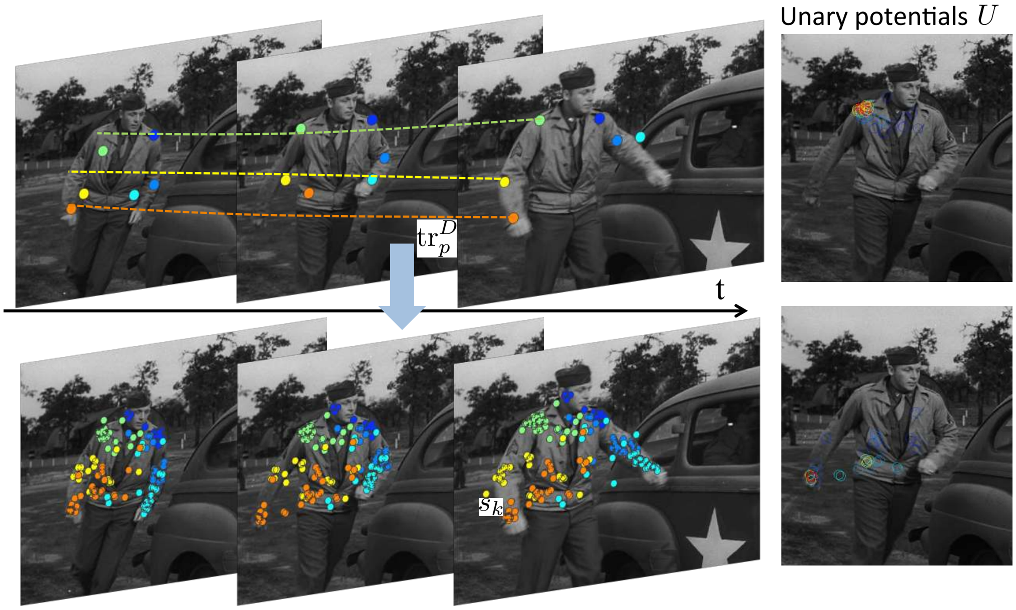


Figure 5.3: Body joint state binding. TopLeft: We track the MAP pose detection in each frame using optical flow. The resulting body joint trajectories can jump miss detection gaps. BottomLeft: Resulting pool of candidate body joints. Right: The state potentials U, incorporate information from multiple frames and are indicative of the basic space-time modes of the body joints in the video.

$$u(p) = \sum_{q,\ l_q=l_p} \mathbf{A}_D(p,q)c_q \tag{5.4}$$

$$p^A(p,q) = \sum_{a,b,t_a=t_b} \mathbf{A}_D(p,a)\mathbf{A}_D(q,b)c_q \tag{5.5}$$

$$\tag{5.6}$$

Let $\mathcal{S} = \{s_k, k = 1 \cdots n_S\}$ denote the set of resulting body joint candidates from the trajectory points of $\mathcal{T}^D$. Let f_k, t_k, r_k denote the trajectory index, the frame index and random variable index for body joint candidate s_k. We convert trajectory potentials to state potentials $U \in \mathbb{R}^{n_S \times 1}$, and P^A, $P^T \in \mathbb{R}^{n_S \times n_S}$ as follows:

$$U(k) = u(f_k) \tag{5.7}$$

$$P^A(k,l) = p^A(f_k, f_l) \tag{5.8}$$

$$P^T(k,l) = \begin{cases} 1, & \text{if} \quad f_k = f_l, |t_k - t_l| = 1, \\ \log(-||(x_{f_k}^{t_k}, y_{f_k}^{t_k}) - (x_{f_l}^{t_l}, y_{f_l}^{t_l})||_2, & \text{if} \quad f_k \neq f_l, |t_k - t_l| = 1, \\ 0, \quad \text{otherwise.} \end{cases} \tag{5.9}$$

Body joint candidates on the same trajectory are bind together, they share the same unary scores. We compute inference in the body pose graph via tree decomposition as in Sapp et al. (2011), also depicted in Figure 5.4. We consider 6 tree structured graphs and sum the resulting max marginals for each body joint candidate. We denote body joint candidate max marginal scores as $\mathrm{mxmr}(s_k), k = 1 \cdots n_S$.

The body joint candidate scores U indirectly incorporate large temporal support and clearly delineate the main modes of the space-time body joint candidates, as depicted in Figure 5.3right. Thus, omitting temporal dependencies during the sum of trees relaxation affects less the quality of the approximation.

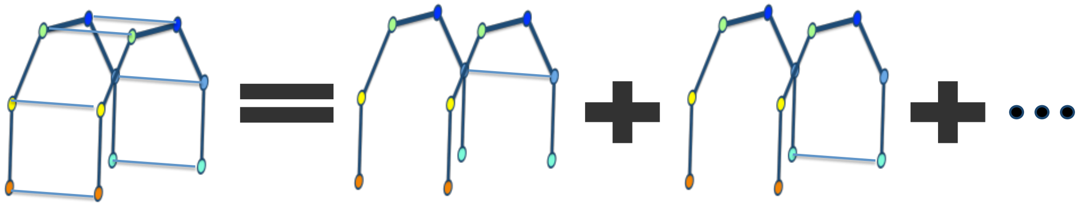


Figure 5.4: Tree decomposition. We approximate inference in the loopy graph by sum of tree inferences.

5.3.3 Trajectory motion graph

Given a video sequence I, we consider a set of dense point trajectories $\mathcal{T} = \{\mathrm{tr}_i, i = 1 \cdots n_T\}$. We compute motion affinities $\mathbf{A}_T \in [0,\ 1]^{n_T \times n_T}$ according to long range motion similarity, as proposed in Brox and Malik (2010b). Let $z \in \{0, 1\}^{n_T \times 1}$ denote the figure-ground trajectory indicator vector, then the goodness of figure-ground trajectory classification is measured by:

$$\mathrm{nassoc(z)} = \frac{z^T \mathbf{A}_T z}{z^T \mathbf{D}_{\mathbf{A}_T} z}, \tag{5.10}$$

where $\mathbf{D}_{\mathbf{A}_T} = \mathrm{Diag}(\mathbf{A}_T \mathbf{1}_{n_T \times 1})$. We remind the reader that maximizing normalized cluster associations are equivalent to minimizing normalized cuts, as discussed in Chapter 2.

Spectral clustering in $G(\mathcal{T}, \mathbf{A}_T)$ often results in over-fragmentation of articulated bodies. Instead of a hard clustering, we will work with soft (undiscretized) embedding trajectory affinities $\tilde{\mathbf{W}} \in [0,\ 1]^{n_T \times n_T}$, depicted in Figure 5.5:

$$\tilde{\mathbf{W}}(i,j) = V_i^T \Lambda V_j / \max(\rho_i, \rho_j) \tag{5.11}$$

$$\rho_i = \max_{j \in \mathcal{N}(i)} (V_i \Lambda V_j^T), \tag{5.12}$$

$$\tag{5.13}$$

where $(V, \Lambda = \mathrm{Diag}(\lambda))$ are the top K eigenvectors and eigenvalues of the normalized affinity matrix $\mathbf{D}_{\mathbf{A}_T}^{-1} \mathbf{A}_T$ and $V_i \in \mathbb{R}^{K \times 1}$ represents the embedding coordinates of tr_i. $\mathcal{N}(i)$ stands for the Delaunay neighborhood of tr_i.

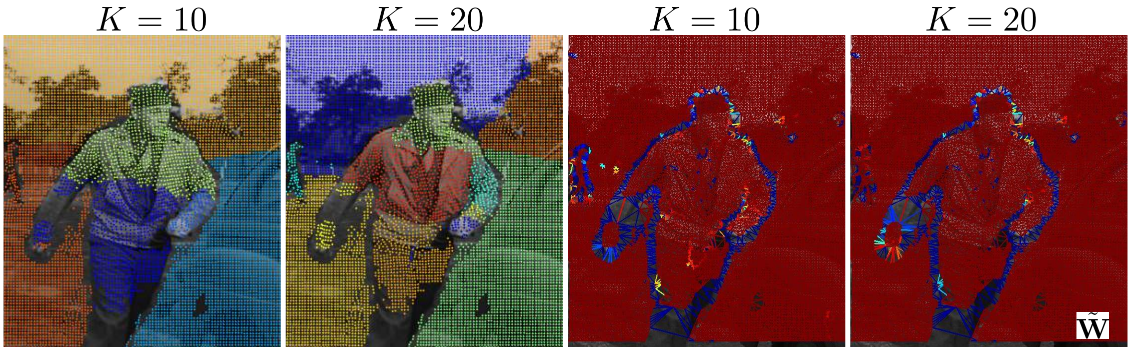


Figure 5.5: Trajectory motion graph. Columns 1, 2: Trajectory spectral clustering for number of eigenvectors $K = 10$, $K = 20$ respectively. Trajectory clustering often over-segments articulated bodies. Columns 3, 4: Normalized embedding trajectory affinities $\tilde{\mathbf{W}}$ for number of eigenvectors $K = 10$, $K = 20$ respectively. Edges correspond to spatially neighboring trajectories and color indicates strength of affinity, with red indicating high affinity and blue low one. The reader can imagine that such pairwise affinities $\tilde{\mathbf{W}}$ exist also between non spatially neighboring trajectories, not shown in the image. Varying the number of eigenvectors K does not dramatically change affinities in $\tilde{\mathbf{W}}$.

5.3.4 Body part to trajectory cross-associations

Let $s_{k\sim l}$ denote the body part with endpoints the body joint candidates s_k, s_l, with $t_k = t_l$ and r_k, r_l are body joint labels neighboring in the per frame articulation tree. Let $M^F(s_{k\sim l})$, $M^B(s_{k\sim l})$ denote foreground and background masks, depicted in Figure 5.6. We establish associations between body parts and trajectories according to spatial overlap:

$$\mathbf{C}^F(k \sim l, i) = \delta(\mathrm{tr}_i \in M^F(s_{k\sim l})), \quad (5.14)$$

$$\mathbf{C}^B(k \sim l, i) = \delta(\mathrm{tr}_i \in M^B(s_{k\sim l})). \quad (5.15)$$

$$(5.16)$$

For easy of notation we will consider the matrices $\mathbf{C}^F, \mathbf{C}^B \in \mathbb{R}^{n_S^2 \times n_T}$ from pairs of body joint candidates to trajectories. Pairs of body joints candidates that do not neighbor in the articulation tree (they do not define a body part), are associated to no trajectories. This will allow us to have an easy transformation between body joint selector y to body part selector $\mathrm{vec}(yy^T)$.

5.3.5 Joint pose and motion figure-ground segmentation

Let $z \in \{0,1\}^{n_T \times 1}$ denote the figure-ground trajectory indicator vector and let $y \in \{0,1\}^{n_S \times 1}$ denote the body joint candidate selection. Joint pose estimation and motion segmentation in videos can be formulated as a maximization of normalized figure-figure and ground-ground trajectory associations (minimizing normalized cuts), body part pose fitting and cross-alignment between selected parts and trajectory foreground:

$$\text{max.} \quad \epsilon(z,y,p) = \underbrace{\frac{z^T\tilde{\mathbf{W}}z}{z^T\mathbf{D}_{\tilde{\mathbf{W}}}z}}_{\text{trajectory clustering}} + \underbrace{y^T\mathbf{P}y}_{\text{pose fit}} + \underbrace{\frac{p^T\mathbf{C}^F z}{p^T\mathbf{D}_{\mathbf{C}^F}p}}_{\text{alignment}} \tag{5.17}$$

$$\text{s.t.} \quad z \in \{0,1\}^{n_T \times 1}, \quad y \in \{0,1\}^{n_S \times 1}, \quad Cy = 1, \quad p = \mathrm{vec}(yy^T). \tag{5.18}$$

A direct way of optimizing Eq. 5.17 is by computing the ncut segmentation score of each pose hypothesis in each frame and compute via dynamic programming the most temporal coherent pose sequence of good segmentability and pose fitting score. We bypass the brute force computation by decomposing the segmentation score of a body pose into body part segmentability scores.

Body part segmentabilities We define the segmentability of a body part candidate $\mathrm{seg}(s_{k\sim l})$ to be:

$$\mathrm{seg}(s_{k\sim l}) = \frac{\mathbf{C}^F(s_{k\sim l},:)\tilde{\mathbf{W}}\mathbf{C}^F(s_{k\sim l},:)^T}{\mathbf{C}^F(s_{k\sim l},:)\tilde{\mathbf{W}}\mathbf{C}^F(s_{k\sim l},:)^T + \mathbf{C}^F(s_{k\sim l},:)\tilde{\mathbf{W}}\mathbf{C}^B(s_{k\sim l},:)^T}, \tag{5.19}$$

where $\mathbf{C}^F(s_{k\sim l},:)^T \in \{0,1\}^{n_T \times 1}$ is the trajectory foreground indicator for part $s_{k\sim l}$. The segmentability of a body part candidate $s_{k\sim l}$ is a measure of its agreement with the underlying motion affinities in the trajectory motion graph $\tilde{\mathbf{W}}$. The higher the segmentability

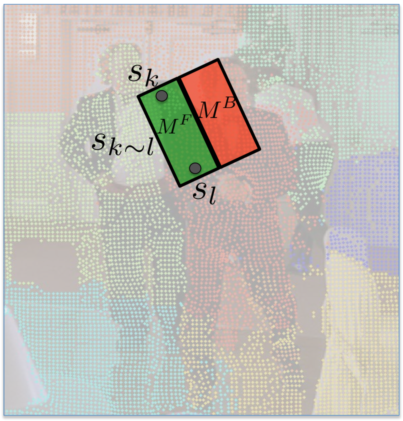


Figure 5.6: Body part segmentability. For each body part candidate $s_{k\sim l}$ we consider one foreground and one background mask M^F, M^B respectively and define segmentability as the fraction of intra-foreground trajectory affinities to the sum of intra-foreground and foreground-background affinities.

the lower the normalized motion affinity cut between its foreground and background trajectories.

We use $l(s_{k\sim l})$ to denote the body part label (left upper arm, left lower arm etc.) of the part candidate $s_{k\sim l}$. For each body part candidate $s_{k\sim l}$ we consider the following set of features:

- Normalized segmentability $\frac{\text{seg}(s_{k\sim l})}{\max\limits_{a,b,r(a)=r(k),r(b)=r(l)} \text{seg}(s_{a\sim b})}$.
- Normalized max marginal score $\frac{\text{mxmr}(s_{k\sim l})}{\max\limits_{a,b,r(a)=r(k),r(b)=r(l)} \text{mxmr}(s_{a\sim b})}$, where we define $\text{mxmr}(s_{k\sim l}) = \min(\text{mxmr}(s_k), \text{mxmr}(s_l))$.
- The tangent $\tan(s_{k\sim l})$ of the angle between the line shoulder-wrist or shoulder-elbow and the vertical direction (depending on whether the type of the part candidate $s_{k\sim l}$ is an upper or lower arm). The larger the angle the more segmentable the part is expected to be.
- Unary potential score $U_k + U_l$.

We use these features to learn a logistic regression for mapping each body part candidate to a desired confidence value estimated from the negative exponential of the Euclidean distance between the two body joint endpoints of the candidate $s_{k\sim l}$ and the groundtruth body joint locations. We compute body joint candidate confidence scores by taking the maximum of all body part confidence across all parts connected to a body joint candidate. We discard body joint candidates with low confidence scores and compute an inference over the pruned state space, where an additional state per random variable indicates occlusion of the corresponding body joint.

We obtain a figure-ground trajectory assignment z by assigning as foreground the trajectories on the foreground of the estimated pose, and as background the rest.

5.3.6 Body pose estimation under interactions

Consensus of segmentation and pose estimation via part segmentabilities works well for large enough video resolution. In small spatial resolution, frequent trajectory fragmentations often times cause a body part not to contain enough trajectories to describe its motion.

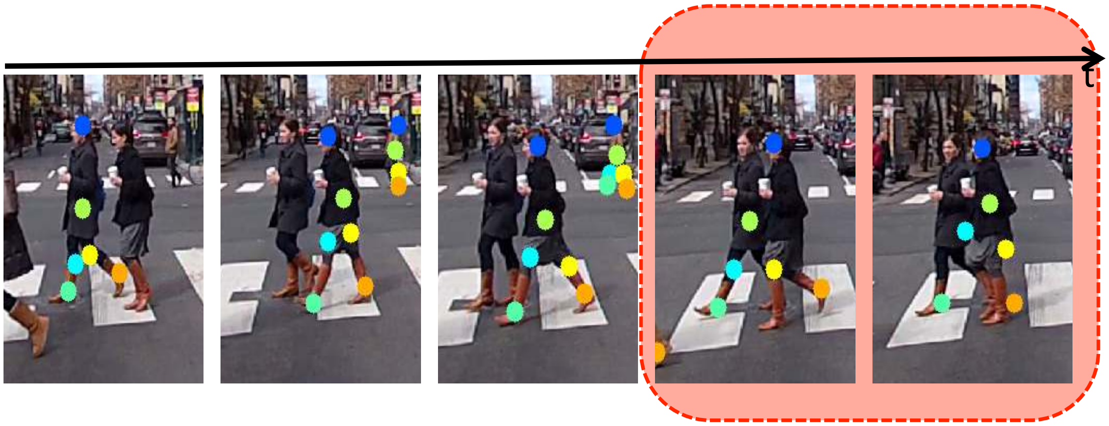


Figure 5.7: Pose detection under occlusions. We show the MAP pose estimates of Yang and Ramanan (2011) in a set of frames. While stylized pedestrian poses are reliably detected, under partial occlusions, the pose estimates span across closeby targets. This is because the pose detector does not model target partial occlusions but rather outputs all body joints each time. Attempts to model partial poses (via, e.g., different detection mixtures) have not been yet very successful due to lack of distinctive features to indicate which occlusion scenario is the correct one.

We propose a two step process for computing consensus between body pose estimates and figure-ground video segmentation that build upon our two-granularity tracking algorithm, described in Chapter 4. Each two-granularity co-cluster provides an approximate figure-ground segmentation mask per target. These figure-ground segmentations are used to estimate assignment of body joint trajectories to targets and eliminate body joints candidates on closeby targets, as shown in Figure 5.8.

Our method is simple. Given a set of K two-granularity tracklets and n_D body joint trajectories, we compute an assignment matrix $P \in \mathbb{R}^{K \times n_D}$ according to maximum closest point distance between the body joint trajectory and the co-cluster trajectories:

$$P(i,j) = \max_{t \in T_i \cap T_j} \left(\min_{\mathrm{tr}_k \in \mathcal{T}_i} \exp(-\frac{1}{\sigma} ||x_k^t - x_j^t, y_k^t - y_j^t||_2) \right), \tag{5.20}$$

where T_j, T_i the frame sets of the body joint trajectory and the co-cluster respectively and $\mathcal{T}_i$ the trajectories of the co-cluster. We double threshold P and discard co-cluster body joint trajectory pairs with scores below a threshold as well as pairs whose score is below 0.3 times the second best co-cluster body joint trajectory match. Trajectory points of the assigned to each target body joint trajectories populate the spate space of the random variables in its MRF. We estimate body pose of people in our scene by inference in each target's MRF using tree decomposition, as already described in Section 5.3.1.

5.4 Experiments

We test our algorithm in pose estimation in 300 randomly sampled videos from the FLIC dataset, introduced in Sapp and Taskar (2013). FLIC contains video sequences from 6 (monocular) movies. Each video sequence is 50-60 frames long. The actors can take a wide range of body poses, are not centered in the middle of the image and may exhibit many different motion patterns: an actor may move abruptly, use only his hands or have very little to no motion whatsoever throughout a video sequence.

We sample static poses in each frame using MODEC, a state-of-the-art pose detector proposed in Sapp and Taskar (2013). MODEC uses a coarse-to-fine pose inference

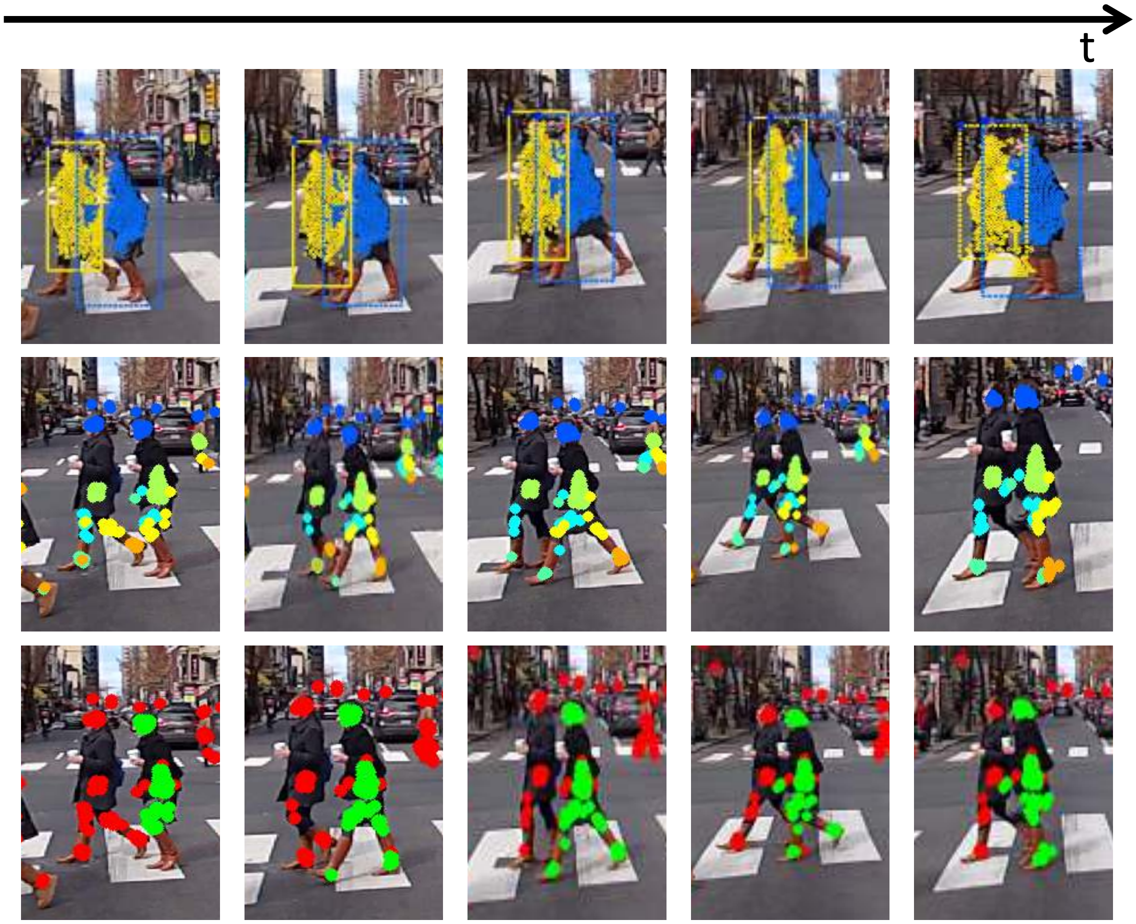


Figure 5.8: Assigning body joint trajectories to tracklets. Top: Two-granularity tracklets. Middle: The state space of our MRF random variables. Color denotes body joint label. Bottom: The body joint trajectories assigned to the lady captured by the blue tracklet are shown in green. Most of the distracting body joints have been discarded. We estimate body pose for each target by inference in our MRF where state space of random variables are populated by the assigned to the target body joint trajectories.

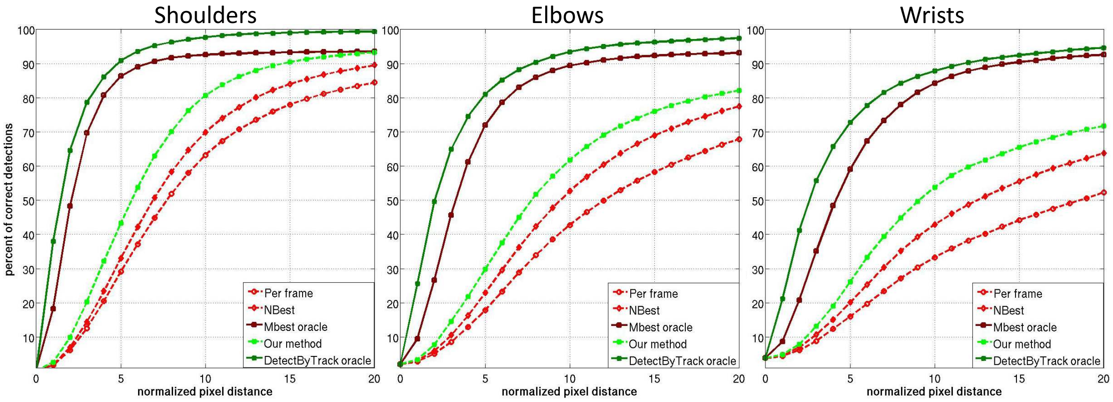


Figure 5.9: Qualitative pose estimation results in FLIC dataset. Our algorithm outperforms the baselines by a large margin for all 3 body joints. Also, the oracle from pose detection-by-tracking (corresponding to the dark green curve) is better than the oracle of standrad multiple pose sampling per frame.

scheme: it detects shoulder hypotheses using poselets of Bourdev et al. (2010) and classifies the patch around them into 32 coarse patch classes, called pose modes. Then computes a refined upper body pose using a tree structured model of Yang and Ramanan (2011), tailored though to the specific pose subspace of each coarse mode. In this way, both the part templates and their pairwise geometric potentials are pose mode specific.

We compare against sample-and-link approaches for video pose estimation that sample multiple pose estimates in each frame, establish temporal potentials across pose samples in consecutive frames, and estimate the most temporally smooth pose sequence using dynamic programming such as Batra et al. (2012); Park and Ramanan (2011). In each video frame we obtain a set of N diverse pose samples, by computing the detailed body pose for each of the 32 (for each arm) coarse pose modes of MODEC. The coarse-to-fine representation of the detector ensures diversity of the resulting poses.

We measure pose detection performance using the evaluation measure of Yang and Ramanan (2011): for any particular joint location precision radius, measured in Euclidean pixel distance scaled so that the ground-truth torso is 100 pixels tall, we report the percentage of correct joints within that radius. For a test set of size M, radius r and particular joint i, this is:

$$acc_i = \frac{100}{MN}\sum_{k=1}^{N} \mathbf{1}(\frac{||y_i^*(x^k) - y_i^k||_2}{\text{torso height}k/100}) <= r,$$

where $y_i(x^k)$ is our model's predicted ith joint location on test frame x^k. We report $acc_i(r)$ for a range of r, resulting in a curve that spans both the very tight and very loose regimes of part localization. We show the result curves for the 3 upper body joints in Figure 5.9. Our method outperforms sample-and-link baselines by a large margin, attributed to our long range temporal potentials that can accumulate information across mis-detection gaps, and mediation with motion segmentation that restricts the figure-ground trajectory assignments and prunes accordingly the part space. We show qualitative results of our algorithm in Figure 5.10 and compare with the Nbest baseline.

To better demonstrate the performance gains and limitations of our method, we evaluate two oracle algorithms, one for our method and one for the baseline, which pick the body joint candidates closest to the ground-truth. Their performance quantifies the percentage of missed detections attributed to ground truth missing from the candidate pool or to failure of the algorithm to select it, despite being present. Our proposed detection-by-tracking strategy that tracks in time the highest scoring pose sample in each frame, provides a better candidate pool than the popular pose sampling in each frame, as shown in Figure 5.9. We also see that both our method and the baseline have a large margin for improvement, upper-bounded by the corresponding oracle performance.

We next test our algorithm in crowded urban scenes. The scale of body parts is too small in this case to reliably compute a trajectory part assignment and segmentability scores. We show qualitative results of our two step inference process, which first computes coarse people tracklets and then detailed body pose for each target, using the spation-temporal coarse target support.

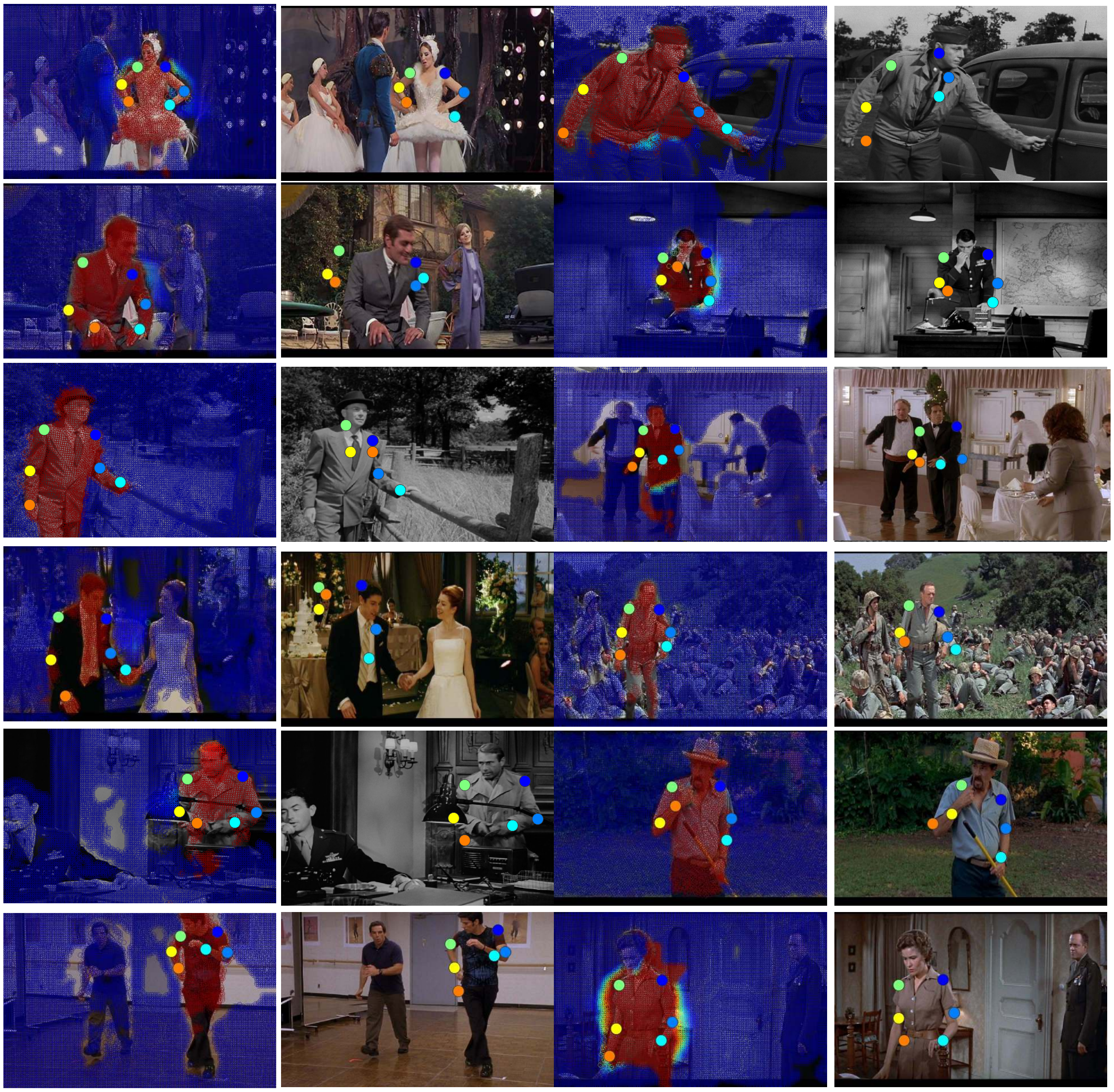


Figure 5.10: Pose estimation results in FLIC dataset of Sapp and Taskar (2013). In frames with no motion, our method still has gains by learning how to penalize easily segmentable poses, such as extended arms, that do not have salient segmentation support and often correspond to false alarms.

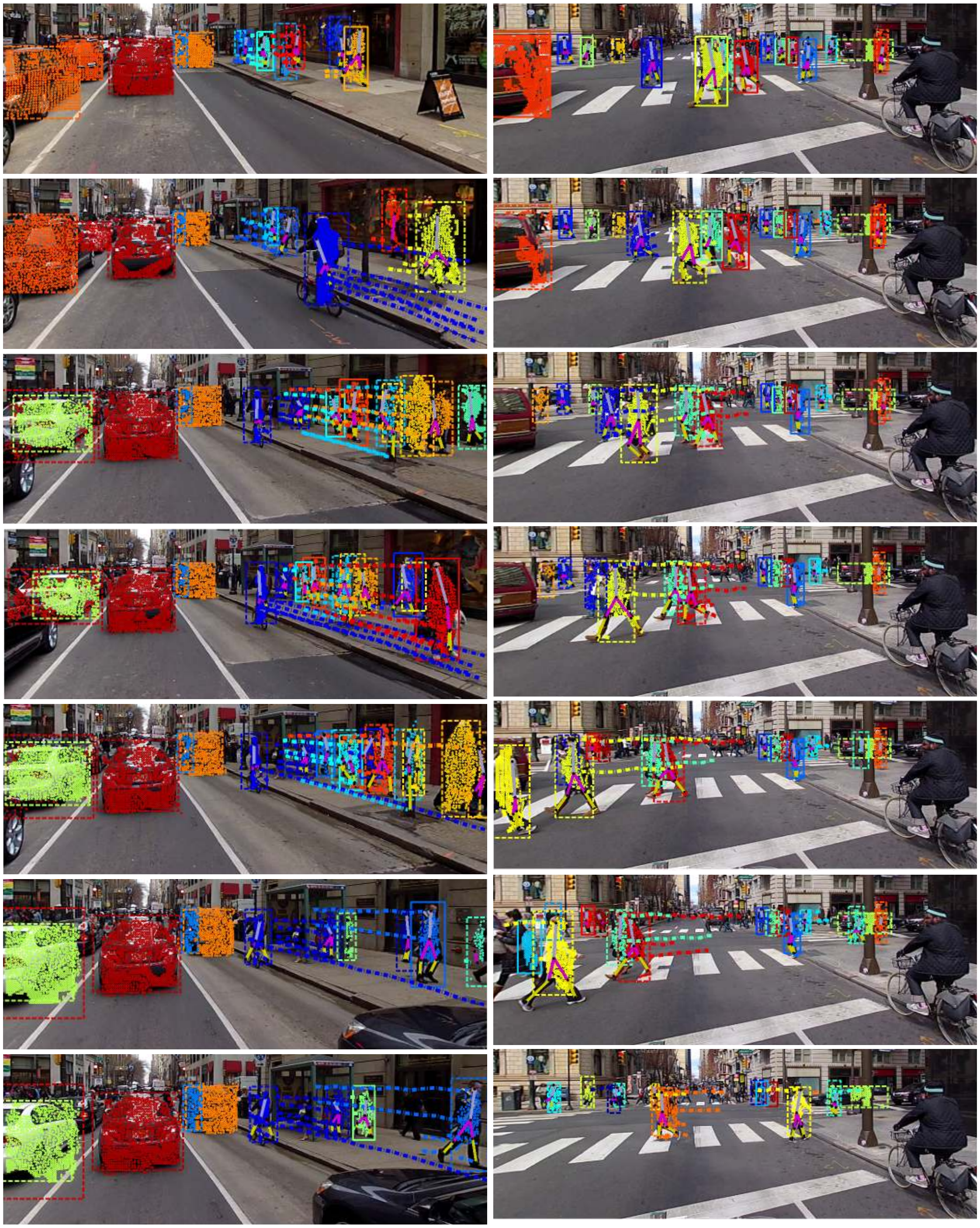


Figure 5.11: Qualitative results of our two-step body pose estimation. We show torso and legs. Wrong poses are result of drifting joint trajectories during leg self occlusions.

5.5 Discussion

We presented a detection-by-tracking approach that represents body joint detections by optical flow trajectories and estimates their unary and pairwise articulation potentials via a motion similarity based voting, that accumulates information across mis-detection gaps. For pose inference under occlusions, we proposed a consensus criterion between pose estimation and motion segmentation, for discarding false alarm body part candidates spanning across nearby targets with different long range motion. Although in large spatial resolution, local normalized cut motion affinity scores can indicate agreement of disagreement of the body part candidate with underlying trajectory motion, in smaller scales, such local measurements are unreliable. We propose a two step optimization of the pose and segmentation consensus, that first computes coarse space time segmentation masks of targets using two-granularity tracking, and then assigns body joint trajectories to targets exploiting long range spatial proximity. In this way, the body joint candidates in the MRF of each target do not contain body joints on closeby targets. We outperform by a large margin the standard sample-and-link approaches for pose estimation in videos, that neglect both the long range nature of bottom-up temporal pixel associations and their spatio-temporal motion based organization.

Chapter 6

Articulated Optical Flow

> All that is important is this one moment in movement. Make the moment important, vital, and worth living. Do not let it slip away unnoticed and unused.
>
> — Martha Graham

In the previous chapters, we used bottom-up optical flow estimation and motion segmentation to aid tracking of objects and their body pose in videos. In this chapter, our goal is human pose detection under wide body deformation. We present a method that tracks deforming human body limbs by employing knowledge of the kinematics of the human body, and in this way improve the optical flow estimation, under fast body motion. Aperture problems and self-occlusions often cause optical flow to fail on deforming human body parts. We present a pose from flow and flow from pose approach, that detects human body pose under large motion. It uses detected articulated joints to incorporate kinematic constraints in optical flow and kinematically track the body limbs in the rest of the video sequence.

6.1 Introduction

Best practices of general object detection algorithms, such as hard mining negative examples Felzenszwalb et al. (2010), and expressive, mixture of parts representations Yang and Ramanan (2011) have recently led to rapid progress in human pose estimation from static images. Parts are the roots of the articulation chains, such as shoulders, are mostly rigid and as such easily detectable. Parts at the end of the articulation chains, i.e., lower arms, are widely deformable and are still hard to detect under unusual body pose, infrequent in the people's pose repertoires as well as in the datasets (see also Figure 6.1). While each rare body pose has low probability of occurrence, the collection of rare body poses occupies a big chunk of the body pose space. This is referred as long tails of the distribution of body poses and visual data in general: they are often comprised by few frequently occuring templates and a large collection of rarely encountered ones.

Rare body poses are often characterized by large, salient motion. Large motion of body parts, though a valuable cue for pose detection, is hard to estimate accurately: body limbs are lost in the coarse pyramid levels of coarse-to-fine optical flow or other gradient based tracking schemes. Furthermore, descriptor matches often slide along the body limb axis due to the 1D nature of body parts, so descriptor augmented optical flow methods also fail under large body motion.

Can bad motion information be useful? Body part motion from optical flow, though not accurate, is often sufficient to segment body parts from their backgrounds. By matching body part segments to shape exemplars, one can improve pose estimation under large body deformations. By estimating pose inversely to current detectors, that is aligning image segmentations to pose exemplars rather than learnt templates to image gradients, we bypass the need for enormous training sets. For such alignment to be possible our method exploits "lucky" segmentations of moving body parts and 1) indexes into a pose space, 2) infers articulated kinematic chains in the image, 3) incorporates kinematic constraints into optical flow tracking. The proposed framework targets rare, widely deformed poses, often missed by pose detectors, and optical flow of human body parts, often inaccurate due to

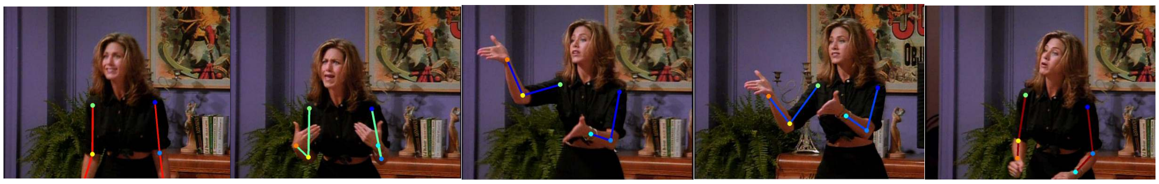


Figure 6.1: Long trails of human body pose distribution. The color of part sticks represents number of training exemplars close in body joint configuration as measured by partial Procrustes distance. While static poses are covered by an abundance of training exemplars (dark red color), widely deformed ones are often rare in the actors' repertoires and in vision datasets (blue color).

clutter and large motion.

Our algorithm segments moving body parts by leveraging motion grouping of saliently moving body parts and figure-ground segregation of reliably detected body parts, e.g., shoulders, in a graph steering framework. Confident body part detections of Bourdev et al. (2010) induce figure-ground repulsions between regions residing in their interior and exterior, and steer region motion affinities in places where motion is not informative. Extracted motion segments with hypothesized body joint locations (at their corners and endpoints) are matched against body pose exemplars close in body joint configuration. Resulting pose labeled segments extract occluding body part boundaries (also interior to the body), not only the human silhouette outline, in contrast to background subtraction works, such as Jiang (2009).

Pose segmentation hypotheses induce kinematic constraints during motion estimation of body parts. We compute coarse piece-wise affine, kinematically constrained part motion models, incorporating reliable pixel correspondences from optical flow, whenever they are available. Our hybrid flow model benefits from fine-grain optical flow tracking for elbows and slowly moving limbs of the articulation chain, while computes coarser motion estimates for fast moving ones. The resulting "articulated" flow can accurately follow large rotations or mixed displacements and rotations of body parts, which are hard to track in the standard optical flow framework. It propagates the pose segmentations in time, from frames of large motion to frames with no salient motion. We show such tracking is robust to pose partial self or scene occlusions.

We evaluate our framework on video sequences of TV shows. Our algorithm can detect people under rare poses, frequently missed by state-of-the-art pose detectors, by proposing a versatile representation for the human body that effectively adapts to the segmentability or detectability of different body parts and their motion patterns.

6.2 Related work

We distinguish two main categories of work combining pose and motion estimation in existing literature: (i) Pose estimation methods that exploit optical flow information; and (ii) part motion estimation methods that exploit pose information. The first class of methods comprises methods that use optical flow as a cue either for body part detection or for pose propagation from frame-to-frame, as in Ferrari et al. (2009a); Sapp et al. (2011). Brox et al. (2006) propose a pose tracking system that interleaves between contour-driven pose estimation and optical flow pose propagation from frame to frame. Fablet and Black (2002) learn to detect patterns of human motion from optical flow.

The second class of methods comprises approaches that exploit kinematic constraints of the body for part motion estimation. Bregler and Malik (1998) represent 3D motion of ellipsoidal body parts using a kinematic chain of twists. Ju et al. (1996) model the human body as a collection of planar patches undergoing affine motion, and soft constraints penalize the distance between the articulation points predicted by adjacent affine models. In a similar approach, Datta et al. (2008) constrain the body joint displacements to be the same under the affine models of the adjacent parts, resulting in a simple linear constrained least squares optimization for kinematically constrained part tracking. Rehg and Kanade (1995) exploit the kinematic model to reason about occlusions.

In the “strike a pose” work of Ramanan et al. (2005a), stylized (canonical) human

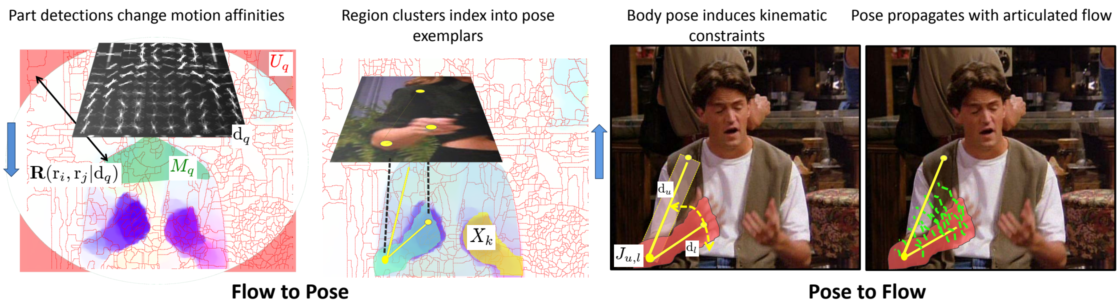

Part detections change motion affinities
Region clusters index into pose exemplars
Body pose induces kinematic constraints
Pose propagates with articulated flow
U_q
d_q
$\mathbf{R}(\mathrm{r}_i, \mathrm{r}_j|\mathrm{d}_q)$
M_q
X_k
d_u
d_l
$J_{u,l}$
Flow to Pose
Pose to Flow

Figure 6.2: Pose from flow. Left: Mediating motion grouping with part detections. Region motion affinities in $\mathbf{A}_R$ change according to confident body part detections that induce repulsions $\mathbf{R}_R$ between regions assigned to their foreground and background. Region clusters index into pose exemplars according to hypothesized joint locations at their endpoints. Right: Pose labelled segmentations propose coarse motion models coupled at the articulation joint. Coarse motion proposals compute an articulated optical flow field that can deal with large part rotations.

body poses are detected reliably, and are used to learn instance specific part appearance models for better pose detection in other frames. In this work, we follow a "strike a segment" approach by segmenting widely deforming body poses and propagating inferred body pose in time using articulated optical flow. Previously, Mori et al. (2004) have used image segments to extract body parts in static images of baseball players.

6.3 From flow to pose

We use segmentation to help the detection of highly deformable body poses. Stylized body poses are covered by an abundance of training examples in current vision datasets, and can often be reliably detected with state-of-the-art detectors, such as Bourdev et al. (2010). Highly deformable poses appear infrequently in the datasets, which reflects their low frequency in people's body pose repertoires. They are mostly transient in nature, the actor is briefly in a highly deformed pose, away from the canonical body configuration. It is precisely their transient nature that makes them easily detectable by motion flow.

There is an asymmetry of motion segmentability among the parts of the human body due to its articulated nature. Parts towards the ends of the articulated chains often deform much faster than the main torso (root of the body articulation tree). Lack of motion may cause ambiguities in motion segmentation of root body parts. However, such root parts can often be reliably detected thanks to their rigidity.

We exploit detectability and segmentability across different body poses and parts in a graph theoretic framework which combines motion-driven grouping cues of articulated parts and detection-driven grouping cues of torso like parts. Similar to the graph steering framework of Section 4.4, detection-driven figure-ground repulsions of torso parts correct (steer) ambiguous motion-based affinities. We segment arm articulated chains by constrained normalized cuts in the steered region graph.

Resulting segmentations with hypothesizing body joints at their corners and endpoints infer body pose by matching against pose exemplars. While detectors would need many training examples to learn to extract a deformed pose from background clutter as noted in Johnson and Everingham (2011), our pose segmentations are already extracted from their backgrounds. We use contour matching between extracted segmentations and pose exemplars to select the right kinematic chain configurations.

6.3.1 Region motion affinities

We pursue a single frame segmentation approach from “lucky” frames that contain non-zero motion, rather than a multi-frame segmentation, as decribed in Chapter 3. Large per frame deformations of lower body limbs though, often prevent optical flow to be reliable: in coarse-to-fine optical flow schemes, motion that is larger than the spatial extent of the moving structure cannot be recovered, since the structure is lost at the coarser levels of the image pyramid Brox et al. (2004). As such, we will integrate per frame optical flow estimates on region spatial support to segment frames with large motion, as measured from a bounding box around a shoulder activation.

We describe the motion of an image region in two ways: i) with the set of point trajectories, if any, overlapping with the region mask, ii) with an affine model fitted to the optical flow displacements of the region pixels. Affine motion fitting allows motion representation in places of ambiguous optical flow anchoring and sparse trajectory coverage. It only takes into account per frame motion estimates and in that sense it is weaker than multi-frame trajectory affinities.

Given a video frame I_t of video sequence I , let $\mathcal{P}$ denote the set of image pixels and let $\mathcal{R} = \{\mathrm{r}_i, i = 1 \cdots n_R\}$ denote the set of image regions. We will use r_i to refer to both the region r_i and its corresponding pixel set. Let $\mathcal{T} = \{\mathrm{tr}_a, a = 1 \cdots n_T\}$ denote the set of point trajectories of video sequence I. Between each pair of trajectories tr_a, tr_b we compute motion affinities $\mathbf{A}_T(a, b)$ encoding their long range motion similarity. Each region r_i is characterized by i) an affine motion model $\mathbf{w}_i^R : \mathcal{P} \to \mathbb{R}^2$, fitted to its optical flow estimates, that for each pixel outputs a predicted displacement vector (u, v), and ii) a set of point trajectories $\mathcal{T}_i$ overlapping with its pixel mask.

We set motion affinities between each pair of regions r_i, r_j to be:

$$\mathbf{A}_R(i,j) = \begin{cases} \dfrac{\sum\limits_{a \in \mathcal{T}_i, b \in \mathcal{T}_j} \mathbf{A}_T(a,b)}{|T_i||T_j|}, & \text{if} \quad \frac{|\mathcal{T}_i|}{|r_i|}, \frac{|\mathcal{T}_j|}{|r_j|} > \alpha, \\ \dfrac{\sum\limits_{p \in r_i \cup r_j} \exp\left(-\frac{1}{\sigma} ||\mathbf{w}_j^R(p) - \mathbf{w}_i^R(p)||_2\right)}{|r_i \cup r_j|}, & \text{o/w}, \end{cases}$$

where $|S|$ denotes cardinality of set S and α a density threshold that depends on the trajectory sampling step. The first case measures mean trajectory affinity between regions, used if both regions are well covered by trajectories. The second case measures compatibility of region affine models, being high in case the two regions belong to the projection of the same 3D planar surface.

6.3.2 Region detection-driven repulsions

We consider a set of poselet shoulder detections of Bourdev et al. (2010), $\mathcal{D} = \{\mathrm{d}_q, q = 1 \cdots n_D\}$. Let mask M_q denote the pixel set overlapping with d_q. We show mask M_q

of a shoulder detection in Figure 6.2. Each detection $\mathrm{d}_q \in \mathcal{D}$ induces implicitly figure-ground repulsive forces between the regions associated with its interior and exterior. Let $x_q^F, x_q^B \in \{0,1\}^{n_R \times 1}$ denote foreground and background region indicators for detection d_q and let U_q denote the pixel set outside a circle of radius that upper-bounds the possible arm length, as estimated from shoulder distance, shown also in Figure 6.2. We have:

$$\begin{aligned} x_q^F(i) &= \delta\left(\frac{|r_i \cap M_q|}{|r_i|} > 0.9\right), \quad i = 1 \cdots n_R,\ q = 1 \cdots n_D \\ x_q^B(i) &= \delta\left(\frac{|r_i \cap U_q|}{|r_i|} > 0.5\right), \quad i = 1 \cdots n_R,\ q = 1 \cdots n_D. \end{aligned} \tag{6.1}$$

Repulsions are induced between foreground and background regions of each detector response:

$$\mathbf{R}_R(i,j|\mathcal{D}) = \max_{q|\mathrm{d}_q \in \mathcal{D}} x_q^F(i) x_q^B(j) + x_q^B(i) x_q^F(j).$$

Let $\mathcal{S}(\mathcal{D})$ denote the set of repulsive edges:

$$\mathcal{S}(\mathcal{D}) = \{(i,j) \text{ s.t. } \exists\, \mathrm{d}_q \in \mathcal{D},\ x_q^F(i) x_q^B(j) + x_q^B(i) x_q^F(j) = 1\}.$$

6.3.3 Steering cut

We combine motion-driven affinities and detection-driven repulsions in one region affinity graph by canceling motion affinities between repulsive regions:

$$\mathbf{W}^{\text{steer}}(\mathcal{D}) = (\mathbf{1}_{n_R \times n_R} - \mathbf{R}_R(\mathcal{D})) \cdot \mathbf{A}_R. \tag{6.2}$$

Inference in our model amounts to selecting the part detections $\mathcal{D}$ and clustering the image regions $\mathcal{R}$ into groups that ideally correspond to the left and right upper arms, left and right lower arms, torso and background. In each video sequence, we infer the most temporally coherent shoulder detection sequence given poselet shoulder activations in each frame. This works very well since people are mostly upright in the TV shows we are working with, which makes their shoulders easily detectable. As such, instead of simultaneously optimizing over part selection and region clustering as we did in Chapter 4, we fix the detection set $\mathcal{D}$ during region clustering.

Let $X \in \{0,1\}^{n_R \times K}$ denote the region cluster indicator matrix, X_k denote the kth column of X, respectively, and K denote the total number of region clusters. Let $\mathbf{D}_{\mathbf{W}^{\text{steer}}}$ be a diagonal degree matrix with $\mathbf{D}_{\mathbf{W}^{\text{steer}}} = \text{Diag}(\mathbf{W}^{\text{steer}} \mathbf{1}_{n_T})$. We maximize the following constrained normalized cut criterion in the steered graph:

Steering Cut:

$$\begin{aligned} \max_{X}. \quad & \epsilon(X|\mathcal{D}) = \sum_{k=1}^{K} \frac{X_k^T \mathbf{W}^{\text{steer}}(\mathcal{D}) X_k}{X_k^T \mathbf{D}_{\mathbf{W}^{\text{steer}}(\mathcal{D})} X_k} \\ \text{s.t.} \quad & X \in \{0,1\}^{n_R \times K}, \quad \sum_{k=1}^{K} X_k = \mathbf{1}_{n_R}, \\ & \forall (i,j) \in \mathcal{S}(\mathcal{D}), \quad \sum_{k=1}^{K} X_k(i) X_k(j) = 0. \end{aligned} \tag{6.3}$$

The set of constraints in the last row demand regions connected with repulsive links in $\mathcal{S}(\mathcal{D})$ to belong to different clusters.

We solve the constrained normalized cut in Eq. 6.3 by propagating information from confident (figure-ground seeds, saliently moving regions) to non-confident places, by iteratively merging regions close in embedding distance and recomputing region affinities, similar in spirit to the multiscale segmentation in Sharon et al. (2000). Specifically, we iterate between:

1. Computing embedding region affinities $\hat{\mathbf{W}} = V \Lambda V^T$, where $(V, \Lambda = \text{Diag}(\lambda))$ are the top K eigenvectors and eigenvalues of $\mathbf{D}_{\mathbf{W}^{\text{steer}}}^{-1} \mathbf{W}^{\text{steer}}$.

2. Merging regions $\mathrm{r}_{\tilde{i}}, \mathrm{r}_{\tilde{j}}$ with the largest embedding affinity, $(\tilde{i}, \tilde{j}) = \underset{(i,j) \notin \mathcal{S}(\mathcal{D})}{\arg\max}\, \hat{\mathbf{W}}(i,j)$. We update $\mathbf{W}^{\text{steer}}$ with the motion affinities of the newly formed region.

Matrix $\mathbf{W}^{\text{steer}}$ shrinks in size during the iterations. In practice, we would merge multiple regions before recomputing affinities and the spectral embedding of $\mathbf{W}^{\text{steer}}$. Iterations terminate when motion affinities in $\mathbf{W}^{\text{steer}}$ are below a threshold. We extracted region clusters X_k with high normalized cut scores $\frac{X_k^T \mathbf{W}^{\text{steer}} X_k}{X_k^T \mathbf{D}_{\mathbf{W}^{\text{steer}}} X_k}$ even before the termination of iterations. While upper arms are very hard to delineate from the torso interior, lower arms would often correspond to region clusters, as shown in Figure 6.2. Foreground and background shoulder seeds help segmenting lower limbs by claiming regions of torso foreground and background, which should not be linked to the lower limb cluster. This is necessary for reliably estimating the elbow from the lower limb endpoint, as described in Section 6.3.4.

We compute steered cuts in graphs from multiple segmentation maps $\mathcal{R}$ by thresholding the output of globalPb at 3 different thresholds. Note that in coarser region maps, a lower limb may correspond to one region.

6.3.4 Matching pose segmentations to exemplars

For each region cluster $X_k \in \{0,1\}^{n_R}$ we fit an ellipse and hypothesize joint locations J_k^1, J_k^2 at the endpoints of the major axis. Using J_k^1, J_k^2 and detected shoulder locations, we select pose exemplars close in body joint configuration as measured by the partial Procrustes distance between the corresponding sets of body joints (we do not consider scaling). We compute a segment to exemplar matching score according to pixelwise contour correspondence between exemplar boundary contours and segment boundary contours, penalizing edgel orientation difference. For this we adapted Andrew Goldberg's implementation of Cost Scale Algorithm (used in the code package of Arbelaez et al. (2009)) to oriented contours. We also compute a unary score for each segmentation proposal, independent of exemplar matching, according to i) chi-square distance between the normalized color histograms of the hypothesized hand and the detected face, and ii) optical flow magnitude measured at the region endpoints J_k^1, J_k^2, large motion indicating a hand endpoint. We combine the 3 scores with a weighted sum.

Confidently matched pose segments recover body parts that would have been missed by the pose detectors due to overwhelming surrounding clutter or misalignment of pose. We select the two segmentations with the highest matching scores, that correspond to left and right arm kinematic chains. Each kinematic chain is comprised of upper and lower arms $\mathrm{d}_u, \mathrm{d}_l$ connected at the elbow body joint $J_{u,l}$, as shown in Figure 6.2.

6.4 From pose to flow

We use the estimated body pose to help motion estimation of lower limbs. Human body limbs are hard to track accurately with general motion estimation techniques, such as optical flow methods, due to large rotations, deformations, and ambiguity of correspondence along their medial axis (aperture problems). These are challenges even for descriptor augmented flow methods Brox and Malik (2010a); Xu et al. (2012c) since descriptor matches may "slide" along the limb direction.

We incorporate knowledge about articulation points and region stiffness in optical flow. Articulation points correspond to rotation axes and impose kinematic constraints on the body parts they are connected to. They can thus suggest rotations of parts and predict occlusions due to large limb motion.

6.4.1 Articulated optical flow

We use our pose labelled segmentations to infer dense displacement fields for body parts, which we call articulated flow fields. Given an arm articulated chain (left or right), let $M_u,\ M_l$ denote the masks of the corresponding upper and lower arms d_u, d_l, linked at the elbow location $J_{u,l}$. Let $\mathbf{w} = (\mathrm{u}, \mathrm{v})$ denote the dense optical flow field. Let $\mathbf{w}_u^D, \mathbf{w}_l^D$ denote affine motion fields of parts d_u and d_l i.e. functions $\mathbf{w}_u^D : M_u \rightarrow \mathbb{R}^2$. Let $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$, $\epsilon = 0.001$ denote the frequently used convex robust function, and $\phi_u(\mathbf{x}) = \exp(-|(I_2(\mathbf{x} + \mathbf{w}_u^D(\mathbf{x})) - I_1(\mathbf{x}))|^2/\sigma)$ the pixelwise confidence of the affine field $\mathbf{w}_u^D$.

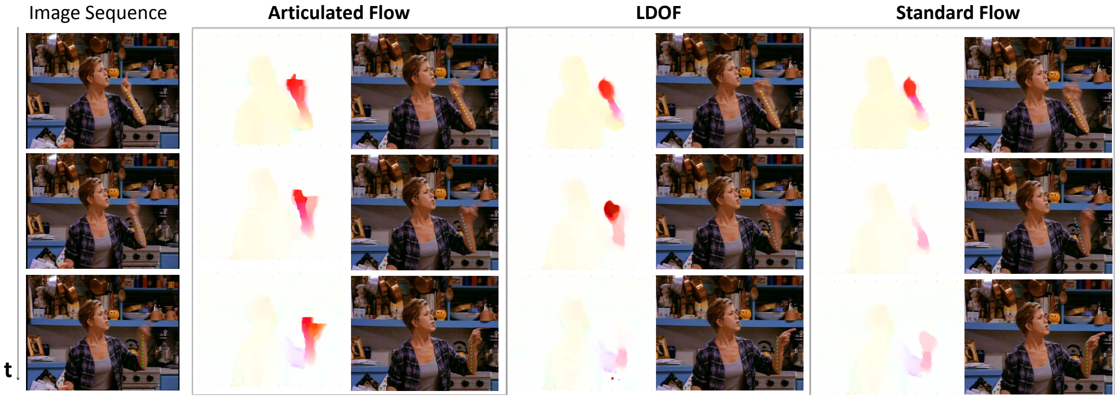


Figure 6.3: Articulated optical flow. Left: A video sequence ordered in time with fast rotations of left lower arm. Right: Motion flow is displayed as a) color encoded optical flow image, and b) the warped image using the flow. We compare the proposed articulated flow, Large Displacement Optical Flow (LDOF) Brox and Malik (2010a) and coarse-to-fine variational flow of Brox et al. (2004). The dashed lines in the warped image indicate the ideal position of the lower arm. If the flow is correct, the warped arm will be aligned on the dashed line. Standard optical flow cannot follow fast motion of the lower arm in most cases. LDOF, which is descriptor augmented, recovers correctly the fast motion in case of correct descriptor matches. However, when descriptors capture the hand but miss the arm, hand and arm appear disconnected in the motion flow space (2nd row). Knowing the rough body articulation points allows to restrict our motion model to be a kinematic chain along the body parts. The resulting articulated motion flow is more accurate.

The cost function for our articulated optical flow reads:

$$\min_{\mathbf{w},\mathbf{w}_u^D,\mathbf{w}_l^D}. \quad E(\mathbf{w},\mathbf{w}_u^D,\mathbf{w}_l^D) = \int_\Omega \Psi(|I_2(\mathbf{x}+\mathbf{w}(\mathbf{x})) - I_1(\mathbf{x})|^2)d\mathbf{x} \tag{6.4}$$

$$+\gamma \int_\Omega \Psi(|\nabla \mathrm{u}(\mathbf{x})|^2 + |\nabla \mathrm{v}(\mathbf{x})|^2)d\mathbf{x}+ \tag{6.5}$$

$$\beta \sum_{e\in\{u,l\}} \int_{M_e} \phi_e(\mathbf{x})\Psi(|\mathbf{w}(\mathbf{x}) - \mathbf{w}_e^D(\mathbf{x})|^2)d\mathbf{x} \tag{6.6}$$

$$+ \sum_{e\in\{u,l\}} \int_{M_e} \Psi(|I_2(\mathbf{x}+\mathbf{w}_e^D(\mathbf{x})) - I_1(\mathbf{x})|^2)d\mathbf{x} \tag{6.7}$$

$$\text{s.t.} \quad \mathbf{w}_u^D(J_{u,l}) = \mathbf{w}_l^D(J_{u,l}). \tag{6.8}$$

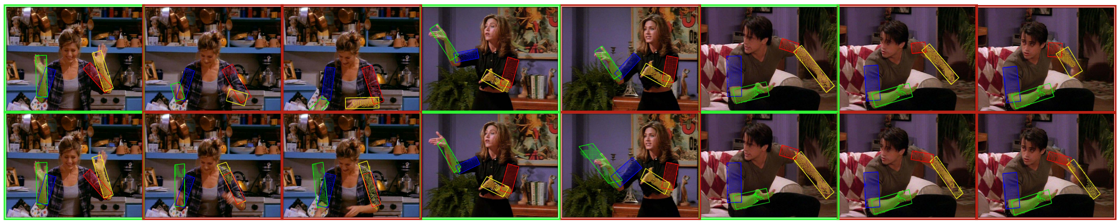


Figure 6.4: Comparison of articulated flow and standard optical flow. Top Row: Pose propagation with articulated optical flow. Bottom Row: Pose propagation with affine motion fitting to the optical flow estimates of Brox and Malik (2010a). Green outline indicates frames with pose detection and red outline indicates frames with the propagated pose. Limb motion is often too erratic to track with standard optical flow schemes, which drift to surroundings under wide deformations.

The first two terms of Eq. 6.4 correspond to the standard pixel intensity matching and spatial regularization in optical flow, as in Brox et al. (2004). For brevity we do not show the image gradient matching term. The third term penalizes deviations of the displacement field $\mathbf{w}$ from the affine fields $\mathbf{w}_u^D, \mathbf{w}_l^D$, weighted by the pixelwise confidence of the affine displacements $\phi_u(\mathbf{x}), \phi_l(\mathbf{x})$. The forth term measures the fitting cost of the affine fields. The constraint requires the affine displacements predicted for the articulated joint by the two affine fields to be equal.

We solve our articulated flow model in Eq. 6.4 by computing coarse affine models for upper and lower arms and then injecting their affine displacements as soft constraints in an optical flow computation for the kinematic chain. For computing the two kinematically constrained affine fields we use "hybrid" tracking: for upper arms or the background, standard optical flow displacements are often reliable, since their motion is not erratic. We use such flow displacements to propagate foreground and background of the arm kinematic chain from the previous frame, and compute an affine motion field for the upper arm $\mathbf{w}_u^D$. Such propagation constrains i) the possible displacement hypotheses of the articulation point $J_{u,l}$, and ii) the possible affine deformations of the lower limb d_l. We enumerate a constrained pool of affine deformation hypotheses for the lower limb: it cannot be part of the background and should couple at the articulation joint with $\mathbf{w}_u^D$. We evaluate such hypotheses according to a figure-ground Gaussian Mixture Model on color computed in the initial detection frame, and Chamfer matching between the contours inside the hypothesized part bounding box and the body part contours of the previous frame, transformed according to each affine hypothesis. The highest scoring deformation hypothesis is used to compute our lower limb affine field $\mathbf{w}_l^D$. Notably, we also experimented with the method of Datta et al. (2008) but found that it could not deal well with self-occlusions of the arms, frequent under wide deformation, as also noted by the authors.

Given part affine fields $\mathbf{w}_u^D, \mathbf{w}_l^D$, Eq. 6.4 is minimized with respect to displacement field $\mathbf{w}$ using the coarse-to-fine nested fixed point iteration scheme proposed in Sundaram et al. (2010). The affine displacements $\mathbf{w}_u^D, \mathbf{w}_l^D$ receive higher weights at coarse pyramid levels and are down-weighted at finer pyramid levels as more and more image evidence is taken into account, to better adapt to the fine-grain details of part motion, that may deviate from an affine model. We show results of the articulated flow in Figure 6.3. Articulated flow preserves the integrity of the fast moving lower arm and hand. In descriptor augmented optical flow of Sundaram et al. (2010) the motion estimate of the arm "breaks" in cases of missing reliable descriptor match to capture its deformation. Standard coarse-to-fine flow misses the fast moving hand whose motion is larger that the its spatial extent.

We propagate our body segmentations in time using articulated optical flow trajectories, as shown in Figure 6.4. The fine grain trajectories can adapt to the part masks under occlusion while the coarse affine models prevents drifting under erratic deformations. We compare with affine fitting to standard flow estimates in Figure 6.4. Ambiguities of limb motion estimation due to self occlusions, non-discriminative appearance and wide deformations cause flow estimates to drift, in absence of pose informed kinematic constraints.

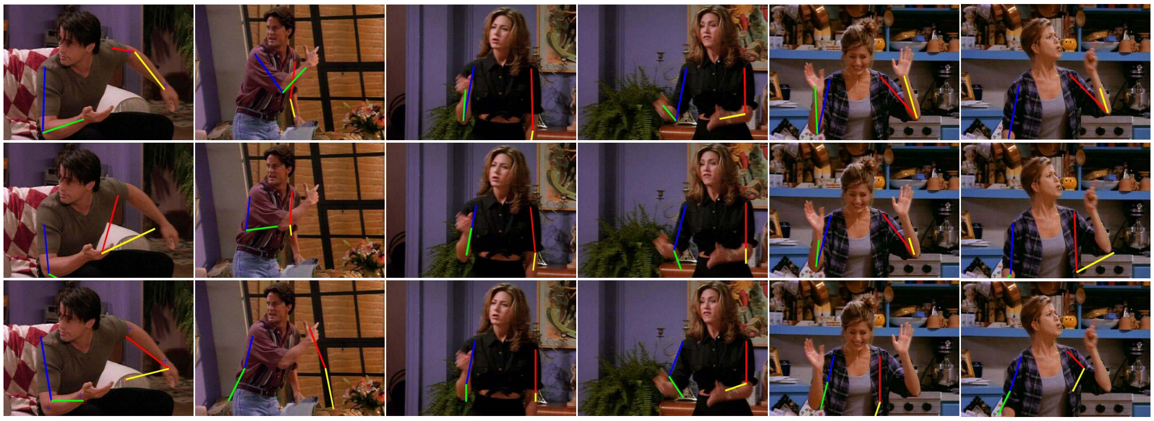


Figure 6.5: Pose detection results in the Friends dataset. Top Row: Pose from flow under wide body deformation. Middle Row: Results of Sapp et al. (2011). Bottom Row: Results of Park and Ramanan (2011).

6.5 Experiments

We test our method on video clips from the popular TV series "Friends", part of the dataset introduced in Sapp et al. (2011). We use 15 video sequences with widely deformed body pose in at least one frame. Each sequence is 60 frames long. The characters are particularly expressive and use a lot of interesting gestures in animated conversations.

In each video sequence, we infer the most temporally coherent shoulder sequence using detection responses from the poselet detector of Bourdev et al. (2010). This was able to correctly delineate the shoulder locations in each frame. We held out a pose exemplar set from the training set of the Friends dataset, to match our steered segmentation proposals against. For each exemplar we automatically extract a set of boundary contours lying inside the ground truth body part bounding boxes of width one fifth of the shoulder distance. We evaluate our full method, which we call "flow→ pose → flow", as well as our pose detection step only, without improving the motion estimation, but rather propagating the pose in time by fitting affine motion models to standard optical flow Brox and Malik (2010a). We call this baseline "flow → pose".

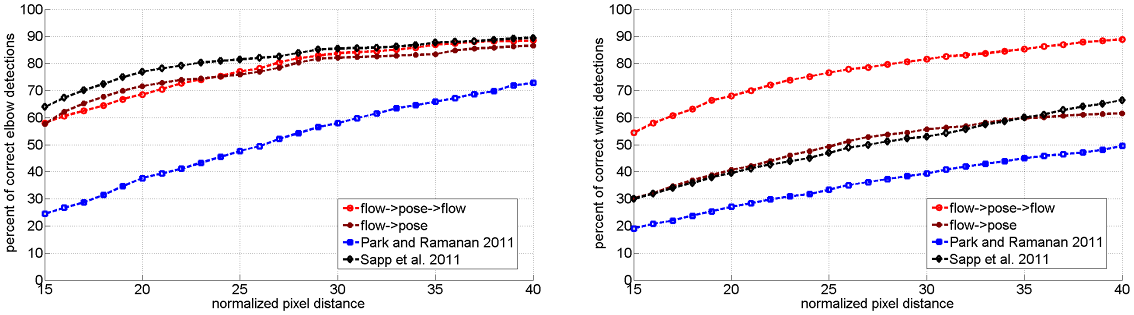


Figure 6.6: Quantitative pose detection results in the Friends dataset. Evaluation of elbow and wrist localization.

We compare against two approaches for human pose estimation in videos: 1) the system of Sapp et al. (2011). It uses a loopy graphical model over body joint locations in space and time. It combines multiple cues such as Probability of Boundary, optical flow edges and skin color for computing unary and pairwise part potentials. It is trained on a subset of the Friends dataset. It assumes the shoulders positions known and focuses on lower arm detection. 2) The system of Park and Ramanan (2011). It extends the state-of-the-art static pose detector of Yang and Ramanan (2011) for human pose estimation in videos by keeping N best pose samples per frame and inferring the most coherent pose sequence across frames using dynamic programming. We retrained the model with the same training subset of Friends as Sapp et al. (2011) but the performance did not improve due to the low number of training examples.

Our performance evaluation measure is percentage of elbows and wrists within a radius from ground truth locations, same as in Section 4.5. We show in Figure 6.6 the percentage of correct wrists and elbows as we vary the radius threshold. The flow→pose→flow and pose→flow methods perform similarly in tracking the detected elbows since upper arms do not frequently exhibit erratic deformations. The two methods though have a large performance gap when tracking lower arms, whose wide frame-to-frame deformations cause standard optical flow to drift. This demonstrates the importance of improving the motion estimation via articulation constraints for tracking the pose in time.

Our method provides accurate spatial support for the body parts, robust to intra-body and scene occlusions. In contrast to standard pose detectors, and also our baseline systems, our method does not require all body parts to be present in each frame. The lack of specified wrist and elbow detectors makes our wrist and elbow localization occasionally poor (see last column of Figure 6.5) while lying inside the body part.

6.6 Discussion

We proposed an approach that detects human body poses by steering cut on motion grouping affinities of lower limbs and figure-ground repulsions from shoulder detections. We focus on detecting rare, transient in nature poses, often under-represented in the datasets and missed by pose detectors. Our segmentations extract lower limbs from their surrounding intra-body and background clutter. Arm articulated chains resulting from matching such segmentations to exemplars, are used to provide feedback to dense body motion estimation about articulation points and region stiffness. Resulting flow fields can deal with large per frame deformations of body parts and propagate the detected pose in time, during its deforming posture. Our flow to pose to flow process is able to infer poses under wide deformations that would have been both too hard to detect and too hard to track otherwise.

Chapter 7

Conclusion

Never mistake motion for action.

— Ernest Hemingway

We have presented methods that combine temporal correspondences and spatial affinities of video pixels with object detectors for parsing video configurations that are rare, i.e., non repeatable in the training sets. Our goal is not to segment detected objects but to use spatio-temporal segmentation to improve object detection and pose estimation in videos.

Video segmentation as a task on its own is important for semi-supervised or unsupervised learning of objects or activities. We envision a system that automatically asks human Mechanical Turk workers to label segments with low detection confidence scores. Video segments and their saliency can cast attention to important parts of a video scene and discard usually uninteresting static backgrounds. Further, motion provides a strong cue for delineating object boundaries, potentially saving a lot of time from human labelers: it often suffices to provide a box around the object for a local figure-ground motion segmentation to provide the right object support and propagate it in time. To this end, we see video segmentation and labeling as a means of obtaining large amounts of labeled training data in a never-ending type of learning setup. This knowledge can be used for parsing more difficult sensor input, such as still images.

Current literature on motion estimation is divided between 1) works that assume a closed world and use physical or statistical constraints to estimate the motion of known video content, e.g., a human face in Garg et al. (2013) or a pair of interacting hands in Oikonomidis et al. (2012), and 2) works that are oblivious to object knowledge, such as optical flow methods, that estimate temporal correspondences from pixel appearance. We envision a hybrid system where motion models and physical constraints are employed on-the-fly to resolve ambiguities of appearance based tracking in open world applications. Learning how people move from large amounts of multi-view 3D data, where groundtruth temporal correspondences can be obtained automatically, and developing compact spatio-temporal representations that can be utilized in unconstrained video input, is important for dealing with ambiguities in motion estimation and is a goal of our future work.

Information about scene functionality and surrounding objects is necessary for better tracking of people and their body pose and interpreting their actions and intentions. We do not model scene context mostly as an effort for the methods to remain as general as possible. There is a natural trade-off between contextual information to be used and generality of an approach, since there will always be videos where information about the scene or surrounding objects is not easy to recover. Coupling the representations developed in this thesis with scene and object functionality for activity understanding is an interesting path of future work.

While in this thesis our focus is to disentangle the people and their body pose under interactions, often times, especially under meaningful interactions for collaborative activity, e.g., shake hands, hugging etc. the entangled ensemble template is stable and useful. We plan to explore a less semantically meaningful and more data driven detector design from image and optical flow gradients which we think is necessary for progress in understanding people's interactions.

Bibliography

Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1442–1456, 2011. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.201. 22, 26

Arnon Amir and Michael Lindenbaum. Grouping based non-additive verification. *TPAMI*, 20, 1998. 54, 83

Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008. 3, 59, 77

Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009. 33, 35, 36, 45, 92, 122

S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV*, 1995. 25

A. Ayvaci and S. Soatto. Detachable object detection: Segmentation and depth ordering from short-baseline video. *TPAMI*, October 2012. 27

A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *International Journal of Computer Vision*, 97(3), May 2012. 27

Andrew D. Bagdanov, Alberto Del Bimbo, Fabrizio Dini, Giuseppe Lisanti, and Iacopo Masi. Compact and efficient posterity logging of face imagery for video surveillance. 2012. 80, 81

Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. Diverse m-best solutions in markov random fields. In *Proceedings of the 12th European conference on Computer Vision - Volume Part V*, ECCV'12, pages 1–16, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-33714-7. doi: 10.1007/978-3-642-33715-4_1. URL http://dx.doi.org/10.1007/978-3-642-33715-4_1. 93, 94, 98, 108

J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 59

Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *J. Image Video Process.*, 2008. 80

C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV*, 2008. 58

M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996. 21

Eran Borenstein and Shimon Ullman. Combined top-down/bottom-up segmentation. *TPAMI*, 30. 54

Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010. URL http://www.eecs.berkeley.edu/~lbourdev/poselets. 3, 55, 77, 91, 94, 108, 115, 117, 119, 127

Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision*, 70(2):109–131, November 2006. ISSN 0920-5691. doi: 10.1007/s11263-006-7934-5. URL http://dx.doi.org/10.1007/s11263-006-7934-5. 18

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23, 2001. 10, 12, 16, 18

Matthieu Bray, Pushmeet Kohli, and Philip H. S. Torr. Posecut: simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *Proceedings of the 9th European conference on Computer Vision - Volume Part II*, ECCV'06, pages 642–655, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33834-9, 978-3-540-33834-5. doi: 10.1007/11744047_49. URL http://dx.doi.org/10.1007/11744047_49. 95

Christoph Bregler and Jitendra Malik. Tracking people with twists and exponential maps. In *CVPR*, 1998. 7, 93, 116

Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009. 4, 54, 59

William Brendel and Sinisa Todorovic. Video object segmentation by tracking regions. In *ICCV*, pages 833–840, 2009. 25

William Brendel, Mohamed R. Amer, and Sinisa Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011. 4, 54, 58, 59

Gabriel J. Brostow and Roberto Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *CVPR*, 2006. 22, 26

T. Brox, B. Rosenhahn, D. Cremers, and H.-P. Seidel. High accuracy optical flow serves 3-D pose tracking: exploiting contour and flow based constraints. In *ECCV*, 2006. 116

Thomas Brox and Jitendra Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *TPAMI*, 2010a. 7, 21, 22, 123, 124, 125, 127

Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*. 2010b. 22, 26, 29, 38, 40, 50, 51, 101

Thomas Brox, Andrs Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. pages 25–36. Springer, 2004. 6, 21, 27, 118, 124, 125

A. Y. C. Chen and J. J. Corso. Propagating multi-class pixel labels throughout video frames. In *Proceedings of Western New York Image Processing Workshop*, 2010. URL http://www.cse.buffalo.edu/~jcorso/pubs/wnyipw2010_video.pdf. xii, 38, 41, 42

Yizong Cheng. Mean shift, mode seeking, and clustering. *TPAMI*, 17, 1995. 12

J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. *ICCV*, 1995. 22, 26

Ankur Datta, Yaser Ajmal Sheikh, and Takeo Kanade. Linear motion estimation for systems of articulated planes. In *CVPR*, 2008. 7, 116, 126

Konstantinos G. Derpanis and Richard P. Wildes. Detecting spatiotemporal structure boundaries: Beyond motion discontinuities. In *Computer Vision — ACCV 2009*, volume 5995, chapter 29. 2010. URL http://dx.doi.org/10.1007/978-3-642-12304-7_29. 27

Chaitanya Desai and Deva Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV (4)*, pages 158–172, 2012a. 92, 95

Chaitanya Desai and Deva Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV (4)*, pages 158–172, 2012b. 58

O. Duchenne, J. Y. Audibert, R. Keriven, J. Ponce, and F. Segonne. Segmentation by transduction. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008. ISSN 1063-6919. doi: 10.1109/cvpr.2008.4587419. URL http://dx.doi.org/10.1109/cvpr.2008.4587419. 16

Marcin Eichner and Vittorio Ferrari. We Are Family: Joint Pose Estimation of Multiple Persons. In *ECCV*. 2010. 95

E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009. 26

A. Ess, B. Leibe, and L. Van Gool. Depth and appearance for mobile scene analysis. In *International Conference on Computer Vision (ICCV'07)*, 2007. 77

Ronan Fablet and Michael J. Black. Automatic detection and tracking of human motion with a view-based representation. In *ECCV*, 2002. 116

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32, 2010. 77, 114

V. Ferrari, M.J. Marn-Jimnez, and A. Zisserman. 2D human pose estimation in TV shows. In D. Cremers et al., editor, *Statistical and Geometrical Approaches to Visual Motion Analysis*, LNCS, pages 128–147. Springer, 1st edition, 2009a. 116

V. Ferrari, M.J. Marn-Jimnez, and A. Zisserman. 2d human pose estimation in tv shows. In D. Cremers et al., editor, *Statistical and Geometrical Approaches to Visual Motion Analysis*, LNCS, pages 128–147. Springer, 1st edition, 2009b. 94

Matthieu Fradet, Philippe Robert, and Patrick Pérez. Clustering point trajectories with various life-spans. In *CVMP*, 2009. 22, 26

Katerina Fragkiadaki and Jianbo Shi. Exploiting motion and topology for segmenting and tracking under entanglement. In *CVPR*, 2011. 11, 22, 26

Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, pages 1846–1853, 2012a. 11, 24

Katerina Fragkiadaki, Weiyu Zhang, Jianbo Shi, and Elena Bernardis. Structural-flow trajectories for unravelling 3d tubular bundles. In *MICCAI (3)*, pages 631–638, 2012b. 11

Katerina Fragkiadaki, Weiyu Zhang, Geng Zhang, and Jianbo Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *ECCV*, 2012c. 11

Katerina Fragkiadaki, Han Hu, and Jianbo Shi. Pose from flow and flow from pose. In *CVPR*, 2013. 11

Fabio Galasso, Roberto Cipolla, and Bernt Schiele. Video segmentation with superpixels. In *ACCV (1)*, pages 760–774, 2012. 36

J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.70. 59, 88

Jean Gallier. Notes on elementary spectral graph theory, applications to graph clustering. In *Unpublished manuscript*, 2013. 31

S. Gammeter, A. Ess, T. Jaeggli, K. Schindler, B. Leibe, , and L. van Gool. Articulated multibody tracking under egomotion. In *European Conference on Computer Vision (ECCV'08)*, LNCS. Springer, October 2008. 95

Dashan Gao, Vijay Mahadevan, and Nuno Vasconcelos. On the plausibility of the discriminant center-surround hypothesis for visual saliency. *Journal of vision*, 8, 2008. 25, 47

Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. June 2013. 21, 131

J. J. Gibson. The perception of the visual world. *The American Journal of Psychology,*, 64: 622–625, October 1951. URL http://www.getcited.org/pub/101498031. 20, 21

Haifeng Gong, Jack Sim, Maxim Likhachev, and Jianbo Shi. Multi-hypothesis motion planning for visual object tracking. In *ICCV*, 2011. 59, 79, 88

Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 4

Leo Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, November 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.233. URL http://dx.doi.org/10.1109/TPAMI.2006.233. 10, 13, 16, 17

Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph based video segmentation. *CVPR*, 2010. 26

Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011. URL http://www.eecs.berkeley.edu/~lbourdev/poselets. 54

Xuming He and Alan Yuille. Occlusion boundary detection using pseudo-depth. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 539–552. Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-15561-1_39. URL http://dx.doi.org/10.1007/978-3-642-15561-1_39. 27

Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *ECCV (3)*, pages 340–353, 2012. 20

Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *ARTIFICAL INTELLIGENCE*, 17:185–203, 1981. 21

Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008a. 59

Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*. 2008b. 4

Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *ICCV*, pages 2220–2227, 2011a. 95

Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *ICCV*, 2011b. 54

Allan D. Jepson, David J. Fleet, and Michael J. Black. A layered motion representation with occlusion and compact spatial support. In *In Proc. of European Conference on Computer Vision*, pages 692–706. Springer-Verlag, 2002. 25

Hao Jiang. Human pose estimation using consistent max-covering. In *ICCV*, 2009. 115

Thorsten Joachims. Transductive learning via spectral graph partitioning. In *In ICML*, 2003. 16

Gunnar Johansson. Visual perception of biological motion and a model for its analysis. In *Percept. Psychophys.*, volume 14, pages 201–211, 1973. 1, 20

Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, 2011. 91, 92, 94, 118

Shanon X. Ju, Michael J. Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *FG*, 1996. 94, 116

Shizuo Kakutani. Markoff process and the dirichlet problem. In *Proc. Japan Acad.*, volume 21, 1945. 17

Kris Kitani, Brian D. Ziebart, J. Andrew (Drew) Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision*. Springer, October 2012. 59, 88

Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321295358. 18

Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. Mrf energy minimization and beyond via dual decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3):531–552, 2011. 10

Xiangyang Lan and D.P. Huttenlocher. A unified spatio-temporal articulated model for tracking. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–722–I–729 Vol.1, 2004. doi: 10.1109/CVPR.2004.1315103. 94

B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3D scene analysis from a moving vehicle. In *CVPR*, 2007. 54

Victor S. Lempitsky, Stefan Roth, and Carsten Rother. Fusionflow: Discrete continuous optimization for optical flow estimation. In *CVPR*, 2008. 21

Anat Levin and Yair Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, 2006. 54

J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*, 2011. 24

Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981. 6, 21, 22

Vijay Mahadevan and Nuno Vasconcelos. Spatiotemporal saliency in dynamic scenes. *TPAMI*, 32, 2010. 25, 47

Michael Maire, Pablo Arbelaez, Charless Fowlkes, and Jitendra Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008. 33

Michael Maire, Stella Yu, and Pietro Perona. Object detection and segmentation from joint embedding of parts and pixels, 2011. 16

Tomasz Malisiewicz and Alexei A. Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *NIPS*, 2009. 58

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 44

Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *NIPS*, 2001. 15

E. Memin and P. Perez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *Image Processing, IEEE Transactions on*, 7(5), 1998. 21

Dennis Mitzel, Esther Horbert, Andreas Ess, and Bastian Leibe. Multi-person tracking with sparse detection and continuous segmentation. In *ECCV*, 2010. 58

Greg Mori, Xiaofeng Ren, Alexei A. Efros, and Jitendra Malik. Recovering human body configurations: combining segmentation and recognition. In *Proc. IEEE Conf. Comput. Vision and Pattern Recogn.*, volume 2, pages 326–333, 2004. 54, 117

Daniel Munoz, J. Andrew (Drew) Bagnell, and Martial Hebert. Stacked hierarchical labeling. In *European Conference on Computer Vision (ECCV)*, September 2010. 16

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001. 15, 33

H. C. Nothdurft. Feature analysis and the role of similarity in preattentive vision. *Perception and Psychophysics*, 52, 1992. 20, 47

Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011. 24

Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, pages 1862–1869, 2012. 131

Kenji Okuma, Ali Taleghani, Nando De Freitas, O De Freitas, James J. Little, and David G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004. 4, 59

Dennis Park and Deva Ramanan. N-best maximal decoders for part models. In *ICCV*, 2011. 4, 93, 94, 98, 108, 127, 128

M. Pawan Kumar, P. H. Torr, and A. Zisserman. Learning layered motion segmentations of video. *Int. J. Comput. Vision*, 76(3):301–319, 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0064-x. URL http://dx.doi.org/10.1007/s11263-007-0064-x. 25

Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc J. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009. 59

Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Occlusion patterns for object class detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Portland, OR, 2013. 58

P.Ochs and T.Brox. Higher order motion models and spectral clustering.

In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. URL http://lmb.informatik.uni-freiburg.de//Publications/2012/OB12. 26, 27

Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä. Segmenting salient objects from images and videos. In *ECCV (5)*, pages 366–379, 2010a. 25, 47

Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkil. Segmenting salient objects from images and videos. In *ECCV*. 2010b. 47

Deva Ramanan, D. A. Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. *CVPR*, 2005a. 116

Deva Ramanan, D. A. Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. *CVPR*, 2005b. 94

S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *CVPR*, 2008. 22, 26

Avinash Ravichandran, Chaohui Wang, Michalis Raptis, and Stefano Soatto. Superfloxels: A mid-level representation for video sequences. In *ECCV Workshops (3)*, 2012. 27

J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV*, 1995. 116

Xiaofeng Ren and Jitendra Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, 2007. 58

Bryan C. Russell, AlexeiA. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 71

Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. 2011. 58

Ben Sapp and Ben Taskar. Modec:multimodal decomposable models for human pose estimation. In *CVPR*, 2013. 6, 91, 93, 94, 106, 110

Benjamin Sapp, David Weiss, and Ben Taskar. Parsing human motion with stretchable models. In *CVPR*, 2011. 94, 100, 116, 127, 128

Eitan Sharon, Achi Brandt, and Ronen Basri. Fast multiscale image segmentation. In *CVPR*, 2000. 121

Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 1154–, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 81-7319-221-9. URL http://dl.acm.org/citation.cfm?id=938978.939083. 23, 25

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *TPAMI*, 2000. 12, 13, 14, 31

Jianbo Shi and Carlo Tomasi. Good features to track. In *CVPR*, 1994. 22

Jamie Shotton, Toby Sharp, Alex Kipman, Andrew W. Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, 2013. 6

Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. Part-based multiple-person tracking with partial occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1815–1821. IEEE, 2012. 58

Leonid Sigal and Michael J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *In CVPR 2006*, pages 2041–2048, 2006. 95

Leonid Sigal, Michael Isard, Horst Haussecker, and Michael J. Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *Int. J. Comput. Vision*, 98(1):15–48, 2012. ISSN 0920-5691. doi: 10.1007/s11263-011-0493-4. URL http://dx.doi.org/10.1007/s11263-011-0493-4. 94

Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, and Dimitris Metaxas. Discriminative density propagation for 3d human motion estimation. In *In CVPR*, pages 390–397, 2005. 94

Stefano Soatto. Motion competition: a variational approach to piecewise parametric motion segmentation. *Int. J. Comput. Vision*, 62:249–265, 2005. 25

Andrew Stein, Derek Hoiem, and Martial Hebert. Learning to find object boundaries using motion cues. In *IEEE International Conference on Computer Vision (ICCV)*, October 2007. 27, 38, 41, 43, 45

Andrew N. Stein, Thomas S. Stepleton, and Martial Hebert. Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In *CVPR*. 27

Deqing Sun, Erik B. Sudderth, and Michael J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *NIPS*, pages 2226–2234, 2010. 25

Deqing Sun, Erik B. Sudderth, and Michael J. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, pages 1768–1775, 2012. 25

Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*. 2010. 22, 28, 98, 126

P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011. 21, 27, 38, 45

William B. Thompson. Exploiting discontinuities in optical flow. *IJCV*, 30, 1998. 23, 35

David A. Tolliver. Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In *In CVPR*, pages 1053–1060, 2006. 33

Carlo Tomasi and Takeo Kanade. shape and motion from image streams: a factorization method. Technical report, IJCV, 1991. 22, 26

Roberto Tron and Ren Vidal. A benchmark for the comparison of 3d motion segmentation algorithms. In *In CVPR*, 2007. 38

Amelio Vazquez-Reina, Shai Avidan, Hanspeter Pfister, and Eric Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*. 2010. 25

Carl Vondrick, Deva Ramanan, and Donald Patterson. Efficiently scaling up video annotation with crowdsourced marketplaces. In *ECCV*, 2010. 50

Huayan Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 2433–2440, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-0394-2. doi: 10.1109/CVPR.2011.5995722. URL http://dx.doi.org/10.1109/CVPR.2011.5995722. 95

Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *ICCV*, 2011. 25

Y. A. Wang and H. Adelson. Representing moving images with layers. *TIP*, 1994. 25

Yang Wang and Greg Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *ECCV (3)*, pages 710–724, 2008. 95

Yair Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. *CVPR*, 1997. 25

M. Wertheimer. Laws of organization in perceptual forms. *A Sourcebook of Gestalt Psycycholgy (Partial translation)*, 1938. 19

Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *IJCV*, 2007. 58

Jiangjian Xiao and Mubarak Shah. Motion layer extraction in the presence of occlusion using graph cuts. *TPAMI*, 2005. 18, 25

Jiangjian Xiao, Hui Cheng, Harpreet Sawhney, Cen Rao, Michael Isnardi, and Sarnoff Corporation. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, 2006. 25

C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012a. 24, 25, 26, 54

Li Xu, Jiaya Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1744–1757, 2012b. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.236. 21

Li Xu, Jiaya Jia, and Yasuyuki Matsushita. Motion detail preserving optical flow estimation. *TPAMI*, 34, 2012c. 123

J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006. 22, 26

Bo Yang and R. Nevatia. An online learned crf model for multi-target tracking. *CVPR*, 2012. 4, 79, 81

Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 3, 91, 94, 105, 108, 114, 128

Omar Javed Yaser Sheikh and Takeo Kanade. Background subtraction for freely moving cameras. In *ICCV*, 2009. 26

Stella Yu and Jianbo Shi. Multiclass spectral clustering. In *ICCV*, 2003. 13, 15, 33

Stella X. Yu and Jianbo Shi. Understanding popout through repulsion. In *CVPR*, 2001. 16, 48

Stella X. Yu, Ralph Gross, and Jianbo Shi. Concurrent object recognition and segmentation by graph partitioning. In *NIPS*, 2002. 16, 71