HUMAN TRACKING AND POSE ESTIMATION IN OPEN SPACES

THIS IS A TEMPORARY TITLE PAGE
It will be replaced for the final print by a version
provided by the service academique.

Thèse n. 6276
présentée le 20 Juin 2014
à la Faculté des Sciences et Techniques de l'Ingénieur
laboratoire Idiap Research Institute
programme doctoral en Génie Électrique
École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences
par

Alexandre Heili

acceptée sur proposition du jury:

Prof. Colin Jones, président du jury
Dr. Jean-Marc Odobez, directeur de thèse
Dr. François Fleuret, rapporteur
Dr. Patrick Pérez, rapporteur
Dr. Tao Xiang, rapporteur

Lausanne, EPFL, 2014

"Thoughts meander like a
Restless wind inside a letter box
They tumble blindly as they make their way
Across the universe."
– Lennon-McCartney –

To my parents.

Acknowledgements

During the last four and a half years, I have somehow felt like a traveler on the open road. Along the way, I have met many people who have made my journey worthwhile and have helped me progress, each in their own way. I would like to express my gratitude to all of them, as well as to some others who have stuck around for a little longer.

First of all, I would like to thank Jean-Marc for guiding me throughout my PhD. His unfailing support, insightful comments and care of details have helped me to methodically overcome the many scientific and technical hurdles along the way. I am also honored to have had Colin Jones, François Fleuret, Patrick Pérez and Tao Xiang as members of my thesis jury.

I thankfully acknowledge the European project VANAHEIM for supporting my research. In this context, I have had the privilege to collaborate with amazing postdocs, both on the human and scientific side. Thanks CC and Adolfo, it has been a pleasure working with you! Thanks as well Rémi for your tips and tricks about computers and life in general.

My deepest appreciation goes to Jagan and everyone at ADSC for hosting me as an intern for three months. My stay in Singapore has proved to be a great cultural and scientific experience.

Kenneth and Samira, you are great friends, and I have enjoyed having you around. The same goes for Kate, Gokul and Majid. I will always remember the good times (and bad jokes) we shared. Life at Idiap would not have been the same without the daily encounters and frequent baby-foot games that brightened up my days. Thanks to Venkatesh, Valérie, Billy, Charles, Nicolae, Leo, Marco, Ivana, Laurent, Elie, Minh-Tri, Ramya, Nesli, Thomas, Harsha and many others. I have also learned a lot from interacting with people from the perception and activity understanding group. Thanks to CC, Jagan, Stefan, Rémi, Kenneth, Adolfo, Samira, Paul, Romain, Elie, Vasil, Dinesh, Rui and Gulcan. Office 308 has a history of hosting awesome people who deserve to be mentioned, including Kate, Dinesh, Stefan, Radu, Chris, Majid, Rémi, CC, Samira, Kenneth, Adolfo, Maryam, Matthieu, and more recently Rui, Cijo, Pranay and James. Thanks as well to Nadine and Sylvie, and everyone from the administrative and system staff at Idiap who are constantly making sure everything is running smoothly. Thanks to Corinne, Chantal, Claude and everyone at EPFL who was involved in the organization of my defense.

Acknowledgements

I am deeply grateful to my friends for their constant support. Francis, you are an inspiration for me. I have always admired your strength and tenacity. Bénédicte and Nicolas, thanks for the good times we spent together in Strasbourg and everywhere else. Thanks to Joe and Moli, my soulmates from classes prépas. Thanks to Agnieszka, Kenneth, Ricardo and all my friends in Lausanne. Thanks Anne for keeping in touch with me after all these years. Thanks Triphon for always channeling good vibes. I would also like to acknowledge the countless music bands who have provided the soundtrack to my thesis, and more generally to my life.

Last but not least, I would like to thank my whole family. Thanks to my parents, Doris and Jean-Jacques, for sticking with me through thick and thin. This thesis is dedicated to you. I would like to thank you, Amiel, for always being there for me. You are the force that drives me forwards. *"I think you're the same as me, we see things they'll never see..."*

Martigny, May 26, 2014 Alexandre

Abstract

In modern civilization, the ever increasing threats of terrorism, theft, vandalism and accidents permanently put persons and infrastructures at risk. Organizations worldwide seek to forestall these potential threats by resorting to smart surveillance systems in order to safeguard their customers, personnel and installations. Surveillance networks can monitor indoor environments like offices occupied by few people and displaying relatively constrained scenarios, or on the other hand deal with much larger places like metro stations, airports or parking lots, that exhibit as well a possibly wider range of behaviors. As sensors are usually rather inexpensive, surveillance systems can benefit from a plethora of such devices, e.g. multiple cameras covering different areas of a transportation network like escalators, corridors, or platforms. However, dealing with sensor data at a large scale quickly becomes overwhelming for the supervising manpower. As a response, smart automated surveillance systems are being designed with the goal of sensing what is happening and taking appropriate action in real time, e.g. by raising an alarm in case of abnormal or antisocial behavior.

In this context, the automatic understanding of human activity is of paramount importance, as it can guide a system to decide on the potential threat or interestingness of a certain event. Such a high-level interpretation typically relies on image data processing and artificial intelligence techniques that should be designed in accordance with the scalability and real-time requirements of a large-scale system. Image processing steps usually consist of motion segmentation and/or object detection in the first place. Subsequent stages (e.g. tracking, pose estimation) exploit the low-level information to get mid-level representations (e.g. trajectories, orientations). These mid-level features can then be fed into high-level artificial intelligence modules that provide expert decisions about ongoing human behaviors in a scene. Specific challenges like low resolution or important depth effects, along with occlusions and poor observation viewpoints typically affect the different algorithmic steps involved in the extraction of the low and middle level representations. Artificial intelligence techniques, on the other hand, strongly depend on the quality of these features.

In this thesis, we focus on designing robust mid-level algorithms, with the motivation that improving their output can in turn benefit behavior analysis at the higher level. First, we design a novel, efficient framework for multi-person tracking, enabling us to consistently and simultaneously follow multiple people over time. Our tracker leverages on the output of a human detector and relies on a Conditional Random Field framework, formulated in terms

of similarity/dissimilarity pairwise factors between detections and additional higher-order potentials defined in terms of label costs. An efficient unsupervised parameter adaptation framework, as well as dedicated optimization algorithms that allow to obtain robust tracking results are presented. Second, we design perceptual algorithms to extract body and head pose cues, and formulate a principled way to jointly estimate them in a temporal filtering framework, leveraging as well on tracking information. We also propose to exploit the structure of the pose features manifold and to handle variations between training and test data through manifold biasing and alignment. Our different contributions are evaluated against several state-of-the-art benchmarks in order to prove their validity and efficiency.

Keywords: multi-person tracking, human pose estimation, surveillance, conditional random field, adaptation, optimization, temporal filtering, manifold alignment.

Résumé

À l'ère moderne, les menaces de terrorisme, de vol, de vandalisme et d'accidents mettent les personnes et les infrastructures constamment en péril. À l'échelle mondiale, différentes organisations cherchent à prévenir ces menaces potentielles en ayant recours à la surveillance intelligente pour protéger leurs clients, leurs employés et leurs installations. Les réseaux de surveillance peuvent concerner des environnements clos comme des bureaux occupés par un faible nombre d'individus, restreignant de fait la variété des scénarios possibles, ou au contraire couvrir des espaces beaucoup plus étendus comme des stations de métro, des aéroports ou des aires de stationnement, dans lesquels une gamme plus variée de comportements est attendue. Les systèmes de surveillance modernes peuvent bénéficier d'une multitude de capteurs peu coûteux comme des caméras vidéo déployées par exemple pour superviser les escaliers roulants, les couloirs et les quais d'un réseau de transport. Cependant, le traitement des données à grande échelle est une tâche difficile étant donné le nombre restreint d'opérateurs analysant ces données. Par conséquent, le développement de systèmes de surveillance dits intelligents s'avère essentiel pour suppléer la main-d'œuvre. Idéalement, de tels systèmes automatisés devraient être capables de comprendre des événements et de réagir en temps réel, par exemple en donnant l'alerte en cas de détection de comportement anormal ou antisocial.

Dans ce contexte, la compréhension automatique des activités humaines est primordiale car elle peut permettre à un système de décider de la dangerosité ou de l'intérêt d'un événement. Une telle interprétation de haut niveau dépend typiquement de techniques de traitement d'image et d'intelligence artificielle qui doivent être conçues en tenant compte de contraintes d'extensibilité et de temps réel. Les étapes de traitement d'image consistent généralement à segmenter le mouvement et/ou à détecter les objets. Les tâches suivantes, comme le suivi de personnes ou l'estimation de leur posture exploitent les informations de bas niveau pour produire des représentations intermédiaires comme des trajectoires ou des orientations. Ces caractéristiques intermédiaires peuvent ensuite êtres fournies en entrée à des modules d'intelligence artificielle qui interprètent à un haut niveau les comportements humains en cours dans une scène. Des défis particuliers comme la basse résolution des personnes dans les images, les effets de profondeur importants, ainsi que les occultations et les mauvais points d'observation affectent les différents algorithmes de la chaîne de traitement ayant pour but d'extraire les représentations de bas niveau et intermédiaires. D'autre part, les techniques d'intelligence artificielle au haut niveau dépendent fortement de la qualité de ces représentations.

Abstract

Dans cette thèse, nous concentrons nos efforts sur la conception d'algorithmes robustes d'extraction de primitives de niveau intermédiaire, avec la motivation que leur amélioration sera également bénéfique à l'analyse de comportements au niveau supérieur. Dans un premier temps, nous mettons en place un système original et efficace de suivi de plusieurs personnes. Notre algorithme de suivi exploite la sortie d'un détecteur et est basé sur un CRF modélisant les similarités/dissemblances entre paires de détections ainsi que des termes d'ordre supérieur définis comme coûts sur l'étiquetage global des détections. Nous présentons également des méthodes non supervisées d'apprentissage de paramètres et des techniques d'optimisation dédiées permettant d'obtenir des résultats de suivi robustes. Dans un second temps, nous nous intéressons à l'estimation de l'orientation de la tête et du corps, et nous présentons une méthode pour les estimer de façon conjointe dans un processus de filtrage temporel, en exploitant également les informations de trajectoires. En outre, nous proposons de tirer profit de la structure de l'espace des caractéristiques de pose et de tenir compte des variations entre données d'apprentissage et de test en biaisant et en alignant les variétés mathématiques correspondantes. Nous démontrons la validité et l'efficacité de nos différentes contributions en les évaluant et en les comparant à des approches de référence.

Mots-clés : suivi de personnes, estimation de la pose, surveillance, CRF, adaptation, optimisation, filtrage temporel, alignement de variétés.

Contents

List of Figures

List of Tables

Glossary

CCTV	Closed-Circuit Television
CHMM	Coupled Hidden Markov Models
CRF	Conditional Random Field
DPM	Deformable Part Model
EM	Expectation Maximization
GMCP	Generalized Minimum Clique Graphs
GMM	Gaussian Mixture Model
HD	High-Definition
HMM	Hidden Markov Model
HOF	Histograms of Flows
HOG	Histogram of Oriented Gradients
ICM	Iterated Conditional Modes
IDS	Identity Switch
JPDAF	Joint Probabilistic Data Association Filter
KLT	Kanade–Lucas–Tomasi
KNN	K Nearest Neighbors
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
MHT	Multi-Hypothesis Tracking
ML	Mostly Lost
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MRF	Markov Random Field
MT	Mostly Tracked
PCA	Principal Component Analysis
PF	Particle Filter
POM	Probabilistic Occupancy Map
PT	Partially Tracked
RJ	Reversible Jump
SO	Standard Output
SVM	Support Vector Machine
SW	Sliding Window

1 Introduction

1.1 Motivation

1.1.1 Surveillance Systems

With the increasing demand for security in the civilian realm, CCTV networks are growing worldwide, with the main purpose of ensuring safety, detecting crime or monitoring flows of crowds [Cristani et al., 2010]. Modern surveillance sensors like cameras and microphones are cheap and can be deployed at large scales, e.g. to monitor a shopping mall, a building, or a transportation network. Their proliferation results in an increasing amount of recorded data that has to be processed. However, the manpower required to monitor and analyze the acquired data is expensive. Often, the recorded streams are merely used as archive for forensics to refer back to an incident which is known to have happened [Ko, 2008].

In addition to passive surveillance, detecting events of interest as they happen brings other valuable information that enables to take appropriate measures in real time. For instance, if detecting that a person is in need of urgent medical care, e.g. after a fall or faintness, a rescue team can be promptly sent on site. As another example, if an abandoned luggage is spotted in a public place, a mine-clearing team of experts has to be deployed[1] to clear any risk that would ensue from a potential parcel bomb exploding. In practice, however, only a few operators are exposed to a large number of data streams, and cannot manage simultaneous monitoring of all the signals they receive. Figure 1.1 shows a typical control room where sensory information from different cameras is displayed on individual screens. In the end, operators can only supervise sparingly and they might miss crucial information, as the probability to watch the right stream at the right time is very limited. In the subway network of Turin[2], for instance, 28 screens are used to monitor the output of 800 cameras. As a consequence, some video streams are never being watched.

[1] This deployment procedure follows a well established protocol in many transport infrastructures and may lead to traffic interruption if the suspicious object is close to a train or a metro line.

[2] GTT - GRUPPO TORINESE TRASPORTI was one of the two technological/scientific assessment infrastructures of the European project VANAHEIM (www.vanaheim-project.eu/)

Figure 1.1: The control room: human operator monitoring a video wall.

Putting surveillance in its current context shows that large-scale sensor data management quickly becomes overwhelming for the supervising manpower. Thus, automatic and intelligent tools that can deal with abundant data and manage large infrastructures prove to be essential. Recognizing interesting events, for example vandalism, people loitering or aggression is easy for human operators, who base their judgment on their social experience in which they have learned to classify behavioral patterns of individuals and groups of people. Similarly, automated surveillance systems can be trained to take decisions based on sensory inputs. For instance, a system could be trained to detect abnormal or antisocial behavior and could raise an alarm accordingly. However, the knowledge of an automatic system is generally very limited, which makes inference difficult. The system might produce many false positives, which can be undesirable, especially if they trigger inopportune alarms. One may want to reduce the volume of false alarms, while keeping a good rate of correct detections. In this context, algorithms can still benefit from human expertise, for instance within a feedback loop where operators provide validation so that the system can continuously adapt and gain robustness.

To address the false alarm issues, an alternative is to use soft alarms, for instance to raise operator awareness of possible incidents. In such stream selection or video-wall management applications, algorithms select relevant audio or video streams to display in the control rooms, but the final judgment is left to a human expert.

Apart from security, public spaces are also subject to capacity issues. Organizations often express the need for the analysis of people dynamics. For instance, it can be important to know their locations, routes, spatio-temporal activities like walking or waiting, as well as their interactions, in order to know for example which are the highly frequented aisles or the common loitering areas of the monitored space. Smart surveillance systems should therefore be able to extract trends of human behavior at an infrastructure level.

Most of the scenarios mentioned previously, whether in event detection applications for safety and security, or environmental reporting for situational awareness require the understanding of human behavior, whether at the individual, group or crowd level. In the end, a complete design of surveillance systems has to be thought about in order to help operators in their daily

tasks and the conception of behavior cue extraction algorithms in open spaces, as investigated in this thesis, lies at the core of this possible evolution.

1.1.2 Behavior Analysis

In the context of smart surveillance, the automatic understanding of human activity is of paramount importance, as it can guide a system to decide on the potential threat or interestingness of a certain event. Gaining such a high-level knowledge typically requires a hierarchical bottom-up process, in which image-level information is first extracted, then used to produce mid-level representations that are finally treated as input by artifical intelligence modules. The typical pipeline of an automated surveillance system is shown in Figure 1.2. Below, we comment on the different bricks of this process and in particular on the potential applications of behavioral cue extraction.

Trajectory Analysis. At the low-level, human detection can be conducted using pixel information. The task of tracking then consists in successfully following each person's location over time, or in other words, to assign consistent identities to the subjects present in the scene. Entries and exits of people also have to be handled. When applying this process to more than one camera view or scene, the problem becomes one of re-identification across disjoint camera views. Human tracking yields trajectories that can be used to understand behaviors. In fact, trajectories provide enough information to know people's locations and routes, as well as some of their simple spatio-temporal activities like walking or waiting. The knowledge given by trajectories can therefore be useful to understand infrastructure usage.

Pose cues also play an important role in the understanding of human behaviors, as they can characterize people's activities and interactions more precisely. Therefore, they can be used in addition to trajectory information as complementary mid-level cues, as shown in the middle brick of the pipeline in Figure 1.2 to help behavior understanding in various applications which we summarize below.

Luggage Attendance Monitoring. Combined with a left luggage detection module, location and pose cues could be used to monitor luggage attendance. Indeed, when observing a single person, his body and head pose indicate which part of the physical space he is facing to, and can be useful to determine his focus of attention. In a public space, if a bag has been dropped for quite some time, it is important for security reasons to know whether it is still attended by its owner or whether it has just been abandoned, as it could conceal a bomb. Another application could be the trigger of a vocal message to warn people about potential theft, if they are standing far from their bags and are not looking at them, as illustrated in Figure 1.3.

Group Detection. Detecting interacting groups has important applications, as it can help for instance to determine the areas where people meet and socialize. When looking at Figure 1.4, we identify the two persons in the front to be part of a group having a social interaction in the

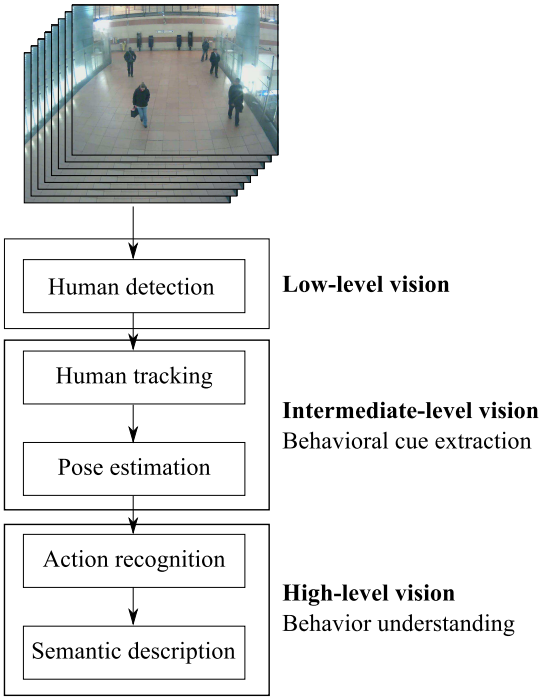


Figure 1.2: Typical hierarchical pipeline of a surveillance system. Performing high-level understanding necessarily involves perceptual algorithms at a lower level, dealing with the elementary bricks of person detection and behavioral cue extraction, e.g. through tracking and pose estimation. (Adapted from [Vishwakarma and Agrawal, 2013])

Figure 1.3: Luggage attendance: in the left image, the person is monitoring his luggage, as he is standing close by and looking towards it. On the other hand, the person in the right image is turning his back to his bag, thus leaving it unattended.

form of a conversation. On the other hand, the person in the background on the left obviously does not belong to their group, as he is standing relatively far and not paying attention to them. As humans, we can identify such high-level information, even from a still image. The question is how to transfer this knowledge to an automated system. It appears that the persons' respective locations and poses give a strong prior on their interactions. For example, in the case of a conversation, we expect spatial proximity as well as concomitant poses. For a walking group, we would expect proximity and poses facing the same moving direction.

Figure 1.4: Interacting group of two people in a metro hall in Turin.

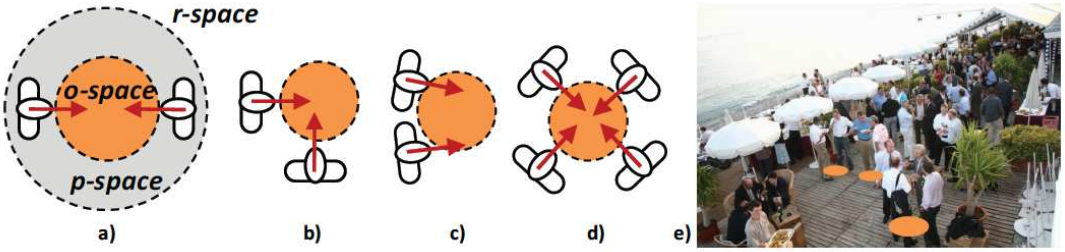


Figure 1.5: Characterizing groups by F-formations. a) Vis-a-vis. b) L shape. c) Side-by-side. d) Circular. e) Cocktail party with o-spaces superimposed. The o-space is a convex empty space surrounded by the people involved in a social interaction, where every participant looks inward into it. The p-space is a narrow stripe that surrounds the o-space, and that contains the bodies of the talking people. The r-space is the area beyond the p-space. (Image taken from [Cristani et al., 2011])

More generally, following the work of Kendon on F-formations [Kendon, 1977] that represent spatial patterns maintained during social interactions by two or more people, researchers have proposed computational models to identify groups. These F-formations rely on the definition of three social spaces: o-space, p-space and r-space, as illustrated in Figure 1.5. In [Cristani et al., 2011] for instance, the authors propose to discover social interactions by detecting o-spaces from people's positions and head pose estimates and a Hough voting procedure, whereas in [Hung and Kröse, 2011], the detection of F-formations is formulated as the task of finding dominant sets in a graph. The authors of [Setti et al., 2013] compare these strategies [Cristani et al., 2011] [Hung and Kröse, 2011] and come up with the interesting conclusion that position information is usually sufficient for defining groups, but that head or body orientation enable improvements in group detection performance, especially in very crowded scenes.

Attention Modeling. In a transportation network, behavioral cues can indicate whether people are interacting with equipments like vending machines or maps. Similarly for shop

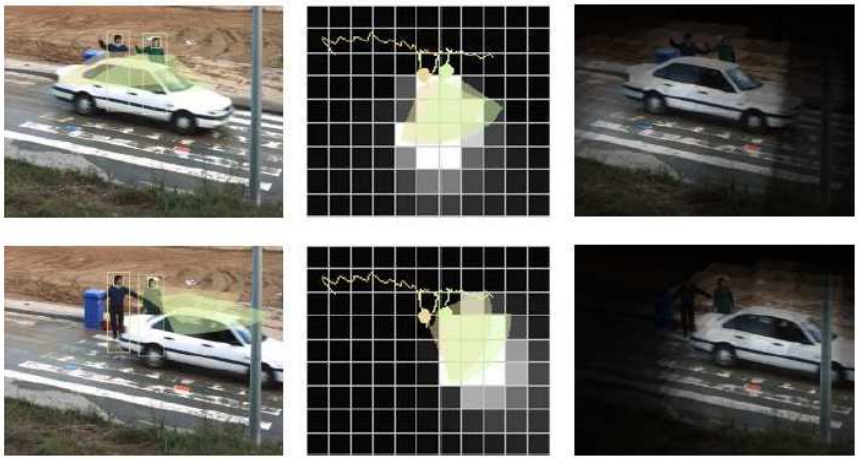


Figure 1.6: Guiding visual surveillance by human attention. The left column shows the original video frames with superimposed gaze directions, the middle column shows the corresponding attention maps (each cell represents a square meter of the ground) and the third column shows the video frame modulated with the projected attention map. (Image taken from [B. Benfold and I. D. Reid, 2009])

customers, their attention towards particular ad placements or products is important and can help for instance to optimize a shop layout. Attention maps can be learned for those purposes. This was proposed in [B. Benfold and I. D. Reid, 2009]. In this approach, pedestrians are tracked and their head poses are exploited to retrieve the parts of the scene that attract most of their attention. Such attention maps can also be used to highlight transient areas of interest, as illustrated in Figure 1.6 and could therefore help to direct the attention of an automated surveillance system (e.g. by guiding a dynamic camera), or to point out interesting areas to a human operator.

1.2 Objectives and Contributions

1.2.1 Goals and Challenges

The different scenarios presented above illustrate how simple behavioral cues like location and pose can help inferring behaviors. Therefore, in this thesis, we propose to answer the two following questions:

- For a given group of persons with different identities at time t, where are each of these individuals at time $t+1$? At time $t+n$?
- What are the body/head pose angles of any given tracked person at a specific instant ?

Our motivation is that robust behavioral cue estimation can in turn benefit behavior analysis at the higher level.

However, estimating these cues is not a trivial task. Typically, computer vision tasks face scene-related problems coming from illumination, difficult viewpoint, or depth effects, as well as sensor-related variabilities, due for instance to intrinsic modalities like sampling frequency, noise, dimensionality or compression techniques. Detectors, even though more and more reliable, still suffer from unreliability and sparsity, meaning they produce false positives and false negatives. Tracking techniques are sensitive to occlusions which typically increase in crowded scenarios. Resorting to multiple cameras with overlapping fields of view can help occlusion reasoning, but is not always an option in surveillance, as it poses the problem of data synchronization, precise joint calibration, and cost. Other issues like unpredictable motion or appearance changes make the tracking task difficult.

Pose estimation is an even more challenging task, due notably to the low resolution of most surveillance data, as well as the large variabilities in face and clothing appearance between different people. Besides, the employed pose classifiers are often trained on a different type of data, making their generalization to the target sequences problematic, and they usually produce noisy observations on body and head pose.

Due to these issues, tracking and pose estimation are still active research topics in the fields of computer vision and pattern recognition. Furthermore, in order to cope with the requirements of modern, large-scale surveillance systems, the developed algorithms should be designed to run in reasonable time, with limited complexity.

1.2.2 Contributions

As motivated in the previous sections, this thesis aims at designing and benchmarking perceptual algorithms for the estimation of human behavioral cues from videos. In the pipeline of Figure 1.2, our algorithms focus on the middle level and concentrate on the task of tracking people and estimating their body and head pose.

In order to improve the state-of-the-art, a central aim of our methods has been the design of algorithms that can leverage on the available test data to obtain better component estimates (models, parameters) and adapt to the scene. For instance, as detailed below, our tracking algorithm naturally adapts to new scenes without additional effort thanks to an unsupervised learning scheme of scene-specific model parameters. Similarly, our pose estimation algorithms can adapt to any scene (or scene part) by using coupling information, or by applying domain adaptation techniques to transfer external knowledge to the target data.

Following this main idea along with other modeling choices, we made the following contributions. On the tracking part:

- **Detection-level CRF model with long-term connectivity.** In [Heili et al., 2011], we formulated our multi-person tracking algorithm as the statistical labelling of the detections produced by a human detector. To this end, we relied on a Conditional Random Field (CRF) approach that models relationships between detections, rather than tracklets as

most CRF systems do. Besides, to derive association measures, the model exploits both similarity and dissimilarity hypotheses for each pair of detections. By contrasting the two hypotheses for each detection pair, the model is more robust to assess the appropriateness of a given association. Our approach also benefits from important temporal context by connecting detection pairs not only between adjacent frames, but between frames within a long time interval, so as to better cope with detector inherent flaws like missed detections and false alarms, and provide larger context for labeling. Within this framework, the association costs are also adapted to the time gaps that separate detection pairs.

- **Image-based dynamics at the detection level, contextual weighting and label costs.** In [Heili et al., 2014a], we proposed to capture the dynamics of targets by exploiting visual motion at the detection level, in contrast to approaches relying on tracklet estimates based on hypothetical associations. Considering spatio-temporal reasoning such as occlusions between detections, we also introduced confidence scores to model the reliability of the features and to efficiently fuse the different observation types. Finally, we defined additional higher-order potentials in the model in terms of label costs penalizing long tracks if those start or end far from a set of pre-defined boundaries.
- **Unsupervised adaptation to a scene.** In [Heili et al., 2011], we proposed to use detection pairs to learn scene-specific and time-sensitive model parameters without supervision, which promoted the generalization capability of our tracker. In [Heili and Odobez, 2013], we improved the framework by adopting an incremental learning process, starting from raw detections, then using intermediate tracking results.

On the pose estimation part:

- **Joint body and head pose estimation in a temporal filtering framework.** We proposed likelihood models for the features extracted from body [Chen et al., 2011a] and head [Chen et al., 2011b] regions under each class representing a discretized range for the pose angle. Using the defined likelihood models, for each frame, noisy body and head pose cues can be estimated. To refine the behavioral cue estimates, we proposed a particle filtering framework to exploit the intra-cue temporal smoothness. To improve the filtering framework, we also introduced soft coupling constraints between cues. First, we proposed in [Chen et al., 2011a] to use velocity information inferred from trajectories as a prior for body pose. To avoid problems when people are static or have only slow movement, we conditioned the coupling on the speed. Second, we enforced a soft coupling between body and head pose in [Chen et al., 2011b], which stems from anatomical constraints.
- **Bridging the gap between training and test data by manifold biasing and alignment.** Until recently, one important limitation of existing pose estimation methods was the use of pre-trained classifiers not adapted to the test data, in spite of obvious variabilities in appearance, viewpoints and illumination. To address this issue, we took inspiration from [Chen and Odobez, 2012], in which classifier adaptation is performed by leveraging

on scene data, coupling between classifiers and pose feature manifold structure. For the latter part, however, rather than relying only on classifier smoothness (similar features get similar labels), we proposed in [Heili et al., 2014b] to weight the feature distance between pairs of samples by a function of their pose angles difference, using either ground truth pose labels or weak labels derived from the velocity (when reliable). In this way, samples are tightly clustered in the feature space as well as the pose angle space. Then, we handled variations between training and test data through semi- or weakly-supervised manifold alignment, based on sparse correspondences.

For both tasks, we conducted experiments on standard public datasets to demonstrate the benefits of our modeling contributions.

The thesis was conducted within the scope of the collaborative European project VANAHEIM, which emphasizes on studying and integrating innovative audio/video analysis tools in a CCTV surveillance platform. A lightweight version of our multi-person tracking algorithm has been integrated within the project demonstrator.

1.3 Thesis Organization

The organization of the manuscript is graphically summarized in Figure 1.7. Below, we briefly explain the content of each Chapter.

Chapter 2. In this literature review, we present and discuss methods dealing with the various blocks making up a surveillance system pipeline, i.e. detection, tracking, pose estimation and behavior analysis. We then motivate our contributions with respect to the state of the art.

Chapter 3. This chapter introduces one of the main parts of the thesis, which is our CRF framework for multi-person tracking, along with its different modeling components. Our model includes similarity/dissimilarity pairwise factors between detections and additional higher-order potentials defined in terms of label costs. Along with position and color features, we propose to use image-based dynamics at the detection level to assess the labeling of detections. Finally, we present a contextual weighting scheme to account for the reliability of the different features.

Chapter 4. In this chapter, we explain how model parameters can be learned in an unsupervised and incremental fashion by gathering observation statistics first from raw detections, then from intermediate tracking results. We also present two algorithms to perform optimization on the graph resulting from our statistical labeling formulation, given the learned models.

Chapter 5. In this chapter, we conduct exhaustive experiments on standard, state-of-the-art datasets to show and discuss the benefits and generalization capability of our detection-based

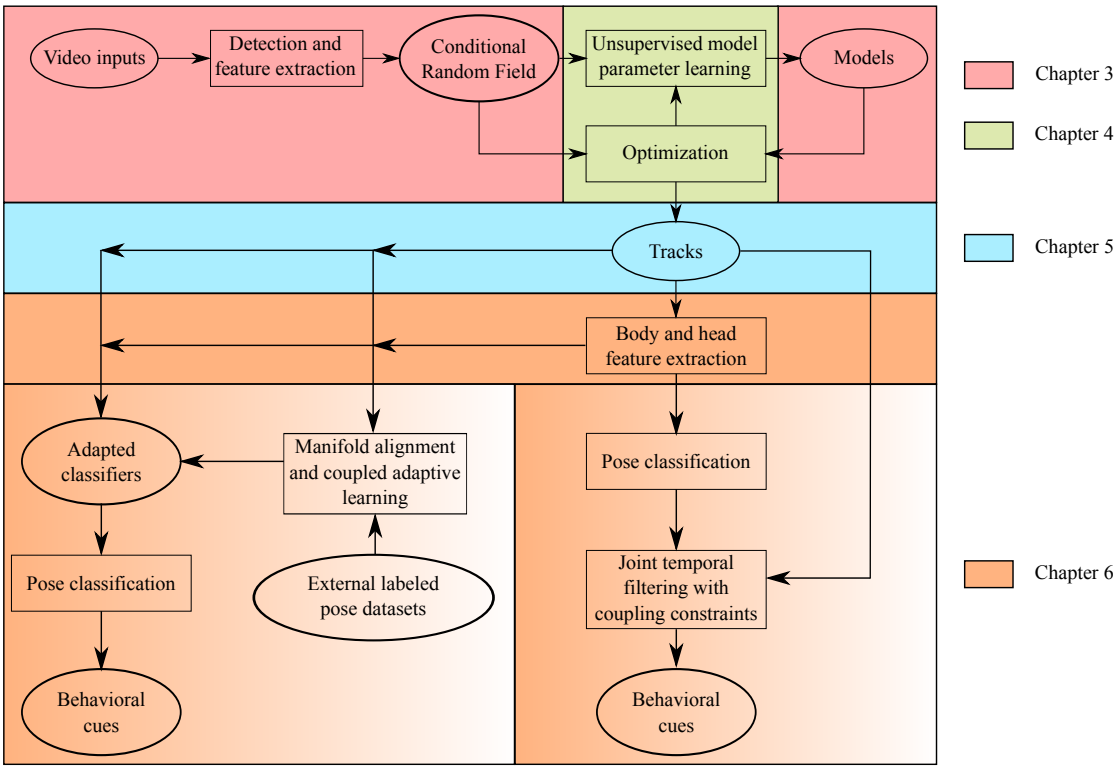


Figure 1.7: Flowchart of the thesis contributions and organization: Chapter 3 introduces our CRF model. Chapter 4 shows how to learn model parameters from data in an unsupervised way (using raw detections or tracks resulting from a first optimization round) and how to optimize the graph resulting from the statistical labeling formulation. Chapter 5 gives extensive results and discussions about the multi-person tracking results. Chapter 6 presents our approaches for the joint estimation of behavioral cues, either in a temporal filtering framework (right); or by addressing domain adaptation through manifold alignment (left).

multi-person tracking approach.

Chapter 6. This chapter presents our research dealing with the extraction of body and head pose information. We first present our temporal filtering approach with soft coupling constraints to obtain smooth pose estimates. We then present our approach to address domain adaptation for better generalization of the classifiers, leveraging on manifold alignment and coupled adaptive learning.

Chapter 7. We summarize the achievements of the thesis, discuss the shortcomings of the current algorithms and propose some directions for future research on the addressed topics.

2 Related Work

2.1 Introduction

In this chapter, we present a brief overview of methods involved in the different components of a surveillance system, and tackling the associated challenges. As the scientific literature on these topics is abundant, we restrict our analysis to works relevant to the objectives of the thesis and motivate our choices with regard to the current state of the art in the domain. More precisely, these are:

- **Tracking:** in Section 2.2, we first present common object representations and detection techniques. We then present and discuss several classes of object tracking methodologies.
- **Pose Estimation:** several state-of-the-art human pose estimation algorithms are exposed in Section 2.3, along with their advantages and limitations.
- **Behavior Analysis:** finally, in Section 2.4, we briefly discuss how mid-level cues can be used to analyze high-level behaviors.

To conclude, Section 2.5 will summarize the different challenges of these topics and put our approach and contributions in their context.

2.2 Tracking

Multi-human tracking, even though it has been explored by the research community for a long time, remains a challenging task. This is especially true in single camera tracking situations, where several challenges notably arise from low image quality, sensor noise, dimension loss due to projection of 3D objects in image planes, clutter, unpredictable motions, appearance changes of people and importantly, occlusions. Multi-camera approaches enable to gain

depth information, to increase the field of view and to facilitate occlusion reasoning, but even in this case, tracking remains difficult when the overlap is small or in high crowding.

Before reaching the vision community, multi-object tracking was applied in pioneering works on radar signals [Blackman, 1986]. However, in this short review, we will focus on visual tracking methodologies. The approaches we will present are dealing with different types of objects, and are interested in following either single or multiple instances of those objects. Recent surveys on object tracking like [Yilmaz et al., 2006] [Jalal and Singh, 2012] [Chau et al., 2013] summarize a large number of methods, which are usually categorized as point tracking, template tracking or silhouette tracking methods, depending on the selected object representation. Therefore, we first present common object representations and features for visual tracking in Section 2.2.1.

Almost all tracking methods imply the use of object detectors, either in the first frame, at every frame or intermittently. In this perspective, a brief overview of the state of the art in object detection is given in Section 2.2.2, with a focus on human detection.

The traditional view of visual tracking, as seen for the last 30 years is based on filtering approaches. We call these *prediction and update* methods, as they jointly estimate object regions and correspondence by iteratively updating object locations and region information obtained from previous frames. These methods can rely on a detector but usually only for track initialization. In a second class of methods, which has recently gained popularity and that we call *tracking-by-detection*, a detector is applied to locate objects and data association is performed to establish correspondences between detections. Finally, we review a class of hybrid tracking methods, which we call *detection-guided prediction and update* methods, and in which the output of a detector is used to guide the state update mechanism of *prediction and update* techniques. These different tracking paradigms are described in Sections 2.2.3, 2.2.4 and 2.2.5, respectively.

2.2.1 Object Representation and Features

Illustrative Example. We start this section by presenting a concrete tracking example, shown in Figure 2.1 and in which the task is to track the face of a person over time. To that end, several points need to be addressed. First, we need to define a state x to represent what we want to infer about the object. In this case, the state is given by the 2D location in the image and the scale of a rectangle region $R(x)$ telling where the face is. Second, we need to define a measurement procedure to extract information about the object. Several features are usually used, and they include color, texture, shape or motion information. In our example, let us assume that we know the initial state x_0 of the object at time t_0, for example by manually picking region $R(x_0)$ or by automatically initializing it with a face detection algorithm. Then, whatever the representation and features, the main question of tracking is how to efficiently estimate hypothesized region(s) $R(x_t)$ at time t and how to tell if a picked region is a good

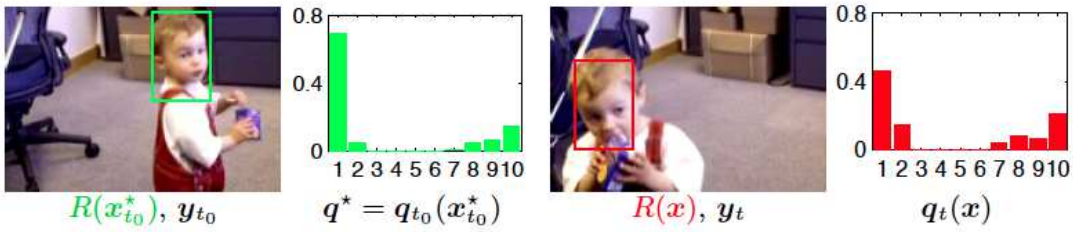


Figure 2.1: Color model for tracking: A reference color histogram $q^\star$ is gathered at time t_0 within a rectangular region R around location $x^\star_{t_0}$. At time t and for a hypothesized state x, the candidate color histogram $q_t(x_t)$ is gathered within the region $R(x_t)$. (Image taken from [Pérez et al., 2002].)

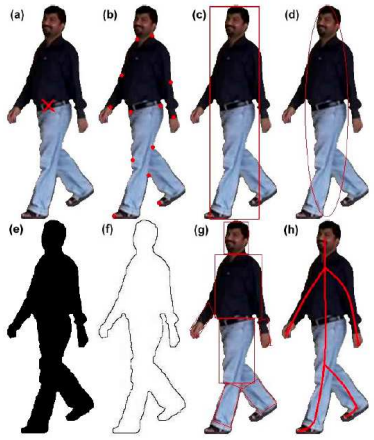


Figure 2.2: Object state representations: a) Single point; b) Multiple points; c) Primitive rectangular shape; d) Primitive ellipsoidal shape; e) Silhouette; f) Contour; g) Articulated shape; h) Skeletal model. (Image taken from [Jalal and Singh, 2012]).

candidate for matching. To pick a candidate region, either the information from previous frames or the output of a detector could be used. To measure the adequacy of matching, an evaluation procedure needs to be employed, for example based on a similarity measure or discriminative information about the object. In the example, hypothesized states are estimated using states from previous frames and a dynamical model. The quality of the matching is evaluated by comparing the color histogram $q_t(x_t)$ extracted from hypothesized region $R(x_t)$ at time t to the reference $q^\star$ gathered at time t_0 within region $R(x_0)$. In the following, we describe common state representations and features used for tracking. We also briefly discuss the design of feature kernels for affinity measures.

State Representations. The first task in tracking is to define an appropriate representation suited to the type of object that has to be tracked. Several representations have been used

in the literature. These include point(s), primitive geometric shapes, silhouette, contour, articulated shape models and skeletal models. Figure 2.2 illustrates common representations used in visual tracking. Depending on the representation, tracking can take a different form. In point tracking, the task is formulated as finding the correspondence between detected objects described by points in consecutive frames [Veenman et al., 2001] [Shafique and Shah, 2005] [Broida and Chellappa, 1986]. In template tracking, the task is to track an object represented by a primitive geometric shape by estimating its motion between consecutive frames, in terms of a parametric transformation that can involve translation, rotation and scaling [Birchfield, 1998] [Schweitzer et al., 2002] [Comaniciu et al., 2003] or by using an optical flow method extended on rectangular regions or patches [Shi and Tomasi, 1994]. Silhouette tracking, on the other hand, either aims at finding the object silhouette in the current frame by shape matching using an edge map [Huttenlocher et al., 1993], or by tracking contour evolution [Terzopoulos and Szeliski, 1993] [Isard and Blake, 1998].

Features. Once a representation has been chosen, features have to be selected to measure intrinsic properties of the object. One property of the employed visual features is their ability to distinguish the object, either from the background or from other objects. Visual features for tracking can be decomposed into shape (e.g. edges or control points), appearance (e.g. color or texture) and motion features (e.g. optical flow) [Yilmaz et al., 2006]. Several methods employ a combination of different types of features to better capture the characteristics of the objects to track [Perez et al., 2004] [Snidaro et al., 2008] [Zelniker et al., 2009] [Chau et al., 2011]. In the end, features are closely related to representations. For instance, contour-based representations usually imply the use of edge or optical flow features, whereas primitive geometric shapes usually imply the use of appearance features.

One can also learn from the re-identification domain to design efficient representations and features, especially for the task of tracking humans. In surveillance, re-identification can basically be seen as an extension of the tracking task to match instances of the same person registered in disjoint camera views, usually assuming that people are wearing the same clothes between sightings. In this context where traditional biometrics such as face, iris or gait recognition cannot be applied due to low resolution, researchers have tried to develop robust representations that capture human characteristics and that are as little as possible sensitive to variations in illumination, pose and viewpoint. The covariance descriptor [Tuzel et al., 2006] is a popular feature that encodes feature variances and correlations and that can absorb rotation and illumination changes, which is useful in re-identification [Bak et al., 2011]. Other approaches advocate the use of salient parts to describe subjects [Farenzena et al., 2010] [Zhao et al., 2013]. In [Farenzena et al., 2010], regions close to symmetry axes are considered to be more salient and features are thus weighted by the distance to these axes so that the effects of pose variations are minimized. In [Zhao et al., 2013], the authors propose an unsupervised way to extract distinctive features by resting upon the assumption that usually, salient parts can be seen from different viewpoints (e.g. colored shirt, bags, carried objects) and that their locations are somehow constrained to remain around a given position (e.g. a backpack should

appear at the same height within the human bounding box, regardless of the viewing angle).

Feature Kernels. An evaluation procedure is then required to assess the quality of any given candidate region for matching, by comparing its feature representation to that of a reference model of the object. In the example of Figure 2.1, the matching is evaluated based on the affinity between color histograms. Instead of defining feature kernels a priori, several multi-object tracking approaches learn appearance and/or motions models that are able to distinguish objects from one another by maximizing inter-class variations, while minimizing intra-class variations. For instance, in a tracking-by-detection context, Kuo et. al. propose to learn discriminative models online [Kuo et al., 2010]. They first collect positive and negative training sample pairs from targets online in a sliding window. They take advantage of several types of local features at multiple locations and scales (color histogram, covariance matrix, HOG) and use the AdaBoost algorithm to learn a strong model which determines the affinity score between two detections such that positive and negative pairs can be discriminatively separated. Yang et. al. further consider discriminative models to distinguish spatially close targets with similar appearances [Yang and Nevatia, 2012a]. Similarly in [Breitenstein et al., 2011], a negative set for learning is sampled from nearby targets so as to learn classifiers that best discriminate difficult pairs. The classifiers are continuously adapted, thus becoming more and more discriminative. In the domain of re-identification, feature kernels should be discriminative to distinguish each individual from the others, and reliable in finding the same person across views. Supervised methods aim at learning distance metrics that maximize the likelihood that a pair of true match has a smaller distance than that of a wrong match [Bak et al., 2012a] [Zheng et al., 2013].

2.2.2 Object Detection

Object detections are almost always required by tracking methods. In prediction and update tracking methods, they can be used for initialization or to guide state update (see Sections 2.2.3 and 2.2.5). In tracking-by-detection approaches, they can potentially be used at every frame (see Section 2.2.4). Zhang et. al. recently presented an exhaustive survey on object detection methodologies [Zhang et al., 2013]. However, we choose to focus here on pedestrian detection in a surveillance context, and like [Li et al., 2012], we categorize pedestrian detection algorithms into model-based and learning-based methods.

Model-based Detection. Model-based detection methods are generative. They consist in building a human model and then looking for the location in the image that produces the best match with the model. This class includes discrete shape models in the form of exemplars that can be used for edge matching [Gavrila, 2007] [Lin and Davis, 2010]. Other generative approaches use continuous, parametric shape models to represent every likely posture [Zhao and Nevatia, 2003] [Ge and Collins, 2009] [Enzweiler and Gavrila, 2008].

Learning-based Detection. Learning-based detection methods are mainly discriminative.

Figure 2.3: Examples of detector outputs showing (left) a missed detection and a false alarm; (right) detection accuracy issues, like legs cut or extended due to projected shadows on the floor. Detections were obtained with the code of [Yao and Odobez, 2008a][2].

Using a large number of training examples, these methods train classifiers to distinguish features coming from the pedestrian class from features belonging to the non-pedestrian class. Various features like HOG [Dalal and Triggs, 2005], integral channel features [Dollár et al., 2009], covariance features [Tuzel et al., 2008] [Yao and Odobez, 2008a] can be used used as input. In video applications, motion contains useful information that can be used for detection, for instance using optical flow information [Dalal et al., 2006], or background subtraction [Yao and Odobez, 2008a]. The most widely used classifiers for pedestrian detection are SVM [Dalal and Triggs, 2005] and Boosting [Tuzel et al., 2008] [Jones and Snow, 2008] [Yao and Odobez, 2008a] [Gualdi et al., 2010].

All detectors produce false negatives and false positives. Besides, imprecise detection localization and size can arise for instance due to the presence of projected shadows or partial occlusion. Some of these detector flaws are illustrated in Figure 2.3. One of the main causes for false negatives is occlusion. Part-based detectors have become popular as they allow to detect discriminatively trained parts and therefore are less sensitive to partial occlusion than holistic detection methods as well as to spatial variations of the discriminative local parts positions within the object bounding box. For instance, in the deformable part models (DPM) of [Felzenszwalb et al., 2010][1], each object is considered as a deformed version of a template and each part represents local visual properties, with spatial relationships.

Apart from the above, multi-camera methods are another way of handling occlusions [Fleuret et al., 2008] [Zeng and Ma, 2011]. Indeed, a person occluded in one camera view might be visible in one or more of the others. Figure 2.4 illustrates the multi-camera detection framework of [Fleuret et al., 2008]. This method reprojects templates from a discretized 3D ground plane and looks for the set of person locations whose reprojection of their 3D body binary templates best fits the computed foreground images in each view.

[1] Code available at www.cs.berkeley.edu/~rbg/latent/
[2] Code available at http://www.idiap.ch/~odobez/human-detection/index.html

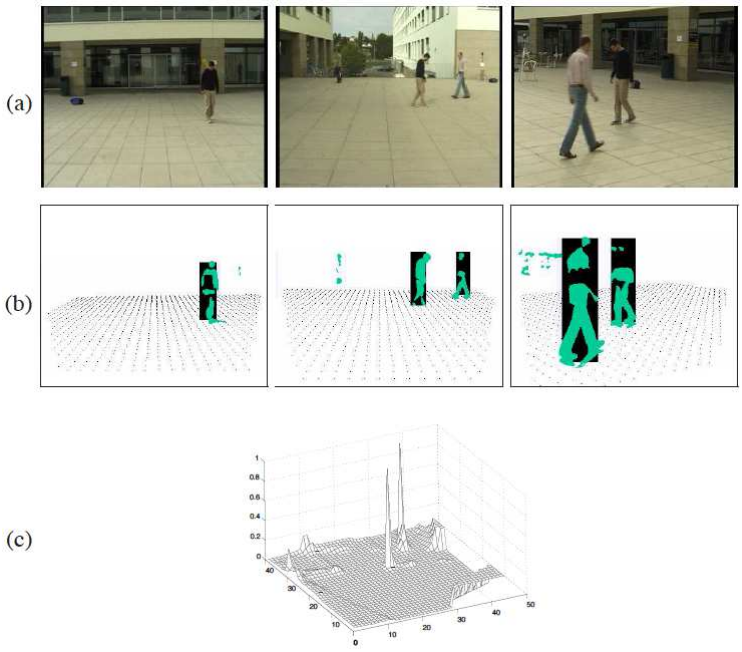


Figure 2.4: The POM detector. a) Original images from several views at a given instant. b) Foreground blobs obtained from background subtraction along with template reprojections (bounding boxes) that best fit the observations. c) Probabilistic occupancy map for each discrete location on the ground plane. (Images taken from [Fleuret et al., 2008].)

2.2.3 Prediction and Update Tracking

Going back to the example of Figure 2.1, one question was how to find the optimal estimate of state x_t at time t. In prediction and update tracking, the state x of the object is recursively inferred in each frame given its history and the current image observations. The *prediction* steps guides the tracking to look for potential matching regions according to some prior given for instance by a dynamical model. The *update* step then exploits observations to refine the predicted state. Below we present two classes of prediction and update tracking methods, namely the deterministic and the probabilistic ones.

Deterministic Methods. Several methods solve the tracking problem deterministically. Methods like template matching [Fieguth and Terzopoulos, 1997] [Birchfield, 1998] [Schweitzer et al., 2002] and mean-shift [Comaniciu et al., 2003] [Avidan, 2005] formulate the task as finding the new state of the object as the one with the highest likelihood in terms of some similarity measure with the object model. In both classes of approaches, a prior that the target is located at the vicinity of its previous location is usually used as a prediction to reduce the search space. The update step is done by selecting the best state by a brute force search in standard template matching, or by finding the mode of a smooth similarity map built in the neighborhood of the preceding location of the tracked object in mean-shift approaches. These

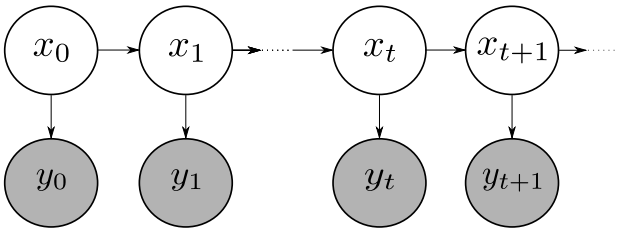


Figure 2.5: Graphical model for Bayesian tracking under a first-order Markov assumption. Shaded nodes are observations, white nodes are hidden states that we wish to infer.

deterministic approaches are sensitive to distraction as they might be driven towards object states with similar appearance than the target and result in tracking failures with little chance of recovery. Probabilistic tracking techniques, that are described next, are an alternative that allows to model uncertainty in the estimates.

Probabilistic Methods. Bayesian methods yield distributions instead of single-point estimates. These techniques are designed to handle uncertainties due to noise and ambiguities. If the hidden state and the data at time t are respectively denoted by x_t and y_t, probabilistic tracking methods typically adopt the following assumptions:

$$p(x_{t+1}|x_{0:t}) = p(x_{t+1}|x_t) \tag{2.1}$$

$$p(y_{t+1}|y_{0:t}, x_{0:t+1}) = p(y_{t+1}|x_{t+1}) \tag{2.2}$$

where $x_{0:t} = (x_0, ..., x_t)$ and similarly for $y_{0:t}$.

Equation 2.1 represents a first-order Markovian prior on the states, defined for example from physical principles like a constant speed model. The second assumption is a conditionally independent observation process, represented by Equation 2.2. The corresponding graphical model is illustrated in Figure 2.5. Under these assumptions, the filtering distribution $p(x_t|y_{0:t})$ follows the recursion given by the two steps in Equations 2.3 and 2.4. *Prediction* (Eq. 2.3) uses the dynamics and the filtering distribution already estimated at the previous time step t to derive the prior distribution of the current state $p(x_{t+1}|y_{0:t})$. The *update* or *correction* step (Eq. 2.4) uses the likelihood of the current observation $p(y_{t+1}|x_{t+1})$ to compute the posterior $p(x_{t+1}|y_{0:t+1})$.

$$p(x_{t+1}|y_{0:t}) = \int_{x_t} p(x_{t+1}|x_t)p(x_t|y_{0:t})\mathrm{d}x_t \tag{2.3}$$

$$p(x_{t+1}|y_{0:t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|y_{0:t}) \tag{2.4}$$

Under the hypotheses of linear operators and independent, normally distributed noises in the process and the measurement, the filtering distribution is analytically tractable, yielding the Kalman filter, which has been used in visual tracking methods [Broida and Chellappa,

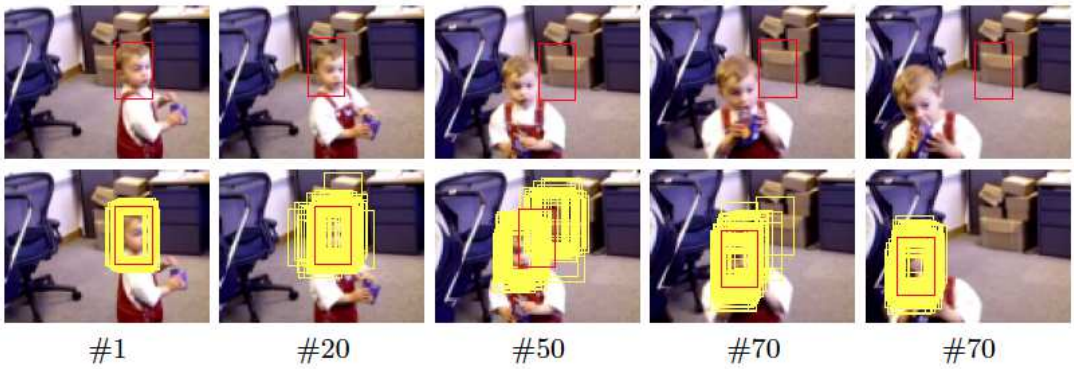


Figure 2.6: Color-based particle filter tracking under distraction. Top: deterministic color-based tracking can get distracted by local minima (e.g. the boxes in the background which might have similar color distributions as skin). Bottom: color-based PF can track momentarily multiple modes like in frame 50 and avoids distraction from the background. (Image taken from [Pérez et al., 2002].)

1986] [Remagnino et al., 1997] [Beymer and Konolige, 1999]. Monte-Carlo approximations like Particle Filters (PF) use a non-parametric representation of the posterior distribution through a set of weighted samples and can be applied without normality constraints. The samples a.k.a particles are moved by sampling from a proposal density function and reweighted so that they approximate the new filtering distribution well. For example, in the case of the bootstrap filter, the proposal density function is simply given by the dynamics $p(x_{t+1}|x_t)$ and the weight of each sample simply corresponds to the likelihood associated to its hypothetical state. PFs require a resampling step according to the weights to avoid degeneracy of the samples. PFs have been used in visual tracking using notably color models [Pérez et al., 2002], shapes with spline parameters [Isard and Blake, 1998] or fusions of different measurement sources [Perez et al., 2004].

One interesting aspect of PFs is that they can momentarily track several modes through the use of samples. This can be useful in the presence of occlusion or distractors, as illustrated in Figure 2.6. Note finally that many probabilistic tracking methods use more complex models than the one shown in Figure 2.5. For instance, Perez et. al. use second-order dynamics [Pérez et al., 2002]. Zelniker et. al. formalize the modeling assumption that the target appearance can change over time and thus follow a generative framework in which they introduce the target appearance a_t, that in addition to the state x_t is also incrementally and recursively estimated [Zelniker et al., 2009][3]. Finally, Odobez et. al. propose a model in which the current observation depends on the current and previous state, as well as on the past observations [Odobez et al., 2006]. By doing so, they introduce a likelihood term that models the temporal correlation between successive images of the same object.

[3] In their paper, the target appearance is denoted by y_t, but we chose a_t to avoid confusion with the notations of our observations.

Extensions to Multi-Object Tracking. In a scene with multiple objects, the objective is to keep track of each of them while avoiding confusions with others. This task is made difficult due to occlusions. Mostly probabilistic methods have been studied to tackle this difficult problem. Running a separate PF for each target is not a viable option, as it does not allow to model their interactions, and failure modes are usually observed in practice when objects are passing close to one another and the tracker associated to one object locks onto a different object. To avoid this, some tracking approaches adopt a joint state space and explicitly model object interactions based for example on pairwise distances [Khan et al., 2003a] or overlapping [Smith et al., 2005].

PFs over a joint state space suffer from exponential complexity in the number of tracked targets, and computational requirements make it impractical to follow more than three or four objects [Khan et al., 2003b]. To alleviate this issue, the authors of [Khan et al., 2003a] propose to efficiently sample from the posterior distribution using Markov chain Monte Carlo (MCMC) sampling. Their joint filter behaves as a set of individual particle filters in the absence of interactions, but efficiently deals with interactions as objects get closer to one another. Very good results are obtained for the task of tracking 20 ants in a closed arena. Note that for this approach, the number of tracked objects needs to be known in advance and has to be fixed.

MCMC-based PFs can be extended to handle a variable number of objects by resorting to a reversible-jump method, first introduced in [Green, 1995] and that allows proposals changing the dimensionality of the space. For example, the authors of [Smith et al., 2005] propose such a trans-dimensional (reversible jump) MCMC algorithm handling the variability in the number of objects by introducing a set of moves allowing track death/birth mechanisms. The authors show good results obtained on outdoor surveillance videos. However, the complexity of these sequences is quite low as the maximum number of persons per frame is 4. Finally, to handle a varying number of objects, the likelihood models have to be designed as global observation models so as to compare configurations involving different numbers of targets. In [Smith et al., 2005] for instance, a pixel-based model comparing the coverage of foreground/background pixels by the multi-object tracking configurations, and another one based on color are proposed as the observation model.

2.2.4 Tracking by Association of Detections

An alternative to perform multi-object tracking is to directly formulate the task as the association of detections into tracks. While some of the previously mentioned methods might require manual re-initialization in case tracking is lost, this is not needed in detection-based tracking as the output of a detector is potentially used at every frame. Besides, on the contrary to prediction and update tracking methods like PFs that explore the state space with samples, tracking-by-detection methods do not require numerous image measurements to search for an optimal state. However, such methods should overcome detector flaws like misdetections,

false alarms and imprecise bounding box localizations.

Historically, data association techniques for tracking started with the works on Multi Hypothesis Tracking (MHT) [Reid, 1979] and Joint Probabilistic Data Association Filter (JPDAF) [Fortmann et al., 1983] for radar or sonar signals. MHT memorizes several association hypotheses on several time steps and defers the correspondence decision so as to potentially solve current ambiguities later. Due to the combinatorial nature of the algorithm, it can usually only be applied on limited-time windows for computational reasons. JPDAF on the other hand is a probabilistic data association method that assigns to each detection the probabilities that it was produced by each target, or clutter. Even though JPDAF does not have exponential complexity like MHT, it is suboptimal as it limits the temporal horizon to one frame. For visual tracking tasks, the same problems due to local ambiguities exist. In the following, we discuss the use of frame-to-frame as opposed to global association methods, we present the associated challenges and how they have been tackled in the literature.

Frame-to-Frame vs. Batch-based Data Association. Conventionally, the deterministic approach to multi-object tracking based on detections consists in attributing a cost to each association between each detected object in frame $t-1$ to each detected object in frame t based on appearance similarity and motion constraints [Yilmaz et al., 2006]. The problem is then one of combinatorial optimization and can be solved for example with optimal assignment methods like the Hungarian algorithm or greedy search methods. However, this one-to-one correspondence scheme is very sensitive to local ambiguities like misdetections or false alarms, and cannot model occlusions, entries or exits, unless we add a number of hypothetical points to cope with these situations.

Instead of solving a frame-to-frame correspondence problem, detection association can also be performed on a batch-of-frames basis, but the complexity of the optimization increases rapidly. Such batch-based methods are often referred to as *global* in the literature and they often model dependencies using graphs. In the next paragraphs, we discuss how several of these techniques address the modeling and optimization of the global association problem over batches of frames.

Graph Partitioning Approaches. Given a graph involving detections over a batch of frames, tracks can be represented by subsets of this graph. Brendel et. al. build a graph in which each node represents a pair of detections in consecutive frames, with a weight representing their similarity [Brendel et al., 2011]. Edges are built between incompatible pairs of nodes based on hard constraints. Tracks are deduced by finding the Maximum Weight Independent Set, which is the heaviest subset of non-adjacent nodes of the graph. Zamir et. al. [Zamir et al., 2012] describe another way of finding tracklets using Generalized Minimum Clique Graphs (GMCP), illustrated in Figure 2.7. The GMCP method consists in constructing a dense graph within a time window and performing data association iteratively by finding the minimum clique of the graph, corresponding to the set of detections with the most consistent and stable appearance features, and the most consistent motion according to a constant velocity model.

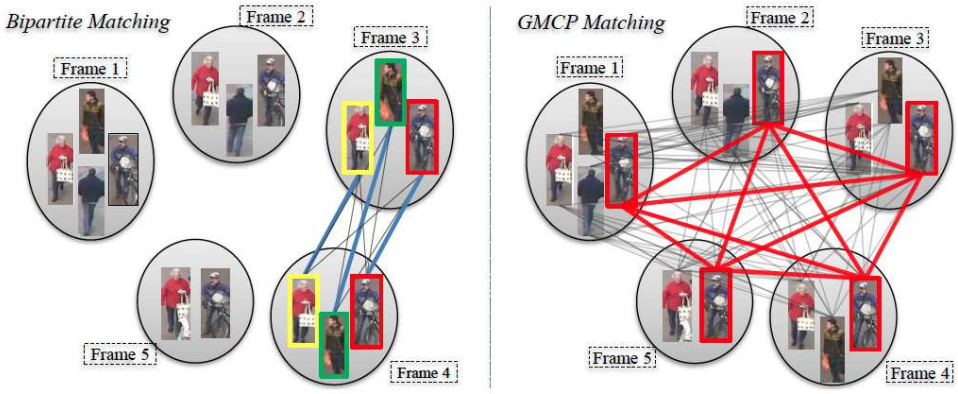


Figure 2.7: Frame-to-frame (left) vs. Generalized Minimum Clique Graphs (GMCP) matching (right). Gray and colored edges represent the input graph and optimized subgraph, respectively. On the contrary to frame-to-frame matching, GMCP matches one object at a time across a larger temporal span by finding the clique of the dense graph with minimum cost. (Image taken from [Zamir et al., 2012].)

This tracklet is then removed from the graph and the optimization process is repeated to find the next person until zero or few nodes remain in the graph. GMCP is a NP-hard problem and optimization is conducted with a local search that is computationally expensive. In the end, each frame is processed in more than 4 seconds, excluding the detection step.

Flow-based Techniques. Another way to see the optimization over global graphs is to find non-overlapping trajectories as edge-disjoint paths in the graph. Efficient network flow algorithms can be used in that context. For instance, in [Zhang et al., 2008], the authors use the same MAP formulation as [Huang et al., 2008] but embed it in a network framework where the min-cost flow algorithm can be applied. Their cost-flow network is illustrated in Figure 2.8.

In another direction, the authors of [Berclaz et al., 2009] directly formulate the problem of tracking as finding the flow of humans on a discrete grid space (representing the ground plane) that minimizes the cost of going through the detections obtained with the POM detector [Fleuret et al., 2008]). Their optimization method relies on a fast k shortest node-disjoint paths algorithm to provide real-time performance. However, their tracking algorithm ignores appearance information and is therefore prone to identity switches in places where the paths of different persons may intersect. The work in [Shitrit et al., 2013] extends the previous method of [Berclaz et al., 2009] by exploiting appearance cues to prevent such identity switches. The methods of [Berclaz et al., 2009] [Shitrit et al., 2013] are among the best performing techniques in multi-camera scenarios but require scene discretization. This poses the problem of precise cross-calibration, that can be difficult in the presence of important distorsion.

Modeling Higher-Order Terms. Working on large temporal spans gives the advantage that

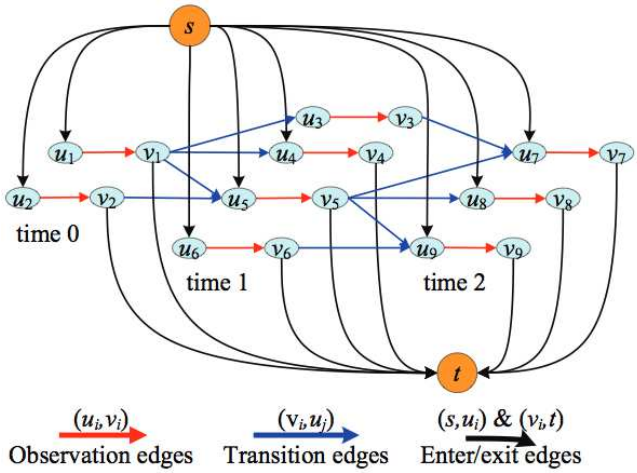


Figure 2.8: Example of cost-flow network with 3 timesteps and 9 observations. Each detection is represented by two nodes (u_i, v_i). Observation edges represent the cost introduced by the fact that a detection might be a false alarm. Transition edges represent linking costs. Enter/exit costs represent the cost to start/end trajectories. (Image taken from [Zhang et al., 2008].)

high-order costs can be taken into account. However, some global methods like network flows are not adapted to handle such terms, as they can usually deal with pairwise factors only. Butt et. al. propose to extend network flows to encode higher-order motion information computed over three observations [Butt and Collins, 2013]. Their formulation involves extra-constraints such that the problem is no longer equivalent to the min-cost network flow. Therefore, they propose an approximate solution based on Lagrangian relaxation to solve the problem.

Other tracking approaches consider smoothness terms of even higher order, i.e. at the trajectory level. For instance, Collins introduced smoothness terms defined as spline-based snake energies over entire trajectories [Collins, 2012]. In his formulation, a method similar to Iterated Conditional Modes (ICM) is cycling through pairs of adjacent frames, looking for block-optimal two-frame assignments, until convergence. In another direction, Andriyenko et. al. tackle the tracking problem by alternating between discrete data association and continuous trajectory estimation using global label costs incorporating a dynamic model to enforce long, persistent trajectories, and that penalize tracks that start or end far from the image border, as well as the total number of targets [Andriyenko et al., 2012]. These approaches all propose approximate algorithms that do not guarantee convergence. Moreover, the last two methods above rely solely on trajectories and do not involve the appearance of objects. This limitation can become critical when trajectories of people are approaching one another or in case of crossings.

Hierarchical Approaches. In order to overcome the limitations of frame-to-frame tracking, global methods mentioned previously conduct optimization over batches of frames. However,

such methods usually scale poorly because the complexity of the association rapidly increases depending on the size of the considered time window and the number of detections. To reduce the computational cost and progressively increase the temporal range for correspondences, hierarchical approaches can be considered. There, short and reliable tracklets (a few detections that can confidently be associated to a single person) are first generated at the low level and then merged at a higher level. As they consist of groups of detections, tracklets can be exploited to compute motion estimates, notably for prediction and linking. Besides, better representations of a person can be extracted from tracklets than from single detections. In the end, as there are fewer tracklets than detections, the computational burden is decreased.

One important drawback of such hierarchical approaches is that any wrong association made at the low level is then propagated to the next level of the hierarchy. In [Huang et al., 2008], the lower level associates pairs of detections in adjacent frames based on their similarity in position, size and appearance. The resulting tracklets are then fed into a Maximum A Posteriori (MAP) association problem which is solved by the Hungarian algorithm, and further refined at a higher level to model scene exits and occluders. Zamir et. al. perform GMCP in temporal windows of around 2 seconds and then perform tracklet association between consecutive windows at the next level [Zamir et al., 2012]. The main reasons for a hierarchical approach are the fact that optimization algorithms are often not efficient at handling longer time segments, and that the motion model may not be the same in the short and long term. Indeed, in the short term, a person is more likely to have a constant velocity vector, whareas in the long term, more complex motion patterns, with possible direction changes cannot be discarded.

Relying on different techniques, Shitrit et. al. also propose a hierarchical treatment to simplify their problem [Shitrit et al., 2013]. The algorithm starts with the tracks obtained by the method of [Berclaz et al., 2009]. These tracks are split at ambiguous points in terms of proximity with other tracks so as to obtain confident tracklets. In the next processing level, these tracklets become the nodes of the new graph taking into account image-appearance cues, and optimization is performed on this reduced graph, relieving the computational burden.

CRF Approaches. In order to model affinities as well as dependencies among observations, Conditional Random Field (CRF) tracking-by-detection approaches have been proposed. For instance, the hierarchical methods of Yang et. al. [Yang et al., 2011] [Yang and Nevatia, 2012a] take as input the low-level tracklets obtained with the method of [Huang et al., 2008] and represent each pair of linkable tracklets as a node in their CRF model. The energy of each node (unary term) depends on global tracklet affinity models based on appearance and motion. In [Yang and Nevatia, 2012a], the CRF formulation is also used to distinguish spatially close targets with similar appearances. For that purpose, on the contrary to most methods that only rely on pre-defined models, this work proposes to learn discriminative, target-specific appearance models online. The energy of each edge (pairwise term) between any nodes that have tail-close or head-close tracklet pairs is based on these discriminative pairwise models, as illustrated in Figure 2.9.

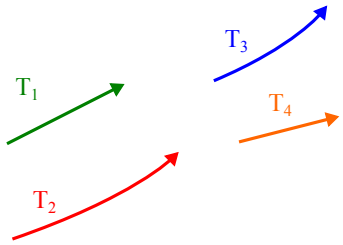


Figure 2.9: Illustration of head-close and tail-close tracklets. Tracklets T_1 and T_2 are tail-close, tracklets T_3 and T_4 are head-close. As tracklets T_1 and T_2 are tail-close, an appearance model that best distinguishes the two targets without considering other targets or the background is learned online. Positive and negative samples only concerned with these two tracklets are online collected and an appearance model based on color, texture and shape that is most discriminant for these two targets is learned with the Real Boost algorithm. Similarly, for the head-close tracklets T_3 and T_4, discriminative models are learned online for these two targets.

2.2.5 Detection-Guided Prediction and Update Tracking

Some methods take advantage of both prediction and update tracking and human detectors to follow multiple people. Indeed, resorting to fast and powerful human detectors is an essential step for good track initialization and state update. Reliable track initialization is indeed important, as object models like color histograms are usually extracted in this phase. Below, we review some methods using a human detector to suggest accurate positions to update current tracks at any given time instant, or that use prediction and udpate tracking to temporally propagate detections.

Data Association Between Detections and Tracks. When using independent prediction and update trackers for each target, if we want to apply the prediction/update methods mentioned in the single-object case, the measurements of interest must be uniquely identified to each object, that means that state estimation has to be coupled with data association techniques. In the method of [Breitenstein et al., 2011], a separate PF is automatically initialized for each person detected with high confidence. Then, a data association step is proposed to decide which new detection should guide which tracker, so that each detection is associated to at most one tracker. The tracker-detection matching score is based on a classifier trained on the tracklet at that is evaluated on the detection. The distance between the detection and the tracker, as well as motion direction are used to assess the association. Rather than only using high-confidence detections, the method also uses the continuous confidence of pedestrian detection outputs within the PF, as this provides a good indication of where a person might be under partial occlusion, lower contrast with background, unusual pose, etc.

Guiding Moves with Detections. When using prediction and update trackers over the joint state space, detections can be used to guide the possible trans-dimensional jumps. The authors of [Yao and Odobez, 2008b] propose to address the task of multi-person tracking in

surveillance scenarios using multiple, partially overlapping cameras. Their approach is very similar to [Smith et al., 2005] in spirit, as they model the problem using a joint state space allowing interaction modeling, and they apply Reversible Jump MCMC sampling for efficiency reasons and to handle a varying number of targets.

One contribution of [Yao and Odobez, 2008b] is the proposal to rely on a human detector to guide the visual tracker by driving the birth and update moves in the RJ-MCMC. The birth move proposes to randomly add one of the detected humans whose positions are far enough from the existing objects in the current configuration. The human detection output is also used in the update move, by potentially sampling a new location around one of the provided detections if it is close enough to the selected target.

Propagating Detections with Prediction and Update Tracking. Tracking-by-detection is strongly constrained by detection results. For instance, missed detections in the beginning or the end of a track cannot be recovered by standard tracking-by-detection approaches. Similarly, tracklets with long time gaps between them are hard to associate, especially if a common linear motion model is assumed. Moreover, applying a detector at every frame can be computationally expensive. To address these issues, researchers have proposed to combine detectors with prediction and update tracking methods.

In the context of head tracking, Benfold et. al. apply a head detector intermittently [Benfold and Reid, 2011a]. Then, they propagate each detection up to 4 seconds in the future and in the past relying on a KLT feature tracking method, allowing to obtain stable and accurate cropping of pedestrians' heads. The resulting data association is then formulated within a Bayesian framework over a sliding window of frames and solved through MCMC. A multi-threaded approach allows them to obtain real-time performance. In [Yang and Nevatia, 2012b], prediction and update tracking is employed to extrapolate tracklets so as to make tracklet associations more robust. Reliable tracklets are first defined as long tracklets with a reasonable number of valid (non-occluded) part features and then propagated by generating samples around the predicted positions estimated with a linear motion model. Potential extensions are selected as samples with the highest part feature similarities.

2.3 Pose Estimation

Tracking provides useful cues to estimate people's behaviors. Location-based analysis can be useful for forensics applications, for example in a tag-and-track scenario, where a person of interest is identified and the system is then asked to automatically retrieve the whereabouts of that individual over time. However, trajectories alone tell very little about interactions, with maybe the exception of spatial proximity which can indicate whether people belong to the same group. Additional behavioral cues like body or head pose provide richer information about human-to-human or human-to-environment interactions, as explained in Chapter 1. In this Section, we summarize several methods addressing human pose estimation in different

contexts and scenarios, along with their associated challenges. We then briefly review some techniques that aim at adapting classifiers to a target scene, or that address the dependence of the features to image location or person identity.

2.3.1 Scenarios and Methods

The level of details to which pose information can be extracted depends mainly on the resolution of the persons in the images and on the level of occlusion. For instance in good resolution images, the different body parts of people are usually large enough to extract distinctive characteristics from them so that refined pose representations can be estimated. These include articulated or skeletal representations. On the other hand, in a crowded surveillance setting with low resolution, the level of occlusion usually implies that pose estimation can only be conducted on parts of the pedestrians like the heads and be parameterized by only few parameters, e.g. simply the orientation. Below, we briefly review some popular methods in these different contexts, starting with a short explanation about poselets that are used by several pose estimation techniques.

Poselets. Poselets are a powerful representation of human pose which captures two aspects of people: the 3D coordinates of body joints in configuration space, and the resulting appearance (pixel values) in a 2D image [Bourdev and Malik, 2009]. A poselet describes a part of one's pose and is defined as being tightly clustered in both appearance and configuration space. In their paper [Bourdev and Malik, 2009], the authors introduce the thoroughly annotated H3D dataset, from which they retrieve poselets by first scanning rectangular windows at different positions and scales within human annotations. Then, based on a residual distance, closest match queries are used to get poselet clusters. This process produces semantically similar yet visually different examples. For each cluster, representing one poselet, a SVM classifier based on HOG features is trained. Responses of such poselet detectors within a human bounding box can provide a distributed representation of the pose, indicating to which degree each poselet is present based on the person's appearance [Maji et al., 2011]. This poselet activation vector can then be used as input of a regression engine for torso and head pose estimation. The drawback of all poselet-driven approaches for pose estimation is that they require thorough annotations. For instance, the H3D dataset consists of annotations of 2000 samples with 3D locations, 19 keypoints and pixel-level labels.

Fine Pose Estimation Approaches. In good resolution images with non-occluded persons, pose estimation methods can rely on sophisticated representations. For instance, Pishchulin et. al. propose to use poselets to get a refined estimation of human pose in terms of an articulated representation [Pishchulin et al., 2013]. Sample results are shown in Figure 2.10. Their approach selects a pictorial structure out of a collection, where a pictorial structure represents the body configuration as a collection of rigid parts with pairwise part connections. In another direction, Lee et. al. propose a generative model comprising articulated structure, shape and clothing models to estimate upper body pose in middle resolution images [Lee and Cohen,

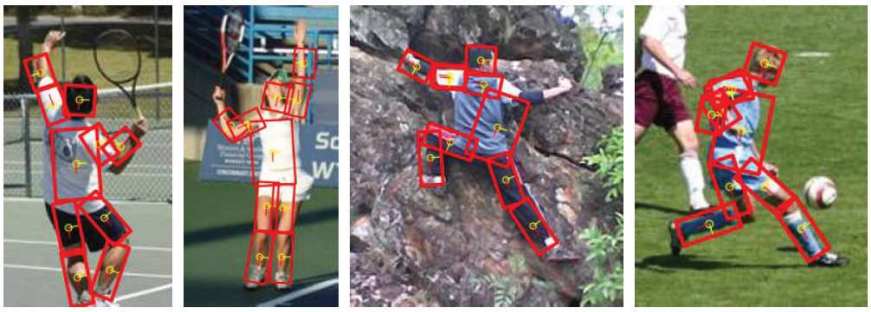


Figure 2.10: Sample articulated pose estimation results obtained by [Pishchulin et al., 2013].

Figure 2.11: Sample upper-body pose estimation results obtained by [Lee and Cohen, 2004]: original images (top), estimated pose (bottom).

2004]. Sample results are shown in Figure 2.11. To search the solution space, the authors propose to use the MCMC framework. Similarly, Chen et. al. address the pose estimation of upper bodies [Chen et al., 2011c]. For that purpose, they learn structured discriminative regression models taking features as input and producing structured coordinates as output, representing the 2D locations of 6 upper body parts. On the contrary to other methods working on still images, the approach of [Andriluka et al., 2010] considers videos, in which temporal coherency can be exploited. In a first step, 2D human detection is applied to each frame and 2D pose estimation is conducted on still images, using pictorial structure models. In a second stage, tracking-by-detection is applied on a small number of consecutive frames, and temporal coherency is exploited to refine 2D detections and viewpoint estimates. The method performs well in realistic street conditions.

Coarse Pose Estimation Approaches. Pose estimation in surveillance scenarios has recently garnered some attention. The main difficulty in this context is the low resolution of images making sophisticated pose models like the ones presented above ill-suited to the problem. In these conditions, many approaches rather concentrate on head pose, or body and head pose

Figure 2.12: Sample body and head pose estimation results obtained by [Gong et al., 2010]: Bounding box colour and text indicate body pose and magenta dials indicate head pose.

estimation. Robertson and Reid address the task of head pose classification in low resolution video images, using feature vectors based on skin detection [Robertson and Reid, 2006]. Yaw head pose angles are discretized in 8 classes. Their main assumption is that the proportion of skin to non-skin pixels within a head patch under a given pose is a relatively invariant cue between different people. The descriptors of the test head patches are matched to the closest training data using an efficient binary-tree framework. Interestingly, the authors also propose to exploit the direction of motion as a guide to resolve ambiguity and improve matching. This method relies critically upon good segmentation of skin and hair in the head images and on head location. Moreover, the coupling between motion direction and pose is constant regardless of the speed, and thus provides bad information at low magnitude, since speed orientation is highly noisy in such cases.

In the same context, Orozco et. al. propose a method that does not need explicit segmentation of skin and hair regions [J. Orozco et al., 2009]. They model features as the maximum Kullback Leibler divergence between each input image and multiple pose mean templates learned from training data. Such descriptors are robust, even though texture distributions are not modeled explicitly. In another direction, Gong et. al. propose to jointly detect pedestrians and estimate their body and head pose thanks to an ensemble of pose-sensitive body models, which are modified versions of the deformable part-based models of [Felzenszwalb et al., 2010]. Sample body and head pose results are shown in Figure 2.12. However, they perform detection and pose estimation separately on the body and the head, notwithstanding the fact that these body parts are coupled by anatomical constraints.

2.3.2 Adaptation

Until recently, one important limitation of existing methods was the use of pre-trained classifiers not adapted to the test data, in spite of obvious variabilities in appearance, viewpoint and illumination between the two sets. To address this issue, some authors have proposed to perform scene adaptation, while others tackled the problem of the dependence of the pose features to location and person identity.

Scene Adaptation. Appearance-based pose estimation methods require test and ground truth

data to come from the same or similar scenes. However, it would be tedious to annotate images for each new scene. To address this issue, Chamveha et. al. propose to automatically collect training data from the test scenes using the walking direction as ground truth [Chamveha et al., 2011]. Unreliable walking direction samples are rejected and oversampling is applied to handle the potentially imbalanced generated dataset. With a similar philosophy, the authors of [Benfold and Reid, 2011b] leverage on weak labels given by the velocity direction in the test set to learn a scene-specific head pose classifier. Recently, Chen and Odobez proposed a coupled classifier adaptation framework [Chen and Odobez, 2012] for body and head pose estimation in surveillance videos. Similarly to [Benfold and Reid, 2011b, Chamveha et al., 2011], reliable tracking motion information within the test set is used for body pose adaptation. Differently, however, in addition to the coupling of body pose with motion, they also use the natural coupling between body and head pose and propose to leverage on external labeled pose datasets to anchor estimation with sufficient amount of data. To that end, within a kernel framework, they formulate adaptation as the task of jointly learning body and head pose classifiers that perform well on external labeled pose datasets as well as on the test data with weak motion labels and providing close (coupled) output for body and head pose.

Location Adaptation. Position-induced appearance changes due to distorsions in appearance owing to camera perspective and scale can have an effect on the extracted pose features. In the context of head pose estimation in a multi-camera setting, the authors of [Yan et al., 2013] actually observe that a pose classifier trained in a specific part of the physical space (one quadrant Q_1 of the room) has a worse performance when applied to test samples coming from the other quadrants (Q_2,Q_3,Q_4) as opposed to when it takes samples coming from Q_1 as input. To tackle this problem, region-specific classifiers can be learned [Yan et al., 2013], with the drawback that this requires more training data, and that these training data should be available for all the discrete regions for which classifiers are learned; or features can be corrected as if they were all coming from the same location, for example by warping them to a fixed, reference location leveraging on 3D geometry [Rajagopal et al., 2013].

Person Independence. Usually, pose detectors are learned independently of the persons' identities, which is equivalent to assuming that pose data features of different individuals lie in a single, continuous manifold. On the contrary to this hypothesis, the pose data space can instead be considered as a union of submanifolds which characterize different persons. Yan et. al propose a supervised submanifold embedding algorithm for person-independent head pose estimation, where the main idea is to find a subspace where the features of different individuals are aligned [Yan et al., 2009]. In other words, samples from different persons yet with similar pose will be projected to close points in the low-dimensional space. This embedding promotes the generalization capability of the training data. One limitation of this approach, however, is that it requires explicit identity labels within the training set.

2.4 Behavior Analysis

The ultimate goal of a commercial surveillance system would be to provide situation awareness, i.e. to understand and predict behaviors of people or crowds as they move around the network of cameras, e.g. for forensics applications. Leveraging on location and pose information, perceptual algorithms can be designed to analyze human behavior from videos. In this Section, we briefly describe some works dealing with behavior analysis from videos. We then focus on methods explicitly exploiting behavioral cues like trajectories or pose to infer human behavior.

2.4.1 Behavior Analysis for Surveillance

Applications. The surveys [Hu et al., 2004] and [Candamo et al., 2010] review the challenges that have to be overcome to perform intelligent surveillance in open spaces. The task of behavior analysis is often separated in three different levels: single person, multiple-person interactions and person-facility/object/location interactions. People interactions may include for example behaviors like following, meeting, splitting or moving as a group. The way people interact with objects, for example their luggage could be of great importance in typical metro station surveillance. Furthermore, in the context of surveillance, spotting abnormal or antisocial behaviors can be practically relevant. For example, if a system could automatically detect such events, an alarm could be triggered to alert the monitoring personnel.

Approaches. Behaviors of single persons include actions such as walking, running, picking up an object, etc. Approaches in the literature recognize such behaviors based on features related to individuals. For instance, in [Danafar and Gheissari, 2007], detection bounding boxes are partitioned heuristically in three body parts where quantized optical flow is computed on each part and fed to an SVM classifier to recognize simple behaviors. Human interaction recognition has been addressed by using various techniques like Event Calculus [Artikis and Paliouras, 2009], Coupled Hidden Markov Models (CHMMs) [Oliver et al., 2000] or model-based motion tracking combined with multi-layer finite state automata [Park and Aggarwal, 2003]. The authors of [Artikis and Paliouras, 2009] present a rule-based method in which long-term behaviors are defined as pre-defined combinations of short-term behaviors like walking or running, used as input to the system. In [Oliver et al., 2000] [Park and Aggarwal, 2003], interactions are represented in a probabilistic way as sequences of states, but the presented applications involve two people only. Wang et. al. present a method to recognize behaviors involving a higher number of interacting people [Wang et al., 2006]. Finally, the understanding of human interactions can also be cast in a multi-camera network situation where we can exploit view switching for occlusion handling [Park and Trivedi, 2008], and multi-modal cues (e.g. audio and video) for behavior recognition can also be useful, as they have been shown to outperform any single modality for the detection of unusual events [Kuklyte et al., 2009].

2.4.2 Exploiting Tracking and Pose for Behavior Analysis

Exploiting Trajectories. Trajectories contain location and motion information that can give useful information about people's behaviors. As an example, in [Takahashi et al., 2010], several features are extracted from the individual trajectories obtained by a tracker, and discriminative methods are used to recognize behaviors. For example, running is easily recognized as opposed to meeting and putting an object down, for which an additional verification step is needed for discrimination.

Exploiting Richer Behavioral Cues. Other behavioral cues can give richer information about behaviors. For instance, Maji et. al. perform action recognition in still images [Maji et al., 2011] based on poselet activation vectors [Bourdev and Malik, 2009], which give a distributed representation of the pose. In [Robertson et al., 2007], the probability distributions of positions, velocities and head and posture information are fused within a Bayesian network to infer spatio-temporal behaviors. Actions are deduced by taking the Maximum Likelihood estimate of all possible spatio-temporal actions at each time step. Hidden Markov Models (HMMs) are then used to encode known rules about behavior as sequences of actions.

Adding the Depth Modality. Most of the behavior analysis approaches in the literature take visual information as input. Recently, traditional video signals have been complemented with the depth modality to improve behavior recognition. In [Ni et al., 2011], a new color-depth dataset (RGBD-HuDaAct) is introduced. The authors demonstrate the superiority of multi-modality (depth and color) as compared to uni-modality (color) for action recognition. This work focuses on healthcare applications for seniors at home. Similarly, the approach of [Ni et al., 2013] is interested in recognizing human-to-human as well as human-to-objects interactions in RGB+D modalities. Detection and tracking are first performed. Depth-induced constraints help improve these two steps. Key poses are extracted and used as a primitive representation for human actions. In practice, each tracklet is assigned a unary attribute based on its key poses. Then, tracklet pairwise interactions based on relative distance, relative velocity and temporal ordering are modeled, and action recognition is conducted within a Bayesian network.

Beyond. Finally, we should always keep in mind that, as anthropologists say, *space speaks* [Cristani et al., 2010], namely that people, even if they seem to move randomly, do in fact respect some patterns governed mainly by social mechanisms. Therefore, computer vision and pattern recognition approaches should not work independently of these social constructs associated to behaviors and space occupancy, but indeed use them. Instead of using detection, tracking and pose estimation as elementary bricks upon which behavior analysis is built, we can incorporate knowledge from social constructs as advised by [Cristani et al., 2010] to help these steps gain flexibility and robustness. For instance, the authors of [Antonini et al., 2006] give some behavioral priors for detection and tracking of pedestrians using discrete choice models based on available space and space occupancy and show their importance in terms of

performance improvement.

2.5 Conclusion and Perspective

2.5.1 Tracking

Summary. Our literature review indicates that detectors are powerful tools that are essential for the task of tracking. Indeed, observation models in prediction and update tracking methodologies are not able to properly initialize a track and are subject to drift, which encourages the use of a detector to guide the tracking process [Yao and Odobez, 2008b] [Breitenstein et al., 2011]. Many recent state-of-the-art tracking methods rely on the output of detectors at every frame [Zhang et al., 2008] [Collins, 2012] [Zamir et al., 2012] [Andriyenko et al., 2012] [Yang and Nevatia, 2012a] and formulate the task of tracking directly as a data association between detections. On the contrary to prediction and update tracking methods, features only need to be extracted within each detection, and no multiple image measurements to search for an optimal state are involved.

Tracking techniques are sensitive to occlusions which typically increase in crowded scenarios and hamper the performance. Indeed, when a person is not visible, or partially visible, the extraction of its descriptors, e.g. color histograms might be corrupted and lead to unreliable association scores. To address this issue, we could rely on a multi-camera setting [Berclaz et al., 2009] [Shitrit et al., 2013]. However, in surveillance applications, multi-camera scenarios are not the common case, and pose the problem of data synchronization between several sensors. Furthermore, in the context of open spaces, camera calibration can be difficult to achieve in presence of important distorsion. In another direction, context before and after the occlusion can also be important to solve ambiguities. Several approaches thus conduct *global* data association within *long-term* temporal windows. Long-term temporal spans are also useful to define global costs, that can penalize trajectories based on smoothness [Collins, 2012], or if they start or end far from exit zones like image borders [Andriyenko et al., 2012]. Finally, motion information at the temporal proximity of the occlusion can be used for prediction and linking.

Tracking-by-detection is strongly constrained by detection results. Missed detections in the beginning or the end of a track cannot be recovered by standard tracking-by-detection approaches. Similarly, long time gaps in the middle of a track make the association difficult. Moreover, detectors are usually good at detecting upright humans only. This limitation poses the problem of finding people sitting or in any other unusual pose. In general, complementing detections with prediction and update tracking is a good way of dealing with the sparse and unreliable output of detectors [Benfold and Reid, 2011a] [Yang and Nevatia, 2012b].

Our Approach. In this thesis, we adopt a tracking-by-detection framework. In our approach, we focus on monocular scenarios, which are currently closer to real open space situations. On

the contrary to standard approaches modeling short-term affinities between detections, like the model depicted in Figure 2.13, we propose instead to:

- Define a cost taking all links into account, differently from flow-based approaches that would evaluate solutions based only on the costs selected by the hypothesized trajectories (see right part of Figure 2.13). In our framework, the validity of a trajectory is not only evaluated based on how similar the detections within the track are, but as well on how dissimilar they are from other detections. For that purpose, like [Lathoud and Odobez, 2007], which addresses the problem of tracking sound sources in a one dimensional space, we do not consider only a similarity hypothesis but also a dissimilarity hypothesis for each pair of detections. By contrasting the two hypotheses for each detection pair, the model is more robust to assess the appropriateness of a given association.
- Consider sufficiently large temporal windows so as to overcome the sparsity in the output of detectors and also to better deal with occlusions. In practice, we benefit from important temporal context by connecting detection pairs not only between adjacent frames, but between frames within a long time interval T_w.
- Exploit visual motion at the detection level. As opposed to hierarchical approaches that would propagate any wrong association to the next stage of the hierarchy, our framework sticks to the detection level. By doing so, we can exploit visual motion directly estimated from image measurements instead of relying on motion estimates computed from hypothetical associations (tracklets).

We formulate our tracking framework with a CRF model. Yang and Nevatia also use a CRF model in a tracking-by-detection framework [Yang and Nevatia, 2012a]. However, their approach is different insofar as they model the affinities and dependencies between tracklets and do not work at the detection level. Apart from the larger connectivity between pairs of detections, our CRF framework also differs in that we consider confidence scores for the features, as well as higher order potentials in the form of label costs. Our confidence scores are a way to model the reliability of the features in terms of inter-person occlusion, and can therefore be related to [Yang and Nevatia, 2012b], which performs explicit occlusion reasoning. Similarly to [Andriyenko et al., 2012], we introduce global costs that penalize long tracks starting or ending far from pre-defined boundaries. Finally, another of our contributions is an unsupervised way of learning scene-specific model parameters.

Details about our tracking framework are given in Chapters 3, 4 and 5.

2.5.2 Pose Estimation

Summary. Although many successful approaches handle only head pose estimation in smart room, multi-camera settings [Yan et al., 2013] [Rajagopal et al., 2013], or articulated pose estimation in still images with good resolution [Lee and Cohen, 2004] [Chen et al., 2011c] [Pishchulin et al., 2013], some other methods explicitly tackle the problem of pose estimation in monocular surveillance videos, where challenges from occlusions and low resolution arise

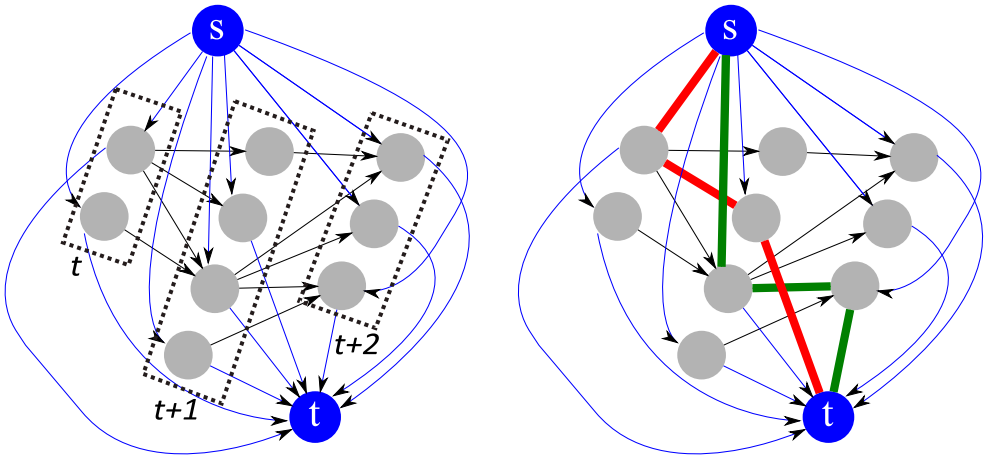


Figure 2.13: Illustration of a typical flow-based model for tracking-by-detection. Left: graphical model linking detections in each frame to adjacent detections and to a source and target node, representing the events of starting and ending a trajectory. Right: two possible paths are highlighted in green and red; the solution is only evaluated using the costs of the highlighted links, ignoring all the others.

[Robertson and Reid, 2006, J. Orozco et al., 2009, Gong et al., 2010, Benfold and Reid, 2011b, Chamveha et al., 2011, Chen and Odobez, 2012]. In this context, most of the works concentrate on the head, and only few approaches deal with the learning of body and head pose [Gong et al., 2010, Chen and Odobez, 2012]. When estimating pose in videos, tracking information is useful to refine pose estimates thanks to temporal coherency reasoning [Andriluka et al., 2010], as well as to perform scene adaptation, as walking direction usually provides a good prior on the pose [Benfold and Reid, 2011b, Chamveha et al., 2011, Chen and Odobez, 2012]. Another challenge is that pose features usually are sensitive to location, due to perspective effects [Yan et al., 2013, Rajagopal et al., 2013] and identity [Yan et al., 2009].

Our Approach. Given the potentially low resolution of surveillance images, articulated pose estimation is not realistic. Instead, we focus on single camera estimation of body and head pose yaw angles, which already are informative behavioral cues in this context. Despite the obvious link between those two cues, they are mostly treated separately in the literature. Alternatively, we propose to jointly estimate them in a temporal filtering framework, leveraging as well on trajectory information provided by our tracker.

We propose to use velocity information inferred from trajectories as a prior for body pose. Such a coupling between movement direction and pose has been exploited in previous work [Robertson and Reid, 2006], but problems remain when people are static or have only slow movement. To tackle this problem, we condition the coupling on the speed.

Chen et. al. [Chen and Odobez, 2012] present an interesting framework, which addresses both coupling and classifier adaptation, and gives state-of-the-art pose estimation performance on

several datasets. Scene adaptation is achieved by constraining classifiers to produce similar outputs on similar features. However, we argue that the features might not be aligned between the training and test datasets, for instance due to viewpoint or illumination differences. Therefore, we propose to align these manifolds to improve the classifier outputs. Manifold alignment, and especially semi-supervised alignment have been addressed in the literature, notably by [Ham et al., 2005, Wang et al., 2012].

Details about our joint temporal filtering framework for pose estimation, as well as our manifold alignment approach are given in Chapter 6.

3 A CRF Model for Detection-Based Multi-Person Tracking

3.1 Introduction

Tracking-by-detection methods have become increasingly popular in the vision community. Our literature review presented in Section 2.2 indeed showed the relatively recent and growing research interest for this paradigm. Several issues have to be taken into account when tracking multiple targets in an open space, some of which like occlusions, unpredictable motions or appearance changes have been exposed in Section 2.2. These difficulties are especially hard to handle in single camera situations. Tracking-by-detection approaches must deal with detectors' caveats, as detectors usually produce false alarms and misdetect objects. False positives create spurious instances that add a noise component to the problem. On the other hand, missed detections, mainly due to occlusions between people and by scene occluders, or unusual poses like people sitting or bending, create gaps that make data association more difficult.

Most existing tracking-by-detection approaches initially link detections to build tracklets and then find an optimal association of such tracklets. Although obtaining impressive results on several datasets, these approaches ultimately rely on low-level associations that are limited to neighboring time instants and reduced sets of features, like color and adjacency. Hence, a number of higher-level refinements with different sets of features and tracklet representations are required in order to associate tracklets into longer trajectories.

In this thesis, we explore an alternative approach that relies on longer-term connectivities between pairs of detections for multi-person tracking. In Section 3.2, we give an overview of our approach. Sections 3.3 to 3.6 give a detailed description of the components of our framework, while Section 3.7 summarizes the main elements of our model and concludes the chapter.

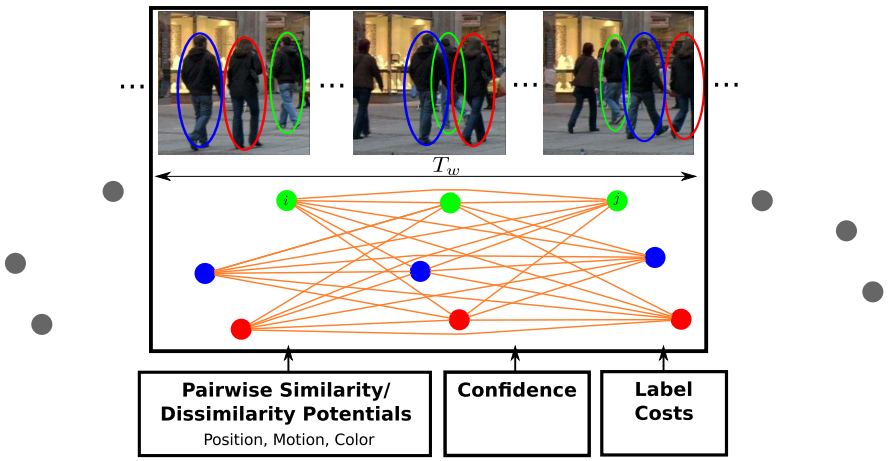


Figure 3.1: Overview of the proposed tracking-by-detection approach. Detections in incoming frames are represented as observation nodes. Pairs of labels/observations within a temporal window T_w are linked to form the labeling graph, thus exploiting longer-term connectivities (note: for clarity, only links having their two nodes within the shown temporal window are displayed). Pairwise feature similarity/dissimilarity potentials, confidence scores and label costs are used to build the energy function to optimize for solving the labeling problem within the proposed CRF framework.

3.2 Approach Overview

In this thesis, we propose a detection-based approach to multi-person tracking. We formulate tracking as a labeling problem in a Conditional Random Field (CRF) framework, where we target the minimization of an energy function (see Section 3.3.3) defined upon pairs of detections and labels, as well as higher-order terms. Our approach is summarized in Figure 3.1, and has the following characteristics:

- In contrast to most existing approaches, we model the association at the detection level and therefore do not resort to potentially erroneous association hypotheses. In our framework, the pairwise links between detections are not limited to pairs of detections in adjacent frames, but between frames within a time interval T_w (from ±0.5s to ±2s). Hence, the notion of tracklets is not explicitly needed to compute features for tracking, allowing us to keep the optimization at the detection level.
- A novelty of our approach is to directly use the visual motion computed from the video sequence for data association. Indeed, while position and color are important features, motion becomes a discriminative feature in more crowded situations, especially when considering detections 1 to 2 seconds apart, as is our aim. Previous methods resorted to tracklet creation or cumbersome tracklet hypothesizing and testing optimization to obtain discriminative motion information. Here, we propose that such information can conveniently be replaced by direct observations.
- Another differential trait of our method is the form of energy potentials, formulated here not only in terms of similarity but also dissimilarity between pairs of detections,

leading to a more discriminative model. Moreover, the proposed potentials depend not only on sets of features, but also on the time interval between two detections. In this way, we model how discriminative a feature is given the observed distance in the feature space and the time gap between pairs of detections, an important characteristic when considering long-term connectivity. The modeling of pairwise factors is described in Section 3.4.

- In order to take into account not only the actual feature distance value but also its reliability, we exploit a set of confidence scores per feature to characterize how trustable the pairwise distances are. For instance, visual cue distances are given a lower confidence whenever one of the detections is possibly occluded. These scores ultimately allow to re-weight the contribution of each feature based on spatio-temporal cues, and to rely on the most reliable pairwise links for labeling. This is important near occlusion situations, where thanks to long-term connectivity, the labeling can count on cleaner detections just before or after occlusion to propagate labels directly to the noisier detections obtained during occlusion, instead of through adjacent drift-prone frame-to-frame pairwise links only. The confidence weighting procedure is explained in Section 3.5.
- Finally, compared to some existing CRF approaches for tracking [Yang et al., 2011, Yang and Nevatia, 2012a] a novel aspect of our framework is that the energy function includes higher order terms in the form of label costs. The aim of such label costs is to model priors on label fields. In our tracking framework, this translates into penalizing the complexity of the labeling, mostly based on the fact that sufficiently long tracks should start and end in specific areas of the scene. We are interested in static camera settings, in which scene-specific maps can be defined for that purpose. Details about label costs are given in Section 3.6.

3.3 Problem Formulation

We start this section by introducing the features we selected to describe each detection. We then formulate our statistical labeling framework in terms of a CRF model, give the energy formulation of the problem and finish by summarizing the notations.

3.3.1 Data Representation

Let us define the set of detections of a video sequence as $R = \{r_i\}_{i=1:N_r}$, where N_r is the total number of detections. The features we choose to represent our detections are articulated around three cues: position, motion and color. More precisely, each detection is defined as

$$r_i = (t_i, \mathbf{x}_i, \mathbf{v}_i, \{\mathbf{h}_i^b\}^{b\in\mathscr{B}}) \tag{3.1}$$

which comprises the following features:

- t_i denotes the time instant at which the detection occurs;

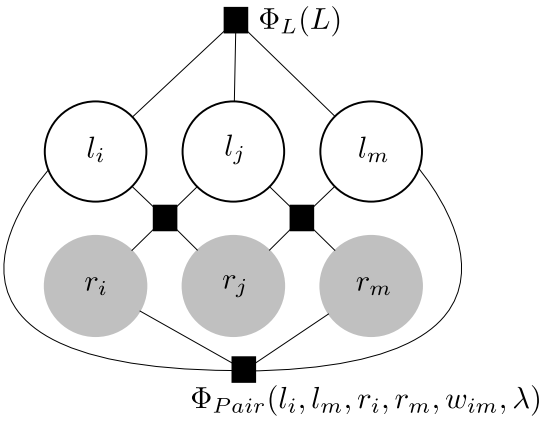


Figure 3.2: Factor graph illustration of our Conditional Random Field model. A factor graph describes the way in which a distribution decomposes into a product of local functions. Each factor (black square) represents a pairwise or higher-order potential which depends only on the variables it is connected to. Shaded nodes represent the observations (detections) and white nodes represent the hidden labels we want to infer.

- $\mathbf{x}_i$ denotes the 2D image or ground-plane position depending on the availability of calibration information;
- $\mathbf{v}_i$ denotes the 2D image plane visual motion computed from the video sequence;
- $\mathbf{h}_i^b$ with $b \in \mathscr{P} = \{\text{whole}, \text{head}, \text{torso}, \text{legs}\}$ denotes a set of multi-resolution color histograms extracted from a set $\mathscr{P}$ of body parts.

Note that, in contrast to existing approaches, each detection has an associated motion vector $\mathbf{v}_i$, which is independent of the label field, i.e., motion is not derived from tracklets but from detections. In our case, a robust estimation of this motion is conducted by performing a weighted average of the displacement estimated at several body part patches resulting from a part-based human detector, where the weight of each displacement vector indicates the motion reliability based on the matching distance and how uniform the patch is. For the color descriptors, we define parts that represent the whole detection region as well as three different spatial regions (head, torso, legs) to take advantage of both a holistic representation and heuristically defined body parts. Implementation details about cue extraction are given in the experimental Section 5.3.

3.3.2 CRF Modeling

We formulate multi-object tracking as a detection labeling problem, in which we seek for the optimal label field $L = \{l_i\}_{i=1:N_r}$, where l_i denotes the label of detection r_i, so that detections within a same track should be assigned the same label. Labels can take their values in $\mathbb{N}$ as we do not know in advance the number of objects in the scene. A generative approach would try

to model the joint distribution of the hidden labels and the observations as:

$$p(L,R) = p(R|L)p(L) \tag{3.2}$$

This formulation attempts to model how hidden labels *generate* the observed features, or in other words how to *sample* features given the label. In our approach, we use a discriminative model instead, by directly modeling the conditional distribution $p(L|R)$, which is enough for the labeling task. The main advantage is that the discriminative formulation does not need an explicit model of $p(R)$, which can be hard to design if the observation features are highly dependent.

To model $p(L|R)$ and solve the labeling task, we rely on a CRF formulation. A CRF is a statistical modeling considered as a structured prediction method, in the sense that it takes into account *neighborhood* information to label the samples. Furthermore, in a CRF, the hidden states are globally conditioned on the observations. A formal definition of a CRF is given in [Lafferty et al., 2001]. Translated in our context, the definition is the following.

Definition. (R,L) *is a CRF in the case, when conditioned on* R*, the random variables* l_v *obey the Markov property with respect to the graph:* $p(l_v|R,\{l_w\}, w \neq v) = p(l_v|R,\{l_w\}, w \sim v)$*, where* $w \sim v$ *means that* w *and* v *are neighbors in the graph.*

A CRF can be represented as a factor graph which factorizes the conditional distribution over the dependent subsets of variables. Our CRF model is illustrated in Figure 3.2 and we explain its different modeling components in the following and the other sections of this chapter. Learning and optimization will be discussed in the next chapter.

We model the posterior probability of the label field given all the observations as follows:

$$p(L|R,\lambda) = \frac{1}{Z(R)}\Phi_{Pair}(L,R,\mathcal{W},\lambda)\Phi_L(L) \propto \left(\prod_{(i,j)\in\mathcal{I}}\prod_{k=1}^{N_f}\Phi_k(l_i,l_j,r_i,r_j,w_{ij}^k,\lambda^k)\right)\Phi_L(L) \tag{3.3}$$

where $\mathcal{I}$ denotes the set of connected detection pairs, and for each detection pair we introduce N_f factor terms Φ_k to account for different pairwise feature similarity/dissimilarity measurements; the variable $\lambda = \{\lambda^k\}$ denotes the set of parameters associated with each of these factors, and $\mathcal{W} = \{w_{ij}^k\}$ with $w_{ij}^k \in [0,1]$ denotes the set of confidence scores associated with each factor and detection pair. The above formulation also incorporates a prior Φ_L over label fields in terms of higher-order potentials.

Factor modeling. The factors Φ_k are modeled using a *long-term*, *two-hypothesis*, *time-interval dependent* and *confident* pairwise approach, as explained in the following. Namely, we can define the graph representing our model with the following characteristics:

- **Long-term links:** we establish links between detection pairs (r_i, r_j) using a *long-term*

connectivity, i.e. the set of connected pairs is defined as:

$$\mathscr{I} = \{(i,j) \ / \ 1 \leq \Delta_{ij} = |t_j - t_i| \leq T_w\}. \tag{3.4}$$

where T_w is our long-term window size.

- **Two-hypothesis model:** for each factor term, a feature function $f_k(r_i, r_j)$ is defined that computes a distance measure between detection characteristics. Then, the corresponding CRF pairwise factor is defined as:

$$\Phi_k(l_i, l_j, r_i, r_j, w_{ij}^k, \lambda^k) \stackrel{\Delta}{=} p(f_k(r_i, r_j)|H(l_i, l_j), \lambda_{\Delta_{ij}}^k)^{w_{ij}^k}. \tag{3.5}$$

where the symbol $\stackrel{\Delta}{=}$ means by definition. This factor depends on the distribution $p(f_k|H, \lambda_\Delta^k)$ of the feature distance f_k under *two different hypotheses* corresponding to whether the labels are the same or not, that is:

$$H(l_i, l_j) = \begin{cases} H_0 & \text{if } l_i \neq l_j \\ H_1 & \text{if } l_i = l_j \end{cases} \tag{3.6}$$

- **Time-sensitive factors:** the feature distribution under the two hypotheses is *time-interval sensitive*, in the sense that we define such a distribution for each time interval Δ that can separate two detections. This allows to take into account the evolution of the feature according to this time parameter. In the model, the dependency is introduced thanks to the use of different sets of parameters λ_Δ^k for each interval Δ.

- **Confidence weighting:** the factor Φ_k defined by Eq. 3.5 also accounts for the *confidence* w_{ij}^k we have between detection pairs by powering the feature distribution with w_{ij}^k. Intuitively, lower confidence values will flatten the distribution of a feature leading to a less discriminative potential, lowering the factor difference under the two hypotheses. At the limit, if $w_{ij}^k = 0$, the factor of a given feature distance will be identical (equal to one) under the two hypotheses.

In addition to the feature factors, our model also comprises a higher-order term prior Φ_L that penalizes labeling solutions that do not meet some criteria defined from prior knowledge. For instance, labeling solutions producing tracks that start or end in improbable locations of the scene should be penalized. The penalization procedure is based on costs that, on the contrary to the pairwise potentials, enable to reason at the global level, and that we call label costs.

In the following, we first present the equivalent energy minimization formulation of the problem of Equation 3.3. The description of the design of the different cue-dependent pairwise feature functions f_k that we have considered along with their associated distributions and the parameters that characterize them is deferred to Section 3.4. Details about the confidence weights w^k that are used to weight the contribution of each factor term of a detection pair are given in Section 3.5. Finally, we define label costs in Section 3.6.

3.3.3 Energy Minimization

We wish to maximize the expression of Equation 3.3, which represents the conditional probability of the label field given the obervations (detections) and the model parameters. If we replace the CRF pairwise factors of Equation 3.3 with their definition given by Equation 3.5, we get the following maximization problem:

$$\underset{L}{\operatorname{argmax}}\, p(L|R,\lambda) = \underset{L}{\operatorname{argmax}} \left(\prod_{(i,j)\in\mathscr{I}} \prod_{k=1}^{N_f} p(f_k(r_i,r_j)|H(l_i,l_j),\lambda^k_{\Delta_{ij}})^{w^k_{ij}} \right) \Phi_L(L) \tag{3.7}$$

The maximization problem of Equation 3.7 is equivalent if we divide the expression by the following quantity, which is a constant with respect to the label field L[1]:

$$C_0(R) = \prod_{(i,j)\in\mathscr{I}} \prod_{k=1}^{N_f} p(f_k(r_i,r_j)|H_0,\lambda^k_{\Delta_{ij}})^{w^k_{ij}} \tag{3.8}$$

By further taking the negative logarithm of the resulting expression, the problem boils down to the following minimization:

$$\underset{L}{\operatorname{argmin}} \left(\sum_{(i,j)\in\mathscr{I}} \sum_{k=1}^{N_f} w^k_{ij} \log \left(\frac{p(f_k(r_i,r_j)|H_0,\lambda^k_{\Delta_{ij}})}{p(f_k(r_i,r_j)|H(l_i,l_j),\lambda^k_{\Delta_{ij}})} \right) \right) - \log \Phi_L(L) \tag{3.9}$$

As a binary hypothesis is considered for each detection pair, either the denominator within the logarithm will contain $H(l_i,l_j) = H_0$ (if $l_i \neq l_j$), cancelling the pairwise term, or it will contain $H(l_i,l_j) = H_1$ (if $l_i = l_j$). In the end, Equation 3.9 can thus be rewritten as:

$$\underset{L}{\operatorname{argmin}} \left(\sum_{(i,j)\in\mathscr{I}} \sum_{k=1}^{N_f} w^k_{ij} \log \left(\frac{p(f_k(r_i,r_j)|H_0,\lambda^k_{\Delta_{ij}})}{p(f_k(r_i,r_j)|H_1,\lambda^k_{\Delta_{ij}})} \right) \delta(l_i - l_j) \right) - \log \Phi_L(L) \tag{3.10}$$

where $\delta(.)$ denotes the Kronecker function ($\delta(a) = 1$ if $a = 0$, $\delta(a) = 0$ otherwise).

The maximization of Equation 3.3 can then be equivalently conducted by minimizing the following energy:

$$U(L) = \left(\sum_{(i,j)\in\mathscr{I}} \sum_{k=1}^{N_f} w^k_{ij} \beta^k_{ij} \delta(l_i - l_j) \right) + \Lambda(L) \tag{3.11}$$

where the Potts coefficients for each pairwise link and each feature distance are defined as:

$$\beta^k_{ij} = \log \left[\frac{p(f_k(r_i,r_j)|H_0,\lambda^k_{\Delta_{ij}})}{p(f_k(r_i,r_j)|H_1,\lambda^k_{\Delta_{ij}})} \right], \tag{3.12}$$

and the term $\Lambda(L) = -\log \Phi_L(L)$ represents the label cost.

[1] In Equation 3.8, the dissimilarity hypothesis is picked for all detection pairs, as $H(l_i,l_j)$ is set to H_0.

3.3.4 Interpretation of the Potts Coefficients

As can be seen from their definition (Equation 3.12), the Potts coefficients, for each feature, are defined by the log-likelihood ratio of the feature distance of a detection pair under the two hypotheses. Since in the energy of Equation 3.11, the terms for pairs having different labels ($l_i \neq l_j$) vanish and only those for which $l_i = l_j$ remain, the Potts coefficient can be seen as costs for associating a detection pair within the same track.

When $\beta_{ij}^k < 0$, the more *negative* this coefficient will be, the more likely the pair of detections *should* be associated, so as to minimize the energy in Eq. 3.11. Reversely, when $\beta_{ij}^k > 0$, the more *positive* this coefficient will be, the more likely the pair of detections *should not* be associated, so as to minimize the energy in Eq. 3.11. When $\beta_{ij}^k = 0$, there is no preference for associating or not the pairs.

The Potts coefficients can also be seen as the output of a classifier in the Bayes sense. Indeed, the Bayes classifier in our binary hypothesis framework would be formulated, for a given feature index k and a given detection pair (r_i, r_j) as[2]:

$$\begin{cases} p(l_i = l_j | f_k(r_i, r_j)) = \frac{p(f_k(r_i,r_j)|l_i=l_j)p(l_i=l_j)}{p(f_k(r_i,r_j)|l_i=l_j)p(l_i=l_j)+p(f_k(r_i,r_j)|l_i\neq l_j)p(l_i\neq l_j)} \\ p(l_i \neq l_j | f_k(r_i, r_j)) = 1 - p(l_i = l_j | f_k(r_i, r_j)) \end{cases} \tag{3.13}$$

Assuming we have an equal prior that $p(l_i = l_j)$ than $p(l_i \neq l_j)$, we have:

$$p(l_i = l_j | f_k(r_i, r_j)) = \frac{1}{1 + \frac{p(f_k(r_i,r_j)|l_i\neq l_j)}{p(f_k(r_i,r_j)|l_i=l_j)}} = \frac{1}{1 + e^{\beta_{ij}^k}} \tag{3.14}$$

The Potts coefficients are thus implicitly related to the Bayes classifier. The same interpretations as above hold. When β_{ij}^k is very negative, the probability $p(l_i = l_j | f_k(r_i, r_j))$ to assign the same label to the pair tends towards 1, whereas it tends towards 0 when β_{ij}^k becomes very positive. When $\beta_{ij}^k = 0$, we have $p(l_i = l_j | f_k(r_i, r_j)) = p(l_i \neq l_j | f_k(r_i, r_j)) = 0.5$.

3.3.5 Notations

The symbols related to our model, their description and the sections where they are defined are summarized in Table 3.1. In the following sections, the specific features, factor models, confidence scores and label costs will be defined and illustrated.

3.4 Time-Sensitive Pairwise Similarity/Dissimilarity Factors

In practice, we use $N_f = 7$ feature functions constructed around three cues: position, motion and color. Their definitions are provided in the next paragraphs. Note that the focus of this

[2] For simplification of notations, we removed the parameter $\lambda_{\Delta_{ij}}^k$ from the expression.

Section is on the design of the similarity distributions, and that parameter learning will be described later in Chapter 4.

3.4.1 Position Cue Similarity Distributions

The position feature is defined for $k = 1$ as:

$$f_1(r_i, r_j) = \mathbf{x}_i - \mathbf{x}_j \tag{3.15}$$

We assume that its probability follows a Gaussian distribution with 0 mean and whose covariance depends on the two label hypotheses H_0 or H_1 and also *on the time gap* $\Delta_{ij} = |t_i - t_j|$ between the detection pairs:

$$p(f_1(r_i, r_j) = f | H(l_i, l_j) = H, \lambda^1) = \mathcal{N}(f\ ;\ 0, \Sigma^H_{\Delta_{ij}}) \tag{3.16}$$

Figure 3.3 illustrates for two different time intervals the learned models in the form of the Potts coefficient β in function of the distance (dx, dy) between detection pairs. As expected, β is highly negative for distance features close to 0 and increases with the distance. The iso-contours of the β surface are also shown. Amongst them, the zero-contour is a good indicator of the learned model, as it shows the frontier between the domain where hypothesis H_1 prevails and the one where H_0 prevails. Figure 3.3 displays them centered around one detection r_0 for two different values of Δ. After $\Delta = 3$ frames, any detection that falls within the blue contour will vote strongly for the association with r_0 (negative cost). After $\Delta = 15$ frames (around 2 seconds in this case), the model is more relaxed and favors association within the green contour.

3.4.2 Visual Motion Cue Similarity Distributions

The position distance similarity alone does not exploit any directional information and can lead to ambiguities. In order to use an estimation of movement direction at the detection level, we propose to exploit visual motion. In our formalism, the visual motion information is represented by two feature functions f_k with $k \in \{2, 3\}$ defined as follows:

$$f_2(r_i, r_j) = \frac{\mathbf{v}_i . \mathbf{d}_{ij}}{\|\mathbf{v}_i\| \|\mathbf{d}_{ij}\|} \text{ and } f_3(r_i, r_j) = \frac{\mathbf{v}_j . \mathbf{d}_{ij}}{\|\mathbf{v}_j\| \|\mathbf{d}_{ij}\|} \tag{3.17}$$

where $\mathbf{d}_{ij} = \mathbf{x}^{im}_j - \mathbf{x}^{im}_i$ denotes the displacement between the image positions $\mathbf{x}^{im}_i$ and $\mathbf{x}^{im}_j$ of the detections[3] and $\mathbf{v}_i$ or $\mathbf{v}_j$ correspond to their respective visual motion. Given a pair of detections (with $t_j > t_i$), they represent the cosine between their image displacement (as measured by $\mathbf{d}_{ij}$) and the visual motion $\mathbf{v}_i$ or $\mathbf{v}_j$. Intuitively, for detections belonging to the same track, these vectors should be aligned (with a cosine close to 1). The computation of the visual motion vectors will be presented in the experimental Section 5.3.

[3] Note that $\mathbf{x}^{im}$ corresponds to $\mathbf{x}$ when no calibration is available, see Subsection 3.3.1.

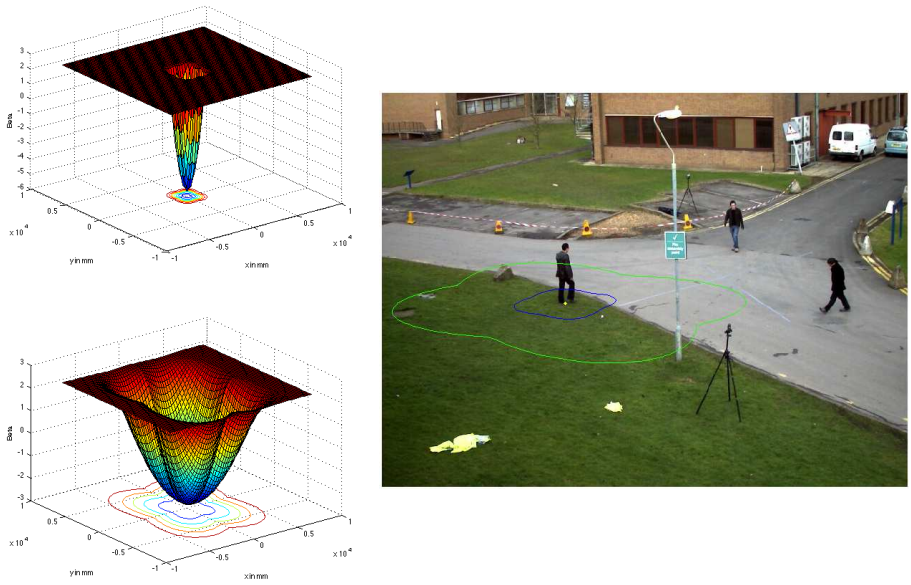


Figure 3.3: Left: the β surface and iso-contours (below) for the position model for $\Delta = 3$, i.e. ~0.45 second (top) and $\Delta = 15$, i.e. ~2 seconds (bottom). Right: the iso-contour of value 0 of the β surface for $\Delta = 3$ (blue) and $\Delta = 15$ (green), centered around one detection.

For the motion feature distribution, we discretize the cosine and use a non-parametric approach by assuming for each time gap Δ and hypothesis H that the features follow a multinomial distribution of parameters $\alpha_{\Delta,H}$:

$$p(f_k(r_i, r_j) = f \mid H(l_i, l_j) = H, \lambda^k) = \alpha_{|t_i - t_j|,H}(b(f)) \tag{3.18}$$

where $b(f)$ denotes the bin index associated with the cosine after quantization. Since f_2 and f_3 play exactly the same role, we use the same model and parameters for both of them.

The intuition is illustrated in Figure 3.4: detections with the same labels are unlikely to fall outside a 2D cone spanned by observed motion vectors. This is confirmed by the β curves automatically learned from data that are shown in Figure 3.5, and which favor association when motion and detection displacements are aligned (cosine near 1) and become more positive as the cosine becomes lower than ≈ 0.5, discouraging association. Interestingly, we see that the model is more discriminative for larger time gaps Δ, when the uncertainty about the displacement (measured from the detected position) is lower.

3.4.3 Color Cue Similarity Distributions

Finally, we propose an appearance similarity measure based on Bhattacharyya distances D_h between color histograms. The pairwise color features are defined for $k \in [4, 7]$ as:

$$f_k(r_i, r_j) = D_h(\mathbf{h}_i^{g(k)}, \mathbf{h}_j^{g(k)}) \tag{3.19}$$

Figure 3.4: Role of the visual motion for tracking. Left: detection r_i at time t_i along with its estimated visual motion $\mathbf{v}_i$ (green ellipse). Right: in subsequent frames, the motion cost associated to this detection at t_i favors associations with other detections located in the direction of motion (shaded area) and penalizes associations in opposite directions (example of blue person, gray ellipse).

where g is a mapping between color feature indices and corresponding body parts:

$$g : k \in [4,5,6,7] \rightarrow g(k) \in [\text{whole}, \text{head}, \text{torso}, \text{legs}] \tag{3.20}$$

Then, the distribution of each feature f_k for a given hypothesis H and time gap Δ is assumed to follow a Gaussian mixture model (GMM) given by:

$$p(f_k(r_i,r_j) = f | H(l_i,l_j) = H, \lambda^k) = \sum_{n=1}^{N_{mix}} \pi^{H,k}_{\Delta_{ij},n} \mathcal{N}(f | \mu^{H,k}_{\Delta_{ij},n}, \sigma^{H,k}_{\Delta_{ij},n}) \tag{3.21}$$

with $\Delta_{ij} = |t_j - t_i|$ and $N_{mix} = 10$ represents the number of mixture components. In practice, the GMM parameters $\lambda^{H,k}_{\Delta} = \{\pi^{H,k}_{\Delta,n}, \mu^{H,k}_{\Delta,n}, \sigma^{H,k}_{\Delta,n}, n \in [1,\ldots,N_{mix}]\}$, i.e. weights, means and variances, are estimated using Expectation-Maximization from appropriate training data (cf. the unsupervised parameter learning Section 4.2 in the next chapter).

Figure 3.6 illustrates the resulting learned β models for different body parts under a time interval Δ of respectively 3 and 15 frames, corresponding to around 0.45s and 2s in this dataset. It can be seen that for small Bhattacharyya distances between detection pairs, the association cost is negative and progressively rises as the distance increases, reaching positive values where it disfavors association. The torso and legs regions exhibit almost no difference in their learned β curves. The head region shows less discrimination, which might be understandable since at the considered resolution, the heads of people contain few distinctive color feature. We can also observe from the figure that color models exhibit time-interval dependencies. Color models for all body parts indeed become less discriminative as Δ increases. Under a higher time interval, even if the Bhattacharyya distance between two detections is slightly larger than between closeby frames, we might still consider their association, though with more uncertainty (flatter β) as people's appearance might change over time, for example due to bounding box localization differences or slight pose variations.

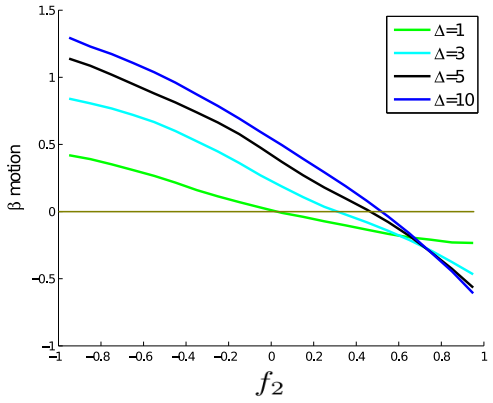


Figure 3.5: Learned β curves for the motion feature on the CAVIAR dataset for different time intervals Δ. When f_2 is close to 1, meaning that the detection displacement is aligned with the visual motion, β is negative, encouraging association. The less these two vectors are aligned (decreasing f_2), the less likely the association, until $f_2 = -1$ (vectors in opposite directions). The $\beta = 0$ iso-line is shown in yellow.

3.5 Pairwise Factor Contextual Weighting

The energy terms defined previously rely on feature distance distributions whose parameters are learned in an unsupervised way as explained in the next chapter. These distributions, however, are global and only reflect the overall feature distance statistics and their discriminative power. To leverage on the local context during test time, we have introduced the weights w_{ij}^k in the definition of our factor terms and of the resulting energy function (3.11). For each feature k and detection pair r_i and r_j, they allow to modulate the previously defined energy terms according to the knowledge of the detection's spatial surroundings.

For instance, when some detection bounding boxes overlap within a frame, the collected color measurements might be corrupted. Hence, we should strongly downvote the color feature contribution of the occluded detections according to the importance of the coverage. Similarly, the visual motion is measured from pixel displacements and such detection overlaps can lead to inaccurate motion estimates that we do not want to rely on for association. By downweighting the contribution of the color and motion features in such cases, we avoid taking into account unreliable features, but can still rely on more accurate measurements done before or after the occlusion and on the position feature to track a partially occluded object. Following the above intuition, the weights have been defined as described below.

Color factor weighting. Let us define the confidence $c(r_i)$ of the visual cues of a detection r_i

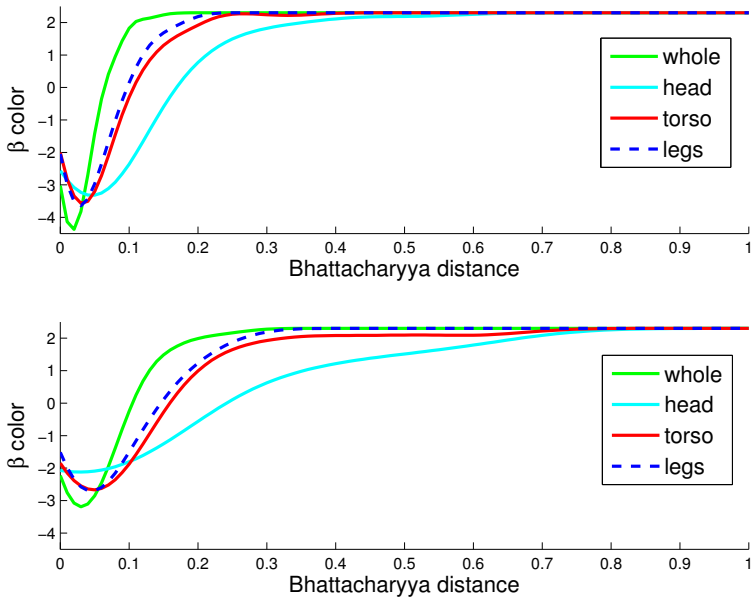


Figure 3.6: Learned β curves for the color feature on the PETS dataset for different body parts subject to a time interval of $\Delta = 3$ frames, i.e. ~0.45 second (top) and $\Delta = 15$ frames, i.e. ~2 seconds (bottom).

based on the overlap with the other detections occurring in the same frame t_i as:

$$c(r_i) = 1 - \min\left(1, \sum_{\substack{r_j \neq r_i \\ t_j = t_i}} \frac{A(r_i \cap r_j)}{A(r_i)}\right) \tag{3.22}$$

where $A(r)$ denotes the area defined by the region associated with the detection r. As can be seen, this confidence is maximum (equal to 1) when the detection does not overlap with any other detection, and decreases in function of the degree of overlap. Accordingly, for each of the color cues ($k = 4,5,6,7$), we simply define the pairwise confidence score as the geometric average of the individual detection confidences, divided by 4 (the number of features for the color cue) to have a normalized confidence score per cue:

$$w_{ij}^k = \frac{\sqrt{c(r_i)c(r_j)}}{4}, \forall k \in \{4,5,6,7\}. \tag{3.23}$$

The evolution of the confidence weight in function of the overlap is illustrated in Figure 3.7.

Motion factor weighting. We use a similar approach for this cue. However, since the reliability of an estimated motion $\mathbf{v}_i$ only depends on the region of the detection r_i it is computed on, we have defined the confidence score for the motion feature implying $\mathbf{v}_i$ ($k = 2$) and $\mathbf{v}_j$ ($k = 3$)

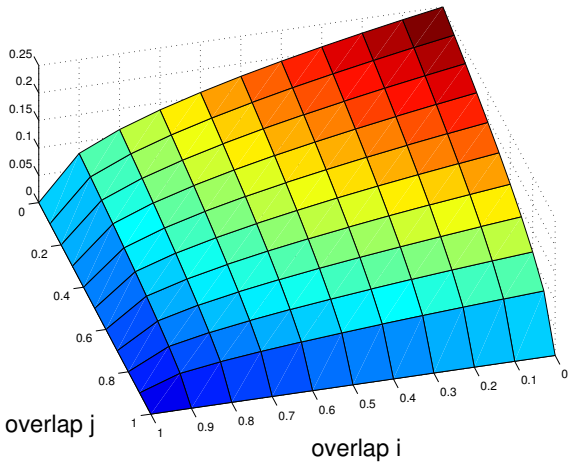


Figure 3.7: Confidence weights w_{ij}^k on the pairwise color feature for each body part in function of the amount of overlap on each detection of the pair. If both detections are not overlapped by any other in their respective frame (overlap=0), the confidence weight on the pairwise color feature is maximal. As soon as one detection of the pair is overlapped, the confidence to rely on the color cue decreases.

as follows:

$$w_{ij}^2 = \frac{c(r_i)}{2} \text{ and } w_{ij}^3 = \frac{c(r_j)}{2}. \tag{3.24}$$

Position factor weighting. Finally, we also introduce a confidence score aiming at downscaling the position energy term for large time intervals. Indeed, as the time difference Δ between two detection increases, the reliability of the position similarity for associating them decreases. This is particularly true in crossing scenarios or when two persons follow each other: in both cases, one of the person's trajectory passes near the other person's previous locations Δ time steps ago, and these small distances tend to vote in favor of association. In order to avoid this effect, we reduce the contribution of the energy term for larger time intervals by defining the confidence score of the position model as:

$$w_{ij}^1 = \frac{1}{1 + e^{|t_i - t_j| - \theta_f}} \tag{3.25}$$

where θ_f denotes the time separation at which the confidence starts to decrease: below θ_f, the confidence is near 1; at θ_f it is equal to 0.5, and beyond it tends to 0 as the time gap $|t_i - t_j|$ increases, as illustrated in Figure 3.8.

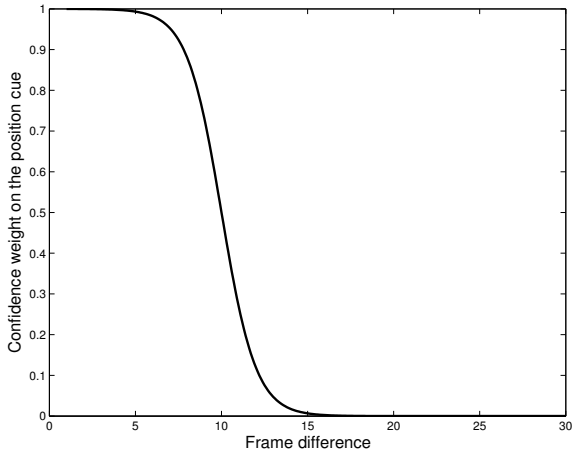


Figure 3.8: Confidence weights on the pairwise position feature in function of the frame difference, with a forgetting factor $\theta_f = 10$ frames. The reliability of the position similarity for associating a pair of detections decreases as the time difference Δ between them increases.

3.6 Label Costs

The energy terms defined earlier concerned detection pairs and did not allow to reason at the global level. The label cost $\Lambda(L)$ we introduced in our energy function of Eq. 3.11 allows to do so by penalizing model complexity. That is, its goal is to avoid having too many labels and obtain coherent tracks from the scene viewpoint. Intuitively, this means that real tracks should start and end near scene entrances/exits (scene boundaries), and that therefore, tracks should be penalized for starting or ending within the scene. Note however, that this is true only for long-enough tracks: short ones, that are less reliable and that are likely to correspond to false alarms should not be penalized.

Before defining the label cost, let us introduce the following notations. For each unique label l, we can define its associated track $\tau_l = \{r_i \ / \ l_i = l\}$ along with its main characteristics: its start time $t_l^s = \min\{t_i \ / \ r_i \in \tau_l\}$, its end time $t_l^e = \max\{t_i \ / \ r_i \in \tau_l\}$, its duration $d_l = t_l^e - t_l^s$, and finally its start and end locations defined by $\mathbf{x}_{t_l^s} = \{\mathbf{x}_i \ / \ r_i \in \tau_l, t_i = t_l^s\}$ and $\mathbf{x}_{t_l^e} = \{\mathbf{x}_i \ / \ r_i \in \tau_l, t_i = t_l^e\}$, respectively.

Then, to achieve the objectives qualitatively stated earlier, we have defined the label cost as follows:

$$\Lambda(L) = \rho \sum_{l \in \mathscr{U}(L)} \left(C^s(\tau_l) + C^e(\tau_l) \right) \tag{3.26}$$

where $\mathscr{U}(L)$ denotes the set of unique labels comprised in the label field L, the parameter ρ controls the importance of the label cost with respect to the pairwise energies, and the start

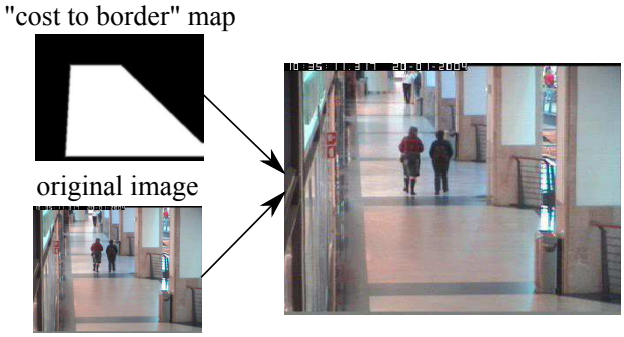


Figure 3.9: Label cost illustration for the CAVIAR data. Long enough tracks starting or ending in the light regions will be penalized. See text for more details.

and ending costs of an individual track are defined as:

$$\begin{aligned} C^s(\tau_l) &= D(d_l)B(\mathbf{x}_{t_l^s})S(t_l^s - t_0;\theta_{tm}) \\ C^e(\tau_l) &= D(d_l)B(\mathbf{x}_{t_l^e})S(t_{end} - t_l^e;\theta_{tm}) \end{aligned} \tag{3.27}$$

where θ_{tm} is a temporal parameter related to the proximity to the start t_0 and end t_{end} of the sequence, and the different terms of this expression that we explain below implement the intuition described earlier.

First of all, the function $B(\mathbf{x}) \in [0,1]$ represents the cost of starting or ending a track at location $\mathbf{x}$, and is illustrated in Fig. 3.9. In practice, we define some scene border regions inside which starting or ending a track has no cost ($B(\mathbf{x}) = 0$) (dark region in Fig. 3.9). On the contrary, tracks that start or end far from these borders have a higher cost ($B(\mathbf{x}) = 1$) (light regions in Fig. 3.9). Smooth transitions between these regions are obtained through filtering. However, since people may already be in the scene at the beginning of the sequence, tracks that start far from the border at this moment should not be penalized. This is achieved thanks to the sigmoid term:

$$S(t_l^s - t_0;\theta_{tm}) = \frac{1}{1 + e^{-\left((t_l^s - t_0) - \theta_{tm}\right)}}$$

which is close to 0 for t_l^s near t_0 and tends to 1 as t_l^s increases. A similar treatment is done for tracks that end by the end of the sequence, since people might still be in the scene at that moment.

Finally, since short tracks that are less reliable might be due to false alarms, they should not be too much penalized to avoid encouraging their association. Thus the overall cost is modulated according to the track duration:

$$D(d_l) = \min(d_l, d_{\max}) \tag{3.28}$$

where $d_{\max}$ is a saturation value beyond which a track is considered long enough to be reliable, and all tracks are penalized in the same way.

3.7 Model Summary and Conclusion

To summarize, in this chapter we have addressed the multi-person tracking problem within the tracking-by-detection paradigm. We have proposed a CRF framework that models pairwise factors as well as additional higher-order potentials defined in terms of label costs. We have shown that our problem could equivalently be expressed as the minimization of an energy function. In our graph, links are created between detection pairs within a long-term interval to take advantage of temporal context. Indeed, this denser graph involving both similarity and dissimilarity pairwise measures reinforces clusters having consistent features over different time intervals, which can help solving temporary ambiguities, like in the case of missing detections.

The focus of this chapter was on the design of several factor potentials in our model, based on position, color and visual motion cues. We have also introduced a set of confidence scores for each feature-based potential and pair of detections that model the reliability of the feature extraction process taking into account occlusions between detections.

For the similarity and dissimilarity functions, we have introduced parametric representations, whose parameters $\lambda = \{\lambda^k\}$ defined for each feature k as $\lambda^k = \{\lambda^k_\Delta, \Delta = 1 \dots T_w\}$, are summarized below:

- $\lambda^1_\Delta = \{\Sigma^{H_0}_\Delta, \Sigma^{H_1}_\Delta\}$ for the position feature (k=1).
- $\lambda^k_\Delta = \{\alpha_{\Delta,H_0}, \alpha_{\Delta,H_1}\}$ for the motion feature (k=2,3).
- $\lambda^k_\Delta = \{\lambda^{H_0,k}_\Delta, \lambda^{H_1,k}_\Delta\}$ for the color feature (k=4,5,6,7).

It is worth emphasizing that each factor is time-interval sensitive, as the parameters depend on the time between the detection pairs.

The remaining tasks include the learning of these parameters, and the optimization of the energy function given by Equation 3.11. These technical points are not trivial and will be investigated in details in Chapter 4.

Section 3.2.	T_w	Long term interval in which pairwise terms are considered.
Section 3.3.1.	r_i	A detection.
	N_r	Number of detections.
	R	Set of detections.
	t_i	Time of occurrence of detection r_i.
	$\mathbf{x}_i$	Position of detection r_i.
	$\mathbf{v}_i$	Image plane visual motion estimate of detection r_i.
	$\mathbf{h}_i^b$	Multi-resolution color histogram extracted from body part $b \in \mathscr{P}$ of detection r_i.
	$\mathscr{P}$	Set of heuristically defined body parts for color extraction.
Section 3.3.2.	l_i	Label of detection r_i.
	L	Label field.
	$\mathscr{I}$	Set of connected detection pairs.
	Φ_k	CRF factor term for feature k.
	N_f	Number of features.
	f_k	Feature function for feature k.
	$\lambda = \{\lambda^k\}$	Factor parameters.
	Δ	Time difference.
	H	Binary hypothesis on label pair (either the same (H_1) or different (H_0)).
	w_{ij}^k	Confidence score associated to detection pair (r_i, r_j) for feature k.
	$\mathscr{W}$	Set of confidence scores for all pairs (i, j).
	Φ_L	Prior over label field in terms of higher-order potentials.
Section 3.3.3.	$U(L)$	Objective energy function to minimize.
	$\Lambda(L)$	Label cost.
	β_{ij}^k	Potts coefficient between detection pair (r_i, r_j) for feature k.
Section 3.4.2.	$\mathbf{d}_{ij}$	Displacement vector between detection pair (r_i, r_j).
	$\mathbf{x}_i^{im}$	Image position of detection r_i.
Section 3.5.	$c(r_i)$	Visual cue confidence of detection r_i based on overlap with detections at the same instant.
	θ_f	Time separation parameter involved in the definition of the confidence weight of the pairwise position factor.
Section 3.6.	τ_l	Track associated to unique label l.
	t_l^s	Start time of track τ_l.
	t_l^e	End time of track τ_l.
	d_l	Duration of track τ_l.
	$\mathbf{x}_{t_l^s}$	Start location of track τ_l.
	$\mathbf{x}_{t_l^e}$	End location of track τ_l.
	$\mathscr{U}(L)$	Set of unique labels.
	ρ	Parameter controlling the importance of the label cost with respect to the pairwise energies.
	$C^s(\tau_l)$	Start cost of track τ_l.
	$C^e(\tau_l)$	End cost of track τ_l.
	t_0	Start time of sequence.
	t_{end}	End time of sequence.
	θ_{tm}	Temporal parameter related to the proximity to the start t_0 and end t_{end} of the sequence.
	$B(\mathbf{x})$	Cost of starting or ending a track at location $\mathbf{x}$.
	$D(d_l)$	Cost on track duration d_l.
	$d_{\max}$	Saturation value for duration beyond which all tracks are penalized in the same way.

Table 3.1: Model notations.

4 Unsupervised Parameter Learning and CRF Optimization

4.1 Introduction

In the preceding chapter, we described our CRF for multi-person tracking and defined several types of factor terms within the model. The appropriate setting of the model parameters is of crucial importance for achieving good tracking results, but can be a tedious task. We remind that since distributions exhibit time dependencies, parameters need to be defined for each feature and each time interval up to T_w. Moreover, parameters also depend on the two-fold hypothesis H, so that ultimately, we have a large parameter space size. Therefore, manual setting of these parameters is not an option.

Several works in the literature discuss how to estimate the parameters of a CRF [Sutton and McCallum, 2012]. Many of these methods are supervised. Given labeled data, one way to perform the learning is to select the parameters such that the training data has the highest likelihood under the model. However, for CRFs with complex structures, exact maximum likelihood training is intractable. In that case, approximate procedures like Belief Propagation or MCMC can be used. CRFs have been used in the tracking literature, notably in [Yang et al., 2011], where the authors present an offline, supervised approach for learning. Similarly to us, an energy function is defined, though over a graph of tracklets. Given some labeled tracks, they first create alternative labelings by randomly introducing labeling errors. Then, they formulate the training task as finding the best energy function so that the preferred global associations (labelings) have lower energy costs.

In practice, however, one would like to avoid supervised learning, as this would require tedious track labeling for each scene or camera. In Section 4.2, we propose an approach for learning the factor parameter set in an unsupervised fashion, meaning it only requires detections without ground truth labels as input. As pre-learned global affinity models usually do not work well under all scenarios (e.g. due to differences in resolution, viewpoint or illumination), we propose to learn scene-specific parameters from training videos coming from the same scene as the test data.

At test time, given the learned parameters, the goal is to infer the label field L that maximizes the CRF posterior $p(L|R)$, or equivalently that minimizes our energy function $U(L)$. Standard inference algorithms like forward-backward or the Viterbi algorithm can be applied to simple types of CRFs, like linear-chain CRFs, which are discriminative analogues of HMMs [Sutton and McCallum, 2012]. For more complex CRFs, approximate inference procedures like Monte Carlo or variational algorithms have been studied.

Often in multi-object tracking, researchers resort to approximate and iterative procedures for optimization. For instance, in [Yang and Nevatia, 2012a], although the energy formulation of their CRF model only contains unary and pairwise terms, they showed that their energy function is not submodular and hence cannot be solved using standard optimization techniques like graph cuts. Therefore, they propose a heuristic Iterated Conditional Modes (ICM) optimization algorithm which makes changes to the labeling by switching labels, and keeps the changes if they decrease the energy. Although not in a CRF framework, the authors of [Andriyenko et al., 2012] also formulate multi-person tracking as the minimization of an energy containing unary, pairwise, and global costs. To tackle the issue of a challenging optimization with global costs, they propose an iterative, approximate algorithm that temporarily disregards label costs. Heuristic approaches like the ones described above do not guarantee optimality, but can provide good solutions in reasonable time.

In the following, we show that our objective function does not follow the submodularity principle and hence cannot be solved using standard optimization techniques (see Section 4.3.1). Therefore, we propose two heuristic algorithms that act as an iterative approximate solution to find a good labeling. In Section 4.3, we propose to minimize the energy objective with an online optimization algorithm, Sliding Window (SW) that works at the detection level (see Section 4.3.2); and a block-level algorithm, Block ICM, which considers label costs and can possibly correct wrong labels from the SW stage (see Section 4.3.3).

The symbols related to our learning and optimization frameworks, their description and the sections where they are defined are summarized in Table 4.1.

4.2 Unsupervised Parameter Learning

4.2.1 Learning Overview

The overall procedure of unsupervised learning and tracking is summarized in the block diagram of Figure 4.1. As shown in this figure, the first step is to learn model parameters by relying directly on the raw detections within training videos of a given scene. The procedure of learning from detections is explained in Section 4.2.2. For convenience, we denote with a $\star$ superscript the notations that apply to these initial models (for instance, these models are learned up to $T_w^\star$). These models can be used for tracking on the training videos, and, provided we use a low $T_w^\star$ value, can lead to pure tracklets, as shown in the experiments of Chapter 5.

Section 4.2.2.	$(.)^\star$	Superscript referring to initial models learned from detections.
	$\mathscr{C}^\star$	Set of closest detection pairs.
	$\mathscr{S}^\star$	Set of second closest detection pairs.
Section 4.2.3.	$\mathscr{C}$	Set of detections collected within tracklets.
	$\mathscr{S}$	Set of detections collected between tracklets.
Section 4.3.1.	U^{ij}_{pair}	Pairwise energy term.
Section 4.3.2.	$\mathbf{A}^{SW}$	Association matrix for Sliding Window optimization.
	N_t	Number of detections in the current frame.
	D	Set of indices of detections within the current frame.
	L^B_a	Set of active labels that have been assigned in the past T_w instants.
	N^B	Number of active labels in the past.
	$B^B_{\gamma(m)}$	Set of indices of past detections with active label $\gamma(m)$.
Section 4.3.3.	$\mathbf{A}^{BI}$	Association matrix for Block ICM optimization.
	L^A_a	Set of active labels that have been assigned in the future T_w instants, including time t.
	N^A	Number of active labels in the future.
	$B^A_{\zeta(n)}$	Set of indices of future detections with active label $\zeta(n)$.

Table 4.1: Learning and optimization notations.

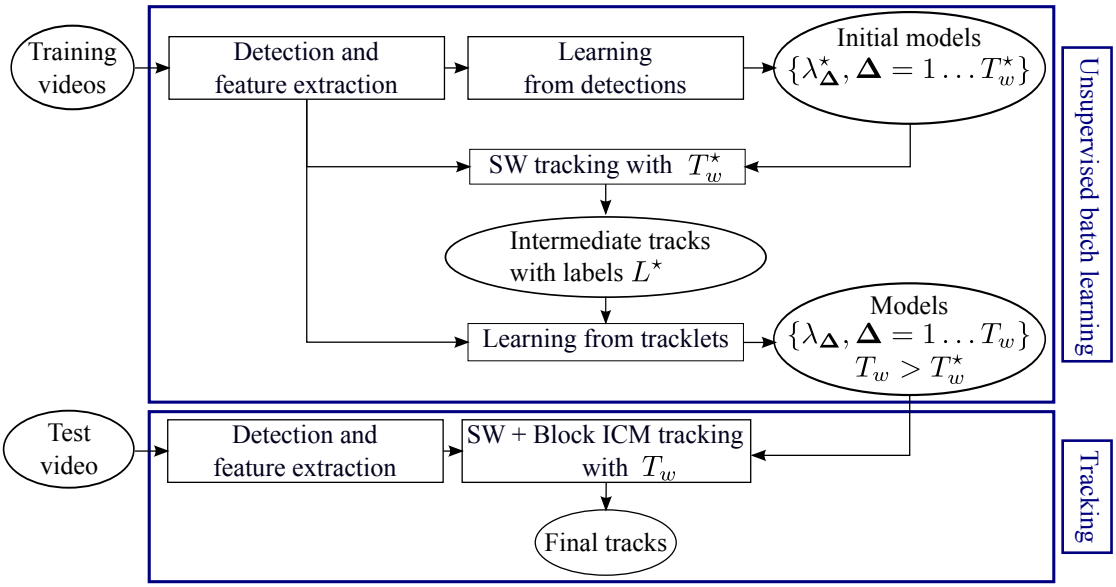


Figure 4.1: Flowchart of the unsupervised batch learning and subsequent tracking procedure. Detections and features are extracted on scene-specific training videos. Initial models up to $T^\star_w$ are learned from detections (Section 4.2.2). Tracking is performed with these models to obtain an intermediate labelling $L^\star$, which is in turn used to relearn more accurate models up to $T_w > T^\star_w$ (Section 4.2.3). Finally, given detections and features of a test video, these refined models for the scene are used to perform tracking. Sliding Window (SW) and Block ICM are two optimization steps that are explained in Sections 4.3.2 and 4.3.3.

Thus, in a second step, these tracklets corresponding to an intermediate labelling $L^\star$ can be conveniently used to refine model parameters and learn parameters for larger T_w values, as explained in Section 4.2.3. The two learning steps are illustrated in the top part of Figure 4.1, and described below.

4.2.2 Learning from Detections

Learning the model parameters λ can be done in a fully unsupervised way using a sequence of detection outputs. When no labels are provided, the intuition for learning consists of collecting training data as follows: for a given detection at time t, the closest detection amongst the detections at time $t+\Delta$ should statistically correspond to a detection of the same track, while the second closest detection[1] would correspond to a different person. Thus, for each time gap Δ, we collect for each detection its closest and second closest detection Δ frames away and construct the set of closest $\mathscr{C}_\Delta^\star$ and second closest $\mathscr{S}_\Delta^\star$ detection pairs. This procedure is summarized in Algorithm 1. These sets can then be used to learn model parameters under each model hypothesis for each feature and time interval. More precisely:

- For the position model, the means of the 2D Gaussian distributions are constrained to zero and the covariances $\lambda_\Delta^{\star 1} = \{\Sigma_\Delta^{\star H_0}, \Sigma_\Delta^{\star H_1}\}$ are learned as a 2-component GMM from features of $\mathscr{C}_\Delta^\star \cup \mathscr{S}_\Delta^\star$ through EM. From the resulting covariances, the smallest one (as measured by the determinant magnitude) is taken as the covariance $\Sigma_\Delta^{\star H_1}$ and the largest one as $\Sigma_\Delta^{\star H_0}$. Hence, the 2D Gaussian for hypothesis H_1 is much peakier than the one representing H_0, meaning that a pair of detections (r_i, r_j) within a close distance will be more likely under $H_1 (l_i = l_j)$ than under $H_0 (l_i \neq l_j)$.
- Since we used a non-parametric model for the motion, using a mixture model like above is impossible. Thus, in a more direct way, the multinomials with parameters $\lambda_\Delta^{\star 2,3} = \{\alpha_{\Delta,H_0}^\star, \alpha_{\Delta,H_1}^\star\}$ are obtained as the quantized distributions of the features, extracted separately from $\mathscr{S}_\Delta^\star$ and $\mathscr{C}_\Delta^\star$, respectively, then smoothed by moving average to account for the limited amount of training data.
- To estimate the color model parameters $\lambda_\Delta^{\star 4,5,6,7} = \{\lambda_\Delta^{\star H_0,4,5,6,7}, \lambda_\Delta^{\star H_1,4,5,6,7}\}$, we fit a GMM with $N_{mix} = 10$ mixture components separately to the set of Bhattacharyya distances computed from the pairs in $\mathscr{S}_\Delta^\star$ (for H_0) and $\mathscr{C}_\Delta^\star$ (for H_1).

4.2.3 Learning from Intermediate Tracking Results

The assumption that parameters can be learned from the closest and second closest detections holds reasonably well for small values of Δ or low crowding, but might not be verified for larger temporal gaps. However, we show in the experimental Chapter 5 that pure tracklets with few identity switches can be obtained by our tracking framework with a relatively small T_w when using models learned as above. We thus propose to use these intermediate tracklets to collect

[1] In principle, all non-closest detections would correspond to different persons. However, we used the second closest detection to obtain more discriminative models, especially for the position feature.

Algorithm 1 Collection of $\mathscr{C}_\Delta^\star$ and $\mathscr{S}_\Delta^\star$ from detections.

for $\Delta = 1$ to $T_w^\star$ **do**
  Initialize empty sets $\mathscr{C}_\Delta^\star$ and $\mathscr{S}_\Delta^\star$
  for $i = 1$ to N_r **do**
    $j = \arg\min_k |\mathbf{x}_i - \mathbf{x}_k|,\ s.t.\ |t_k - t_i| = \Delta$
    $m = \arg\min_k |\mathbf{x}_i - \mathbf{x}_k|,\ s.t.\ t_k = t_j$ and $k \neq j$
    Add pair (r_i, r_j) to set $\mathscr{C}^\star{}_\Delta$ and pair (r_i, r_m) to set $\mathscr{S}^\star{}_\Delta$
  end for
end for

Algorithm 2 Collection of $\mathscr{C}_\Delta$ and $\mathscr{S}_\Delta$ from intermediate labelling $L^\star$.

for $\Delta = 1$ to T_w **do**
  Initialize empty sets $\mathscr{C}_\Delta$ and $\mathscr{S}_\Delta$
  for each unique label $l \in L^\star$ **do**
    for $(r_i, r_j) \in \tau_l$ with $|t_i - t_j| = \Delta$ **do**
      Add pair (r_i, r_j) to set $\mathscr{C}_\Delta$
    end for
  end for
  for each pair of unique labels (l, l') with $l \neq l'$ **do**
    for $r_m \in \tau_l$ and $r_n \in \tau_{l'}$ with $|t_m - t_n| = \Delta$ **do**
      Add pair (r_m, r_n) to set $\mathscr{S}_\Delta$
    end for
  end for
end for

more reliable data for each hypothesis and learn more discriminative model parameters, up to a higher value of T_w.

This is illustrated in Figures 4.2 and 4.3 for the torso color model, learned from raw detections and tracklets, respectively. For the models learned from detections shown in Figure 4.2, we can observe that for small time gaps ($\Delta = 1$) the Bhattacharyya distance distributions are well separated under the two hypotheses. However, as T_w increases (e.g. for $\Delta = 15$), the collected feature sets $\mathscr{C}_\Delta^\star$ and $\mathscr{S}_\Delta^\star$ from the detections do not correspond to the assumption any more and become more blended w.r.t. the H_1 or H_0 hypothesis, resulting in non-discriminant parameter estimates. Instead, we propose to collect new sets $\mathscr{C}_\Delta$ and $\mathscr{S}_\Delta$ of detection pairs for learning, using the intermediate track information, i.e. the current labelling $L^\star$. The procedure of collecting these sets from tracklets is summarized in Algorithm 2[2]. Using the two collected sets, the learning is done similarly as in Section 4.2.2, except that now, for the position model, a 2D Gaussian is fitted separately on $\mathscr{S}_\Delta$ and $\mathscr{C}_\Delta$.

When using the tracking results obtained with $T_w^\star = 8$ (and model parameters learned from the raw detections) to collect training data, we obtain more accurate and sensible (and still discriminative) distributions, especially for large values of T_w, as can be seen in Figure 4.3 compared to the distributions learned from detections, in Figure 4.2. We also illustrate the

[2]Note that here, in order to build $\mathscr{S}_\Delta$, we consider for each detection belonging to a given tracklet all the detections Δ frames apart and coming from any other tracklet, not just from the closest one.

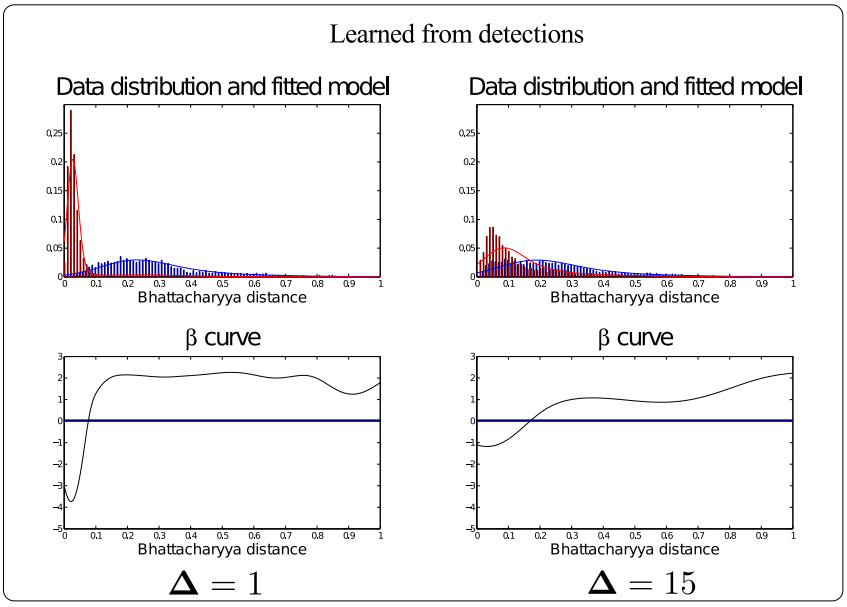


Figure 4.2: Parameters learned from detections for the color potential of the torso. Top: pairwise color feature (Bhattacharyya distance) distribution fitted on PETS data under the H_1 hypothesis, i.e. labels are supposed to be the same (red curve), and H_0 (blue curve), for two different values of Δ, relying on training sets collected from raw detections. Bottom: corresponding β curves of color model. For small time gaps ($\Delta = 1$) the Bhattacharyya distance distributions are well separated under the two hypotheses. For larger time gaps ($\Delta = 15$), the training feature sets become more blended w.r.t. the H_1 or H_0 hypothesis, resulting in non-discriminant parameter estimates.

distributions learned from detections as opposed to the ones learned from tracklets for the position and motion features, in Figures 4.4 and 4.5, respectively. Note that the method is unsupervised and the relearned models are still global (i.e. not specific to any track or detection).

Robust estimates. The above approach assumes that we obtain representative training sets for both hypotheses. While this might be true for the dissimilar hypothesis H_0, we actually miss large measurements for the similar case H_1, since tracks might actually be broken (fragmented) at places with high feature distances, and lead to an overconfident model for H_1. We alleviated this issue as follows. Let us denote by $\hat{p}(f_k|H_h,\lambda_\Delta^k)$ the feature distributions learned using the training sets collected as above. Then, we defined:

$$\begin{cases} p(f_k|H_1,\lambda_\Delta^k) = 0.9\,\hat{p}(f_k|H_1,\lambda_\Delta^k) + 0.1\,\hat{p}(f_k|H_0,\lambda_\Delta^k) \\ p(f_k|H_0,\lambda_\Delta^k) = \hat{p}(f_k|H_0,\lambda_\Delta^k) \end{cases} \tag{4.1}$$

as actual feature distributions in the tracking framework. Intuitively, the above heuristic implicitly assumes that some measurements in the H_0 training set are actually coming from

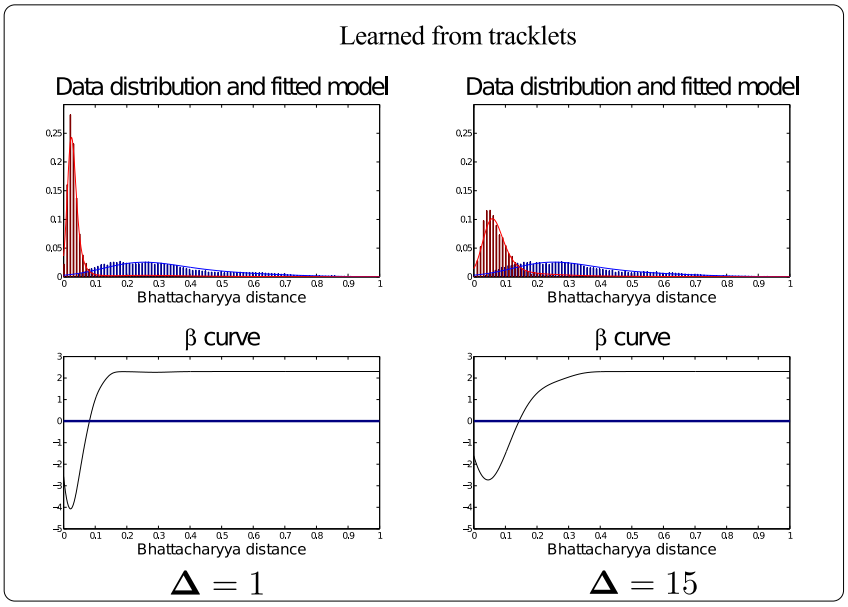


Figure 4.3: Parameters learned from tracklets for the color potential of the torso. Top: pairwise color feature (Bhattacharyya distance) distribution fitted on PETS data under the H_1 hypothesis, i.e. labels are supposed to be the same (red curve), and H_0 (blue curve), for two different values of Δ, relying on training sets collected from tracklets. Bottom: corresponding β curves of color model. When using intermediate tracking results to collect training data, we obtain accurate and sensible distributions.

the same person tracks and thus should be incorporated in the H_1 distribution. In practice it leads to the saturation effect shown on β curves.

4.3 CRF Optimization

We formulated multi-person tracking as finding the label field $L = \{l_i\}_{i=1:N_r}$ minimizing the energy function presented in Eq. 3.11[3], and which we recall below:

$$U(L) = \sum_{(i,j)} \underbrace{\sum_{k=1}^{N_f} w_{ij}^k \beta_{ij}^k \delta(l_i - l_j)}_{U_{pair}^{ij}(l_i,l_j)} + \Lambda(L) \tag{4.2}$$

Many problems in computer vision can be expressed in terms of an energy minimization. However, the optimization remains a challenging problem when the form of the energy is too complex and many researchers resort to heuristic methods for optimization. In the following, we first demonstrate why our formulation does not meet a necessary condition to

[3]Remember that we want to infer the label l_i of each detection r_i and that labels can take their values in $\mathbb{N}$ as we do not know in advance the number of persons in the scene.

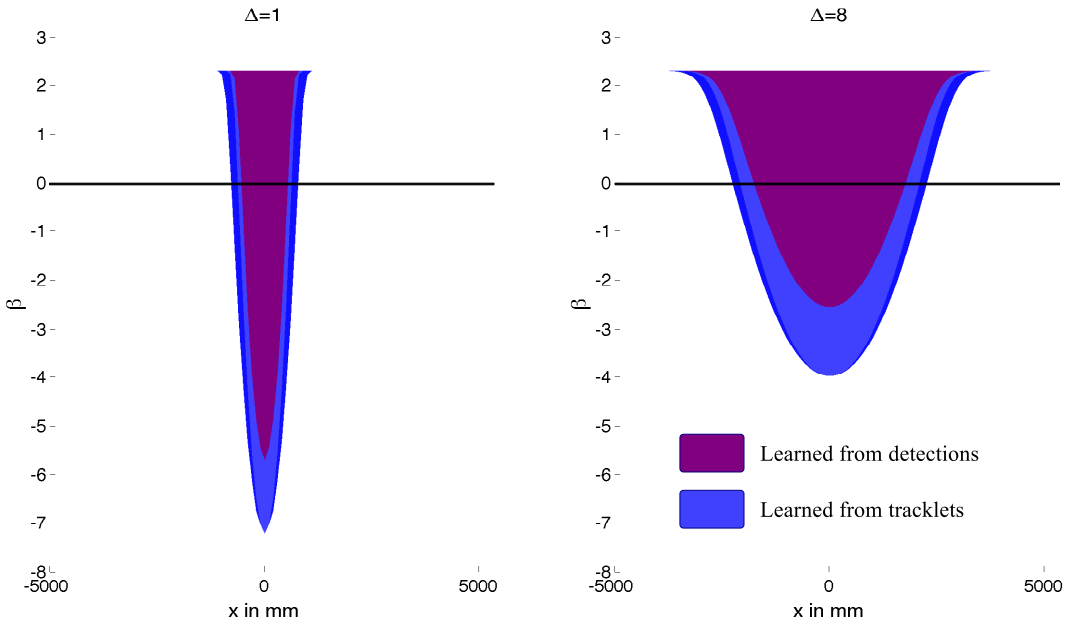


Figure 4.4: Unsupervised parameter learning for the position potential on PETS data. Cut-away view of the β surface for $\Delta = 1$ (left), for $\Delta = 8$ (right). Models are either learned from detections (purple) or from tracklets (blue). Crossings with the zero iso-surface occur around the same values of x, but models learned from tracklets are more discriminative, especially for larger values of Δ.

be minimized optimally by graph cuts. Then, we present an iterative approximate solution adapted to our problem.

4.3.1 Submodularity

Graph cut techniques construct a graph such that the minimum cut on the graph also minimizes the energy. These techniques are convenient because they guarantee some theoretical quality, like optimality or near optimality of the solution. However, graph cuts cannot be applied to any kind of energy. The authors of [Kolmogorov and Zabih, 2004] provide a necessary condition for any function of binary variables to be minimized via graph cuts, which can be generalized for a larger number of labels. In particular, for functions from the class $\mathcal{F}^2$ they demonstrate the following theorem[4]:

Theorem. *Let E be a function of n binary variables from the class $\mathcal{F}^2$, i.e.,*

$$E(x_1,\ldots,x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{ij}(x_i,x_j) \tag{4.3}$$

[4] Functions from the class $\mathcal{F}^2$ can be written as a sum of functions of up to two binary variables.

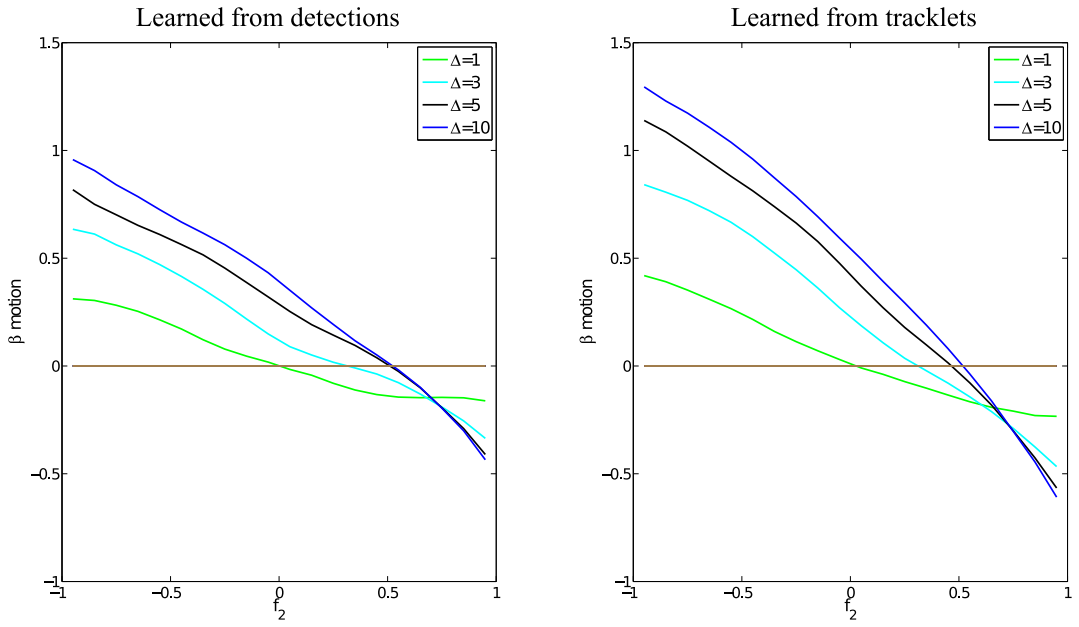


Figure 4.5: Unsupervised parameter learning for the motion potential on CAVIAR data for different values of Δ: β curves of models learned from detections (left) and from tracklets (right). Crossings with the zero iso-line occur around the same values of f_2, but models learned from tracklets are more discriminative, especially for larger values of Δ.

Then, E is graph-representable if and only if each term E^{ij} satisfies the inequality

$$E^{ij}(0,0) + E^{ij}(1,1) \leq E^{ij}(0,1) + E^{ij}(1,0) \tag{4.4}$$

In other words, the *regularity* or *submodularity* defined by Equation 4.4 is a necessary and sufficient condition for an energy function E to be solved by graph cuts. Our energy (cf. Equation 4.2) is decomposed into two components: the sum of feature-specific pairwise terms (Potts coefficients) weighted by their confidence, and higher-order cost terms (label costs). If we forget about the label costs $\Lambda(L)$, our energy function is from the class $\mathscr{F}^2$, with:

$$\begin{cases} E^i(l_i) = 0, \ \forall i \in \{1 : N_r\} \\ E^{ij}(l_i, l_j) = U^{ij}_{pair}(l_i, l_j), \ \forall (i,j) \in \{1 : N_r\}^2 \end{cases} \tag{4.5}$$

Assuming that we restrict our labeling problem to a binary one (i.e. we would look for 2 tracks only), submodularity imposes that the pairwise term U_{pair} in Equation 4.2 must satisfy for each term:

$$U_{pair}(l_i = 1, l_j = 0) + U_{pair}(l_i = 0, l_j = 1) \geq U_{pair}(l_i = 1, l_j = 1) + U_{pair}(l_i = 0, l_j = 0) \tag{4.6}$$

The left term of Eq. 4.6 is zero because the Kronecker function cancels the pairwise energy term when labels are different (cf. definition of U^{ij}_{pair} in Equation 4.2). The right term is equal

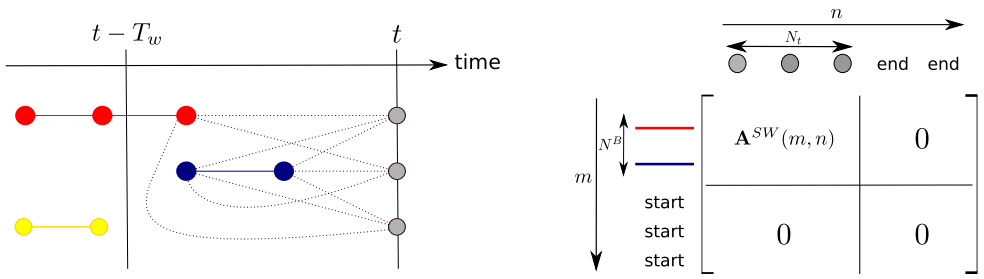


Figure 4.6: Sliding Window at time t: Grey nodes are the current detections. Red and blue nodes are the detections associated with the active labels. (Left): Dotted lines show the pairwise links used to compute the assignment costs for SW. (Right): Corresponding assignment matrix, showing the cost to assign labels to current detections. If a detection is associated to *start*, a new label is created for that detection. There is one *start* possibility for each detection so that each one of them can have a new label. Similarly, there is one *end* possibility for each active label so that each track can potentially be discontinued in the current frame.

to $2\sum_{k=1}^{N_f} w_{ij}^k \beta_{ij}^k$. In the end, we cannot guarantee that $0 \geq 2\sum_{k=1}^{N_f} w_{ij}^k \beta_{ij}^k$ for each pairwise term because the Potts coefficients can be negative or positive. Subtracting a constant to the Potts coefficients so as to make them all negative does not produce an equivalent optimization energy. Therefore, in our framework, even when simplified to its pairwise energies only, submodularity is not verified and exact graph cuts cannot be applied.

4.3.2 Sliding Window Optimization

As graph cut techniques cannot be applied, we introduce an iterative approximate algorithm to find a good labeling solution. More precisely, we start the labeling process by applying a Sliding Window approach. Then, in a second step we perform a more global block Iterated Conditional Modes (ICM) optimization. The Sliding Window (SW) technique is an online, iterative labeling method. At every time instant t, the labeling of the current detections is done leveraging on the links with past detections. The idea is the following: given a past window of already labeled detections, we wish to locally optimize the labels in the current frame. For that purpose, we try different labelings for the current detections, by picking labels that have appeared before in the window, as well as new labels to account for the possible apparition of a new person in the scene or a false alarm. The most likely combination of labels in terms of local pairwise energies in the window is retained, and we impose the constraint that the same label cannot appear more than once in the frame. We describe the SW algorithm more formally below.

Let us define the set of N_t current detection indices as $D = \{i \mid t_i = t\}$. The set of N^B *active* labels that have already been assigned in the past T_w instants is given by $L_a^B = \{l : \exists i \mid l_i = l \text{ and } t - T_w \leq t_i < t\}$. The left part of Figure 4.6 illustrates how the T_w window is used to define active labels. The dotted lines show the links that will be used to compute the assignment costs for SW.

Each of the current detections can potentially take a label among the N^B active labels. We also allow the possibility for each of these detections to take a new label and thus start a new track. In other words, each new detection can either extend an existing track, or start a new track, while existing tracks are either extended or not. Therefore, we formulate SW at time t as the assignment between labels (including new ones) and current detections. We also wish to allow active labels not to be assigned in the current frame. Note here that a track not being extended at current time instant t does not mean that it is ended for good, as it could be reassigned in future frames due to the long-term connectivity between detections.

For convenience, let us define the two following mappings γ and ν:

$$\begin{cases} \gamma : m \in [1 : N^B] \rightarrow \gamma(m) \in L_a^B \\ \nu : n \in [1 : N_t] \rightarrow \nu(n) \in D \end{cases} \tag{4.7}$$

where γ maps active label indices to the actual label values and ν maps indices in the current frame to the corresponding detection indices.

The assignment between active labels and the current detections is represented by the association matrix $\mathbf{A}^{SW} \in \mathbb{R}^{(N^B+N_t)\times(N^B+N_t)}$, which is illustrated in the right part of Figure 4.6. For each index $m \in [1 : N^B]$, we denote by $B^B_{\gamma(m)} = \{i \mid t - T_w \leq t_i < t \text{ and } l_i = \gamma(m)\}$ the set of indices of past detections that have been previously assigned with active label $\gamma(m)$. The association matrix $\mathbf{A}^{SW}$ is then defined as:

$$\mathbf{A}^{SW}(m,n) = \begin{cases} \sum\limits_{i \in B^B_{\gamma(m)}} \sum\limits_{k=1}^{N_f} w_{ij}^k \beta_{ij}^k & \text{if } (m \leq N^B, n \leq N_t), \text{ with } j = \nu(n) \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

This formulation basically optimizes the energy locally within the sliding window, without taking into account label costs, since we do not want to penalize ending old tracks or starting new ones in order to avoid initial identity switches. The local energy is only affected by the labeling of the current detections, the labels of past detections in the window being fixed. Therefore, the only terms affecting the assignment are the pairwise energies shown in the left part of Figure 4.6. The elements of $\mathbf{A}^{SW}$ defined in Equation 4.8 account for those energy terms that are affected by the assignment. The cost to start any new label or to not continue any track is set to zero, so that it does not favor, nor disfavor such a situation.

The optimal assignment is performed with the Hungarian algorithm [Burkard et al., 2009], the labels are assigned for frame t, then the sliding window is shifted by one frame and the procedure is repeated for the detections in frame $t+1$ and so on, until the end of the sequence is reached. The constraint that the same label cannot be assigned more than once in the frame is implicitly handled by the assignment algorithm. As will be shown in the experiments, this SW

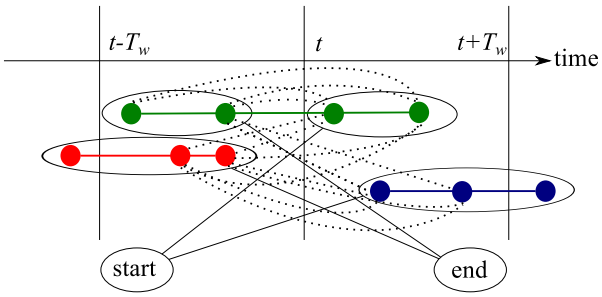


Figure 4.7: Block ICM at time t. Current tracks before and after t are associated so as to minimize block-wise β costs between pairs (dotted lines) and label costs related to the start and ending of tracks.

optimization already leads to very good results given the use of the long-term connectivities, and generally produces pure but potentially fragmented tracks, depending on the value of T_w.

4.3.3 Block ICM Optimization

One limitation of SW is that it performs optimization locally and taking into account only past information. For Block ICM, the granularity is different as we add the constraint that all labels inside blocks (defined as the nodes with the same label in the current results within a window of duration $2T_w$) should be changed (or not) simultaneously. The procedure is similar to [Collins, 2012] and our approach is designed to deal with our higher-order label costs. In this step, optimization is thus conducted at a more global level. Similarly to the SW approach, we can define the set L_a^B of N^B active labels that exist within T_w instants *before* t and the set L_a^A of N^A active labels that exist within T_w instants *after* t. An association matrix $\mathbf{A}^{BI} \in \mathbb{R}^{(N^B+N^A)\times(N^B+N^A)}$ is built such that it comprises all terms that depend on the assignment: the pairwise terms, which involve only links within a temporal neighborhood of T_w around t (hence the Block ICM terminology), and the global start and end label cost terms. The labeling can be seen as an assignment problem between past and future tracks. In this sense, SW was a particular case of Block ICM where the future blocks were constrained to be single detections within the optimized instant t, and which did not take label costs into account. With Block ICM, past tracks can be extended or stopped, and future tracks can extend a track or start a new one.

For convenience, let us define an additional mapping ζ:

$$\zeta : n \in [1 : N^A] \rightarrow \zeta(n) \in L_a^A \tag{4.9}$$

which maps indices of active labels in the future to the actual label values.

Like before, for each index $m \in [1 : N^B]$, we denote by $B^B_{\gamma(m)} = \{i \mid t - T_w \leq t_i < t \text{ and } l_i = \gamma(m)\}$ the set of indices of past detections that have been previously assigned with active label $\gamma(m)$. Similarly, for each index $n \in [1 : N^A]$, we denote by $B^A_{\zeta(n)} = \{j \mid t \leq t_j < t + T_w \text{ and } l_j = \zeta(n)\}$ the set of indices of future detections (beyond t) with active label $\zeta(n)$.

Finally, for each label $l \in L_a^B \cup L_a^A$, we introduce the notation of tracks *before* and *after* t, respectively, such that $\tau_l = \tau_l^B \cup \tau_l^A$:

$$\begin{cases} \tau_l^B = \{r_i \mid l_i = l \text{ and } t_i < t\} \\ \tau_l^A = \{r_j \mid l_j = l \text{ and } t_j \geq t\} \end{cases} \tag{4.10}$$

Note that these tracks are not limited to the blocks but are defined as complete trajectories, to which label costs can be applied.

Then, the assignment matrix $\mathbf{A}^{BI}$ is defined as follows:

$$\mathbf{A}^{BI}(m,n) = \begin{cases} \left(\sum\limits_{\substack{i \in B^B_{\gamma(m)} \\ j \in B^A_{\zeta(n)}}} \sum\limits_{k=1}^{N_f} w_{ij}^k \beta_{ij}^k \right) + \\ \rho \times (C^s(\tau^B_{\gamma(m)} \cup \tau^A_{\zeta(n)}) + C^e(\tau^B_{\gamma(m)} \cup \tau^A_{\zeta(n)})) & \text{if } (m \leq N^B, n \leq N^A)^5 \\ \rho \times (C^s(\tau^B_{\gamma(m)}) + C^e(\tau^B_{\gamma(m)})) & \text{if } (m \leq N^B, n > N^A)^6 \\ \rho \times (C^s(\tau^A_{\zeta(n)}) + C^e(\tau^A_{\zeta(n)})) & \text{if } (m > N^B, n \leq N^A)^7 \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

In essence, the upper left $N^B \times N^A$ submatrix gathers costs for connecting tracks (i.e. blocks of detections) existing before and after t. The upper right $N^B \times N^B$ submatrix represents the costs of ending tracks existing before t, whereas the lower left $N^A \times N^A$ submatrix represents the costs of starting tracks after t.

The optimal assignment is solved with the Hungarian algorithm, the labels are updated and the procedure is then applied at time $t+1$ and so on, until the end of the sequence is reached.

Several sweeps through the sequence are possible to refine the labeling. One limitation of Block ICM is that it does not perform online optimization, on the contrary to SW. When exploited in an online system processing incoming video streams, the above strategy could be adapted to address this issue. For instance, SW could be applied at every frame (using a sliding window size T_w of typically a few seconds), while Block ICM could be invoked from time to time to correct SW labeling within a larger sliding window (typically about 10 seconds). In that case, t_0 of Section 3.6 would refer to the start of this larger sliding window used by Block ICM[9], while t_{end} would be the end of this larger window, i.e. would correspond to the latest available frame of the video stream.

[6] Connecting tracks.
[7] Ending tracks.
[8] Starting tracks.
[9] Or the corresponding effect could be neglected since the start of the video is far in the past.

4.4 Conclusion

To summarize, we proposed a way to learn scene-specific model parameters in an unsupervised and incremental fashion, starting from detections, then using intermediate tracking results:

- We proposed a criterion to first collect relevant detection pairs to measure their similarity/dissimilarity statistics and learn model parameters that are sensitive to the time interval between detection pairs.
- Then, at a successive optimization round, we can leverage on intermediate track information to gather more reliable statistics and exploit them to estimate accurate model parameters.

In this way, model parameters are automatically adapted to a scene, which promotes the generalization capability of our tracker. Note that besides scene adaptation, local spatio-temporal context can also be useful to improve the association. In our framework, this is implicitly handled by the confidence weighting presented in Section 3.5. Even though contextual information is not directly used to act on the model parameters, it still influences the data association by weighting the pairwise factors, leading to a modulation of the energy terms according to the knowledge of each detection's spatio-temporal surroundings.

We then showed that our energy function was not submodular and could not be solved by standard optimization procedures. Therefore, we introduced an iterative approximate algorithm to find a good labeling solution by first applying an online optimization, then a block-level algorithm:

- First, SW is applied. It is an online procedure that labels the detections of the current frame given a set of previously labeled tracks within a sliding window, and therefore does not correct the labels of other detections within the sliding window.
- Then, in a second step, Block ICM is applied. This algorithm considers the whole sequence and makes successive block-optimal two-frame assignments. Block ICM is able to correct mistakes done at the SW level, due to its use of label costs and of both past and future observations at a given frame.

Note that both optimization steps operate at the detection level, even though Block ICM is able to update the labels of several detections simultaneously. We also tried to apply a simple ICM procedure at the node level just after SW. However, we observed that this step only brought marginal changes to the labeling and did not contribute to improve the tracking performance. This observation is in accordance with the results of [Lathoud and Odobez, 2007], where the authors noted that SW alone provided close-to-optimal results in terms of energy.

In the next chapter, extensive experiments conducted on standard public datasets show the benefit of the different modeling components introduced in Chapters 3 and 4 in terms of tracking performance compared to recent state-of-the art methods.

5 Multi-Person Tracking Experiments

5.1 Introduction

In this chapter, we conduct exhaustive experiments to show the benefits of the different modeling components of our multi-person tracking framework and to benchmark it against reference approaches on state-of-the-art datasets. We conduct experiments on five different datasets, described in Section 5.2. Experimental details are given in Section 5.3 and our protocol is introduced in Section 5.4. We demonstrate the impact and benefit of the different modeling components of our framework in Section 5.5, before comparing it to state-of-the-art tracking approaches in Section 5.6. Qualitative results are shown in Section 5.7 while Section 5.8 discusses complexity and speed of the algorithms. Finally, Section 5.9 gives some concluding remarks.

5.2 Datasets

We used five public datasets for which bounding box annotations are available (see sample frames in Figure 5.1). For all datasets, unless specifically mentioned, we are using the official ground truth files.

PETS 2009. PETS'09 S2.L1 is a video of 795 frames recorded at 7 fps[1]. It presents a moderately crowded scene where 20 pedestrians are often crossing each other's trajectories, creating inter-person occlusions. People are also often occluded by a street light in the middle of the scene, creating misdetections. Although several views of the same scenario are available, we are working solely in View 001. As there is no official ground truth available for PETS, we are using the one provided by [Shitrit et al., 2011].

TUD. It consists of three short videos recorded at 25 fps. We focus on the two longest ones, which are also the ones presenting the most occlusions: TUD-Crossing (201 frames, 13 pedes-

[1] www.cvg.rdg.ac.uk/PETS2009/

Figure 5.1: Tracking datasets. Each dataset brings its own challenges. In PETS, the street light in the middle of the scene causes many occlusions. Most people are wearing dark clothes. On CAVIAR, many inter-person occlusions occur due to the low viewpoint. Another difficulty comes from the many entry/exit points (shops on the left, corridor towards the right). TUD sequences present an even lower viewpoint and heavy occlusions. In the Parking Lot dataset, there are many inter-person occlusions, and some occlusions by the white car. Notice as well that many persons are wearing the same (blue) clothes. TownCentre is a challenging scene with many inter-person occlusions as well as partial occlusions by the bench on the left.

trians) and TUD-Stadtmitte (179 frames, 10 pedestrians), showing respectively a pedestrian crossing and a town-centre pedestrian area. These videos have a low viewpoint, on the contrary to the PETS sequence.

CAVIAR. This corpus contains 26 monocular videos of a corridor view recorded at 25 fps[2]. The average video length is 1500 frames. To compare our performance to competitive approaches,

[2] groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

we use the same subset of 20 videos as [Zhang et al., 2008, Huang et al., 2008], containing 140 people, along with their selected ground truth, in which fewer persons are annotated as compared to the complete CAVIAR ground truth. Challenges in this dataset arise from reflections on the floor, projected shadows, occlusions, and numerous possible entry and exit points.

Parking Lot. The Parking Lot dataset used by [Zamir et al., 2012] is a 1000-frame video recorded at 29 fps, containing 14 pedestrians walking in queues. Challenges in this dataset include long-term inter-object occlusions, and appearance similarities between several subjects.

Town Centre. The Town Centre dataset introduced by [Benfold and Reid, 2011a] is a high-definition surveillance video of a busy town centre street recorded at 25 fps. This dataset is challenging because it contains a large number of people frequently occluding each other, especially on the left sidewalk and behind the bench. Bounding box annotations are given for 3 minutes of this video.

5.3 Experimental Details

5.3.1 Human Detection

In tracking-by-detection approaches, the tracking performance is subject to the detection accuracy. In the literature, different authors often apply different detectors suited to their techniques on a given dataset. For instance, on the PETS dataset, Ben Shitrit et. al. [Shitrit et al., 2011] use the POM detector [Fleuret et al., 2008] which exploits multi-camera information, Breitenstein et. al. [Breitenstein et al., 2011] use the HOG detector [Dalal and Triggs, 2005], Andriyenko et. al [Andriyenko et al., 2012] use a detector exploiting both HOG [Dalal and Triggs, 2005] and relative optical flow (HOF) [Walk et al., 2010] features within SVM classification. Similarly to us, Zamir et. al. [Zamir et al., 2012] use the part-based model detector [Felzenszwalb et al., 2010]. Hence, it is currently very difficult to have fair comparisons by re-using available detection results, as pointed out in [Milan et al., 2013].

The entry to our tracking-by-detection framework is the output of the part-based detector [Felzenszwalb et al., 2010] using the human deformable model trained on the INRIA person dataset[3]. In this work, we relied on a recently proposed accelerated version of DPM [Dubout and Fleuret, 2012] which exploits Fast Fourier transforms to speed up the per-part convolutions required by the algorithm. Benchmarked on the VOC dataset, the algorithm was shown to provide a speed-up of one order of magnitude over the DPM baseline. As mentioned in Section 3.3.1, adopting a part-based detector is an algorithmic choice allowing to extract motion from discriminatively trained parts (details given below) and to be able to use the same detector and method for all experiments and all datasets. It also presents the advantage of relying on a publicly available detector. Note that the part-based model detector that we use does not

[3] http://pascal.inrialpes.fr/data/human/

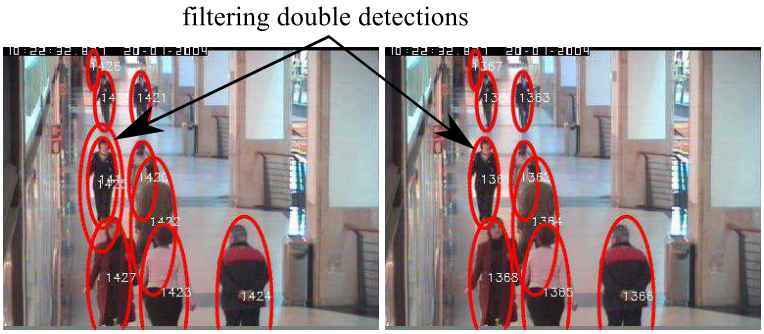


Figure 5.2: Filtering double detections. Based on a simple coverage test, double detections are removed. Left: before filtering. Right: after filtering.

completely solve the detection problem by itself. Indeed, as shown in Tables 5.5 and 5.9, our detector gives us similar input detection accuracies as compared to other approaches. Hence, the results shown in the manuscript are based on input detections that are affected by severe occlusions, false positives and misses.

5.3.2 Detection Filtering

As a pre-processing step, we filter detections so as to remove those of unrealistic size that correspond to false alarms. On PETS, we use calibration information to remove detections that do not match the typical height of a person given the location. On CAVIAR, we apply a simple criterion assuming a linear relationship between the image vertical position (of the bottom of the bounding box) and the bounding box height (in pixels). If a detection bounding box does not correspond (with a certain tolerance) to the theoretical height at a given vertical position, it is discarded. On the other datasets, size filtering is simply done by setting a lower and upper threshold on the height of bounding boxes in pixels. On TownCentre, we apply another filtering based on image location to remove the many false alarms created by a mannequin in the shop window (cf. shop in the bottom left corner of sample Town Centre image in Figure 5.1)[4].

Typically, the output of the detector can give double detections, especially if the detection threshold is low. An example of double detection is illustrated in the left part of Figure 5.2. Double detections produce overlapping bounding boxes that designate the same person and this redundancy can make the association task more difficult. Therefore, we choose to remove those repetitions based on an overlap criterion. The right part of Figure 5.2 shows the detections after this filtering step. More precisely, for any two detections (r_i, r_j) in a given frame, we compute $F_i = \frac{A(r_i \cap r_j)}{A(r_i)}$ and $F_j = \frac{A(r_i \cap r_j)}{A(r_j)}$, where $A(r)$ denotes the area defined by the region associated with the detection r. If $\max(F_i, F_j) > 0.6$, we assume there is a double detection and in this case, we remove the bigger detection of the two (the one with the smaller F).

[4] Note that pedestrians will not appear at these locations so we are sure we do not delete true detections.

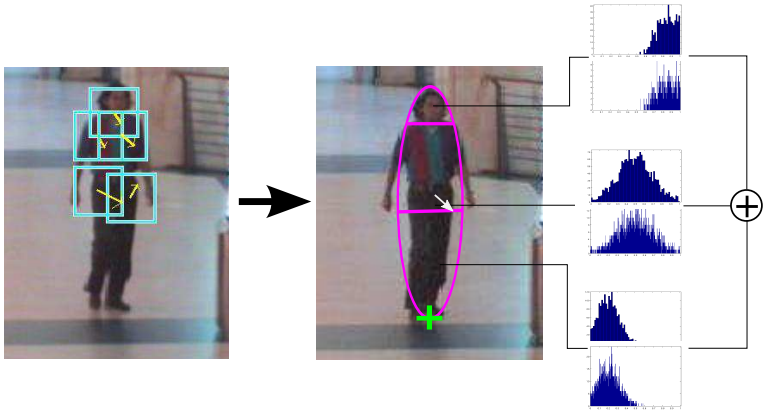


Figure 5.3: Extracted features for representing detections. Left: upper-body parts obtained from the deformable parts model (cyan bounding boxes) and estimated motion on each part (yellow arrows). Right: position (green cross), final motion feature (white arrow) and color histograms obtained from different pre-defined parts (head, torso, legs and fullbody).

5.3.3 Feature Computation

Position Feature. For the PETS and CAVIAR datasets, camera calibration and ground-plane homography are available, respectively. Using this information, position models are defined in the ground plane. On the other datasets, we defined the position models in the image plane.

Motion Computation. Several techniques could be applied to extract the motion vector $\mathbf{v}_i$ of a detection r_i. In this work, it is extracted by estimating an affine motion model on each of the 5 upper-body parts of the deformable part model (see Figure 5.3) using the robust multi-resolution approach by [Odobez and Bouthemy, 1995a], which provides individual part motion along with a confidence weight (as explained in Section 3.3.1). The overall motion is then obtained as the weighted average of these upper-body parts motions. Note that these upper body parts are not the limbs, but the head, shoulders and lower torso. We observed that their motion is in general similar. Confidence weights given by [Odobez and Bouthemy, 1995a] contribute to lower the scores of parts with unreliable motion.

Color Histograms. To avoid taking into account too many pixels from the background, we only consider the elliptical region enclosed within each bounding box. The parts are defined by vertically partitioning the ellipse into three parts, with the top 20% aiming at capturing the head, the 40% and 40% left in the middle and the bottom aiming at capturing the torso and the legs, respectively, as illustrated in Figure 5.3. As color descriptors $\mathbf{h}_i^b$ for each of the 4 pre-defined parts $b \in \mathscr{P} = \{\text{whole}, \text{head}, \text{torso}, \text{legs}\}$, we used RGB multi-resolution histograms (at resolutions $4 \times 4 \times 4$ and $8 \times 8 \times 8$) to reduce quantization effects.

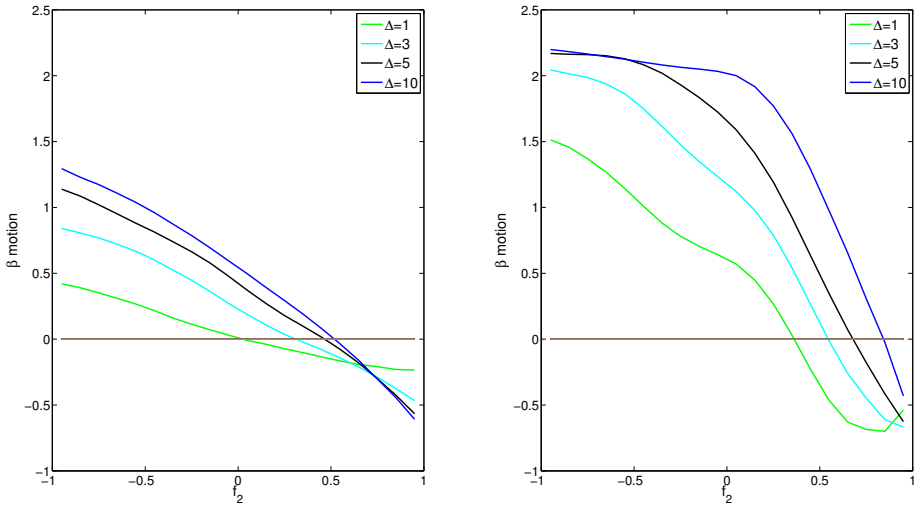


Figure 5.4: β curves of motion models learned from tracklets on different scenes. Left: β curves on CAVIAR for several values of Δ. Right: β curves on Town Centre for the same values of Δ. Both datasets have the same framerate. We can observe that the models for Town Centre are more discriminative than the ones for CAVIAR. This can be explained by the fact that most people in CAVIAR move along the camera axis direction, producing little discriminative motion as compared to Town Centre where people move obliquely with respect to the camera.

5.4 Experimental Protocol

5.4.1 Parameters

As illustrated in Figure 5.4 for the motion feature, models can vary a lot between different scenes, which advocates the learning of scene-specific models. Since we consider rather short sequences for testing, unsupervised learning of the parameters $\lambda = \{\lambda^k\}$, defined for each feature k as $\lambda^k = \{\lambda^k_\Delta, \Delta = 1 \dots T_w\}$ is performed in batch mode directly on the test sequence, i.e. the training set is the whole test sequence, except for the CAVIAR dataset, in which we use as training videos the set of 6 videos that are not used in the test.

Besides λ which are learned automatically, the same following parameters were used in all sequences: $\theta_f = 10$ frames for the position model forgetting factor (Section 3.5); $d_{\max} = 10$ frames and $\theta_{tm} = 3$ frames to define the label cost (Section 3.6). Besides, unless stated otherwise, unsupervised learning of interval sensitive parameters from tracklets was conducted, all features (including motion) were used, and SW optimization followed by Block ICM exploiting label cost with $\rho = 1$ was applied. Finally, we vary the size T_w of the temporal window to analyze the impact of connectivity.

5.4.2 Post Processing

After the tracks have been obtained by our optimization procedure, and before evaluation, we apply two simple post-processing steps. First, we remove short tracks from the solution, as these most likely correspond to false alarms. In practice, on all datasets, tracks shorter than 1.2 seconds are deleted.

Then, in order to fill the holes created by temporary missed detections, for each track of the final results, we linearly interpolate detections within the gaps.

5.4.3 Performance Measures

Detection. In order to compare our input detections to the ones used by other authors, when available, we report Det. Prec. and Det. Rec, which are respectively the frame-based precision and recall of the raw detections. The precision is defined as the number of correctly matched detections over the total number of detection outputs. The recall is defined as the number of correctly matched detections over the total number of ground-truth objects. On all datasets, these measures are computed following the criterion of the PASCAL Visual Object Classes (VOC) challenge on the intersection over union for matching[5]. We also provide recall and precision after tracking (Rec. and Prec.) by using tracking information to interpolate tracks and remove short ones.

Tracking. In the multiple person tracking literature, different existing evaluation metrics are not consistently used by competing approaches [Milan et al., 2013]. To achieve a fairer comparison with existing approaches, we use two types of measures to perform our evaluations.

Measures introduced in [Li et al., 2009] indicate how correct the tracks are in terms of fragmentation and confusion between different people. Namely, Frag is the number of times that a ground truth trajectory is interrupted in the tracking result, while IDS is the total number of identity switches, i.e. it indicates the number of times an output track is associated to several ground truth targets. We also report the number of tracker outputs SO after post-processing, the percentage of tracks that are tracked for more than 80% of their duration MT (Mostly Tracked), the percentage of tracks that are tracked between 20% and 80% of their duration PT (Partially Tracked) and the percentage of tracks that are tracked less than 20% of their duration ML (Mostly Lost).

Since the above metrics are not adopted by several competing sate-of-the-art tracking methods, we additionally use the CLEAR MOT metrics MOTA and MOTP [Bernardin and Stiefelha-

[5] If the intersection over union between a ground truth object and a detection bounding box is above a threshold, which is 0.2 in our experiments, then we consider that they match, otherwise not.

gen, 2008], which are defined as:

$$\text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{5.1}$$

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{5.2}$$

where, at time t, c_t is the number of matches found between ground truth objects and tracker estimates, d_t^i is, for each match, the distance between the object and its corresponding tracker estimate, g_t is the number of ground truth objects, and m_t, fp_t and mme_t are the number of misses, of false positives, and of mismatches, respectively. *Multi-Object Tracking Accuracy* (MOTA) thus combines missed detections, false positives and identity switches into a single evaluation measure. On the other hand, *Multi-Object Tracking Precision* (MOTP) gives a measure on bounding boxes localization accuracy.

In our evaluations, we use ground truth files in which occluded persons are annotated, otherwise their annotation is deduced using linear interpolation. For consistency, we use our interpolated tracks (see Section 5.4.2) to compute the matches between ground truth and tracking estimates.

5.5 Results - Component Analysis

In the following, we demonstrate quantitatively on some datasets the benefit of the different modeling factors, which we call *components*. Additional experimental results supporting the effectiveness of each component on other datasets are given in Appendix A.

5.5.1 Unsupervised Learning

Table 5.1 demonstrates the effect of learning model parameters from tracklets rather than from detections, as explained in Section 4.2. In practice, on the PETS data, we used tracklets obtained with models learned from detections with $T_w = 8$ (~1.1 second)[6] to relearn models from tracklets up to $T_w = 16$ (~2.2 seconds). We can observe that the refinement of model parameters using tracklets has almost no effect on the performance for $T_w = 8$, showing that the assumption of using the closest and second closest sets of detection pairs to learn models is valid for small values of T_w. However, with a larger association window ($T_w = 16$), using the default models leads to precise but very fragmented tracklets (92 different labels, 27 Frag). This fragmentation can be dramatically reduced by using the refined parameter estimates obtained from tracklets, showing the benefit and validity of our approach.

[6] First line of Table 5.1.

T_w	*MET*	Rec	Prec	SO	MT	PT	ML	Frag	IDS
8	Off	0.84	0.95	40	70%	25%	5%	13	1
8	On	0.84	0.95	39	70%	25%	5%	12	0
16	Off	0.82	0.95	92	60%	35%	5%	27	0
16	On	0.87	0.94	25	70%	25%	5%	3	0

Table 5.1: Unsupervised learning. SW optimization for PETS using model parameters estimated from tracklets (MET = "On"), or not (MET = "Off").

T_w	*TW*	Rec	Prec	SO	MT	PT	ML	Frag	IDS
16	Off	0.86	0.94	26	70%	25%	5%	6	3
16	On	0.87	0.94	25	70%	25%	5%	3	0

Table 5.2: SW optimization output for PETS sequence using time-interval sensitive models (TW = "On") or not (TW = "Off") for the color and motion models.

5.5.2 Time Interval Sensitivity

One might argue that learning motion and color similarity models that depend on the time gap between detection pairs may have no impact on the results, since within our association windows, motion and appearance patterns of an individual are likely to stay similar. However, Table 5.2 demonstrates empirically that exploiting such time-interval dependent models indeed helps reaching better tracking performance, and confirms the dependencies observed on the learned β curves (see Figures 3.5 and 3.6). When the motion and color features between pairs of detections are collected from tracklets regardless of their time difference[7] ($TW = 0$), worse results are obtained, resulting in 3 more fragmentations and IDS.

5.5.3 Temporal Context

The benefit of using a longer temporal connectivity between detection pairs is demonstrated in Table 5.3, where we observe that larger T_w values reduce fragmentations. This is due to two main reasons. First, note that tracks for which there are long intervals with no detections (beyond T_w) can not receive the same label, since no link is created between the detections before and after the misdetection interval. Hence, increasing T_w can solve these misdetection situations (e.g. due to short occlusion). This is mainly illustrated in PETS where people tend to get occluded by the street lamp for more than 10 frames. By increasing T_w to a value of 16, the number of fragmentations gets significantly reduced (e.g. from 12 to 3 when using all features). The second reason is that a longer temporal connectivity that relies on all pairwise links leads to an energy that is better conditioned for optimization, or in other words, that provides a better temporal context for labeling. This is illustrated in Figure 5.5 in an example

[7] The position model is learned normally.

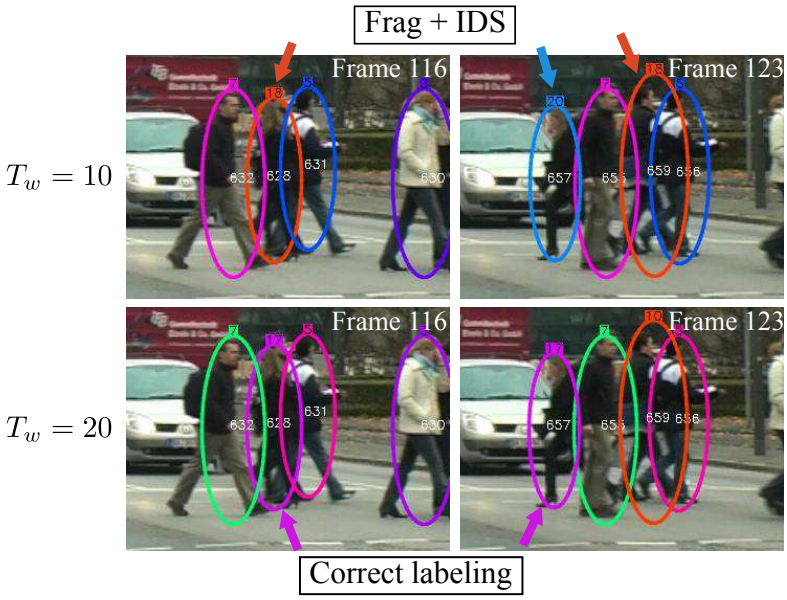


Figure 5.5: Temporal context effect. First row: Even though the occluded person with the orange label (#18) reappears less than $T_w = 10$ frames later, the links do not provide enough context to reassign her with the correct label. Bottom row: when a longer context is available ($T_w = 20$) more pairwise comparisons are available, allowing to maintain a correct labeling.

from TUD-Crossing.

5.5.4 Visual Motion Cue

Table 5.3 also demonstrates the usefulness of the motion feature at solving ambiguities and therefore reducing the number of identity switches. In practice, these ambiguities happen mainly when people with similar appearance are crossing trajectories and there are important misdetection periods and badly framed detections (i.e. encapsulating parts of the two people). The position model that does not favor any movement direction and the color model might not be discriminant enough to solve the association in these cases, and the motion feature adds the complementary information. Note here that confidence weighting is important, as motion estimates at the near proximity of the crossing might be unreliable because bounding boxes tend to get blended together, but previous motion estimates are then prevailing in the energy term because of their higher confidence (the same goes for the color models). In the end, by using the motion feature and a sufficiently large value of T_w, we are able to obtain pure tracklets with few IDS in general.

5.5.5 Label Costs and Block ICM Optimization

We evaluated the benefit of using label costs with a more global optimization to improve performance. On PETS data, where the Sliding Window approach already provides very good

	T_w	*motion*	Rec	Prec	SO	MT	PT	ML	Frag	IDS
PETS	8	Off	0.84	0.95	38	70%	25%	5%	13	2
	8	On	0.84	0.95	39	70%	25%	5%	12	0
	16	Off	0.87	0.94	23	70%	25%	5%	4	3
	16	On	0.87	0.94	25	70%	25%	5%	3	0
TUD	10	Off	0.77	0.98	20	70%	30%	0%	6	2
	10	On	0.77	0.99	20	70%	30%	0%	6	2
Stadtmitte	20	Off	0.79	0.98	19	70%	30%	0%	5	2
	20	On	0.79	0.99	19	70%	30%	0%	4	1

Table 5.3: SW optimization output on PETS and TUD-Stadtmitte sequences with different temporal window sizes and using the motion feature or not. Using the motion feature (motion="On") and larger temporal window T_w provides better results. On PETS, $T_w = 8$ frames ~1.1 second. On TUD, $T_w = 10$ frames = 0.4 second.

T_w	*BlockICM*	Rec	Prec	SO	MT	PT	ML	Frag	IDS
20	Off	0.79	0.99	19	70%	30%	0%	4	1
20	On	0.81	0.99	18	70%	30%	0%	1	0

Table 5.4: Effect of Block ICM with label costs for TUD-Stadtmitte.

tracking results with only 3 fragmentation and 0 IDS, no improvement was observed. However, results on TUD-Stadtmitte with $\rho = 3$ (Table 5.4) shows that several errors can be corrected, allowing us to reach a very good performance of just 1 Frag and 0 IDS. A similar benefit could be observed on CAVIAR data, where Block ICM and label cost acted towards fragmentation reduction while solving some IDS ambiguities as well. From our experience, it stands out that Block ICM with label costs can be useful to correct some mistakes through the incorporation of track start and end penalizations leveraging on scene-specific knowledge to define prior label information.

5.6 Results - Comparisons with State-of-the-Art Approaches

Tables 5.5, 5.6, 5.7, 5.8 and 5.9 show the comparison with recent state-of-the-art algorithms for the different datasets, when available. Although there are public methods for tracking evaluation, as mentioned earlier, there is a lack of a unique standard procedure: some authors use MOT metrics while others use fragmentation and IDS. This makes fair comparison against several methods, including recent ones, difficult, as pointed out by Milan et. al. [Milan et al., 2013]. Here, we evaluate our performance with different existing metrics to allow comparison with existing approaches that have some similarities to our proposal. Note as well that, as discussed in Section 5.3, different authors often use different detectors. For the sake of having more detailed comparisons, we also report and discuss the input detection recall and precision of our detections and compare them to those of the detections provided by the

different authors, when available.

CAVIAR. On the CAVIAR dataset, Table 5.5 compares our results obtained with an association horizon of 1.5 second ($T_w = 38$) and default parameters, with the approaches from [Huang et al., 2008] and [Zhang et al., 2008]. Note first that our detector delivers lower performance, with a worse detection recall for a comparable detection precision. Nevertheless, the table shows that we outperform [Huang et al., 2008] in terms of Frag and IDS. As compared to the network flow formulation of [Zhang et al., 2008] (algo. 1), we reach an almost identical number of IDS (8 vs. 7) but with much less fragmented tracks (38 vs. 58). When adding an explicit occlusion model on top of the flow model (algo. 2), the method in [Zhang et al., 2008] reduces the number of fragmentations to 20, but this is at the cost of a higher number of IDS (15). Our approach thus offers a good tradeoff between their methods.

TUD. For the TUD and PETS datasets, we report our results obtained with $T_w = 20$ (0.8 second) and $T_w = 16$ (~2.2 seconds)[8], respectively. In the TUD-Crossing sequence which contains heavy occlusions, we obtain 1 Frag and 0 IDS, outperforming the method of [Breitenstein et al., 2011] (2 IDS) and we equal [Zamir et al., 2012] in terms of IDS. However, they both present a better MOTA score. This can be explained by the fact that MOTA takes into account not only IDS, but also tracking precision and recall. In this sequence, people are often occluded because they walk next to each other, and this translates into low detection recall. For instance, by the end of the sequence we miss a subject due to such an occlusion, as is illustrated in Figure 5.6. Since the proposed method does not attempt to propagate detections nor extrapolate tracklets, such misdetections penalize the tracking recall, and ultimately the MOTA. The methods of [Breitenstein et al., 2011] and [Zamir et al., 2012] generate candidate detections by using particles and virtual nodes, respectively, potentially overcoming problems with missing detections due to occlusion. Despite the lack of detections, in this sequence our method obtains pure tracklets, with only 1 fragmentation.

On TUD-Stadtmitte, we outperform [Andriyenko et al., 2012] both in terms of Frag, IDS and MOT metrics. We reach similar results as [Zamir et al., 2012] and [Yang and Nevatia, 2012a], with 1 Frag and 0 IDS. However, we outperform [Zamir et al., 2012] in terms of MOT metrics.

PETS. On PETS, we clearly outperform other techniques insofar as we reach 0 IDS. The authors of [Zamir et al., 2012] obtain comparable MOT metrics but with a much higher number of 8 IDS. It can be noted that one of our fragmentations is due to the fact that a person going out of the scene and coming back later is annotated as one single ground truth object. This situation is out of the scope of this thesis, as we do not tackle the re-identification problem. Another fragmentation is due to a very long occlusion by the street lamp (more than 10 seconds) and is shown in Figure 5.7.

Parking Lot - Town Centre. Finally, we compare our tracking results to state-of-the-art meth-

[8] As the TUD sequences are quite short, it is difficult to learn parameters for larger time intervals.

Figure 5.6: Effect of missed detections near the end of the sequence on TUD-Crossing. A person is detected for the last time in frame 168 and is assigned label #21. The person is not re-detected before the end of the video. As we do not propagate detections nor extrapolate tracklets, tracking is lost for that person. Images were cropped to highlight the misdetection.

	Det. Rec	Det. Prec	Rec	Prec	Frag	IDS
Huang et. al. [Huang et al., 2008]	0.88	0.70	0.86	-	54	12
Zhang et. al. [Zhang et al., 2008] algo 1	0.88	0.70	-	-	58	**7**
Zhang et. al. [Zhang et al., 2008] algo 2	0.88	0.70	-	-	**20**	15
Ours	0.82	0.69	0.78	0.93	38	8

Table 5.5: Comparison with state-of-the-art approaches on CAVIAR.

ods on Parking Lot and Town Centre. On both sequences, we use a temporal connectivity of $T_w = 40$. These results are summarized in Table 5.9. We obtain a similar MOTA than [Zamir et al., 2012] on the Parking Lot sequence. However, our tracking precision is higher. On the Town Centre sequence, we outperform [Benfold and Reid, 2011a] and [Zamir et al., 2012] both in terms of MOTA and MOTP. Note that on these datasets, the recall and precision of our detections are similar to those of the detections provided by the authors of [Zamir et al., 2012] for Parking Lot, and [Benfold and Reid, 2011a] for Town Centre[9].

5.7 Qualitative Results

Qualitatively, Figures 5.8, 5.9, 5.10, 5.11 and 5.12 show that even in the presence of multiple occlusions and ambiguities, our algorithm is able to maintain correct tracks throughout time[10]. Figure 5.13 shows an example of a correct labeling after a challenging trajectory crossing where one person is occluded for a relatively long time. On the other hand, Figure 5.14 shows a failure case, where an IDS occurs. In this case, it is difficult to recover because of the little amount of temporal connections (the person exiting the shop gets almost immediately occluded) and because of imprecise bounding box localization (cf. frame 911).

[9] We recall that Zamir et. al. [Zamir et al., 2012] also use the part-based model detector on all datasets.
[10] Tracking videos can be found online www.idiap.ch/~aheili/tracking.html

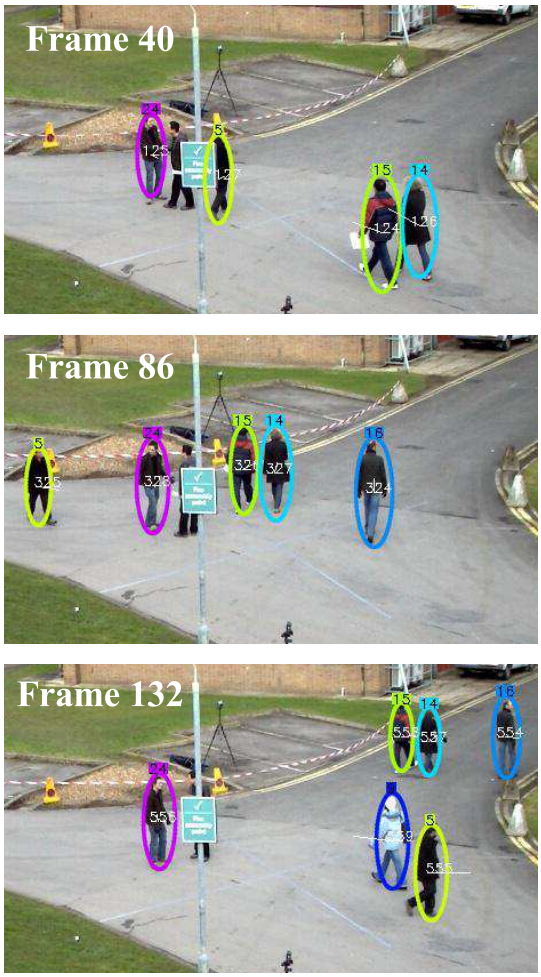


Figure 5.7: Long occlusion by scene occluder on PETS data. The person occluded by the street lamp for a long time will not be recovered by our tracker when it will be re-detected. However, persons occluded in the short term, e.g. person with id #5 can be tracked throughout occlusions. Images were cropped to highlight the region around the street lamp.

5.8 Complexity and Speed

Detection-based tracking approaches can basically be described as two processing steps: detection and association. While the main cost of the human detector is very proportional to the size of the input image and is rather independent of its content, the tracker cost comes from the appearance and motion information extraction, the graph construction, and the graph optimization. Appearance and motion feature extraction is done once for every detection, and is thus not affected by the amount of temporal connectivity. Pairwise β term computation to build the graph, however, depends directly on the connectivity, but relies on simple distances between feature vectors whose computation cost is rather small or that can be easily optimized.

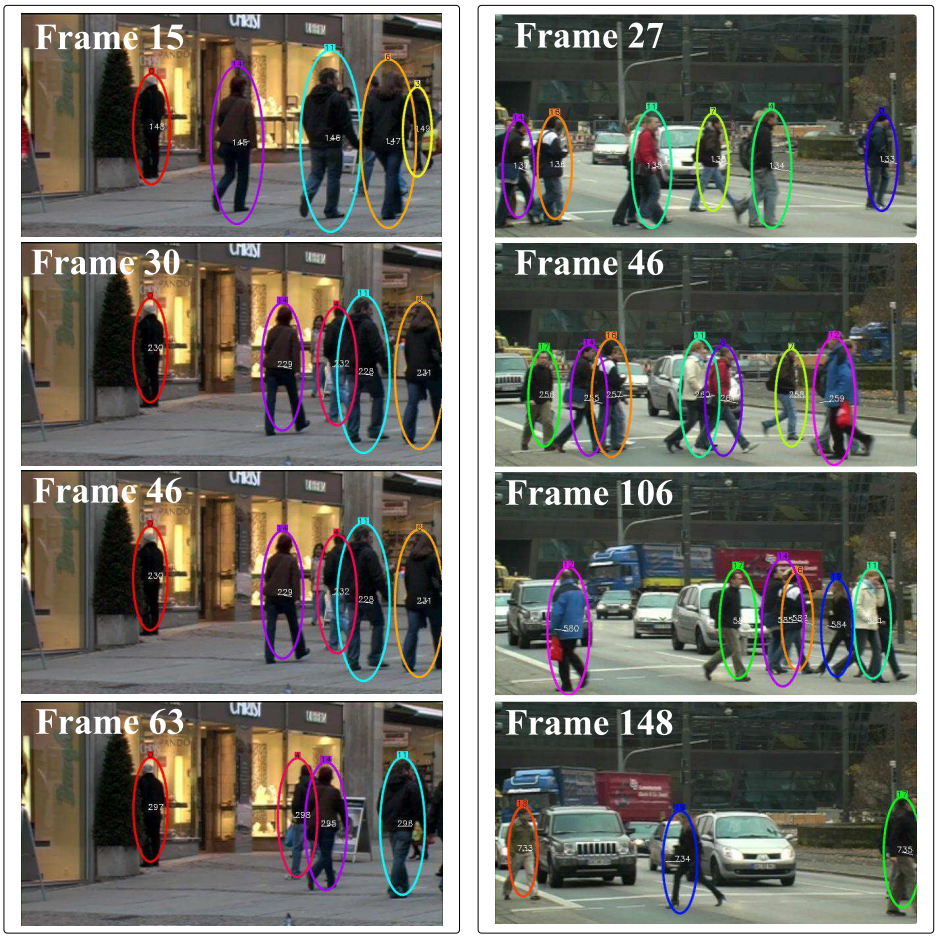


Figure 5.8: Visual results on TUD-Stadtmitte (1st column) and TUD-Crossing (2nd column). Images were cropped to highlight interesting regions.

Figure 5.9: Visual results on PETS S2.L1 sequence (View 001). Images were cropped to highlight interesting regions.

Figure 5.10: Visual results on CAVIAR.

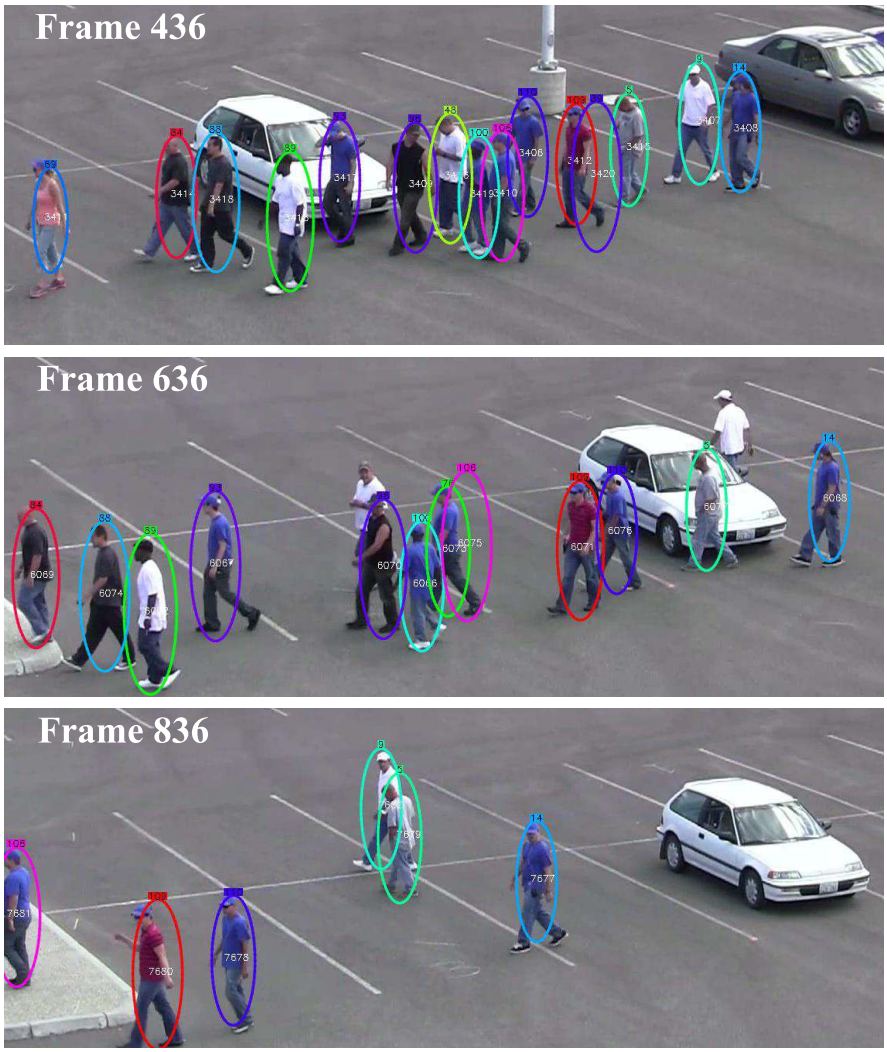


Figure 5.11: Visual results on Parking Lot. Images were edited to highlight interesting regions.

Figure 5.12: Visual results on Town Centre. Images were cropped to highlight interesting regions.

	Rec	Prec	Frag	IDS	MOTA	MOTP
Breitenstein et. al. [Breitenstein et al., 2011]	-	-	-	2	0.84	0.71
Zamir et. al. [Zamir et al., 2012]	0.93	0.99	-	**0**	**0.92**	0.76
Ours	0.89	0.93	**1**	**0**	0.79	**0.78**

Table 5.6: Comparison with state-of-the-art approaches on TUD-Crossing.

	Rec	Prec	Frag	IDS	MOTA	MOTP
Andriyenko et. al. [Andriyenko et al., 2012]	-	-	**1**	4	0.62	0.63
Yang et. al. [Yang and Nevatia, 2012a]	0.87	0.97	**1**	**0**	-	-
Zamir et. al. [Zamir et al., 2012]	0.81	0.96	-	**0**	0.78	0.63
Ours	0.81	0.99	**1**	**0**	**0.90**	**0.84**

Table 5.7: Comparison with state-of-the-art approaches on TUD-Stadtmitte.

As for the optimization, since the SW algorithm depends on the Hungarian algorithm, its complexity is polynomial in $O(n^3)$, where n is the sum of the number of detections in the current frame and the number of current tracks in the sliding window. Therefore, longer term connectivity does not necessarily imply an increase in complexity. Indeed, as there are typically fewer fragmentations (and thus fewer tracks) when using longer temporal windows, the complexity might even be reduced. Similarly, Block ICM is optimized using the Hungarian algorithm, and its complexity is polynomial in the sum of the number of tracks before and after the currently optimized frame in the ICM sweep.

To give an idea about the computational complexity of our tracking algorithm, we report the following average processing times per frame on the medium crowded scene of PETS 2009 with an association horizon T_w of 2 seconds, tested on a 2.9 GHz Intel Core i7 laptop with 8GB of RAM and assuming detections are available: 150ms for visual motion estimation and color features extraction; 180ms for computing the pairwise β terms; 60ms and 280ms for SW and Block ICM optimization, respectively. Note that we have an unoptimized implementation in Python with no threading. Online tracking processing could be achieved by optimizing algorithmic steps, for instance using simple and quick procedures to trim unnecessary links in the graph, e.g. by not creating links between detection pairs that are separated by unrealistic distances. This link pruning procedure based on distance could depend on the time interval between detections as well as on the local detection density. Additionally, motion information could be used to guide the pruning, such that links between detections with incompatible motion are preferrably discarded. Another way of reaching online processing could involve a selection of the time steps at which applying Block ICM could be useful, or through code optimization (programming language, multi-threading, etc.) as well as by processing videos at a lower framerate[11].

[11] We indeed observed no loss in tracking accuracy on the CAVIAR dataset when processing only one frame out of five.

	Rec	Prec	Frag	IDS	MOTA	MOTP
Andriyenko et. al. [Andriyenko et al., 2012]	-	-	8	10	0.89	0.56
Shitrit et. al. [Shitrit et al., 2011]	-	-	-	9	-	-
Breitenstein et. al. [Breitenstein et al., 2011]	-	-	-	-	0.80	0.56
Zamir et. al. [Zamir et al., 2012]	0.96	0.94	-	8	**0.90**	**0.69**
Ours	0.87	0.94	**3**	**0**	0.89	0.66

Table 5.8: Comparison with state-of-the-art approaches on PETS S2.L1.

		Det. Rec	Det. Prec	MOTA	MOTP
Parking Lot	Zamir et. al. [Zamir et al., 2012]	0.86	0.96	**0.90**	0.74
	Ours	0.91	0.96	0.89	**0.85**
Town Centre	Benfold et. al. [Benfold and Reid, 2011a]	0.77	0.88	0.65	0.80
	Zamir et. al. [Zamir et al., 2012]	-	-	0.76	0.72
	Ours	0.73	0.90	**0.79**	**0.82**

Table 5.9: Comparison with state-of-the-art approaches on Parking Lot and Town Centre.

5.9 Conclusion

In this experimental chapter, we demonstrated the benefits of the different components of our multi-person tracking framework. The main advantage of using longer-term connectivity is the availability of more pairwise observations which better condition the energy to minimize. When exploited in conjunction with the other contributions which proposed to use visual motion, confidence weights, time-sensitive parameters, unsupervised learning from tracklets and higher-order costs, our algorithm provides tracking results that are on par with or better than recent competing approaches.

However, like all tracking-by-detection approaches, our performance is subject to detection accuracy. In our experiments, we did not study the effect of the detection threshold on the tracking performance. On the one hand, a low detection threshold could potentially introduce many false alarms that can be hard to deal with. For instance, in the presence of numerous and recurrent false positives, it would be difficult to remove spurious tracks a posteriori based on their duration. Moreover, having more detections overall would increase the complexity of the data association. On the other hand, a high detection threshold could potentially introduce a high level of misdetections that can also have a negative impact on our algorithm. In the end, what is needed by our tracker is a good tradeoff between the precision and recall of input detections. In order to overcome the potential lack of detections, our algorithm could be improved by propagating detections in the past and in the future, similarly to what [Benfold and Reid, 2011a] [Yang and Nevatia, 2012b] do. Furthermore, to handle long occlusions, tracklet-level appearance factor terms potentially relying on online learned discriminative models could be solved at another level of hierarchy.

Figure 5.13: People tracked through trajectory crossing on CAVIAR. People with id labels #93 and #94 cross trajectories, but our tracker successfully maintains a correct labeling throughout the occlusion thanks to long-term connectivities and the confidence weighting scheme that downvotes the feature contributions at the temporal proximity of the occlusion so that the tracker can rely more on reliable feature information from before and after the occlusion. Images were zoomed in to show the occlusion of interest. (Interpolated detections are shown here; the person in the back is not detected around frame 3206.)

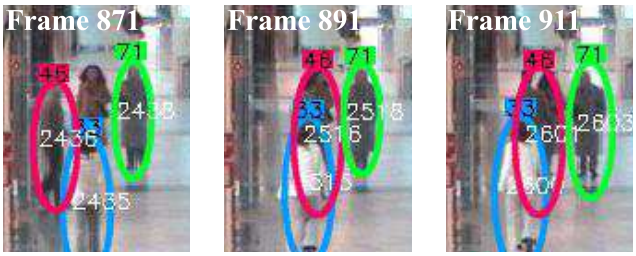


Figure 5.14: Identity switch example on CAVIAR. Person with id #46 comes out of a shop then gets occluded in the next frames. Another person that was previously occluded takes his label. Images were zoomed in around the location where the IDS takes place.

6 Body and Head Pose Estimation

6.1 Introduction

In the previous chapters, we introduced an original framework for multi-person tracking, which we showed could give robust tracking results on standard surveillance datasets. The obtained tracks can give valuable information to a surveillance system, as they tell where people are and how they are moving in a scene.

In this chapter, we study the problem of estimating people's body and head poses in open spaces, where low resolution typically prevents from estimating richer pose representations, e.g. with complicated articulated models. As was introduced in Chapter 1, these orientations can provide complementary cues to the trajectories that can help understand behaviors and interactions better.

Two approaches were investigated. The first one presented in Section 6.2 formulates body and head pose estimation in a Bayesian filtering framework. First, head and body pose likelihood models are constructed using sparse or template feature representations. Then, location obtained from the tracker, body and head pose are jointly estimated within a particle filtering framework to exploit the temporal coherency of these cues. This temporal filtering takes into account two soft coupling constraints. The first is a coupling between body orientation and velocity direction which is speed dependent, i.e. it takes into account whether people are moving or static. The second is a coupling between head and body orientation, stemming from anatomical constraints.

In Section 6.3, we present a second approach. Indeed, a limitation of the previous approach and in general of pose classification techniques is that they train classifiers on data that is different from the target scene. Such discrepancies can be due to different viewpoints or different appearances. This poses the problem of the generalization capability of the learned classifiers. To address this issue, the method of [Chen and Odobez, 2012] adapts pose classifiers so that they perform well on external labeled datsets (based on *ground-truth* labels), as well as on the test data from the target scene (based on *weak* labels derived from track velocity

information). In addition, the manifold structures, formulated as similar features should be classified with similar poses, are used to constrain the outputs of the classifiers. In our work, we propose to improve their method by better constraining the manifold structures and performing alignment between training and test manifolds in a semi- or weakly-supervised way.

Finally, Section 6.4 draws some conclusions about the problem.

6.2 Joint Body and Head Pose Estimation in a Temporal Filtering Framework

6.2.1 Approach Overview

The workflow of our first approach is summarized in Figure 6.1. First, we employ our multi-person tracker to generate continuous tracks, where each track contains a noisy bounding box sequence in the image for a person identity. Typical noisy trajectories are shown in Figure 6.2. Then, for each bounding box, we perform some static analysis. Namely, the head is localized, and body and head features are extracted and used to evaluate the body and head pose likelihood under a set of discrete poses. Finally, based on these likelihoods, we perform a joint estimation of all the cues in a particle filtering framework. The joint estimation takes into account the smoothness of cues over time, which is ensured by the temporal filtering itself, and the dependency between the cues, including a soft and speed dependent coupling between motion direction, body orientation and head orientation. More details about our temporal filtering approach are provided in the following subsections:

- We first briefly describe our method to localize the head within a human bounding box in Section 6.2.2.
- We then describe the representations, features and likelihood models used for body and head pose estimation, respectively in Sections 6.2.3 and 6.2.4.
- We present our particle filter for temporal filtering, along with how it models the soft coupling constraints in Section 6.2.5.
- Finally, the experimental Section 6.2.6 evaluates our approach and shows its validity.

6.2.2 Head Localization

Prior to extracting head pose observations, we first estimate the head localization and size from the human body bounding box. Here, routines such as face detection or skin color detection can not be exploited since they will fail when the head is not faced towards the camera. To design a robust pose-independent head localization algorithm, we rely on a HOG-based feature and Adaboost classifier. As the focus of this chapter is on pose estimation, we just give a brief overview of our head localization strategy. More details can be found in [Chen et al., 2011b].

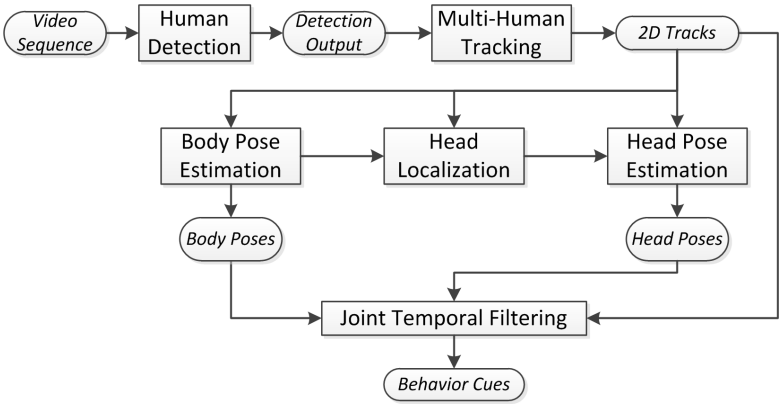


Figure 6.1: Workflow of our approach. Our multi-person tracking-by-detection algorithm is applied on a video sequence. From each obtained track, body and head features are extracted and fed into pose classifiers. The estimated poses, along with the track locations are then refined jointly in a temporal filtering framework.

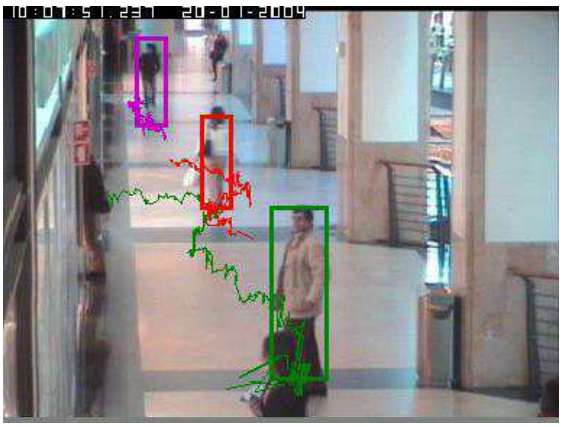


Figure 6.2: Trajectories obtained by the tracking algorithm are noisy sequences of bouding box locations. These noisy locations, along with the noisy output of body and head pose classifiers will be used to refine trajectory and pose estimates in a joint temporal filtering framework.

First, given a training dataset of positive (head) or negative (non-head) samples, we learn a head detector using an Adaboost classifier. As feature, we use a HOG (Histogram of Oriented Gradients) feature vector.

Then, at test time, we apply the head classifier to different head localization configurations (locations, sizes, aspect ratios) within an upper region of the detected person bounding box. Furthermore, to filter multiple local detection hits, we use the real-valued score of the detector to build a confidence map on the possible head locations, and perform non-maximum suppression to find local maxima as localized heads. The detector scores of those local maxima are used to accept or reject the detection, but we always assume the presence of at least one head. Finally, to select the single head location used for further processing, we apply a separate and simple temporal filtering on the head location candidates by enforcing a

Figure 6.3: Head localization results. Dashed blue boxes are the human detection outputs. Solid red boxes are the head localization outputs (first candidate for each bounding box). Failure cases are shown in the last row.

head location smoothness over time.

Figure 6.3 shows some head localization examples. Our method successfully localizes most heads, but fails on some examples as illustrated in the last row of the figure, mainly due to a badly detected human bounding box. In the following, we extract body features from the human detection bounding boxes and head features from the localized head patches. These features will be used to train pose classifiers.

6.2.3 Body Pose Likelihood Modeling

Body Pose Representation. Given the low resolution images, we quantize the body orientation in the image into eight directions (See Figure 6.4): N, NE, E, SE, S, SW, W, NW[1]. Note that at

[1]The naming of these directions is just for convenience and has nothing to do with the real directions such as north/south.

Figure 6.4: Eight body pose classes.

this stage body pose classification is performed in the 2D images, and no 3D information (e.g. the camera's tilting angle) is inferred. We make a reasonable assumption that the camera tilt is not too large ($< 30°$) and that the pose classification can be conducted without explicitly considering the tilt.

Defining the Body Pose Feature $\mathbf{z^b}$. For each human bounding box in the image, we calculate a multi-level HOG feature vector. The bounding box is divided into non-overlapping blocks at three different levels: 1×3, 2×6 and 4×12. Each block in turn consists of 4 cells. We quantize the gradient orientation into 9 unsigned directions, and each pixel votes to the corresponding direction using the gradient magnitude as weight. In this way, for each human region we end up with a 2268 dimensional feature vector.

Body Pose Likelihood Model $\mathbf{p^b}$. To model the pose likelihood, we rely on a sparse representation technique, which has been shown to be effective in image analysis and face recognition [Wright et al., 2009]. Let $\{(\mathbf{z}_i^\mathbf{b}, l_i)\}$ $(1 \leq i \leq N)$ denote the training data, where each $\mathbf{z}_i^\mathbf{b}$ is a multi-level HOG feature vector, and l_i is the corresponding pose label. For a novel feature vector $\mathbf{z}^\star$, the pose label l can be inferred from its relation to the training data. Specifically, $\mathbf{z}^\star$ can be approximated as a linear combination of the training features:

$$\mathbf{z}^\star \approx a_1 \mathbf{z}_1^\mathbf{b} + ... + a_N \mathbf{z}_N^\mathbf{b} = \mathbf{Za}, \tag{6.1}$$

where $\mathbf{Z} = [\mathbf{z}_1^\mathbf{b}, ..., \mathbf{z}_N^\mathbf{b}]$, and $\mathbf{a} = [a_1, ..., a_N]^T$ is the vector of reconstruction weights subject to the non-negativity constraint $a_i \geq 0$. It is reasonable to assume that $\mathbf{z}^\star$ will be well approximated by using only the part of training data with the same inherent pose label, which means the reconstruction vector $\mathbf{a}$ is sparse. To seek for the sparse solution, an L_1 term is used for regularization and our goal is to find:

$$\mathbf{a}^\star = \arg\min \left\| \mathbf{z}^\star - \mathbf{Za} \right\|_2^2 + \gamma \|\mathbf{a}\|_1, \tag{6.2}$$

where $\|.\|_2$ and $\|.\|_1$ are the L_2 norm and L_1 norm, respectively, and γ is the parameter controlling the importance of the L_1 regularizer. In principle, the non-zero elements of $\mathbf{a}^\star$ will be concentrated on the training data point with the same pose label as $\mathbf{z}^\star$. Therefore, we propose to use the normalized sum of the weights of the samples belonging to a pose class k as likelihood of the observation under this class. More precisely, we define $p^\mathbf{b}(\mathbf{z}^\star | k)$ as follows:

	Accuracy 1	Accuracy 2
[Andriluka et al., 2010] (SVM)	0.42	0.70
[Andriluka et al., 2010] (SVM-adj)	0.35	**0.76**
Multi-Class SVM	0.48	0.75
Sparse representation	**0.55**	**0.76**

Table 6.1: Comparison of body pose classification performance. Accuracy 1 denotes the ratio of exact hits to the total number of classified samples, while Accuracy 2 also considers adjacent pose hits as correct.

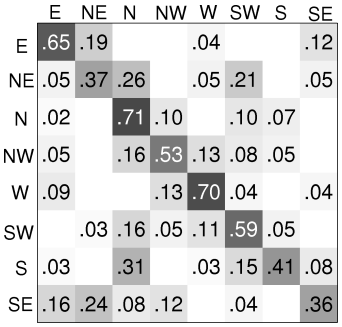


Figure 6.5: Confusion matrix for body pose classification.

$$p^{\mathbf{b}}\left(\mathbf{z}^{\star}|k\right) = \sum_{l_i=k} a_i^{\star} \Big/ \left\|\mathbf{a}^{\star}\right\|_1 \tag{6.3}$$

Evaluating the Body Pose Likelihood Model. We use the TUD Multiview Pedestrians dataset [Andriluka et al., 2010] providing train and test sets to evaluate our novel sparsity-based body pose likelihood model used as body pose classifier. For sparse representation, we use the toolbox as in [Kim et al., 2007] which utilizes the truncated Newton interior-point method. We compare our performance with that reported in [Andriluka et al., 2010]. As classifier, we also tried a multi-class SVM (Support Vector Machine) on multi-level HOG features. Table 6.1 shows the details. As in [Andriluka et al., 2010], we report two performance measures: Accuracy 1 where we only consider exact hit, and Accuracy 2 where an adjacent pose hit is also considered as correct.

As can be seen, our method generates more concentrated pose predictions than [Andriluka et al., 2010]. Figure 6.5 shows the classification confusion matrix of our method, where each row is the distribution of predicted labels over a ground-truth group. There is a concentration on the diagonal of the confusion matrix. Note that quite a few errors are introduced by assigning an adjacent rather than the exact direction (which is not a complete error). Also, there are some confusions between symmetric poses (i.e. N and S, NE and SW).

6.2.4 Head Pose Likelihood Modeling

Head Pose Representation. In this section, we represent head pose as pan β and tilt γ angles in the image plane[2]. Considering the resolution of surveillance video, we discretize the pan into 12 angles with 30° interval[3], and we discretize tilt angle into 3 classes: up ($\gamma > 30°$), middle ($-30° < \gamma < 30°$) and down ($\gamma < -30°$). Therefore, we have a set of 36 poses (β_m, γ_m).

Defining the Head Pose Feature $\mathbf{z^h}$. We use both texture and color features for head pose estimation. As texture feature, we use again a multi-level HOG descriptor. The head patch is divided into non-overlapping blocks at two levels: 2×2 and 4×4 blocks. Each block in turn consists of 4 cells. The gradient orientation is quantized into 9 unsigned bins, and the 4×9 entries of a block are normalized to 1. In this way, for each head patch we end up with a 720 dimensional feature vector. For the color feature, we use the histogram-based skin color detector proposed in [M. J. Jones and J. M. Rehg, 2002] to detect the skin region in the head patch. Then, the head patch is resized into a 20×20 binary skin mask as our 400 dimensional color feature.

Head Pose Likelihood Model $\mathbf{p^h}$. Learning the likelihood is conducted assuming training data with known head poses. For each class m, we calculate the mean texture feature $\mathbf{r}_m^{\text{text}}$ and mean color feature vectors $\mathbf{r}_m^{\text{col}}$. Then, the likelihood of a head patch observation $\mathbf{z^h} = \left(\mathbf{z}^{\text{text}}, \mathbf{z}^{\text{col}}\right)$ for a given pose class m is expressed as:

$$p^{\mathbf{h}}\left(\mathbf{z}^{\mathbf{h}}|m\right) = p^{\text{text}}\left(\mathbf{z}^{\text{text}}|m\right) p^{\text{col}}\left(\mathbf{z}^{\text{col}}|m\right) \tag{6.4}$$

where each component likelihood is in turn expressed as:

$$p^F\left(\mathbf{z}^F|m\right) = \exp\left(-\lambda^F d^F\left(\mathbf{z}^F, \mathbf{r}_m^F\right)\right), \tag{6.5}$$

where $F = \{\text{text}, \text{col}\}$ is the feature type, λ^F is a parameter, and $d^F(.)$ is the distance between the observed feature and the mean feature. For the texture feature, we use the L_2 distance. For the color feature, we use the L_1 distance.

6.2.5 Temporal Filtering with Coupling Constraints

Up to now, for each human detection output, we have extracted the body and head 2D location, as well as body and head pose features. Using the defined likelihood models, for each frame, we could estimate the optimal body and head poses. However, such estimates would be quite noisy. For example, the bounding box jumps in the image due to uncertainties of the human detector, and the body/head pose estimation can be wrong due to poorly localized bounding

[2] That is, the pose is defined with respect to the viewing direction, which means that $(\beta, \gamma) = (0, 0)$ corresponds to a person looking straight at the camera, whatever his image position.

[3] Unlike some works where the head pose is only estimated in frontal and profile views, we allow the 360° full pan range to include back view.

boxes or partial occlusion. To improve the accuracy, we use temporal filtering to exploit the intra-cue temporal smoothness, and the estimation is conducted jointly to also exploit the inter-cue dependencies.

Particle Filtering Framework. Our estimation problem is formulated in a Bayesian framework, where the objective is to recursively estimate the filtering distribution $p(\mathbf{s}_t|\mathbf{z}_{1:t})$, where $\mathbf{s}_t$ is the state at time t and $\mathbf{z}_{1:t}$ denotes the set of measurements from time 1 to time t. Under standard assumptions that the state sequence is Markovian and the observations are independent given the states, the recursion is given by:

$$p(\mathbf{s}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{s}_t) \int p(\mathbf{s}_t|\mathbf{s}_{t-1}) p(\mathbf{s}_{t-1}|\mathbf{z}_{1:t-1}) d\mathbf{s}_{t-1} \tag{6.6}$$

In non-linear, non-Gaussian cases, it can be solved using sampling approaches, also known as particle filters (PF). The idea behind PF consists of representing the filtering distribution using a set of weighted particles defined as $\{\mathbf{s}_t^n, w_t^n, n = 1, ..., N\}$. In general, given the particle set of the previous time step, configurations of the current step are drawn from a proposal distribution $\mathbf{s}_t^n \sim q(\mathbf{s}|\mathbf{s}_{t-1}^n, \mathbf{z}_t)$. The weights are then computed as:

$$w_t^n \propto w_{t-1}^n \frac{p(\mathbf{z}_t|\mathbf{s}_t) p(\mathbf{s}_t|\mathbf{s}_{t-1}^n)}{q(\mathbf{s}_t|\mathbf{s}_{t-1}^n, \mathbf{z}_t)} \tag{6.7}$$

In this work, we use the Boostrap filter, in which the dynamics are used as proposal. Then, three terms which are defined below are important to define our filter: the state model defining the abstract representation of our object, the dynamical model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ governing the temporal evolution of the state, and the likelihood $p(\mathbf{z}_t|\mathbf{s}_t)$ measuring the adequacy of the observations given our state configuration. Figure 6.6 provides the graphical model of our approach, highlighting the dependency assumptions between variables.

State Space. The state vector is defined as $\mathbf{s}_t = [\mathbf{x}_t, \dot{\mathbf{x}}_t, \theta_t, \alpha_t]^T$, where $\mathbf{x}_t = [x_t, y_t]$ is the body position in the 3D world coordinate frame (on the ground plane), $\dot{\mathbf{x}}_t = [\dot{x}_t, \dot{y}_t]$ is the velocity, $\theta_t (0° \leq \theta_t < 360°)$ is the body orientation angle, $\alpha_t (0° \leq \alpha_t < 360°)$ is the 3D head pan angle. Note that all the elements in the state vector are defined with regard to the 3D world coordinate frame.

Dynamical Model. We use a first-order dynamical model which, given adequate conditional independence assumptions, decomposes as follows:

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}) = p(\mathbf{x}_t, \dot{\mathbf{x}}_t|\mathbf{x}_{t-1}, \dot{\mathbf{x}}_{t-1}) \, p(\theta_t|\theta_{t-1}, \dot{\mathbf{x}}_t) \, p(\alpha_t|\alpha_{t-1}, \theta_t) \tag{6.8}$$

Location dynamics: The first term of Equation 6.8 describes the position and velocity evolution,

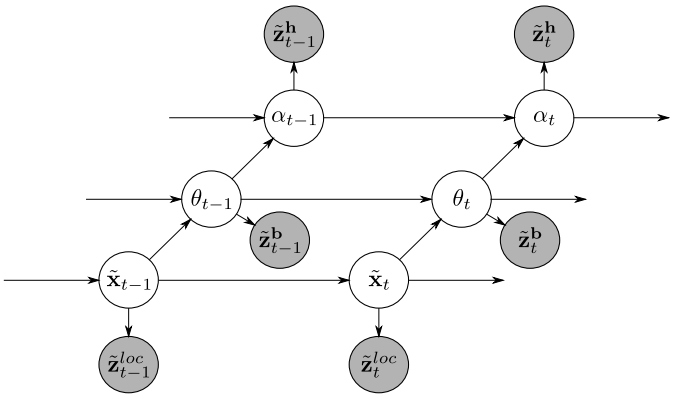


Figure 6.6: Graphical model of the joint temporal filtering approach. State variables are unshaded, and observation variables are shaded. Our model exploits the dependencies between the different cues, as well as their temporal smoothness.

and for this we use a linear dynamical model:

$$p\left(\mathbf{x}_t, \dot{\mathbf{x}}_t | \mathbf{x}_{t-1}, \dot{\mathbf{x}}_{t-1}\right) = \mathcal{N}\left(\tilde{\mathbf{x}}_t; \mathbf{H}\tilde{\mathbf{x}}_{t-1}, \mathbf{Q}_t\right) \tag{6.9}$$

where $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$ denotes the Gaussian probability distribution function (pdf) with mean μ and variance Σ, $\tilde{\mathbf{x}}_t = [\mathbf{x}_t, \dot{\mathbf{x}}_t]^T$ is the composite of position and velocity, $\mathbf{H}$ is the 4×4 transition matrix corresponding to $\mathbf{x}_t = \mathbf{x}_{t-1} + \dot{\mathbf{x}}_{t-1}\delta_t$ (with δ_t the time interval between successive frames), and $\mathbf{Q_t}$ is the system variance.

Body pose dynamics and coupling with motion direction: The second term of Equation 6.8 describes the evolution of body pose over time. It is in turn decomposed as:

$$p\left(\theta_t | \theta_{t-1}, \dot{\mathbf{x}}_t\right) = \mathcal{V}\left(\theta_t; \theta_{t-1}, \kappa_0\right) \mathcal{V}\left(\theta_t; \text{ang}\left(\dot{\mathbf{x}}_t\right), \kappa_{\dot{\mathbf{x}}_t}\right) \tag{6.10}$$

where ang(.) is the angle of the velocity vector (in ground plane), $\mathcal{V}(\theta; \mu, \kappa) \propto e^{\kappa \cos(\theta - \mu)}$ denotes the pdf function of the von Mises distribution over the angle θ parameterized by its mean orientation μ and its concentration parameter κ, and κ_0 is the system concentration parameter for body pose. Equation 6.10 sets two constraints on the dynamics of body pose. The first term expresses that the new body pose at time t should be distributed around the pose at previous time $t-1$. The second term, illustrated in Figure 6.7, imposes that the body orientation should be somewhat aligned with the moving direction of the body. The body pose dependency concentration, $\kappa_{\dot{\mathbf{x}}_t}$, is dependent on the magnitude of the velocity and is defined as:

$$\kappa_{\dot{\mathbf{x}}_t} = \begin{cases} 0, & \text{if } \|\dot{\mathbf{x}}_t\| < \tau \\ \kappa_1 \left(\|\dot{\mathbf{x}}_t\| - \tau\right)^2, & \text{otherwise} \end{cases} \tag{6.11}$$

This means that if the speed is below some threshold τ, then the person is treated as static and the prior on body pose from velocity is completely flat. When the speed is above τ, however, a larger speed introduces a tighter coupling of the body pose around the moving direction.

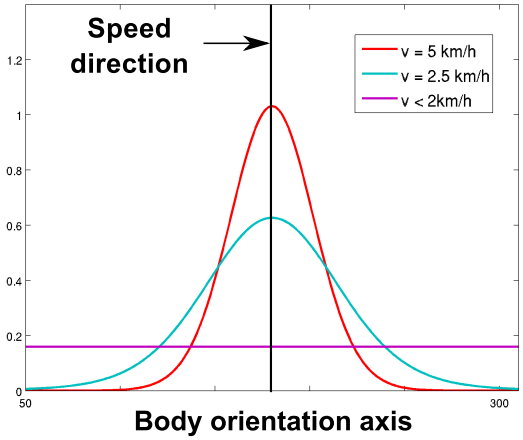


Figure 6.7: Speed conditioned coupling between body pose and motion direction. The higher the speed, the tighter the coupling. When the velocity is below a threshold, the prior on the body pose from motion direction is completely flat.

Note that such coupling had already been used in Robertson et. al. [Robertson and Reid, 2006]. However, in their framework, the coupling was independent of the speed and was thus failing to predict accurate poses at low speed magnitudes. In our framework, the coupling is dependent on the speed: when a person is moving fast, his body orientation is more sharply aligned to the movement direction, whereas the prior on body pose from velocity is completely flat if the speed is below a pre-defined threshold.

Head pose dynamics and coupling with body pose: The third term of Equation 6.8 describes the evolution of the head pose over time. It is decomposed as:

$$p(\alpha_t|\alpha_{t-1},\theta_t) = \mathcal{V}(\alpha_t;\alpha_{t-1},\kappa_1)\,\mathcal{V}(\alpha_t;\theta_t,\kappa_2) \tag{6.12}$$

Similarly to Equation 6.10, Equation 6.12 sets two constraints on the dynamics of head pose. The first term ensures the temporal smoothness of the head pan evolution, wheras the second term models the soft coupling between the head and body orientations. However, in this case, the concentration parameter κ_2 is constant (and lower than κ_1 since the coupling is looser than between the body orientation and motion direction).

Observation Model. Given the human detection bounding box output associated with a human track at time t, the observation at time t is defined as $\mathbf{z}_t = (\mathbf{z}_t^{\text{loc}}, \mathbf{z}_t^{\mathbf{b}}, \mathbf{z}_t^{\mathbf{h}})$, with:

- $\mathbf{z}_t^{\text{loc}} = [u_t, v_t]$, which denotes the bottom-center position of the body bounding box in the image plane.
- $\mathbf{z}_t^{\mathbf{b}}$, the body pose feature.
- $\mathbf{z}_t^{\mathbf{h}}$, the head pose feature.

Under observation conditional independence assumptions, the observation likelihood is given

by:

$$p\left(\mathbf{z}_t|\mathbf{s}_t\right) = p\left(\mathbf{z}_t^{\text{loc}}|\mathbf{s}_t\right) p\left(\mathbf{z}_t^{\mathbf{b}}|\mathbf{s}_t\right) p\left(\mathbf{z}_t^{\mathbf{h}}|\mathbf{s}_t\right) \tag{6.13}$$

where each term is defined as follows. The position likelihood is calculated as:

$$p\left(\mathbf{z}_t^{\text{loc}}|\mathbf{s}_t\right) = \mathcal{N}\left([u_t, v_t]; \mathbf{C}\left(\mathbf{x}_t\right), \Sigma_{\text{loc}}\right) \tag{6.14}$$

where $\mathbf{C}$ is the homography from ground plane to image plane, and Σ_{loc} is the uncertainty of the detected location (in pixels) in the image plane. This term simply expresses that the detected location should be close to the (projected) estimated state.

For the pose observations, we can rely on the likelihood models introduced previously. Since these likelihood models are defined for pose values expressed in the local image frame coordinate system, we first transform the body pose angle θ_t and head pose pan angle α_t from the 3D world coordinate frame to the local image coordinate (note that this depends on the person's position). Then, body and head observation likelihoods are simply defined as the data likelihood given the closest body or head pose class, i.e.:

$$\begin{cases} p\left(\mathbf{z}_t^{\mathbf{b}}|\mathbf{s}_t\right) = p^{\mathbf{b}}\left(\mathbf{z}_t^{\mathbf{b}}|k^{\text{clo}}(\mathbf{s}_t)\right) \\ p\left(\mathbf{z}_t^{\mathbf{h}}|\mathbf{s}_t\right) = p^{\mathbf{h}}\left(\mathbf{z}_t^{\mathbf{h}}|m^{\text{clo}}(\mathbf{s}_t)\right) \end{cases} \tag{6.15}$$

where $k^{\text{clo}}(\mathbf{s}_t)$ returns the body pose class label whose orientation angle is the closest to the (transformed) state orientation $\mathbf{s}_t$ (and similarly for $m^{\text{clo}}(\mathbf{s}_t)$), and $p^{\mathbf{b}}(.)$ and $p^{\mathbf{h}}(.)$ are the body and head likelihoods defined in Sections 6.2.3 and 6.2.4.

6.2.6 Experiments

We tested our joint tracking approach on surveillance videos acquired in a metro station (with head pose models learned from the CHIL data of CLEAR 2007 [Stiefelhagen et al., 2007]). Figure 6.8 is the legend of the illustration, and Figure 6.9 shows some sample results. To save space, the images are cropped and only the region around the active person is shown. We show the human detection bounding boxes as dash rectangles, and the head localization outputs as small solid rectangles. To provide a 3D perception sense, we display two 3D horizontal circles of radius 50cm centered on the bottom-center of the person and the head position, respectively. The body poses and head poses (in 3D space) are shown using radial lines within the circles. More precisely, the body/head poses estimated directly from the feature are shown using radial lines without arrows[4], and the body/head poses returned by the temporal filtering are shown using thicker lines and arrows.

Qualitative Results. Figure 6.9 shows the result on a clip with interaction between two persons. The two persons walk, meet, discuss and then separate. We show the results separately for the

[4]The body/head pose classes with the highest likelihood are shown.

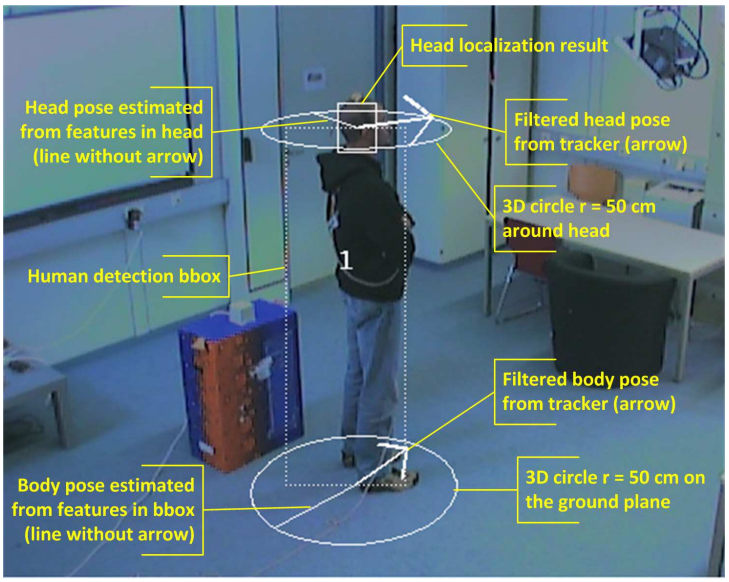


Figure 6.8: Legend for the illustration of results shown in Figures 6.9 and 6.11.

woman (pose estimations shown in green) and the man (pose estimations shown in yellow) in the figure. At $t = 20,220$ there is no result for the man either because he is outside the camera view range or the tracking is lost due to occlusion. Note how our joint filtering approach successfully manages to extract accurate body and head poses from noisy observations, even when people are almost static.

Figure 6.10 illustrates the same video clip as Figure 6.9 in a top-down bird view. Here, the 3D body and head poses can be more easily interpreted and their importance for interaction analysis becomes obvious. Four representative frames are shown. At $t = 60$, both persons are walking with a notable speed, and according to our model, the speed direction provides a good prior for the body pose (and head pose indirectly). At $t = 140$, the persons are talking, with body and head oriented towards each other. In this case, the speed magnitude is very small, resulting in a noisy speed direction which is ignored by our method for the body and pose estimation. At $t = 200$, although the two persons are close to one another, we can still infer that the interaction just stopped because they are not facing each other. At $t = 230$, the two persons have separated.

Quantitative Results. We conducted quantitative evaluation on the CHIL dataset. The dataset contains annotated data for 10 persons (id 6-15) where people in the videos are turning their body and head orientation. For each frame, the ground-truth head poses are provided by a magnetic field location and orientation tracker. We used the person id 6-11 for training the head pose model, and 12-15 for testing. For body pose evaluation, we manually labeled the body orientation of 100 randomly selected frames using a 3D interface.

As performance measure, we use pose accuracy defined as the average error of the pan

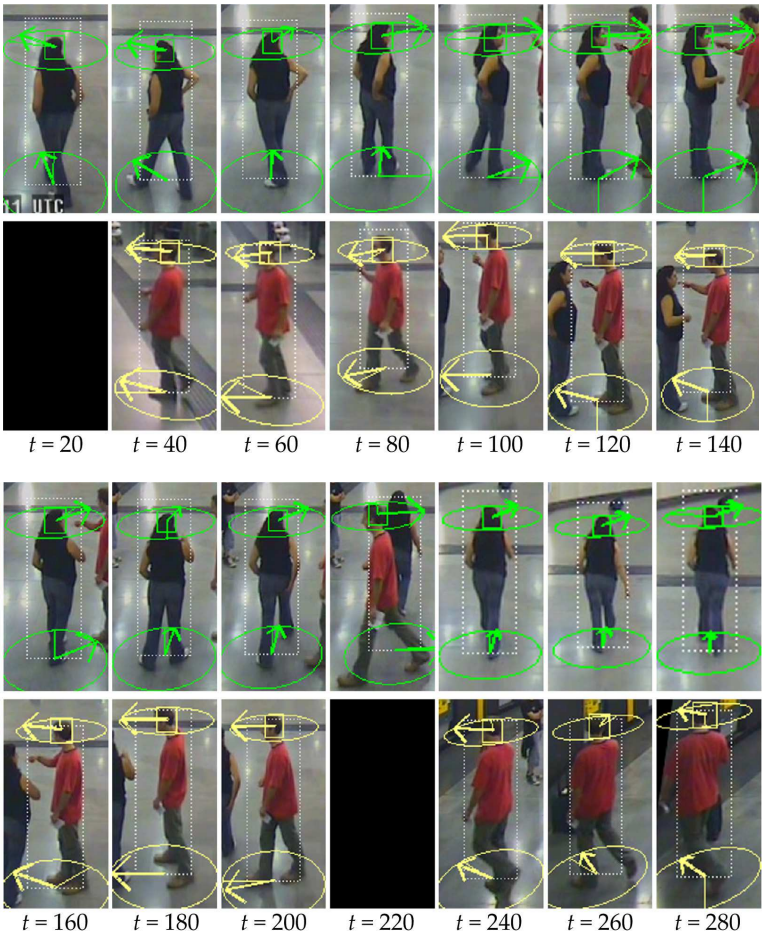


Figure 6.9: Results on a metro station surveillance video with human interaction. Image resolution is 486×363.

angle between the predicted and ground-truth pose angles in the 3D space. To evaluate the effectiveness of our joint tracking approach, we compare our method with the results obtained on a per-frame basis, and with a baseline where body and head pose are filtered separately without exploiting the soft coupling between them (i.e. we have $\kappa_2 = 0$ in Eq. (6.12)). The comparison is depicted in Table 6.2. It can be seen that our joint filtering method significantly outperforms the separate filtering approach, and that the accuracy is quite high given that only one camera is used.

Figure 6.11 illustrates the results on a sample test clip. Here it is straightforward to see the advantage of our approach. By exploiting the soft coupling between body pose and head pose, we can get better accuracy for both. For example, at $t = 600$ and $t = 1200$, the incorrectly estimated head pose is corrected by the body pose. At $t = 1000$, $t = 2000$ and $t = 2600$, the body pose is corrected by the head pose. On the other hand, our soft coupling remains loose enough to still allow some discrepancy between body pose and head pose, which is useful when the head is turning away from the body orientation (e.g. $t = 1400$ and $t = 2600$).

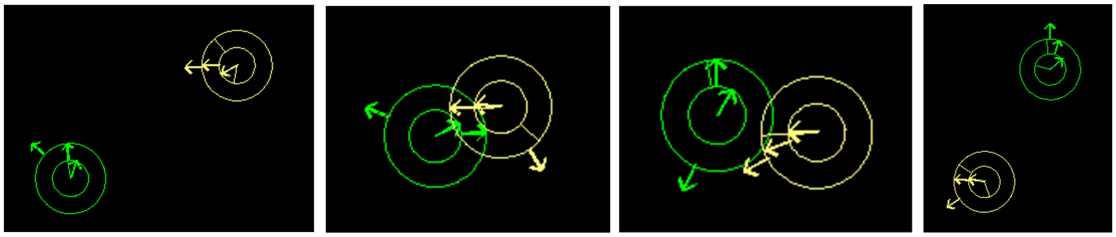


Figure 6.10: Top-down view illustration (same clip as in Figure 6.9). Each person is represented by two circles and three arrows. The arrows indicate (from outer to inner) the tracked body speed, body pose, and head pose.

	per frame	separate filtering	joint filtering
body pose	41.7	32.9	21.9
head pose	30.3	25.8	17.6

Table 6.2: Evaluation on the joint tracking approach. Pose estimation is conducted either on a per-frame basis, or in a filtering framework, without or with coupling between body and head pose. Mean errors in the pan angles are reported in degrees.

6.2.7 Conclusion

We have presented an approach for the joint tracking of behavioral cues in surveillance videos. Given the tracks generated by our detection-based multi-person tracker, we first localize the head and extract body and head pose features. These features are used to jointly estimate the body position, body pose and head pose in 3D space using a particle filtering approach that reliably exploits the conditional coupling between body movement direction and body pose, and the soft coupling between body pose and head pose. We have shown that exploiting temporal coherency and soft coupling improved the estimation of the behavioral cues.

6.3 Domain Adaptation through Manifold Alignment

Until recently, one important limitation of existing pose estimation methods was the use of pre-trained classifiers that were not adapted to the test data, in spite of obvious appearance variabilities in the extracted features. To address this issue, some authors have proposed to perform classifier adaptation [Benfold and Reid, 2011b] [Rajagopal et al., 2013]. Most notably, the framework presented in [Chen and Odobez, 2012] addresses cue coupling, as we did in our temporal filtering approach (see Section 6.2), but also classifier adaptation. One characteristic of their approach is to use manifold information to constrain samples with similar features to be assigned similar pose labels. However, in practice, although the continuity constraint applies to samples coming from both the training and test set (and therefore should help propagating label information to test samples), one can notice that neighbors of training samples often come from the training set and neighbors of test samples come from the test

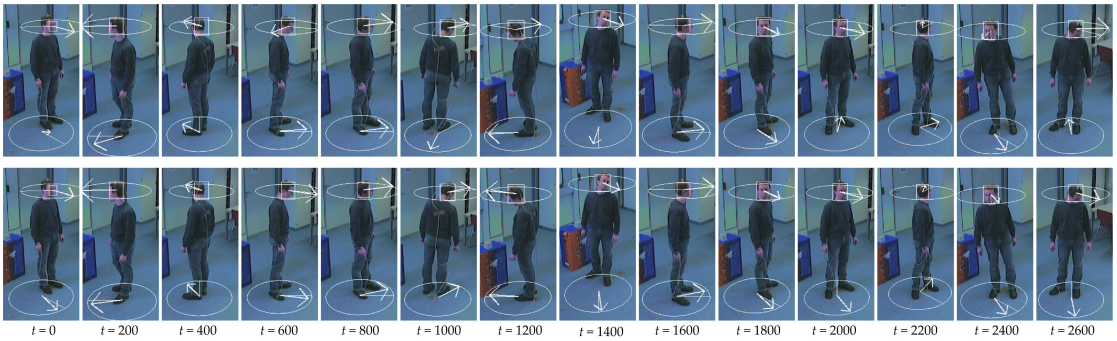


Figure 6.11: Comparison on a CHIL sequence. Image resolution is 640×480. First row: without soft coupling. Second row: our approach.

set. This is mainly due to the fact that HOG features are not semantic descriptors, and that the Euclidean distance in this feature space is subject to important variations due to viewpoint or appearance changes. We propose two ways of addressing this problem:

- by using the more semantic poselet activation descriptors with the hope that the Euclidean distance in this feature space will be more invariant.
- by ensuring that the pose manifolds in feature space between train and test data are aligned, i.e. that samples with the same pose are tightly clustered in feature space.

In the following, we first review the framework of [Chen and Odobez, 2012] in Section 6.3.1. Then, we briefly investigate the use of poselet activation vectors to replace the HOG features in Section 6.3.2. We then present our manifold learning and alignment strategy in Section 6.3.3, with experimental details and results in Section 6.3.4. Finally, Section 6.3.5 summarizes our contributions on domain adaptation for pose estimation.

6.3.1 Baseline Approach

Notations. Let $\mathscr{D}^b = \{(\mathbf{x}_i^b, \mathbf{y}_i^b), i = 1 : n_b\}$ denote the prior labeled dataset for body pose where, $\mathbf{x}_i^b \in \mathbb{R}^{K_b}$ is the feature vector from the body and $\mathbf{y}_i^b$ is its corresponding ground truth label vector. Since we formulate our estimation problem as a multi-class classification problem, the label vector $\mathbf{y}_i^b \in \{0,1\}^{K_l}$ is a one of K_l vector where all but the j^{th} element, $(1 \leq j \leq K_l)$ are zero. A similar treatment applies to our head pose dataset which is indicated by $\mathscr{D}^h = \{(\mathbf{x}_i^h, \mathbf{y}_i^h), i = 1 : n_h\}$. Our target dataset for adaptation is indicated by $\mathscr{D}^t = \{(\tilde{\mathbf{x}}_i^b, \tilde{\mathbf{x}}_i^h, \mathbf{v}_i, u_i), i = 1 : n_t\}$, where $\tilde{\mathbf{x}}_i^b$ and $\tilde{\mathbf{x}}_i^h$ are body and head features respectively, $\mathbf{v}_i \in \{0,1\}^{K_l}$ is the motion direction expressed in the label space, $u_i \in \{0,1\}$ is a binary value indicating if the object motion is fast enough ($\geq$ 3km/h) or otherwise. In the following, we consider 8 classes for both the body and head pose pan angles ($K_l = 8$).

Problem definition. Our goal is to learn a multi-class classifier $f^b : \mathbb{R}^{K_b} \rightarrow \mathbb{R}^8$ for body pose and similarly, f^h for head pose by leveraging various information sources, i.e. labeled and unlabeled data, head-body-motion coupling and the fact that samples with similar pose lie

closeby in the feature manifold. This is achieved by optimizing an objective function E as follows:

$$E = E_l + \alpha E_m + \beta E_c^{bh} + \gamma E_c^{vb} + \lambda E_r \tag{6.16}$$

where the different terms above model the following constraints[5]:

- **Training error term E_l.** The classifier function should have minimum error on labeled training samples $\mathscr{D}^b$ and $\mathscr{D}^h$. This constraint can be encoded by the following function for the body pose:

$$E_l^b = \frac{1}{n_b} \sum_{i=1}^{n_b} \left\| \mathbf{M} f^b(\mathbf{x}_i^b) - \mathbf{M}\mathbf{y}_i^b \right\|_F^2, \tag{6.17}$$

  where M is a label smoothing matrix[6]. Similarly, we obtain the term E_l^h for the head pose by replacing the superscripts b with h. We have $E_l = E_l^b + E_l^h$.

- **Manifold term E_m.** The classifier function should be smooth over the manifold obtained from labeled and unlabeled samples. In other words, samples close by in the HOG feature space should generate labels that are similar too. To achive this, a binary similarity matrix $\mathbf{S}^{b\tilde{b}}$ is constructed by setting $s_{ij}^{b\tilde{b}} = 1$ if $\mathbf{z}_i^b$ is one of the k nearest neighbors of $\mathbf{z}_j^b$ and 0 otherwise. E_m^b is then defined as the violation of this similarity in the output.

$$E_m^b = \frac{1}{\sum_{i \neq j} s_{ij}^{b\tilde{b}}} \sum_{i \neq j} s_{ij}^{b\tilde{b}} \left\| f^b(\mathbf{z}_i^b) - f^b(\mathbf{z}_j^b) \right\|_F^2, \tag{6.18}$$

  Similarly, we get E_m^h for the head pose. We have $E_m = E_m^b + E_m^h$.

- **Body and head coupling term E_c^{bh}.** Generally, people tend to look into the same direction as their body is faced. Also due to anatomical constraints, we can expect that head pose is mostly aligned with the body pose in our datasets.

$$E_c^{bh} = \frac{1}{n_t} \sum_{i=1}^{n_t} \left\| \mathbf{M} f^b\left(\tilde{\mathbf{x}}_i^b\right) - \mathbf{M} f^h\left(\tilde{\mathbf{x}}_i^h\right) \right\|_F^2 \tag{6.19}$$

- **Velocity and body coupling term E_c^{vb}.** It is also observed that when people are moving, their body pose is oriented in the moving direction. Therefore, in the target data $\mathscr{D}^t$, the body pose is constrained to be mostly aligned with the velocity direction, when $u_i = 1$,

[5] In the following, we use $\mathbf{z}_i^b = \mathbf{x}_i^b$ for $i \in [1 : n_b]$ and $\mathbf{z}_i^b = \tilde{\mathbf{x}}_{i-n_b}^b$ for $i \in [n_b + 1 : n_b + n_t]$.

[6] We use $\mathbf{M} = \begin{bmatrix} 11000001 \\ \dots \\ 10000011 \end{bmatrix}$ to *diffuse* the label, posing less penalty on adjacent misclassifications (e.g. classifying "left" as "left-front") than complete mistakes (e.g. classifying "left" as "right").

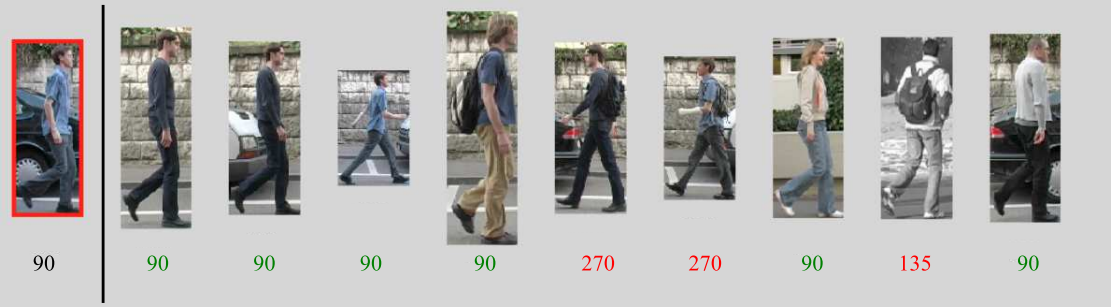


Figure 6.12: Illustration of KNN for a query image in the original HOG feature space. Among the 9 closest neighbors retrieved, 3 have a different pose, which shows that proximity in the feature space does not imply proximity in the label space.

i.e. when speed is reliable.

$$E_c^{vb} = \frac{1}{\sum u_i} \sum_{i=1}^{n_t} u_i \left\| \left(\mathbf{M} f^b \left(\tilde{\mathbf{x}}_i^b \right) - \mathbf{M} \mathbf{v}_i \right) \right\|_F^2 \tag{6.20}$$

Within a kernel-based framework in which features are mapped to a high dimensional space, learning classifiers f^b and f^h reduces to the learning of weight parameters, whose complexity is controlled by the regularization factor E_r. The non-negative parameters α, β, γ and λ control the effect of the constraints. The objective function of Equation 6.16 is then convex and has a closed-form solution. We refer to [Chen and Odobez, 2012] for a more detailed explanation of the model and its optimization.

The model described above benefits from adaptation and coupling which have been shown to positively contribute to pose estimation performance on several datasets. However, one important limitation is that the cost term in Equation 6.18 is used so that the classifier predicts similar labels for samples that lie close by in feature space. However, in practice, we can observe that proximity in the feature space does not imply proximity in the label space, as illustrated in Figure 6.12. In this example, for a query sample, we retrieved its 9 nearest neighbors (NN) in feature space using Euclidean distance. We see that three of the neighbors have different pose angles w.r.t the query sample.

To address this problem, we first tried to adopt a more semantic feature. The approach is described in the next section.

6.3.2 Experiments with Poselet Activation Vectors

We first tried to replace the HOG features in the framework of [Chen and Odobez, 2012] with more semantically meaningful representations. We adopted the poselet activation feature, which is based on the output of poselet classifiers within a human bounding box and that we presented in Chapter 2. This vector indicates to which extent each poselet is present in the observed human image. The motivation behind this choice is that such features, as they are based on higher-level representations of the pose, could be less scene-dependent than

HOG and therefore could lead to a better use of the manifold term for classifier adaptation. Below, we present body and head pose estimation results using 3 different datasets as well as 3 methods.

Datasets. The CHIL dataset [Stiefelhagen et al., 2007] contains videos of static people, rotating around a fixed point and moving the head freely. We consider 4 of these subjects to test the methods. In a surveillance context, we used 3 short clips of a metro station in Turin. The total number of people we consider for pose estimation in these three videos is 4. We also used the TownCentre dataset [Benfold and Reid, 2011b], which is a high resolution video of a busy city street, in which we considered 15 people. For all the datasets, we use the ground truth (head and body pose annotations) provided by [Chen and Odobez, 2012] for evaluation.

The same bounding boxes are used in all the different experiments.

Methods. We investigated the three following methods:

- **Method 1: Regression based on poselet activation vector.** Poselet activation vectors were introduced by [Maji et al., 2011]. The authors trained 1200 poselets on the PASCAL train 2010[7] and the H3D trainval dataset [Bourdev and Malik, 2009]. Poselet activation vectors can be constructed by considering all poselet detections whose predicted bounding box overlaps the bounding box of the person, defined by the intersection over union higher than 0.20. In their paper, the authors propose a regression engine to estimate body and head pose from these vectors. This method does not exploit any coupling and is pre-trained on an external dataset with labeled poses[8].
- **Method 2: Coupled adaptive learning based on HOG features** . This method is the baseline of [Chen and Odobez, 2012] with HOG descriptors as input and the following parameters: $\alpha = 1, \beta = 0.5, \gamma = 0.5, \lambda = 0.01$.
- **Method 3: Coupled adaptive learning based on poselet activation vectors**. The idea of this method is to combine poselet descriptors [Maji et al., 2011] with the coupling and classifier adaptation of [Chen and Odobez, 2012] in order to compare the efficiency of a higher-level representation in the form of poselet activations as opposed to low-level HOG features[9]. The same parameters as in Method 2 are used.

The results, reported as mean angle error, are presented in Table 6.3. The first observation we can make is that the method of [Maji et al., 2011] works the best on the CHIL data, which confirms that poselet-driven approaches can perform well on data with good resolution. In CHIL, the coupling between body pose and motion is not exploited because subjects, even if they turn around, remain static around a single ground plane position. In the surveillance datasets, the 2nd and 3rd methods yield better results as they can reliably exploit the velocity of people to impose coupling. However, the results show that using poselet activation vectors

[7] pascallin.ecs.soton.ac.uk/challenges/VOC/

[8] Training images and annotations for body and head pose are given by [Maji et al., 2011].

[9] Poselet activations are extracted as body features but we keep the HOG descriptors as head features.

	Method 1	Method 2	Method 3
CHIL	**32.5**/**32.8**	37.4/41.5	40.2/43.5
Metro station	38.5/42.6	**28.9**/**29.1**	34.6/33.9
TownCentre	33.1/33.7	18.8/19.2	**17.6/18.3**

Table 6.3: Mean body/head pose pan angle error in degrees. Method 1 is the poselet-driven approach of [Maji et al., 2011], which does not exploit coupling. Method 2 is the coupled, adaptive baseline [Chen and Odobez, 2012] with HOG features as input. Method 3 exploits the same coupling and adaptation as Method 2 but takes poselet activation vectors as input.

within the coupled adaptive framework gives similar pose estimation results than when using HOG descriptors. Therefore, the adaptation procedure does not benefit from using these higher-level features, and the problem that the manifold term uses the Euclidean distance in feature space to assess the similarity of samples still needs to be addressed. This is done in the next subsection.

6.3.3 Semi-Supervised Manifold Biasing and Alignment

Instead of directly resorting to the Euclidean distance in the original HOG feature space to assess the similarity of samples, we propose to first learn more effective manifolds of the high dimensional feature space for the training and target sets. This will ensure that neighboring data points in the manifold have similar pose angles. Another limitation of the baseline method is that it considers the training and target set to share a common manifold structure despite changes in point of view, illumination, perspective and object size. To tackle this issue, we propose to align the training and target manifolds by establishing correspondences between samples in the training set and a subset of the target data using their respective pose labels. To address the two proposed tasks, we adopt a joint manifold learning and embedding procedure.

More precisely, we adopt a graph-based manifold learning approach proposed in [Ham et al., 2005, Wang and Mahadevan, 2009] to learn and align the train and target manifolds. Let $\mathbf{X} = \{\mathbf{x}_1^b, \ldots, \mathbf{x}_{n_b}^b\}$ be our training data and $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_1^b, \ldots, \tilde{\mathbf{x}}_{n_t}^b\}$ be the target data[10]. Let us now define the set of corresponding points for alignment between train and target data by the set of index pairs $I_c = \{(i, j) \mid \mathbf{x}_i^b \in \mathbf{X}\,,\ \tilde{\mathbf{x}}_j^b \in \tilde{\mathbf{X}}$ and $\mathbf{x}_i^b$ is in correspondence with $\tilde{\mathbf{x}}_j^b\}$. In practice, I_c is determined by using pose information of a subset of test samples and finding for each of them the training sample with the most similar pose. The task is then to learn the linear mappings $\mathbf{F}$ and $\tilde{\mathbf{F}}$ from the training and test feature spaces $\mathbf{X}$ and $\tilde{\mathbf{X}}$ into the same embedded space, given the similarity matrices $\mathbf{W}$ and $\tilde{\mathbf{W}}$ defined on the training and test set, respectively. The dual

[10] In the following, we detail the procedure for the body feature. The same procedure is applied for the head feature.

learning and alignment problem is solved by minimizing the cost function:

$$C(F,\tilde{F}) = \mu \sum_{(i,j)\in I_c} ||\mathbf{F}^T\mathbf{x}_i^b - \tilde{\mathbf{F}}^T\tilde{\mathbf{x}}_j^b||^2 + \sum_{i,j} ||\mathbf{F}^T(\mathbf{x}_i^b - \mathbf{x}_j^b)||^2\mathbf{W}_{ij} + \sum_{i,j} ||\tilde{\mathbf{F}}^T(\tilde{\mathbf{x}}_i^b - \tilde{\mathbf{x}}_j^b)||^2\tilde{\mathbf{W}}_{ij} \quad (6.21)$$

where the first term penalizes discrepancies between $\mathbf{F}$ and $\tilde{\mathbf{F}}$ on the corresponding pairs, and the second term imposes smoothness of $\mathbf{F}$ and $\tilde{\mathbf{F}}$ on the respective spaces. Fig. 6.13 illustrates the results of the joint learning and alignment procedure.

The method requires the knowledge of the similarity matrices $\mathbf{W}$ and $\tilde{\mathbf{W}}$ between the data points in $\mathbf{X}$ and $\tilde{\mathbf{X}}$, respectively. KNN with Euclidean distance is typically used to compute the entries of these matrices. In our case, we propose to exploit the label information present in the two datasets to bias the distance between two samples. More precisely, our biased distance $D'(\mathbf{x}_i^b,\mathbf{x}_j^b)$ between two samples $\mathbf{x}_i^b$ and $\mathbf{x}_j^b$ is given by:

$$D'(\mathbf{x}_i^b,\mathbf{x}_j^b) = \left(\tau + \frac{\rho}{1+e^{r-\delta}}\right) D(\mathbf{x}_i^b,\mathbf{x}_j^b) \quad (6.22)$$

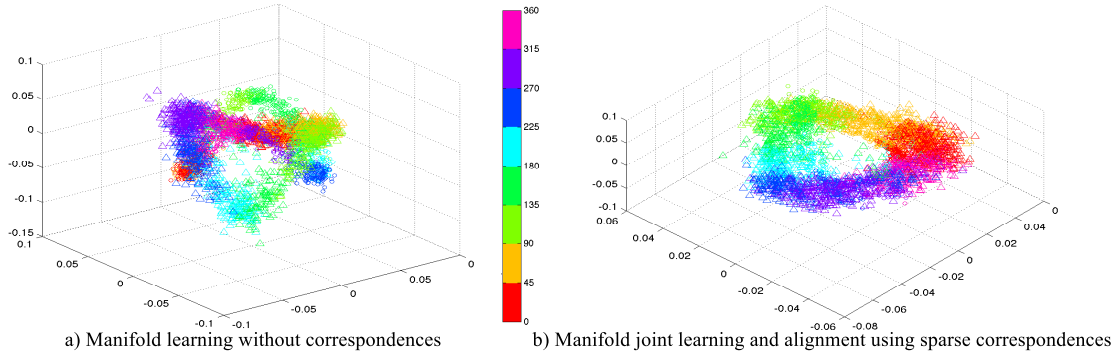
where $D(\mathbf{x}_i^b,\mathbf{x}_j^b)$ is the Euclidean distance in the original feature space and δ is the difference in the pose angles. The bias coefficient, illustrated in Fig. 6.14, is a sigmoid function with parameters τ, ρ and r. We use these parameters to specify the shape of the sigmoid in terms of its upper and lower saturation points, offset, and slope such that the function diminishes the original feature distance when pose differences are within 45° and accentuates this distance when pose differences are more that 45°. In practice, for pose differences more than 90° the distance is doubled. The similarity is then computed based on the biased distance, as a heat kernel parameterized by σ:

$$\mathbf{W}_{ij} = e^{-D'(\mathbf{x}_i^b,\mathbf{x}_j^b)/\sigma} \quad (6.23)$$

Fig. 6.15 (bottom row) shows the positive effect of using the biased distance.

Definition. *We denote by adaptation set the part of the target data that has (weak) labels associated to it and that is used to bias the test manifold and to establish correspondences with the training manifold.*

Method. Within the respective training and test manifold, we compute pairwise distances and use our bias when pose is available. For the test samples without pose information, we propagate labels from neighbors within the adaptation set, by using a simple KNN technique and majority voting scheme, and use the estimated label to compute the bias. We then simultaneously impose smoothness within each manifold and enforce inter-manifold alignment based on sparse correspondences. Figure 6.16 illustrates how the adaptation set is used to establish these sparse correspondences for alignment based on pose distance. Once the projections are learned, we apply them on the features so as to project them on the common manifold where they are aligned and proceed with optimizing Equation 6.16.



a) Manifold learning without correspondences b) Manifold joint learning and alignment using sparse correspondences

Figure 6.13: Illustrations of manifold learning and alignment on CHIL data for the body pose feature. Projection of train and test samples to a 3D subspace with $\mathbf{F}$ and $\tilde{\mathbf{F}}$ learned from Equation 6.21: a) without correspondences ($\mu = 0$); the learned manifolds are not aligned. b) using sparse correspondences; the learned manifolds are aligned. Colors represent the ground-truth pose classes for the pan angle. Triangular symbols represent test samples, circular symbols represent train samples (best viewed in color and zoomed in).

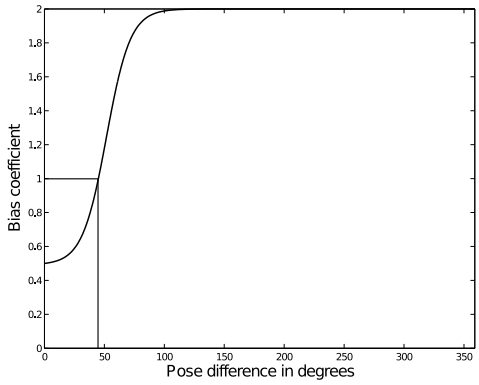


Figure 6.14: Bias coefficient in function of the pose difference. Below 45° of pose difference, the bias reduces the manifold distance, whereas it increases it as the pose angle difference gets larger. The effect of the bias is to bring samples with the same pose closer together, while pushing away samples with different poses.

In the end, in our biased and aligned manifold, samples are tightly clustered in the feature space and in the pose space, such that the different terms of Equation 6.16, especially the manifold term can be exploited more reliably.

6.3.4 Experiments

Datasets. We show the benefits of semi-supervised manifold alignment on pose estimation using two state-of-the-art datasets. We considered 4 subjects from the CHIL dataset for our experiments[11] and 15 pedestrians from the TownCentre surveillance sequence. Similarly to [Chen and Odobez, 2012], we use the TUD Multiview Pedestrians dataset [Andriluka et al.,

[11] We remind that CHIL data contains videos of static people, rotating around a fixed point and moving the head freely.

Figure 6.15: Illustration of KNN for the same query image (highlighted) in original feature space (first row, 2 mistakes) and in the biased manifold (second row, neighbors have the same pose).

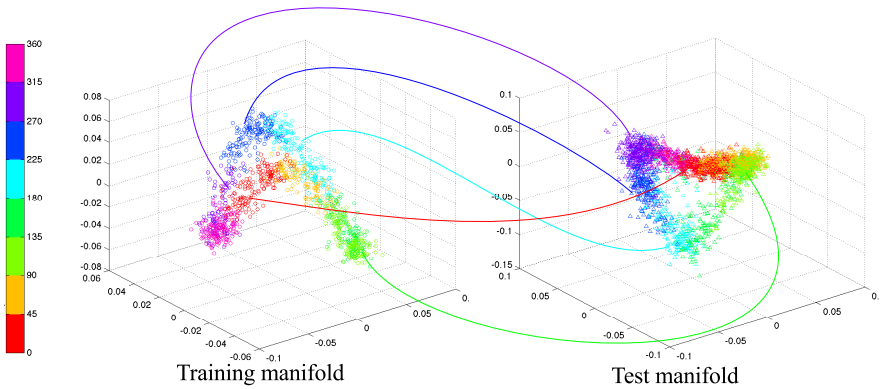


Figure 6.16: Illustration of the use of an adaptation set to build the set of sparse correspondences I_c for alignment, based on pose distance. In this illustration, the training manifold (left) and the test manifold (right) are the same as in Figure 6.13 a) (learned with $\mu = 0$).

2010] and the Benfold dataset [B. Benfold and I. D. Reid, 2009] as the external, labeled datasets for body and head pose features, respectively.

Performance Measure. We report the mean body and head pan angle absolute errors in degrees, w.r.t the ground truth[12].

Procedure. Similarly to [Chen and Odobez, 2012], we extract multi-level HOG features from within body and head bounding boxes obtained from detection, head localization and tracking. The dimensions of each body and head feature is 2268 and 720, respectively. We first apply PCA to reduce the dimension of these features, which will make the following steps faster. We select the principal components so as to keep 90% of the data variance, which leads to new feature dimensions of 445 and 212. Finally, we choose a dimension of 100 for the embedded subspace, for both the body and the head. In the next subsection, we show that partial pose information within the test set (obtained from annotations or weak labels) can be used to align the test features with the ones from the training set in a semi-supervised way, which improves the pose estimation results.

Results. On the CHIL dataset, we partition the data samples in 4 chunks of equal size by taking 1/4 of each track. We then do a 4-fold evaluation, using in turn each data partition as the adaptation set, and the 3 others for testing. Similarly, on the TownCentre dataset, we perform a 3-fold evaluation. The mean performance obtained from the cross-validation sets is reported in Table 6.4. We compare our results with the baseline of [Chen and Odobez, 2012], which was not using any annotations within the test set. Note that in our experiments, we do not use coupling between body pose and velocity ($\gamma = 0$) on the test set, so as to simulate pose estimation on static persons and better evaluate the actual learned and adapted classifiers. This is done for both the baseline [Chen and Odobez, 2012] and our method.

In the CHIL dataset, for each adaptation set we consider, we use the ground truth annotations of these samples to find the alignment with the training manifold. Such an approach can be used in any similar scenario where we can still have access to a few annotated samples within the target data to do the semi-supervised alignment. Table 6.4 shows that partial annotations of the test set (here 25%) can help to gain a significant improvement over the unsupervised baseline.

In most real-world cases however, as manual labelling can be a tedious task, it would be more desirable to avoid such annotations within the target data. For datasets where some people are moving with a reliable velocity, like TownCentre, we propose to use the motion direction of those adaptation samples as weak pose labels[13]. The alignment therefore becomes weakly-supervised and no manual intervention is needed. The last column of Table 6.4 shows

[12] Note that we use the classification scores $\{o_i, i = 1..8\}$ of each class (o_i can be interpreted as classification score for the class angle θ_i) to compute a real-valued angular output using the weighted average vector $\sum_{i=1}^{8} o_i \vec{n}_{\theta_i}$, where $\vec{n}_{\theta_i}$ denotes the unit vector associated with θ_i.

[13] We could not use motion direction as weak labels on CHIL because people remain static around a fixed point and velocity is thus unreliable.

	CHIL	TownCentre
Chen et. al. [Chen and Odobez, 2012]	37.6/41.4	29.0/29.1
Manifold biasing and alignment	**21.3/22.8**	**24.8/23.8**

Table 6.4: Mean body/head pose error in degrees on CHIL and TownCentre datasets. On CHIL, our method uses partial annotations within the test set. On TownCentre, our method uses available motion estimates and does not require manual annotations. Note that on the test folds, the velocity coupling has been set to 0 ($\gamma = 0$) to better evaluate the learned classifiers.

that our biased, weakly-supervised manifold alignment brings an improvement of around 5 degrees on TownCentre.

6.3.5 Conclusion

We presented an improved approach to address classifier adaptation for body and head pose estimation in videos. Our approach leverages on external, labeled training data and some partial labeling information within the test data in the form of some annotations or weak labels from reliable speed direction, when available. The labels of the training set and the (weak) labels within the adaptation set are used to bias and align manifolds. The biasing procedure allows the manifold distance between two samples to be weighted by the difference in the pose angles too. As a result, we learn manifolds where the neighborhood of a sample is constrained to be samples that are closely related in the feature space as well as the pose angle space. Then, the semi-supervised alignment of the training and target manifold compensates for the variabilities between training and test data such that classifier adaptation is better constrained. We have shown that aligning manifolds helps improve the accuracy over an already challenging benchmark for pose estimation.

6.4 Discussion and Conclusion

6.4.1 Computational Aspects

The temporal filtering approach to pose estimation is an online procedure, whereas the classifier adaptation framework works in batch mode. To give an idea of the computational complexity of both methods, we report their average processing times to estimate the body and head pose of one test sample. The filtering approach takes on average 0.6 second to process one sample, while the baseline of [Chen and Odobez, 2012], thanks to an efficient matrix formulation, takes on average 0.1 second per sample. The additional computational cost brought by our joint manifold learning and alignment, which involves the computation of the similarity matrices $\mathbf{W}$, $\tilde{\mathbf{W}}$, and the learning of the linear mappings $\mathbf{F}$, $\tilde{\mathbf{F}}$ depends on the size of the test set, the external training set being fixed in our experiments. On TownCentre, this additional processing time is on average of 0.07 second per test sample.

6.4.2 Conclusion

We have presented two approaches to address body and head pose estimation in videos. Both approaches rely on the output of a multi-person tracker to extract track information, and on velocity information derived from trajectories to be used as prior for body, and indirectly head orientation estimation.

In the first framework, behavioral cues are first extracted on a per-frame basis. Then, temporal coherency within each track is exploited thanks to a temporal filtering framework that enforces soft coupling constraints between the different behavioral cues in the dynamical model. In the second approach, we proposed to address classifier adaptation. We improved an existing state-of-the-art method by better exploiting the structure of the pose features manifolds and leveraging on manifold alignment techniques to bridge the gap between training and test data. We observed that adding temporal filtering on top of this second approach did not benefit the pose estimation results. In fact, within this framework, temporal coherency is implicitly handled by the manifold energy term. Indeed, between successive instances, the features of a person will mostly stay stable and the manifold smoothness constraint will enforce continuity in the estimated pose.

One limitation of the two presented methods is that they do not investigate the issues of wrong estimates due to occlusions (especially for the body part), which can occur frequently in surveillance scenarios. An interesting aspect would be to extend our methods to multi-camera settings in which we could model and resolve ambiguities. Finally, the output of our behavioral cue estimation algorithms could be used at a higher level to model human interactions.

7 Conclusion

The objective of the thesis was to design perceptual algorithms for the estimation of human behavioral cues from videos, which can be useful for many surveillance applications. More precisely, we focused on the challenging tasks of tracking multiple people and estimating their body and head pose. In Sections 7.1.1 and 7.1.2, we remind the challenges of each task and summarize our contributions. In a second part (Section 7.2), we present some perspectives to improve our work.

7.1 Achievements

7.1.1 Multi-Person Tracking

In this thesis, we addressed multi-person tracking in mono-camera situations. This is a challenging problem that is still widely studied and that has not been solved yet. Difficulties arise when people occlude each other or when they wear similar clothes and are hard to distinguish from one another. In this context, we presented a tracking-by-detection algorithm and formulated the tracking task as a statistical labeling of detections. More precisely, we proposed to use the following methodologies:

- We formulated the labeling problem within a CRF framework, modeled at the detection level. Particularly, we considered long-term temporal connectivities between pairs of detections so as to better exploit temporal context.
- We introduced potentials based on pairwise similarities as well as dissimilarities between detection pairs. In that way, the likelihood ratio under the two hypotheses can be seen as a classifier on the links between detections, which leads to better discrimination.
- We used visual motion at the detection level as a cue to guide the labeling. In this way, our motion cues directly come from image measurements and are not derived from hypothetical short-tracks, making them more accurate.
- We introduced higher-order potentials (label costs) to globally constrain the labeling by penalizing unrealistic solutions based on some priors about track duration and

starting/ending locations.

- We proposed an unsupervised way of learning time-sensitive model parameters, which have been shown to be important for the success of our method. Moreover, we have shown these parameters to be scene-dependent. Being unsupervised, our learning procedure can be automatically adapted to any new scene without requiring tedious track annotation.

- The labeling task within our CRF model was shown to be equivalent to an energy minimization problem, for which we introduced an iterative approximate algorithm to find a good solution in reasonable time.

- Extensive experiments on most of the benchmarks in the domain validated the different modeling components of our model.

A lightweight version of our tracking framework was integrated within the demonstrator of the European project VANAHEIM. To allow for live processing, we relied on the Sliding Window optimization only. The integrated tracker receives input in the form of a stream of time-stamped images and time-stamped detections that possibly arrive with a non-fixed framerate[1]. When frames are received with an average framerate between 3 and 5 fps, the tracker module has been shown to have realtime-compatible performances on low or medium crowded scenes (around 10 persons).

7.1.2 Pose Estimation

We also investigated the estimation of body and head pose in videos given the individual trajectories of people in the scene, obtained by tracking. To address this task, we proposed two main approaches:

- We proposed a Bayesian filtering framework solved with a particle filter. The filtering allows to exploit the temporal coherency of the behavioral cues as well as their dependencies. To account for inter-cue dependencies, we introduced a soft coupling constraint between the body pose and the motion direction, conditioned on the speed, and another soft coupling constraint between body and head pose. We demonstrated the good performance of our pose estimation algorithm in real surveillance examples.

- We proposed to study an approach tackling the adaptation of pose classifiers from external labeled pose datasets to the test scene. We used an existing state-of-the-art approach as our baseline and improved it by performing semi- or weakly-supervised manifold alignment prior to pose estimation, so that features from the training and test data are tightly clustered in both pose and feature space. We showed that this alignment procedure can help improve the accuracy over the already challenging baseline for pose estimation.

[1] The visual motion cue was not exploited in the integrated tracker.

7.2 Limitations and Future Work Directions

Although being on par with, or better than recent competing approaches in terms of performance, our algorithms do have shortcomings, that we discuss below. We also give possible future research directions to address these limitations and further improve our approaches.

Online Tracking and Real-Time Compatibility. The full version of our multi-person tracker, i.e. including label costs and Block ICM for optimization works in batch mode and is not real-time. Moreover, unsupervised learning of model parameters is conducted offline in batch mode. To address the first issue, Block ICM could be invoked intermittently to correct the labels within a larger temporal window than the one SW is working on. For the learning part, rather than using the same model parameters for the whole test sequence, unsupervised learning or adaptation of model parameters could be done online by considering detection outputs until the given instant while performing tracking on long videos. To make the computation faster, both in terms of feature extraction and optimization, videos could be processed at a lower framerate[2]. We could as well optimize our code, for instance by selecting appropriate programming languages, resorting to multi-threading, etc.

Discriminative Training. In our tracking framework, we proposed an unsupervised way of learning model parameters characterizing the distribution of pairwise feature distances under two distinct hypotheses. The contrast between these two distributions, given by their log-likelihood ratio, indicates whether the association between a pair of detections should be favored or disfavored. In the future, we could directly learn binary classifiers that best separate pairwise feature distances under the two hypotheses, and the classification scores could be used to weight the confidence about the association or non-association.

Dependence to Detectors. Like all detection-based trackers, the success of our tracking algorithm is dependent on detection accuracy and a high level of misdetections can have a negative impact on its performance. In order to handle the potential high-level of misdetections, short-term forward and/or backward propagations of detections could be generated, similarly to [Benfold and Reid, 2011a] [Yang and Nevatia, 2012b], and directly used as another pairwise association cue. Furthermore, to handle long occlusions, higher order appearance re-identification factor terms potentially relying on online learned discriminative models like [Bak et al., 2012b] could be defined and exploited at another hierarchical level. Detections can also be corrupted due to imprecise localization or overlap with others.

To handle this issue, in addition to the exploitation of reliability factors to handle corrupted features, perspective reasoning as well as finer pixel-level segmentation (e.g. relying on motion [Odobez and Bouthemy, 1995b]) could be used to select only the relevant pixels for computing the appearance and motion descriptors associated with a detection. These

[2] We indeed observed in practice that processing 1 frame out of 5 of videos recorded at 25 fps still yielded good tracking results.

treatments can also be useful for the pose estimation task, as it is sensitive to detection accuracy (imprecise bounding box localization or overlap with an occluder could lead to the extraction of unreliable features).

Other sensors. Both our tracking and pose estimation algorithms could benefit from other sensors. More and more surveillance systems use HD cameras, which can monitor large areas with good resolution. In this case, the dependence of the pose classifiers to location should be investigated, as perspective effects in large scenes might dramatically affect the features according to the location where they are extracted. Larger areas to monitor also usually imply a higher number of people to track, and possibly more crowded scenarios. To this day, most tracking algorithms would only work for medium-crowded situations. Pose estimation in crowds is also a very challenging problem, as usually only the heads of people are (partially) visible. In scenarios of low to medium crowding, multiple cameras could be used to reason about occlusions. Our tracking algorithm could be adapted to such settings, for instance by fusing information from several views, then only extracting visual features from regions that are deemed non-occluded. Our pose estimation algorithms do not take into account the measurement uncertainty of the input features and could also exploit multi-camera environments to resolve ambiguities. Similarly, the depth modality could help occlusion reasoning. However, in the end, algorithms will always depend on crowding, and on sensor placement, as the viewpoint plays an important role in the level of perceived occlusion.

Behavior Models. To improve tracking and better handle crowd and small group moving interactions, high-order dynamical prior model taking into account multiple tracks jointly could be defined like in [Berclaz et al., 2008] and used to constrain the solution space in the global optimization stage. In another direction, we recall that, as was introduced in Chapter 1, the motivation of the thesis was to develop robust techniques for behavioral cues estimation to improve in turn behavior analysis at the higher level. Indeed, as our brief literature review on behavior analysis was further pointing out, extracting mid-level behavioral cues like trajectory and pose information can help understanding higher level semantic concepts, like human-to-human or human-to-environment interactions [Robertson et al., 2007, Ni et al., 2011]. Therefore, it would be interesting to investigate in the future the modeling of human interactions based on the output of our behavioral cue estimation methods, and see whether the level of accuracy is high enough to improve an interaction detection task, for example.

A Complements on Tracker Component Analysis

This Appendix provides additional experimental details on the benefits of:

- Learning parameters from tracklets in Table A.1.
- Having time sensitive models in Table A.2.
- Using an extended temporal connectivity in Table A.3.
- Applying Block ICM optimization after SW in Table A.4.

To empirically show the contribution of each of these components, fragmentations and identity switches are shown on PETS, TUD-Crossing, TUD-Stadtmitte and CAVIAR. In all these experiments, unless stated otherwise, all features (including motion) were used, unsupervised learning of time sensitive parameters from tracklets was applied and followed by SW optimization. Each experiment then consists in varying one particular component while keeping the other components fixed.

Unsupervised learning. In Table A.1, we show the benefit of learning model parameters from tracklets as opposed to learning them from raw detections. Unsupervised learning from tracklets gives better models that allow to significantly reduce the number of fragmentations (except on the rather short TUD-Stadtmitte sequence) while at the same time maintaining or reducing the number of IDS on all datasets.

Time interval sensitivity. In Table A.2, we show the benefit of learning motion and color similarity models that depend on the time gap between detection pairs. The position models are learned normally. On TUD-Crossing, exploiting time-interval dependent models helps reducing the fragmentation, but introduces two switches. Using such time-dependent models on TUD-Stadtmitte keeps the results unchanged. On the longer datasets PETS and CAVIAR, the benefit of similarity models depending on the time gap are better shown as they enable to reduce fragmentation, and to solve 3 IDS on PETS. This confirms the dependencies on Δ observed on the learned β curves.

Temporal context. In Table A.3, we show that extending temporal connectivity is beneficial.

	T_w	*MET*	Frag	IDS
PETS	16	Off	27	0
	16	On	3	0
TUD-Crossing	20	Off	12	3
	20	On	4	3
TUD-Stadtmitte	20	Off	4	2
	20	On	4	1
CAVIAR	20	Off	108	21
	20	On	79	17

Table A.1: Benefit of learning from tracklets as opposed to learning from detections. SW optimization output with models learned from detections (MET = "Off") or from tracklets (MET = "On").

	T_w	*TW*	Frag	IDS
PETS	16	Off	6	3
	16	On	3	0
TUD-Crossing	20	Off	12	1
	20	On	4	3
TUD-Stadtmitte	20	Off	4	1
	20	On	4	1
CAVIAR	20	Off	117	16
	20	On	79	17

Table A.2: Benefit of using time-interval sensitive models. SW optimization output using time-interval sensitive models (TW = "On") or not (TW = "Off") for the color and motion models.

For instance, we can observe on PETS and CAVIAR that increasing temporal connectivity clearly benefits the tracking performance by significantly decreasing the amount of fragmentation. Longer connectivity generates more links, enabling to link detections further apart in time and thus provides more context to assess the labeling.

Block ICM with label costs. Finally, Table A.4 shows that applying Block ICM on the output of SW optimization definitely helps solving ambiguities like IDS, as it is able to break tracks by exploiting long term connectivities and global reasoning within the scene. In the same way, Block ICM is able to merge tracks, thus helping to reduce fragmentations.

	T_w	Frag	IDS
PETS	8	12	0
	16	3	0
TUD-Crossing	10	4	4
	20	4	3
TUD-Stadtmitte	10	6	2
	20	4	1
CAVIAR	20	79	17
	38	64	17

Table A.3: Benefit of larger temporal context. SW optimization output using different temporal connectivities. Using a larger temporal window T_w provides better results.

	T_w	*BlockICM*	Frag	IDS
PETS	16	Off	3	0
	16	On	3	0
TUD-Crossing	20	Off	4	3
	20	On	1	0
TUD-Stadtmitte	20	Off	4	1
	20	On	1	0
CAVIAR	38	Off	64	17
	38	On	38	8

Table A.4: Benefit of Block ICM with label costs ($BlockICM$ = "On") on the output of SW optimization ($BlockICM$ = "Off") .

Bibliography

[Andriluka et al., 2010] Andriluka, M., Roth, S., and Schiele, B. (2010). Monocular 3D pose estimation and tracking by detection. In *CVPR*, pages 623–630.

[Andriyenko et al., 2012] Andriyenko, A., Schindler, K., and Roth, S. (2012). Discrete-continuous optimization for multi-target tracking. In *CVPR*, pages 1926–1933.

[Antonini et al., 2006] Antonini, G., Martinez, S. V., Bierlaire, M., and Thiran, J. P. (2006). Behavioral priors for detection and tracking of pedestrians in video sequences. *INT. J. COMPUT. VIS*, 69(2):159–180.

[Artikis and Paliouras, 2009] Artikis, A. and Paliouras, G. (2009). Behaviour recognition using the event calculus. In *AIAI*, pages 469–478.

[Avidan, 2005] Avidan, S. (2005). Ensemble tracking. In *In CVPR*, pages 494–501.

[B. Benfold and I. D. Reid, 2009] B. Benfold and I. D. Reid (2009). Guiding visual surveillance by tracking human attention. In *BMVC*.

[Bak et al., 2012a] Bak, S., Charpiat, G., Corvée, E., Brémond, F., and Thonnat, M. (2012a). Learning to match appearances by correlations in a covariance metric space. In *ECCV (3)*, volume 7574 of *Lecture Notes in Computer Science*, pages 806–820. Springer.

[Bak et al., 2012b] Bak, S., Chau, D. P., Badie, J., Corvee, E., Bremond, F., and Thonnat, M. (2012b). Multi-target tracking by discriminative analysis on Riemannian manifold. In *ICIP*, pages 1–4.

[Bak et al., 2011] Bak, S., Corvee, E., Bremond, F., and Thonnat, M. (2011). Multiple-shot human re-identification by mean riemannian covariance grid. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 179–184.

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

[Benfold and Reid, 2011b] Benfold, B. and Reid, I. (2011b). Unsupervised learning of a scene-specific coarse gaze estimator. In *ICCV*, pages 2344–2351.

[Berclaz et al., 2008] Berclaz, J., Fleuret, F., and Fua, P. (2008). Multi-camera tracking and atypical motion detection with behavioral maps. In *ECCV*, pages 112–125.

[Berclaz et al., 2009] Berclaz, J., Fleuret, F., and Fua, P. (2009). Multiple object tracking using flow linear programming. In *Winter-PETS*, pages 1–8.

[Bernardin and Stiefelhagen, 2008] Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the CLEAR MOT metrics. *J. Image Video Process.*, 2008:1:1–1:10.

[Beymer and Konolige, 1999] Beymer, D. and Konolige, K. (1999). Real-time tracking of multiple people using continuous detection. *IEEE Frame Rate Workshop*.

[Birchfield, 1998] Birchfield, S. (1998). Elliptical head tracking using intensity gradients and color histograms. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 232–237.

[Blackman, 1986] Blackman, S. (1986). *Multiple-target tracking with radar applications*. Artech House radar library. Artech House.

[Bourdev and Malik, 2009] Bourdev, L. and Malik, J. (2009). Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision (ICCV)*.

[Breitenstein et al., 2011] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. (2011). Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(9):1820–1833.

[Brendel et al., 2011] Brendel, W., Amer, M., and Todorovic, S. (2011). Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280. IEEE.

[Broida and Chellappa, 1986] Broida, T. J. and Chellappa, R. (1986). Estimation of object motion parameters from noisy images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(1):90–99.

[Burkard et al., 2009] Burkard, R. E., Dell'Amico, M., and Martello, S. (2009). *Assignment Problems*. SIAM.

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

[Candamo et al., 2010] Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B., and Kasturi, R. (2010). Understanding Transit Scenes: A Survey on Human Behavior-Recognition Algorithms. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):206–224.

[Chamveha et al., 2011] Chamveha, I., Sugano, Y., Sugimura, D., Siriteerakul, T., Okabe, T., Sato, Y., and Sugimoto, A. (2011). Appearance-based head pose estimation with scene-specific adaptation. In *ICCV Workshops*, pages 1713–1720. IEEE.

[Chau et al., 2013] Chau, D. P., Brémond, F., and Thonnat, M. (2013). Object tracking in videos: Approaches and issues. *CoRR*, abs/1304.5212.

[Chau et al., 2011] Chau, D. P., Brémond, F., and Thonnat, M. (2011). A multi-feature tracking algorithm enabling adaptation to context variations. *CoRR*, abs/1112.1200.

[Chen et al., 2011a] Chen, C., Heili, A., and Odobez, J.-M. (2011a). Combined estimation of location and body pose in surveillance video. In *AVSS*.

[Chen et al., 2011b] Chen, C., Heili, A., and Odobez, J.-M. (2011b). A joint estimation of head and body orientation cues in surveillance video. In *IEEE International Workshop on Socially Intelligent Surveillance and Monitoring*.

[Chen and Odobez, 2012] Chen, C. and Odobez, J.-M. (2012). We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video. In *CVPR*, pages 1544–1551.

[Chen et al., 2011c] Chen, K., Gong, S., and Xiang, T. (2011c). Human pose estimation using structural support vector machines. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 846–851.

[Collins, 2012] Collins, R. T. (2012). Multitarget data association with higher-order motion models. In *CVPR*, pages 1744–1751.

[Comaniciu et al., 2003] Comaniciu, D., Ramesh, V., and Meer, P. (2003). Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577.

[Cristani et al., 2011] Cristani, M., Bazzani, L., Paggetti, G., Fossati, A., Tosato, D., Bue, A. D., Menegaz, G., and Murino, V. (2011). Social interaction discovery by statistical analysis of F-formations. In *BMVC*, pages 1–12.

[Cristani et al., 2010] Cristani, M., Murino, V., and Vinciarelli, A. (2010). Socially intelligent surveillance and monitoring: Analysing dimensions of physical space. In *International Workshop on Socially Intelligent Surveillance and Monitoring (SISM 2010)*.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893.

[Dalal et al., 2006] Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *In European Conference on Computer Vision*. Springer.

[Danafar and Gheissari, 2007] Danafar, S. and Gheissari, N. (2007). Action recognition for surveillance applications using optic flow and svm. In *Proceedings of the 8th Asian conference on Computer vision - Volume Part II*, ACCV'07, pages 457–466, Berlin, Heidelberg. Springer-Verlag.

[Dollár et al., 2009] Dollár, P., Tu, Z., Perona, P., and Belongie, S. (2009). Integral channel features. In *BMVC*.

[Dubout and Fleuret, 2012] Dubout, C. and Fleuret, F. (2012). Exact acceleration of linear object detectors. In *ECCV*, pages 301–311.

[Enzweiler and Gavrila, 2008] Enzweiler, M. and Gavrila, D. (2008). A mixed generative-discriminative framework for pedestrian classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[Farenzena et al., 2010] Farenzena, M., Bazzani, L., Perina, A., Murino, V., and Cristani, M. (2010). Person re-identification by symmetry-driven accumulation of local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2360–2367.

[Felzenszwalb et al., 2010] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645.

[Fieguth and Terzopoulos, 1997] Fieguth, P. and Terzopoulos, D. (1997). Color-based tracking of heads and other mobile objects at video frame rates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 21–27.

[Fleuret et al., 2008] Fleuret, F., Berclaz, J., Lengagne, R., and Fua, P. (2008). Multi-camera people tracking with a probabilistic occupancy map. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):267–282.

[Fortmann et al., 1983] Fortmann, T., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173 – 184.

[Gavrila, 2007] Gavrila, D. M. (2007). A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1408–1421.

[Ge and Collins, 2009] Ge, W. and Collins, R. (2009). Marked point processes for crowd counting. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2913–2920.

[Gong et al., 2010] Gong, S., Xiang, T., and Hongeng, S. (2010). Learning human pose in crowd. In *Proceedings of the 1st ACM International Workshop on Multimodal Pervasive Video Analysis*, MPVA '10, pages 47–52. ACM.

[Green, 1995] Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732.

[Gualdi et al., 2010] Gualdi, G., Prati, A., and Cucchiara, R. (2010). Multi-stage sampling with boosting cascades for pedestrian detection in images and videos. In *Computer Vision – ECCV 2010*, volume 6316 of *Lecture Notes in Computer Science*, pages 196–209. Springer Berlin Heidelberg.

[Ham et al., 2005] Ham, J., Lee, D., and Saul, L. (2005). Semisupervised alignment of manifolds. In *AISTATS*, pages 120–127.

[Heili et al., 2011] Heili, A., Chen, C., and Odobez, J.-M. (2011). Detection-based multi-human tracking using a CRF model. In *ICCV Visual Surveillance Workshop*, pages 1673–1680.

[Heili et al., 2014a] Heili, A., Mendez, A. L., and Odobez, J.-M. (2014a). Exploiting long-term connectivity and visual motion in CRF-based multi-person tracking. *IEEE Transactions on Image Processing*.

[Heili and Odobez, 2013] Heili, A. and Odobez, J.-M. (2013). Parameter estimation and contextual adaptation for a multi-object tracking CRF model. In *PETS*, pages 14–21.

[Heili et al., 2014b] Heili, A., Varadarajan, J., Ghanem, B., Ahuja, N., and Odobez, J.-M. (2014b). Improving head and body pose estimation through semi-supervised manifold alignment. In *IEEE International Conference on Image Processing*.

[Hu et al., 2004] Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34:334–352.

[Huang et al., 2008] Huang, C., Wu, B., and Nevatia, R. (2008). Robust object tracking by hierarchical association of detection responses. In *ECCV*, pages 788–801.

[Hung and Kröse, 2011] Hung, H. and Kröse, B. (2011). Detecting f-formations as dominant sets. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, ICMI '11, pages 231–238. ACM.

[Huttenlocher et al., 1993] Huttenlocher, D., Noh, J., and Rucklidge, W. (1993). Tracking non-rigid objects in complex scenes. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 93–101.

[Isard and Blake, 1998] Isard, M. and Blake, A. (1998). Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28.

[J. Orozco et al., 2009] J. Orozco, S. Gong, and T. Xiang (2009). Head pose classification in crowded scenes. In *BMVC*.

[Jalal and Singh, 2012] Jalal, A. S. and Singh, V. (2012). The state-of-the-art in visual object tracking. *Informatica (Slovenia)*, 36(3):227–248.

[Jones and Snow, 2008] Jones, M. and Snow, D. (2008). Pedestrian detection using boosted features over many frames. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4.

[Kendon, 1977] Kendon, A. (1977). *Studies in the behavior of social interaction*. Studies in semiotics. Indiana University.

[Khan et al., 2003a] Khan, Z., Balch, T., and Dellaert, F. (2003a). An MCMC-based particle filter for tracking multiple interacting targets. In *in Proc. ECCV*, pages 279–290.

[Khan et al., 2003b] Khan, Z., Balch, T. R., and Dellaert, F. (2003b). Efficient particle filter-based tracking of multiple interacting targets using an mrf-based motion model. In *IROS*, pages 254–259. IEEE.

[Kim et al., 2007] Kim, S., Koh, K., Lustig, M., Boyd, S., and Gorinevsky, D. (2007). A method for largescale l_1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, pages 606–617.

[Ko, 2008] Ko, T. (2008). A survey on behavior analysis in video surveillance for homeland security applications. In *Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–8. IEEE Computer Society.

[Kolmogorov and Zabih, 2004] Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81.

[Kuklyte et al., 2009] Kuklyte, J., Kelly, P., Ó Conaire, C., O'Connor, N. E., and Xu, L.-Q. (2009). Anti-social behavior detection in audio-visual surveillance systems. In *PRAI-HBA - Workshop on Pattern Recognition and Artificial Intelligence for Human Behaviour Analysis*.

[Kuo et al., 2010] Kuo, C.-H., Huang, C., and Nevatia, R. (2010). Multi-target tracking by online learned discriminative appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 685–692.

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289. Morgan Kaufmann Publishers Inc.

[Lathoud and Odobez, 2007] Lathoud, G. and Odobez, J.-M. (2007). Short-term spatio-temporal clustering applied to multiple moving speakers. *IEEE Trans. on Audio, Speech, and Language Processing*, 15(5):1696–1710.

[Lee and Cohen, 2004] Lee, M. and Cohen, I. (2004). Human upper body pose estimation in static images. In *Computer Vision - ECCV 2004*, volume 3022 of *Lecture Notes in Computer Science*, pages 126–138. Springer Berlin Heidelberg.

[Li et al., 2012] Li, B., Yao, Q., and Wang, K. (2012). A review on vision-based pedestrian detection in intelligent transportation systems. In *Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on*, pages 393–398.

[Li et al., 2009] Li, Y., Huang, C., and Nevatia, R. (2009). Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, pages 2953–2960.

[Lin and Davis, 2010] Lin, Z. and Davis, L. (2010). Shape-based human detection and segmentation via hierarchical part-template matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):604–618.

[M. J. Jones and J. M. Rehg, 2002] M. J. Jones and J. M. Rehg (2002). Statistical Color Models with Application to Skin Detection. *International Journal of Computer Vision*, 46(1):81–96.

[Maji et al., 2011] Maji, S., Bourdev, L., and Malik, J. (2011). Action recognition from a distributed representation of pose and appearance. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Milan et al., 2013] Milan, A., Schindler, K., and Roth, S. (2013). Challenges of ground truth evaluation of multi-target tracking. In *Proc. of the CVPR 2013 Workshop on Ground Truth - What is a good dataset?*

[Ni et al., 2013] Ni, B., Pei, Y., Liang, Z., Lin, L., and Moulin, P. (2013). Integrating multi-stage depth-induced contextual information for human action recognition and localization. In *FG*, pages 1–8. IEEE.

[Ni et al., 2011] Ni, B., Wang, G., and Moulin, P. (2011). RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *ICCV Workshops*, pages 1147–1153.

[Odobez and Bouthemy, 1995a] Odobez, J. and Bouthemy, P. (1995a). Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348 – 365.

[Odobez et al., 2006] Odobez, J., Gatica-Perez, D., and Ba, S. (2006). Embedding motion in model-based stochastic tracking. *Image Processing, IEEE Transactions on*, 15(11):3514–3530.

[Odobez and Bouthemy, 1995b] Odobez, J.-M. and Bouthemy, P. (1995b). MRF-based motion segmentation exploiting a 2D motion model robust estimation. In *IEEE Int. Conf. on Image Processing*.

[Oliver et al., 2000] Oliver, N. M., Rosario, B., and Pentland, A. (2000). A Bayesian Computer Vision System for Modeling Human Interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843.

[Park and Aggarwal, 2003] Park, S. and Aggarwal, J. K. (2003). Video retrieval of human interactions using model-based motion tracking and multi-layer finite state automata. In *In Lecture Notes in Computer Science: Image and Video Retrieval*. Springer Verlag.

[Park and Trivedi, 2008] Park, S. and Trivedi, M. M. (2008). Understanding human interactions with track and body synergies (TBS) captured from multiple views. *Computer Vision and Image Understanding*, pages 2–20.

[Perez et al., 2004] Perez, P., Vermaak, J., and Blake, A. (2004). Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3):495–513.

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

[Pérez et al., 2002] Pérez, P., Hue, C., Vermaak, J., and Gangnet, M. (2002). Color-based probabilistic tracking. In *ECCV (1)*, volume 2350 of *Lecture Notes in Computer Science*, pages 661–675. Springer.

[Rajagopal et al., 2013] Rajagopal, A., Subramanian, R., Ricci, E., Vieriu, R., Lanz, O., and Sebe, N. (2013). Exploring transfer learning approaches for head pose classification from multi-view surveillance images. In *IJCV*, pages 1–22. Springer.

[Reid, 1979] Reid, D. B. (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24:843–854.

[Remagnino et al., 1997] Remagnino, P., Baumberg, A., Grove, T., Hogg, D., Tan, T., Worrall, A., Baker, K., et al. (1997). An integrated traffic and pedestrian model-based vision system. *Proceedings of the eighth british machine vision conference (BMVC97)*, pages 380–389.

[Robertson et al., 2007] Robertson, N., Reid, I., and Brady, J. (2007). Automatic human behaviour recognition and explanation for cctv video surveillance. *Security Journal*.

[Robertson and Reid, 2006] Robertson, N. M. and Reid, I. D. (2006). Estimating Gaze Direction from Low-Resolution Faces in Video. In *Proceedings of the 2006 European Conference on Computer Vision*.

[Schweitzer et al., 2002] Schweitzer, H., Bell, J., and Wu, F. (2002). Very fast template matching. In *Computer Vision — ECCV 2002*, volume 2353 of *Lecture Notes in Computer Science*, pages 358–372. Springer Berlin Heidelberg.

[Setti et al., 2013] Setti, F., Hung, H., and Cristani, M. (2013). Group detection in still images by f-formation modeling: A comparative study. In *IEEE International Workshop on Image and Audio Analysis for Multimedia Interactive Services*.

[Shafique and Shah, 2005] Shafique, K. and Shah, M. (2005). A noniterative greedy algorithm for multiframe point correspondence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1):51–65.

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600.

[Shitrit et al., 2011] Shitrit, H. B., Berclaz, J., Fleuret, F., and Fua, P. (2011). Tracking multiple people under global appearance constraints. In *ICCV*, pages 137–144.

[Shitrit et al., 2013] Shitrit, H. B., Berclaz, J., Fleuret, F., and Fua, P. (2013). Multi-commodity network flow for tracking multiple people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99:1.

[Smith et al., 2005] Smith, K., Gatica-Perez, D., and Odobez, J.-M. (2005). Using particles to track varying numbers of interacting people. In *CVPR (1)*, pages 962–969. IEEE Computer Society.

[Snidaro et al., 2008] Snidaro, L., Visentini, I., and Foresti, G. (2008). Dynamic models for people detection and tracking. In *Advanced Video and Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on*, pages 29–35.

[Stiefelhagen et al., 2007] Stiefelhagen, R., Bernardin, K., Bowers, R., Rose, R. T., Michel, M., and Garofolo, J. S. (2007). The clear 2007 evaluation. In *CLEAR*, pages 3–34.

[Sutton and McCallum, 2012] Sutton, C. and McCallum, A. (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

[Takahashi et al., 2010] Takahashi, M., Fujii, M., Shibata, M., and Satoh, S. (2010). Robust recognition of specific human behaviors in crowded surveillance video sequences. *EURASIP J. Adv. Signal Process*, 2010:13:1–13:10.

[Terzopoulos and Szeliski, 1993] Terzopoulos, D. and Szeliski, R. (1993). Active vision. chapter Tracking with Kalman Snakes, pages 3–20. MIT Press, Cambridge, MA, USA.

[Tuzel et al., 2006] Tuzel, O., Porikli, F., and Meer, P. (2006). Region covariance: A fast descriptor for detection and classification. In *ECCV (2)*, volume 3952 of *Lecture Notes in Computer Science*, pages 589–600. Springer.

[Tuzel et al., 2008] Tuzel, O., Porikli, F., and Meer, P. (2008). Pedestrian detection via classification on riemannian manifolds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1713–1727.

[Veenman et al., 2001] Veenman, C., Reinders, M. J. T., and Backer, E. (2001). Resolving motion correspondence for densely moving points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(1):54–72.

[Vishwakarma and Agrawal, 2013] Vishwakarma, S. and Agrawal, A. (2013). A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009.

[Walk et al., 2010] Walk, S., Majer, N., Schindler, K., and Schiele, B. (2010). New features and insights for pedestrian detection. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Wang et al., 2012] Wang, C., Liu, B., Vu, H., and Mahadevan, S. (2012). Sparse manifold alignment.

[Wang and Mahadevan, 2009] Wang, C. and Mahadevan, S. (2009). Manifold alignment without correspondence. In *IJCAI*, pages 1273–1278.

[Wang et al., 2006] Wang, Y., Hou, X., and Tan, T. (2006). Recognize multi-people interaction activity by PCA-HMMs. In *Computer Vision – ACCV 2006*.

[Wright et al., 2009] Wright, J., Yang, A., Ganesh, A., Sastry, S., and Ma, Y. (2009). Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227.

[Yan et al., 2009] Yan, S., Wang, H., Fu, Y., Yan, J., Tang, X., and Huang, T. S. (2009). Synchronized submanifold embedding for person-independent pose estimation and beyond. *IEEE Transactions on Image Processing*, 18:202–210.

[Yan et al., 2013] Yan, Y., Ricci, E., Subramanian, R., Lanz, O., and Sebe, N. (2013). No matter where you are: Flexible graph-guided multi-task learning for multi-view head pose classification under target motion. In *ICCV*.

[Yang et al., 2011] Yang, B., Huang, C., and Nevatia, R. (2011). Learning affinities and dependencies for multi-target tracking using a CRF model. In *CVPR*, pages 1233–1240.

[Yang and Nevatia, 2012a] Yang, B. and Nevatia, R. (2012a). An online learned CRF model for multi-target tracking. In *CVPR*, pages 2034–2041.

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

[Yao and Odobez, 2008a] Yao, J. and Odobez, J.-M. (2008a). Fast human detection from videos using covariance features. In *8th European Conference on Computer Vision Visual Surveillance workshop (ECCV-VS)*.

[Yao and Odobez, 2008b] Yao, J. and Odobez, J.-M. (2008b). Multi-camera multi-person 3D space tracking with MCMC in surveillance scenarios. In *European Conference on Computer Vision, workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications (ECCV-M2SFA2)*.

[Yilmaz et al., 2006] Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13.

[Zamir et al., 2012] Zamir, A. R., Dehghan, A., and Shah, M. (2012). GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, pages 343–356.

[Zelniker et al., 2009] Zelniker, E., Hospedales, T. M., Gong, S., and Xiang, T. (2009). A unified bayesian framework for adaptive visual tracking. In *BMVC*.

[Zeng and Ma, 2011] Zeng, C. and Ma, H. (2011). Human detection using multi-camera and 3D scene knowledge. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 1793–1796.

[Zhang et al., 2008] Zhang, L., Li, Y., and Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *CVPR*, pages 1–8.

[Zhang et al., 2013] Zhang, X., Yang, Y.-H., Han, Z., Wang, H., and Gao, C. (2013). Object class detection: A survey. *ACM Comput. Surv.*, 46(1):10:1–10:53.

[Zhao et al., 2013] Zhao, R., Ouyang, W., and Wang, X. (2013). Unsupervised salience learning for person re-identification. In *CVPR*, pages 3586–3593. IEEE.

[Zhao and Nevatia, 2003] Zhao, T. and Nevatia, R. (2003). Bayesian human segmentation in crowded situations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–459–66 vol.2.

[Zheng et al., 2013] Zheng, W.-S., Gong, S., and Xiang, T. (2013). Re-identification by relative distance comparison. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):653–668.

Curriculum Vitae

Alexandre Heili

Personal Information

Born in France, 21/04/1986

email aheili@idiap.ch

website www.idiap.ch/~aheili

professional address Idiap Research Institute

Rue Marconi 19

1920 Martigny, Switzerland

Education

PhD student
2010-Present

École Polytechnique Fédérale de Lausanne (EPFL),

Switzerland

PhD thesis topic: Human Tracking and Pose Estimation in Open Spaces
Doctoral Courses, total 12 ECTS (credit units): Machine Learning, Graphical Models, Computational Perception using Multimodal Sensors

Master of Science
(with honors)
2007–2009

Université de Strasbourg, France

Images, Information Sciences and Technologies
Specialization in Biomedical Engineering

Engineering degree (with honors) 2006-2009 Télécom Physique Strasbourg, Illkirch, France

Computer Science, Electronics, Physics, Signal Processing, Mathematics, Project Management, Robotics

Classes Préparatoires 2004-2006 Lycée Kléber, Strasbourg, France

Post-secondary advanced Mathematics and Physics classes preparing for entrance examinations to the French Grandes Écoles

Experience

Research Assistant 2010 - Present Idiap Research Institute, Switzerland

Research interests: Multi-person tracking, human pose estimation, surveillance data, conditional random fields, data association, particle filters, Monte Carlo techniques, optimization, domain adaptation
Thesis goal: Develop new perceptual algorithms for multi-person tracking and pose estimation in open space scenarios
Advisor: Dr. Jean-Marc ODOBEZ, **Funding**: VANAHEIM project

Research Intern Nov. 2013-Feb. 2014 Advanced Digital Sciences Center (ADSC), Singapore

PhD internship in this joint research center established by the University of Illinois at Urbana-Champaign and the Singapore government agency A*STAR
Subject: Exploring and implementing manifold alignment techniques for domain adaptation in the context of human body and head pose estimation, in MATLAB

Teaching Assistant 1st Term 2012 École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Preparing and correcting assignments, interacting with students for the doctoral course on Computational Perception using Multimodal Sensors

Research Intern 2nd Term 2009 University of Houston, Department of Computer Science, Texas, US

Collaboration with Methodist Hospital, Department of Surgery, Houston
Subject: Mathematical modeling and computational science of a new stress monitoring passive sensor from thermal images, in MATLAB

Research Intern
Summer 2008
Imperial College London, UK

Collaboration with the University College London Psychology department and University of Cardiff Astrophysics group
Subject: Literature survey on interdisciplinary signal recognition: gravitational wave astronomy & automatic speech recognition, implementation of adaptive filters for MRI acoustic noise reduction, in MATLAB

Research Intern
Summer 2007
Observatoire Astronomique de Strasbourg, France

Participation in research activities in Astrophysics
Subject: Literature survey, mathematical modeling and computer simulation of X-ray sources migration in the Milky Way, in C

Academic Honors

Awards

Ranked 3rd out of 70 students graduating in 2009, over the course of the three years of studies, Télécom Physique Strasbourg

Best Paper Award at the PETS Workshop, IEEE Winter Vision Meetings, January 2013

Skills

General

Knowledge in Pattern Recognition and Machine Learning, Computer Vision and Image Processing, Biomedical Engineering, Telecommunications, Physics, Automatic Control

Computer related

Operating systems: Mac OS X, Linux, Windows
Programming languages: Python (used to implement a multi-object tracker for PhD thesis), C/C++, MATLAB
Engineering software: CAE integration platform Altair HyperWorks, from meshing to optimization (HyperMesh, HyperCrash, RADIOSS, HyperView, HyperGraph)
Other: practical experience with OpenCV library, 3D camera calibration, databases and SQL, git, HTML
Desktop applications: Microsoft Office, LaTeX, Inkscape

Languages

French [Mother tongue], English [Proficient user],
German [Independent user], Spanish [Basic knowledge]

Other

Driving licence

Since 2004

Artistic skills

Acoustic and electric guitar, singing, songwriting, recording

Publications

Journal

Alexandre HEILI, Adolfo LÓPEZ-MÉNDEZ, Jean-Marc ODOBEZ, *Exploiting Long-Term Connectivity and Visual Motion in CRF-based Multi-Person Tracking.* In IEEE Transactions on Image Processing, May 2014

Conferences/Workshops

Alexandre HEILI, Jagannadan VARADARAJAN, Bernard GHANEM, Narendra AHUJA, Jean-Marc ODOBEZ, *Improving Head and Body Pose Estimation through Semi-Supervised Manifold Alignment.* In IEEE International Conference on Image Processing, October 2014

Alexandre HEILI, Jean-Marc ODOBEZ, *Parameter Estimation and Contextual Adaptation for a Multi-Object Tracking CRF Model.* In IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS), January 2013, **Best Paper Award**

Cheng CHEN, Alexandre HEILI, Jean-Marc ODOBEZ, *A Joint Estimation of Head and Body Orientation Cues in Surveillance Video.* In IEEE International Workshop on Socially Intelligent Surveillance and Monitoring (ICCV-SISM workshop), November 2011

Alexandre HEILI, Cheng CHEN, Jean-Marc ODOBEZ, *Detection-Based Multi-Human Tracking Using a CRF Model.* In IEEE International Workshop on Visual Surveillance (ICCV-VS workshop), November 2011

Cheng CHEN, Alexandre HEILI, Jean-Marc ODOBEZ, *Combined Estimation of Location and Body Pose in Surveillance Video.* In IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), August 2011