TECHNISCHE
UNIVERSITÄT
DARMSTADT

ENERGY MINIMIZATION FOR MULTIPLE OBJECT TRACKING

Dissertation approved by the
Fachbereich Informatik

in fulfillment of the requirements for the degree of
Doktor-Ingenieur (Dr.-Ing.)

by

ANTON MILAN
Dipl.-Inform.

born in Kiev, Ukraine

Examiner:	Prof. Stefan Roth, PhD
Co-examiner:	Prof. Dr. Konrad Schindler
Co-examiner:	Dr. Ivan Laptev

Date of Submission:	4th of April, 2013
Date of Defense:	16th of May, 2013

Darmstadt, 2014
D17

Anton Milan: *Energy Minimization for Multiple Object Tracking*

ABSTRACT

MULTIPLE target tracking aims at reconstructing trajectories of several moving targets in a dynamic scene, and is of significant relevance for a large number of applications. For example, predicting a pedestrian's action may be employed to warn an inattentive driver and reduce road accidents; understanding a dynamic environment will facilitate autonomous robot navigation; and analyzing crowded scenes can prevent fatalities in mass panics.

The task of multiple target tracking is challenging for various reasons: First of all, visual data is often ambiguous. For example, the objects to be tracked can remain undetected due to low contrast and occlusion. At the same time, background clutter can cause spurious measurements that distract the tracking algorithm. A second challenge arises when multiple measurements appear close to one another. Resolving correspondence ambiguities leads to a combinatorial problem that quickly becomes more complex with every time step. Moreover, a realistic model of multi-target tracking should take physical constraints into account. This is not only important at the level of individual targets but also regarding interactions between them, which adds to the complexity of the problem.

In this work the challenges described above are addressed by means of energy minimization. Given a set of object detections, an energy function describing the problem at hand is minimized with the goal of finding a plausible solution for a batch of consecutive frames. Such offline tracking-by-detection approaches have substantially advanced the performance of multi-target tracking. Building on these ideas, this dissertation introduces three novel techniques for multi-target tracking that extend the state of the art as follows:

The first approach formulates the energy in discrete space, building on the work of Berclaz et al. (2009). All possible target locations are reduced to a regular lattice and tracking is posed as an integer linear program (ILP), enabling (near) global optimality. Unlike prior work, however, the proposed formulation includes a dynamic model and additional constraints that enable performing non-maxima suppression (NMS) at the level of trajectories. These contributions improve the performance both qualitatively and quantitatively with respect to annotated ground truth.

The second technical contribution is a continuous energy function for multiple target tracking that overcomes the limitations imposed by spatial discretization. The continuous formulation is able to capture important aspects of the problem, such as target localization or motion estimation, more accurately. More precisely, the data term as

well as all phenomena including mutual exclusion and occlusion, appearance, dynamics and target persistence are modeled by continuous differentiable functions. The resulting non-convex optimization problem is minimized locally by standard conjugate gradient descent in combination with custom discontinuous jumps. The more accurate representation of the problem leads to a powerful and robust multi-target tracking approach, which shows encouraging results on particularly challenging video sequences.

Both previous methods concentrate on reconstructing trajectories, while disregarding the target-to-measurement assignment problem. To unify both data association and trajectory estimation into a single optimization framework, a discrete-continuous energy is presented in Part III of this dissertation. Leveraging recent advances in discrete optimization (Delong et al., 2012), it is possible to formulate multi-target tracking as a model-fitting approach, where discrete assignments and continuous trajectory representations are combined into a single objective function. To enable efficient optimization, the energy is minimized locally by alternating between the discrete and the continuous set of variables.

The final contribution of this dissertation is an extensive discussion on performance evaluation and comparison of tracking algorithms, which points out important practical issues that ought not be ignored.

ZUSAMMENFASSUNG

Multi-target Tracking beschäftigt sich mit der Problemstellung, mehrere Objekte in einer dynamischen Szene zu verfolgen und ist für eine Vielzahl von Anwendungen relevant. Im Straßenverkehr kann beispielsweise die Absicht eines Fußgängers von einem Fahrzeug aus erkannt werden, um einen unachtsamen Autofahrer zu warnen und somit Verkehrsunfälle zu reduzieren. Ein weiteres Beispiel ist die Navigation autonomer Roboter, die ein Verständnis der dynamischen Umgebung voraussetzt. Schließlich können Todesopfer bei Massenpaniken durch eine automatisierte Analyse von Menschenmassen vermieden werden.

Bei dieser Problemstellung gibt es jedoch zahlreiche Herausforderungen. Zunächst sind visuelle Daten oft mehrdeutig. Beispielsweise können Objekte aufgrund schlechter Kontrastverhältnisse oder bei Verdeckung unerkannt bleiben. Des Weiteren werden durch objektähnliche Strukturen im Hintergrund Fehldetektionen verursacht, die den Trackingalgorithmus stören. Eine zweite Herausforderung entsteht dann, wenn mehrere Messungen nahe beieinander liegen. Das Auflösen der Mehrdeutigkeiten führt zu einem kombinatorischen Problem, dessen Komplexität mit jedem Zeitschritt rasant ansteigt. Zusätzlich sollen physikalische Rahmenbedingungen erfüllt werden, welche sich nicht nur auf einzelne Trajektorien erstrecken, sondern auch auf deren Zusammenspiel.

Diese Dissertation befasst sich mit dem Ansatz der Energieminimierung, um den oben genannten Herausforderungen zu begegnen. Ausgehend von einer Menge an Objektdetektionen wird eine Energiefunktion, welche das vorliegende Problem umschreibt, minimiert, um eine geeignete Lösung für eine vorgegebene Bildsequenz zu finden. Solche Tracking-by-Detection Ansätze haben erheblich zum Fortschritt des Multi-Target-Trackings beigetragen. Diese Arbeit baut auf diesen Grundideen auf und stellt drei neue Methoden vor, die den Stand der Technik wie folgt erweitern:

Der erste Ansatz basiert auf der Arbeit von Berclaz et al. (2009) und formuliert die Energie im diskreten Raum. Die zulässigen Objektpositionen werden dabei auf ein regelmäßiges Gitter beschränkt und die Objektverfolgung wird als ganzzahlige lineare Programmierung formuliert. Im Gegensatz zu früheren Ansätzen beinhaltet die hier vorgestellte Methode ein dynamisches Modell sowie zusätzliche Zwangsbedingungen, die es erlauben, schwächere Hypothesen direkt auf der Ebene der Trajektorien zu unterdrücken. Diese Erweiterungen verbessern die Ergebnisse sowohl qualitativ als auch quantitativ hinsichtlich annotierter Ground-Truth-Daten.

Der zweite technische Beitrag ist eine stetige Energiefunktion, die durch die Diskretisierung entstehende Einschränkungen überwindet. Die kontinuierliche Formulierung kann viele wichtige Aspekte des Multi-Target-Trackings, wie etwa Objektlokalisierung oder Bewegungsschätzung, exakter erfassen. Im Einzelnen werden der Datenterm und Phänomene wie gegenseitige Kollisionen und Verdeckung, das Aussehen, die Dynamik und die Langlebigkeit der Objekte als stetige, differenzierbare Funkionen modelliert. Das daraus resultierende nichtkonvexe Optimierungsproblem wird lokal mittels Verfahren der konjugierten Gradienten in Kombination mit speziell angepassten Sprüngen minimiert. Die sorgfältigere Problembeschreibung stellt ein robustes Verfahren zur Verfolgung mehrerer Objekte dar und zeigt vielversprechende Ergebnisse auf besonders anspruchsvollen Videosequenzen.

Die beiden oben genannten Ansätze fokussieren sich auf die Rekonstruktion der Trajektorien und lassen dabei die Zuweisungsaufgabe außer Acht. Um sowohl das Korrespondenzproblem als auch die Schätzung der Trajektorien in einem Optimierungsproblem zu vereinen, wird im dritten Teil dieser Dissertation eine diskret-kontinuierliche Energie präsentiert. Aktuelle Fortschritte in der diskreten Optimierung (Delong et al., 2012) ermöglichen es, Multi-Target-Tracking auf eine Art zu formulieren, bei der eine diskrete Zuordnung und eine kontinuierliche Repräsentation des Zustands in einer gemeinsamen Zielfunktion vereint werden. Um eine effiziente Optimierung zu ermöglichen, wird die Energie alternierend zwischen den beiden Variablenmengen lokal minimiert.

Im abschließenden Teil werden wichtige Aspekte diskutiert, die beim Evaluieren und beim Vergleich unterschiedlicher Tracking-Methoden auftauchen, und die nicht vernachlässigt werden sollten.

ACKNOWLEDGMENTS

First of all, I would like to thank both my supervisors: Konrad Schindler and Stefan Roth. The circumstances required me to switch the research group during the second year of my PhD. Thanks to the effort from both sides, the transition went smoothly and I could greatly benefit from the joint mentoring and close collaboration. I also thank Ivan Laptev for agreeing to serve as a co-examiner and for his valuable comments. My gratitude also goes to Bernt Schiele and Michael Goesele for providing support and top-class working environments and interaction with their research groups.

It goes without saying that I am very thankful to my office mates, Diane Larlus, Stefan Walk and Uwe Schmidt, as well as to all my colleagues at MIS, ESS, GKmM and GRIS for many fruitful discussions during lunches, coffee breaks, workshops and retreats. A very warm thank you goes to Uschi Paeckel for the constant readiness to instantly manage issues and solve questions of any sort. Moreover, I thank Carola Eichel, Silke Romero and Nils Balke for their administrative and technical support.

I wouldn't have been able to complete my PhD so quickly and efficiently without relying on code provided by my colleagues. Therefore, I would like to thank Stefan Walk for providing his pedestrian detector, which served as basis for all methods developed in this thesis and Christian Wojek for sharing his Kalman filter implementation. Furthermore, I thank Carl Rasmussen, Olga Veksler and Andrew Delong for making their libraries for continuous, respectively discrete optimization publicly available. I am also very grateful to Lauren Uğur, Hsin Nieh, Frank Pagram, Konrad Förstner and Dirk Heiderich for agreeing to proofread parts of my dissertation.

Finally, I truly and deeply appreciate everyone in my life beyond my office. Many thanks go to my parents for their great effort of raising me and teaching me about the importance of education. Thanks to all my friends in Darmstadt for leaving life-long, unforgettable memories. And, clearly, my biggest gratitude goes to my wife and best friend Kinga. Thank you for all the help, advice, encouragement and love throughout all these years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACRONYMS

BPF	boosted particle filter (Okuma et al., 2004)
BP	belief propagation
CCTV	closed-circuit television
CLEAR	Classification of Events, Activities and Relationships
CPU	central processing unit
CRF	conditional random field
DDMCMC	data driven Markov chain Monte Carlo
DP	dynamic programming
DPM	deformable part-based model (Felzenszwalb et al., 2010)
EKF	extended Kalman filter
GLPK	GNU Linear Programming Kit
GMCP	generalized minimum clique problem
GNN	global nearest neighbors
HMM	hidden Markov model
HOG	histogram of oriented gradients (Dalal and Triggs, 2005)
ICM	iterated conditional modes
ICP	iterative closest point
ILP	integer linear program
ISM	implicit shape model (Leibe et al., 2008a)
IP	integer program
JPDAF	joint probabilistic data association filter (Chang and Bar-Shalom, 1984)
KF	Kalman filter
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago
KLT	Kanade-Lucas-Tomasi (Tomasi and Kanade, 1991)
KSP	k-shortest paths
LP	linear program
MAP	maximum a-posteriori
MCMC	Markov chain Monte Carlo
MCMCDA	Markov chain Monte Carlo data association
MDL	minimum description length
MHT	multiple hypothesis tracker
MILP	mixed integer linear program
MOTA	Multiple Object Tracking Accuracy (Bernardin and Stiefelhagen, 2008)
MOTP	Multiple Object Tracking Precision (Bernardin and Stiefelhagen, 2008)

MOT	multiple object tracking
MRF	Markov random field
MWIS	maximum-weight independent set
NMS	non-maxima suppression
PASCAL	Pattern Analysis, Statistical Modeling and Computational Learning
PDA	probabilistic data association
PDAF	probabilistic data association filter
PETS	Performance Evaluation of Tracking and Surveillance
PIRMPT	Person Identity Recognition based Multi-Person Tracking (Kuo and Nevatia, 2011)
PTZ	pan-tilt-zoom
QBP	quadratic boolean program
QPBO	quadratic pseudo-boolean optimization
RANSAC	random sampling consensus
RJMCMC	reversible jump Markov chain Monte Carlo
SCIP	Solving Constraint Integer Programs
SMC	sequential Monte Carlo
SVM	support vector machine
TRW-S	sequential tree-reweighted message passing
UKF	unscented Kalman filter
VATIC	Video Annotation Tool from Irvine, California
VOC	Visual Object Classes

1 INTRODUCTION

No limits, Jonathan? he thought, and he smiled.
His race to learn had begun.

Jonathan Livingston Seagull
Richard Bach

CONTENTS

The rapid advancement in technology has made machines and computers ubiquitous in our everyday lives. With their ever increasing computational capabilities combined with low camera prices, image understanding is now an important part of many applications. *Computer vision*, whose ultimate goal is to design models and develop algorithms that allow computers to perceive and entirely understand the visual world, has thus become a popular research area. Some achievements in computer vision ranging from low-level tasks like image deblurring to seemingly more complex ones like face detection or human pose estimation have been successfully employed in consumer electronics. Nevertheless, human abilities of understanding scenes and interpreting visual information are still superior to current systems. Such high-level tasks include image classification, semantic segmentation and object detection. This dissertation addresses the task of visually following multiple moving objects in a dynamic scene, a task most often referred to as *multiple object tracking*, or equivalently, *multi-target tracking*.

This chapter will introduce the reader to the topic and provide an overview of the entire dissertation. We will first outline the problem in its most general form and present the challenges to be overcome in order to solve it. Before introducing the energy-based approach in

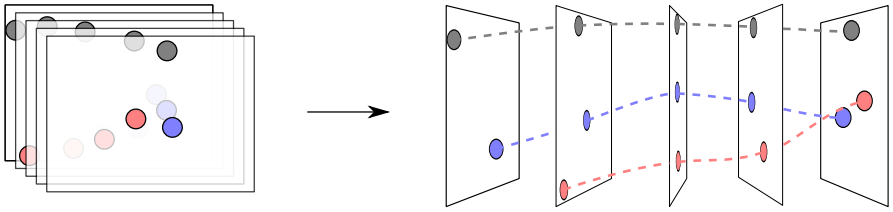


Figure 1.1: A schematic illustration of multi-target tracking. Given a set of video frames, the task is to reconstruct the trajectories of all targets.

Section 1.3, we will motivate the importance of the problem in Section 1.2 on the basis of several examples. Finally, Section 1.4 explicitly states the scientific contribution to the problem at hand and gives a detailed outline of the subsequent chapters.

1.1 THE CHALLENGES OF MULTI-TARGET TRACKING

Visual tracking usually refers to the process of following one single object of interest, *i.e.* inferring its location, in a sequence of video frames. Most methods addressing this task (*e.g.*, Avidan, 2005; Babenko et al., 2009; Kalal et al., 2010; Kwon and Lee, 2011) make two principal assumptions: *i)* the initial location of the target must be known precisely, *e.g.* marked by the user in the first frame, and *ii)* there is exactly one target to be tracked throughout the video. In contrast, in a multi-target tracking setting the number of targets is unknown. Moreover, their number changes over time as targets tend to appear in the field of view and disappear at a later point in time. In addition, a multi-target tracking system is expected to run automatically without manual initialization. One way to define the task at hand is thus:

The target class (e.g. people, cars) is assumed known.

> *Given a video sequence, multi-target tracking equates to precisely reconstructing the trajectory of every single, freely moving target in the scene.*

In other words, the aim is to determine the spatial location and to identify the exact instance of each object of interest in a dynamic scene at every time step. Note that it is not important for now whether the trajectories are described on the image plane or in two-dimensional, respectively three-dimensional world coordinates. This choice is rather application dependent, as is the actual meaning of *target*. Moreover, the notion of a *trajectory* is ambiguous in general and may describe the trajectory of the center of mass, the center of the object's bounding box or any other meaningful point.

What is more important is the information that multi-target tracking provides for scene understanding. Obviously, a trajectory defines the spatial location at any point in time while preserving the unique identifier of a certain object. It also implicitly carries the information about the object's linear velocity (*i.e.* its speed) and acceleration. In

addition, the temporal limits of a trajectory indicate when a target entered and when it exited the scene. Finally, looking at more than one trajectory at a time may provide useful information such as the number of targets at a given time or the interaction between different targets. We will now discuss some of the points that make this task so challenging.

Let us, for now, look at the problem from a probabilistic point of view (we will discuss the relation to energy minimization in more detail later in Section 1.3). One way to approach this problem is to find the most likely state of a predefined model given some observations. This corresponds to computing the maximum a-posteriori (MAP) estimate of the posterior distribution:

MAP estimation is equivalent to energy minimization.

$$\mathbf{X}^{\text{MAP}} = \arg\max_{\mathbf{X}} \; p(\mathbf{X}|\mathbf{I}), \tag{1.1}$$

where $\mathbf{X}$ represents the state, *i.e.* the set of trajectories, and $\mathbf{I}$ is the observed data, *i.e.* a sequence of video frames. If the state is defined in a discrete space, its size usually grows exponentially with the number of frames. Optimization by simple enumeration of all possible combinations is thus not possible in practice. Note that, although the targets are in general assumed to move freely as stated above, a correct formulation of the problem should include dependencies between certain states. In particular, two simultaneous observations that are sufficiently far apart cannot be caused by the same target. Moreover, every observation must explain at most one target. These dependencies arise from the physical constraints that an object cannot be at two places at once and that two objects cannot occupy the same physical space at the same time. In other words, the inference of the problem from Eq. (1.1) amounts to maximizing a function of several variables that are not independent.

The terms observation, detection *and* measurement *are used interchangeably throughout the dissertation.*

Both modeling and inference are complicated even further due to necessarily noisy observations. Not only is there localization uncertainty but also the presence of false alarms caused by clutter and missing evidence due to occlusions or other sources of failure must be taken into account.

It may therefore be somewhat surprising that several formulations claiming to find the globally optimal solution to the problem have been proposed (Jiang et al., 2007; Zhang et al., 2008; Berclaz et al., 2009; Andriyenko and Schindler, 2010; Pirsiavash et al., 2011). However, to achieve this, simplifying assumptions must be made. Perhaps the most common simplification is to significantly restrict the search space. One way is to only regard a short temporal interval at a time, *e.g.* two neighboring frames, and to perform bipartite matching. Another, more common one is to force all objects to only move through a finite set of predefined locations. The resulting combinatorial problems are then formulated such that standard optimization algorithms can be directly applied to find the solution. It is important to note

(a) (b) (c)

(d) (e) (f)

Figure 1.2: Application examples for multi-target tracking. Next to common scenarios involving *(a)* accident prevention, *(b)* surveillance or *(c)* robotics, it can also be used to study animal behavior *(d-e)* (Fletcher et al., 2011) or advance the research in microbiology *(f)* (Keller et al., 2008).

that global optimality, although often achieved in practice, cannot be guaranteed by some of these methods. Nevertheless, theoretical bounds on the optimality may offer valuable information about the quality of the obtained solution.

A different strategy to address the problem is to concentrate on designing more accurate and less restrictive models (Khan et al., 2006; Andriyenko and Schindler, 2011; Andriyenko et al., 2012). While representing trajectories in continuous space enables a more natural description of the problem at hand, it also leads to highly complex optimization problems that can only be solved to local optimality. In the course of this dissertation we will discuss both extremes of the conflict between modeling accuracy and optimization convenience and present three energy functions corresponding to different trade-offs in terms of their domain and their complexity. But before turning to the technical details, let us first motivate the practical importance of multi-target tracking.

1.2 MOTIVATION

Tracking an object over time not only reveals its location at every time step but also allows one to fully reconstruct its trajectory. Based on this information it is possible to analyze the dynamic behavior of an object and to make predictions into the future. Let us briefly motivate why it is important to develop systems that are capable of robustly keeping track of freely moving objects and how it is relevant in science, entertainment and everyday life. The main focus of

this dissertation is on applications related to visual surveillance that are described in Section 1.2.2. One reason is that, next to reducing the number of road accidents, this is probably the most sought-after application, which has a great impact on our lives. In addition, all models presented in this work are designed to analyze a batch of frames at a time leading to offline tracking (*cf*. Section 1.3) and are thus less suitable for time critical application such as driver assistance. Nonetheless, even though all methods are tested on people tracking scenarios, they are by no means limited to this specific target class.

1.2.1 *Road safety*

Although the numbers of road fatalities have been declining over the last decade in developed countries, the risk of being involved in a traffic accident still remains high. In 2010 over 1.3 million people were killed and 50 million people were injured on the roads worldwide according to the latest report by the International Transport Forum (IRT, 2012). Almost all of these accidents were caused by human failure, which is a clear indicator that an advanced technology could help to reduce the road death toll by a large margin. Modern cars are already equipped with high-tech electronics and a range of sensors including radar, sonar and cameras to assist the driver in various situations. Although it may still take several years until completely autonomous cars find their way onto the streets, task-specific systems such as lane control, traffic sign recognition and pedestrian detection are becoming more standard in modern vehicles.

To successfully navigate through the world, it is essential to determine where all the surrounding objects are located and where they are headed. This is exactly where multi-target tracking can be applied. One or multiple cameras that are mounted on a car may capture the surroundings while and a driver assistance system would keep track of nearby cars and pedestrians. Should the vehicle be on a collision course, it will alert the driver about the dangerous situation or even apply the brakes to avoid an accident or at least mitigate its outcome.

1.2.2 *Visual surveillance*

Tracking models developed in this work are best suited for surveillance related applications.

In our modern society we are constantly being watched when moving through public space. The number of surveillance cameras in stores or shopping malls, in train stations or in airports, in parks, on the streets or in parking lots grows constantly. In London, the city with probably the highest concentration of surveillance technology, the number of CCTV cameras is estimated to be close to half a million[1]. Let us for the moment put aside all the privacy issues and the social and ethical aspects associated with CCTV monitoring – these

1 http://www.cctv.co.uk/how-many-cctv-cameras-are-there-in-london

will be thoroughly discussed in section 7.3. It is obvious that manual supervision of all the captured data becomes impossible. As a consequence, the demand for reliable automated people tracking rises.

There are many possible scenarios where observing the motion of pedestrians becomes important. First, let us have a look at crime prevention. Scenes where a person abandons a piece of luggage or a group of people who behave aggressively or pose a danger may be automatically detected. At crowded places like in a metro station or at a soccer stadium a tracking system may detect a mass panic and prevent casualties by automatically opening fire doors or other emergency gates.

People tracking need not always be performed in real-time. Performing offline analysis of large amounts of reconstructed trajectories may be useful for discovering areas where people tend to clump together. This may give new insights on planning escape routes in buildings. Another interesting application from the commercial viewpoint is learning how people move within a grocery store or on a shopping street. Obviously visual surveillance is not limited to monitoring people. A system that is able to track cars on a busy intersection may adjust the traffic lights to enable a smoother flow and prevent traffic jams or simply provide valuable information for long-term traffic development in urban environments.

1.2.3 *Robotics*

'Google's driverless car' is one of the most remarkable achievements in civil robotics.

Research in modern robotics has come a long way from manufacturing helpers to humanoid robots[2]. Although it is still not possible to fully replicate a human being by a machine, autonomous robots are slowly finding their way into our lives. In static environments, simple obstacle detection may be sufficient for navigation. However, in a constantly changing environment it is crucial to keep track of all moving objects to make predictions about the intended motion and avoid collisions. Certain applications, such as personal helper robots for elderly people for instance, will require that the robot identifies and follows one specific person, which makes robust tracking indispensable.

1.2.4 *Life sciences*

ANIMAL BEHAVIOR. Multi-target tracking is also required in many areas of scientific research. For instance, biologists are interested in tracking schools of fish, flocks of birds (Straw et al., 2011) or bats (Betke et al., 2007), ant colonies (Khan et al., 2006; Fletcher et al., 2011) or large groups of fruit flies (Straw et al., 2011; Liu et al., 2012). Such findings may give insight about animal behavior in situations like

2 http://world.honda.com/ASIMO

foraging for food or being attacked by predators. One of the challenges here is the almost identical appearance of all individuals of a certain species, hence, tracking systems must primarily rely on dynamic models to disambiguate between different targets. In addition, the trajectories of flying or swimming targets must be reconstructed in 3D, which necessitates the use of several cameras.

MICROBIOLOGY. Multi-target tracking can even be applied on a molecular level. For example, inferring the precise motion of proteins within a neuron may help us to better understand how our brain works (Al-Bassam et al., 2012). Studying the dynamics and the development of bacteria and other cells is crucial in drug discovery and pathogenesis (Kluepfel, 1993; Xie et al., 2008; Debeir et al., 2005; Lou and Hamprecht, 2011; Kausler et al., 2012). To make tracking possible, microscope videos taken over several hours or days are sped up, which causes poor image quality due to camera shake or accidental defocus. Further challenges include similar appearance, the lack of clearly visible object boundaries and extensive occlusions. Moreover, such tracking algorithms should allow phenomena such as cell division, which are not present in more typical real-world applications.

1.2.5 *Entertainment*

Finally, tracking multiple objects can be used in television and entertainment. Tracking players in soccer or ice hockey gives an objective measure on the physical performance of each player and also allows one to analyze the errors made during the match and develop better playing strategies (Cai et al., 2006; Liu et al., 2009). With powerful handheld devices such as smartphones and tablet computers and soon also wearable head-up displays, the market for augmented reality video games grows quickly. Here, multi-target tracking can be applied to bring real world objects into the game or extend the interface of human-computer interaction.

1.3 ENERGY-BASED MULTI-TARGET TRACKING

As already discussed in Section 1.1, solving the task of tracking multiple targets is not straightforward. After stating the problem and motivating its importance, this section describes the general approach to addressing this challenge that is followed throughout this dissertation. A thorough review of the related literature is presented later in Chapter 2.

Most multi-target approaches follow the so-called *tracking-by-detection* paradigm, which originated several decades ago in the realm of radar tracking (Morefield, 1977; Reid, 1979; Fortmann et al., 1980). In this setting, a radar sweep is executed at discrete time steps providing

each time a set of measurements that approximates the locations of the nearby targets. In visual multi-target tracking the sweeps are replaced by a sequence of frames and a class specific object detector is run on each image to obtain a set of observations. The task of multi-target tracking is then twofold: the first is to determine the source of each observation. A detection may be caused by a target that existed in the previous frame or a newly appearing one, in which case it should be assigned a corresponding ID number of a target that had been visible at some point or be identified as a new target with its own unique ID, respectively. A detection may also be a false alarm caused by background clutter containing no target at all, in which case it should be discarded. This process of uniquely identifying each observation is usually referred to as *data association*. If the detector was flawless at finding all targets and the targets moved slowly relative to the frame rate, then solving the data association problem would be sufficient to determine the location of each target at any point in time. However, object detectors are not perfect in real-world scenarios. The object is usually not localized precisely so that some form of temporal smoothing is required to approximate the natural motion. Moreover, detectors may produce errors, *e.g.* due to clutter or occlusion (*cf.* Section 3.2). A simple interpolation may lead to undesirable effects with implausible or crossing trajectories. In such cases, it is thus necessary to address the second part of the problem: *trajectory estimation*. Its primary objective is to reconstruct the entire trajectory of each target. In practice this means to fill in the gaps where no detections are present but also to adjust the exact course of a trajectory, which tends to deviate from the measurements due to imprecise target localization.

Note that a detector cannot find a completely occluded target.

Tracking-by-detection methods can be coarsely classified into two groups (*cf.* Figure 1.3 *(left, middle)*). The first one includes so-called *non-backscan* or *recursive* methods that follow a (first-order) hidden Markov model (HMM). The state, represented by a probability density function, at any given time is usually estimated only relying on the current observation and on the previous state. Such state estimation techniques are thus often termed *recursive Bayesian estimation* or simply *Bayes filters*. Examples of such online tracking methods include Kalman filters (Kalman, 1960) or particle filters (Doucet et al., 2001). An additional procedure is required to resolve data association, *i.e.* to determine which measurements guide which tracker. Examples of such strategies include, *e.g.*, bipartite graph matching or greedy assignment algorithms.

Online state estimation methods (*e.g.*, Kalman, 1960; Gordon et al., 1993; Breitenstein et al., 2009) that only rely on past observations are desirable for time crucial applications such as pedestrian safety or robot navigation. However, intuitively it seems beneficial to exploit more information for more accurate results. This consideration led

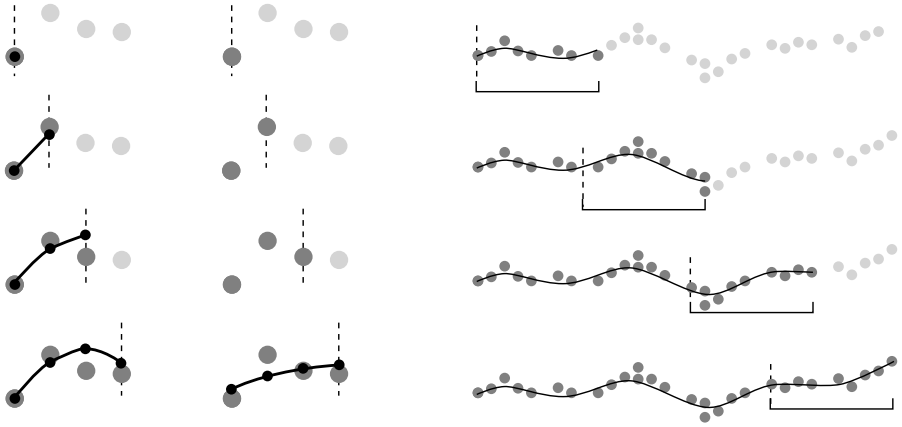


Figure 1.3: Recursive vs. non-recursive state estimation. Recursive filtering methods *(left)* estimate the state (black) one step at a time (dashed line). Batch approaches *(middle)* infer the state for the whole time span at one when all observations (gray) are available. In practice a temporal sliding window is often employed *(right)* to process one batch from a longer sequence at a time. The delay of the output is then equivalent to the length of the temporal window.

to the development of tracking methods that belong to the group of *non-recursive* state estimation, which is also examined in the course of this dissertation. Although batch processing usually requires a time delay between the currently acquired frame and the output of the tracking algorithm, such techniques have proven to be more robust at bridging long-term occlusions because the state for all targets is inferred jointly in a given time window. Moreover, the fact that detections are obtained independently at each time step allows one to avoid irreversible tracker drift. It is important to emphasize one essential distinction between the two approaches. In the former case of recursive state estimation a state prediction step is followed by the state update in the light of new measurements. In contrast, in batch methods all measurements are assumed to be known beforehand and the state at time t depends on the situation before *and* after t. Of course, analyzing extremely long video sequences at a time is neither feasible nor reasonable. Firstly, even if the computational complexity grows linearly with time, there is always a physical memory limit that prevents handling an arbitrarily large amount of input data. Secondly, the current state usually only influences temporally close events and it is thus not necessary to consider situations that are far in the past or in the future. It is therefore common to divide the entire video in several batches and only consider one time window at a time. To ensure consistent trajectories between neighboring temporal windows, a small temporal overlap is accepted where trajectories are constrained to be identical in both solutions.

Non-recursive tracking approaches usually employ the temporal sliding window technique to estimate the state in one batch of frames at a time.

The Boltzmann distribution: $p_i = \frac{1}{Z} e^{-E_i/T}$, where T is the temperature.

In Section 1.1 we posed the task of multi-target tracking as a maximum a-posteriori (MAP) problem that aims at maximizing the posterior over all unknowns of the tracking problem. According to the Boltzmann distribution (Landau and Lifshitz, 1969), the negative logarithm of the probability of a certain state is proportional to the Gibbs energy of the system (Jaynes, 1957). Therefore, MAP estimation can be performed by an inference technique called *energy minimization*. In a nutshell, an energy is a scalar function

$$E : \mathbf{X} \to \mathbb{R} \tag{1.2}$$

that maps every possible state to a real value. The term originally stems from molecular dynamics where the energy describes the entropy of a system of molecules and reaches its minimum at the equilibrium. Many problems in computer vision have also been approached by energy minimization (Mumford and Shah, 1989; Boykov et al., 2001).

Designing an energy function for a specific problem poses two challenges. On the one hand, the energy should describe the problem at hand as accurately as possible, *i.e.* assigning high values to unlikely states and low values to plausible ones, ideally attaining the global minimum at the correct solution. On the other hand, it should be computationally feasible to optimize. In particular, to be globally optimizable, a continuous energy function should be (pseudo-)convex, while a discrete energy should satisfy the submodularity conditions (Kolmogorov and Zabih, 2004). Unfortunately, most real-world computer vision problems including multiple object tracking are too complex to meet these requirements. In this dissertation, various ways of approaching the problem of tracking multiple targets by energy minimization are examined. The individual contributions and the outline of the dissertation are listed in detail in the next section.

1.4 CONTRIBUTIONS AND OUTLINE

CONTRIBUTIONS. The goal of this work is to advance the state-of-the-art in multi-target tracking. To this end, three different types of energies that approach the task from different angles striking the balance between modeling accuracy and the capability of achieving global optimality during optimization are investigated (*cf*. Figure 1.4):

- The first approach (Fig. 1.4(a)) builds on the work of Berclaz et al. (2009) and formulates the tracking problem entirely in a discrete space. Although this results in an ILP which is hard to optimize, a linear program relaxation simplifies the problem such that it can be solved efficiently to (near) global optimality.

  The idea of applying 0-1 integer programming to solve the problem of multi-target tracking itself is not new. However, the ap-

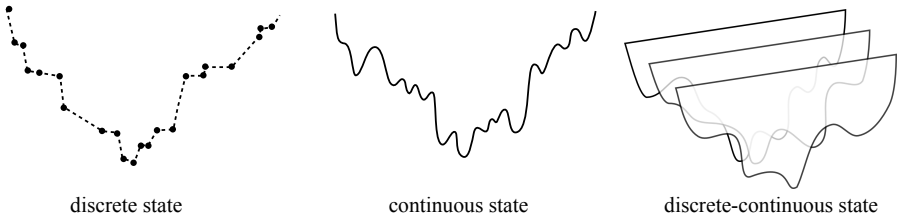


Figure 1.4: Three types of energy functions are investigated in this dissertation. A purely discrete approach *(left)*, a continuous formulation *(middle)* and a discrete-continuous optimization method *(right)*.

proach described in Chapter 4 makes several contributions to this line of thought. In particular, a dynamic model is introduced into grid-based people tracking. To enable this, we propose to discretize the grid in a triaxial way to reduce the effect of aliasing. Finally, we introduce additional constraints to the objective function to shift non-maxima suppression to the level of trajectories, thus allowing for more uncertainty in the likelihood. As a result, the inferred trajectories exhibit a smoother, more natural shape while at the same time producing fewer association errors.

- To surpass any restrictions on the state space we propose a global optimization method that is formulated entirely in continuous space (*cf*. Figure 1.4(b)). The main motivation for this approach was to develop a fairly accurate model that captures all important aspects of multi-target tracking. To this end, trajectories are inferred entirely in continuous space by optimizing a high-dimensional energy function using conjugate gradient descent. To allow a variable number of targets, the optimization is extended by a set of jump moves that help to better explore the energy in various dimensionalities.

  The model includes an observation likelihood and several physically motivated priors, such as the target's dynamics, exclusion and trajectory persistence. In addition, we design a global occlusion model that is seamlessly integrated into the continuous representation by modeling the target occupancy by two-dimensional Gaussians. Similarly, we model the target appearance in the continuous domain to easily fit the framework. Despite the highly non-convex behavior of the resulting energy function we show that our optimization scheme is able to find good local optima. This claim is supported by state-of-the-art performance on several challenging datasets.

- Both the purely discrete energy formulation as well as the continuous one only deal with the task of trajectory estimation because the state is completely described by target locations only. The data association in the classical sense, *i.e*. the assignment of

targets to detections, is solved implicitly and is not directly included in the optimization. On the contrary, our third approach combines both aspects of the multi-target tracking problem into one consistent energy. The state is represented by both discrete and continuous variables. The motivation behind this approach is to tackle both data association *and* trajectory estimation in one unified framework. To achieve this, the first challenge is posed as a graphical model where the solution to a multi-labeling problem is found either by graph cut based optimization or by message passing algorithms, depending on the complexity of the exact formulation. Next to the purely discrete problem of data association, all target trajectories are represented by piecewise cubic splines in continuous space. Through this combination, it is possible to account for both aspects within a single optimization framework. To locally minimize the discrete-continuous energy (*cf*. Figure 1.4(c)), an alternating optimization procedure is employed where one set of variables is being fixed while the other one is being optimized.

- Finally, we present a detailed discussion on the evaluation of multiple target tracking approaches. Although a thorough experimental validation of any method is required in most scientific work, an objective evaluation or comparison of a certain multi-target tracking approach is not straightforward. The reasons for this are varied, ranging from limited available data over ambiguities in ground truth and in evaluation protocols to the strong dependence on the object detector.

OUTLINE. The remainder of this dissertation is structured as follows. Chapter 2 reviews the previous work on multiple object tracking.

The main technical contribution is divided into three parts. Part I (Chapter 4) deals with the discrete energy formulation and renders our contribution to the ILP formulation for multi-target tracking. The technical part of this chapter was previously published in (Andriyenko and Schindler, 2010). Part II (Chapter 5) presents our entire continuous energy framework including global occlusion reasoning and an appearance model. This chapter is mostly based on (Andriyenko and Schindler, 2011; Andriyenko et al., 2011). Part III then goes on to combine both sides of the problem, namely data association and trajectory estimation in a unified discrete-continuous optimization framework. Chapter 6 first presents the main idea of the approach with a simplified energy that can be minimized by graph cuts and a closed-form least-squares solution. This approach was partially presented in (Andriyenko et al., 2012). It then goes on to introduce more sophisticated components such as exclusion on the level of detections to ensure plausible interpretation of the data. We also present a statistical anal-

ysis of various energy components based on annotated data. Finally, we extend the initial label cost formulation by proposing a pairwise label cost. Note that such a graphical model can easily be used for solving other problems that involve a variable number of labels with mutual interaction. This extension of the discrete-continuous energy formulation appeared in (Milan et al., 2013b).

In Chapter 7 we will discuss several issues that go beyond the purely technical contribution of this work concerning more practical aspects that we faced. In particular, we will deal with the problem related to objectively evaluating and comparing multi-target tracking methods. Parts of this chapter appeared in (Milan et al., 2013a). Furthermore, we will point out numerical issues that affect iterative optimization methods presented in Chapters 5 and 6. Moreover, we will discuss some of the known or potential issues that may arise with this kind of technology.

The final chapter summarizes the contributions that were developed in this dissertation and presents a discussion on the relevance and role of each individual approach as well as an outlook for possible future research direction.

2

BACKGROUND AND RELATED WORK

If I have seen further than others, it is by standing upon the shoulders of giants.

ISAAC NEWTON

CONTENTS

THE body of literature dealing with the problem of tracking multiple targets is enormous. Still, tracking and motion estimation continue to be very active research topics. Currently, around 30 new articles presenting novel tracking methods or improving existing ones appear at each major computer vision conference.[1] It is thus not feasible to give a complete review of all work on this topic within the limits of this dissertation. In this chapter we will first look at the task from a different point of view, namely in the context of perception and cognition and briefly review some of the work on human tracking abilities. We will then look at the most important milestones in multiple target tracking, in particular some of the first established data association techniques. Finally, we will provide a thorough overview of the important work closely related to visual multi-target tracking, in particular people tracking. Please note that this chapter provides a general review of the related work. A short discussion of the methods that are relevant for each of our proposed strategies will be presented at the beginning of the respective chapter. Also keep in mind that many of the mentioned strategies overlap at one point or another and it is therefore impossible to clearly categorize each approach using one single keyword. The grouping is performed

1 http://www.cvpr2011.org/statistics

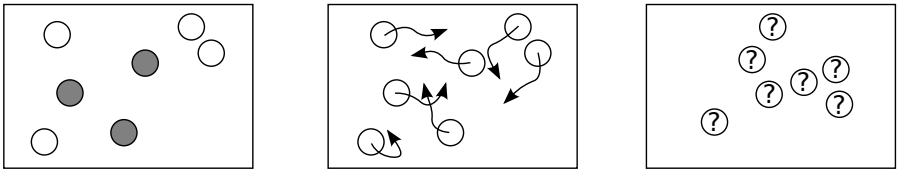


Figure 2.1: In a typical multi-object tracking trial, the subject is shown a number of targets (gray) along with some distractors (white). Then all targets begin to move simultaneously. After a few seconds the subject is asked to identify the targets.

based on the main underlying concept of each method, in particular the nature of the state space representation and its inference.

2.1 TRACKING IN HUMAN PERCEPTION

Before getting into the realm of computer algorithms and models aiming at solving the problem, let us first briefly turn to biological approaches. Tracking moving objects is an ability that is present in almost all living things that posses a visual sense. Although the exact functionality of the human brain is not yet fully understood and we cannot yet precisely point out how the tracking mechanism works on the micro-scale, it is interesting to investigate the question on the limitations of this skill. Research in neurophysiology and cognition studies human attention and perception capabilities. In particular, one is interested in finding out how many objects a human is able to track correctly. This information may be useful for certain professions such as air traffic controllers. To answer this question researchers conduct what is called a multiple object trial (Pylyshyn and Storm, 1988). A test subject observes a number of moving objects having identical appearances for a short period of time and is then asked to identify the targets that had been highlighted in the beginning. This procedure is illustrated in Figure 2.1. Although the maximal number of targets that a human can track reliably strongly depends on the speed and the proximity of targets, several studies suggest that in a reasonable setting this number lies somewhere between four and eight targets (Pylyshyn and Storm, 1988; Alvarez and Franconeri, 2007). In such tests the subject must be fully attentive at all times because all objects look identical and there is thus no possibility of re-identification by matching the appearance.

The efficiency of computer based tracking methods has steadily improved over the recent years. Current approaches reach over 90% accuracy on reasonably crowded datasets showing up to eight pedestrians (Henriques et al., 2011; Andriyenko et al., 2012), thereby achieving results similar to human performance. In more difficult cases with dense crowds, the accuracy of automated systems drops significantly. However, in such scenarios, humans need to considerably

slow down the video to correctly identify all targets by exploiting very subtle cues of motion and appearance. For annotation purposes each frame is examined thoroughly, and possibly several times, by jogging through the sequence back and forth. It is thus reasonable to state that modern computer systems already outperform human capabilities in certain settings when it comes to real-time multi-target tracking tasks.

2.2 RADAR AND SONAR TRACKING

Much of the early work on multi-target tracking originated from air traffic or naval related applications (Morefield, 1977; Reid, 1979; Fortmann et al., 1980). In those early days, usually only radar or sonar sensors were employed to obtain the measurements. The difficulties arising from imperfect measurements such as imprecise object localization or false alarms due to clutter were known and integrated in those approaches. There are, however, two main differences with respect to the data acquired and used in computer vision applications. On the one hand, visual data provides additional information about the appearance of the object. Various cues such as color, texture, shape or size of an object may be used to disambiguate it from other targets or to make the tracking more robust by directly incorporating them into the target's state. On the other hand, radar or sonar devices only provide information about the presence or the absence of an object at a certain position. On the contrary, most object detectors output a scalar value that in one way or another correlates to the certainty about the presence of the object. Although it is not trivial to correctly map this scoring value to the actual probability, it can still be a helpful cue to guide a tracker.

2.3 GUIDED FILTERS

Tracking a target encompasses state estimation from noisy measurements. Probably the most popular approach to this problem was proposed by Kalman (1960) in the sixties and is widely known as the *Kalman filter* (KF). In its native form, given a sequence of measurements, the Kalman filter estimates the optimal state of a system in a least squares sense and under certain assumptions. In particular, the Kalman filter assumes a linear dependency for both the state transition and the observation model. Additionally, zero mean Gaussian noise can be directly integrated into both components. The prediction and update equations can then be derived to solve the problem in closed form. To relax the linearity constraint, extensions such as the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) (Julier and Uhlmann, 1997) have been proposed. The demand for non-Gaussian models later led to the development of stochastic recursive

state estimation techniques called *particle filters* (Gordon et al., 1993; Isard and Blake, 1998; Doucet et al., 2000). Here, the state is represented by a set of particles (or samples) allowing arbitrary multi-modal distributions which are approximated by sequential Monte Carlo (SMC) importance sampling.

Although all of the above filtering techniques are intrinsically designed to handle one dynamic system, *i.e.* one target, they are often employed in multiple-target tracking. To achieve this, the state of each target is estimated independently based only on those observations that are believed to have originated from the same target. In other words, the data association problem is usually solved in a separate procedure. To this end, many different approaches have been proposed, most of which include a so-called *gating* mechanism (Fortmann et al., 1980), motivated by the assumption that the objects' maximum speed is limited by physical constraints. Consequently, targets cannot move arbitrarily far between adjacent scans. This technique allows a significant reduction of the search space, by only considering measurements within a certain distance from the targets for the assignment problem. The distance is usually determined by the estimated uncertainty of the current prediction.

Gating reduces the search space to a small area around each target.

In the following we will give a coarse overview of some of the more popular data association methods and briefly outline their core principles. For a more thorough comparison and discussion as well as formal derivations, please refer to Bar-Shalom and Fortmann (1988); Cox (1993); Blackman and Popoli (1999).

GLOBAL NEAREST NEIGHBORS. One of the simplest data association techniques is global nearest neighbors (GNN) (Blackman and Popoli, 1999). Here, at each incoming data scan, the association hypothesis with the highest probability for all targets is kept and all the other ones are discarded. A similar method was employed by Deriche and Faugeras (1990) to associate and track line segments extracted from object edges in a corridor environment. Obviously, this naive method cannot be globally optimal and performs quite poorly, especially when targets come close to one another or in presence of false alarms or missing detections.

In this context, hypothesis *denotes a feasible measurement-to-targets assignment.*

MULTIPLE HYPOTHESIS TRACKER. An obvious drawback of GNN is that all information about previous measurements is discarded as soon as the current time step has been processed. To remedy this, Reid (1979) proposed a more sophisticated data association method: the multiple hypothesis tracker (MHT). The idea is to keep *all* possible association events from the past observations in memory and to choose the best one at each time step. The motivation behind this exhaustive search algorithm is that any current ambiguity will be resolved at a later point in light of new evidence. This formulation pro-

vides an optimal solution of the objective in theory, but is not feasible in practice because the number of hypotheses grows exponentially with time. A pruning strategy is therefore required. To only keep a finite set of promising hypotheses, several alternatives exist: *i)* only maintain a maximum number of the most probable solutions, *ii)* only maintain solutions above a specified confidence value, or *iii)* discard all hypotheses that reach longer than n scans into the past. Even with these pruning techniques, the MHT algorithm is rarely used today due to its poor complexity behavior in challenging situations with large numbers of targets. Note that although the data is always processed online after each new set of measurements, the MHT belongs to a class of backscan tracking or deferred logic techniques where the final result is only obtained with a fixed temporal delay.

PROBABILISTIC DATA ASSOCIATION. A further class of techniques is called probabilistic data association (PDA). As the name suggests, probabilities for various sources of origin for each detection are accumulated and propagated through time. In the simple case with only one target, the probabilistic data association filter (PDAF), originally proposed by Bar-Shalom and Jaffer (1972), computes and assigns a probability that a certain measurement arose from that target or from clutter. In contrast to the global nearest neighbors (GNN) approach, the probabilistic assignment can much better deal with missing evidence and false alarms. The extension to multiple targets was later presented by Fortmann et al. (1983). The joint probabilistic data association filter (JPDAF) follows a similar principle as the PDAF but assigns an array of values to each measurement where each one corresponds to the probability that a particular target, respectively background clutter gave rise to that measurement. One serious limitation of the classical JPDAF is that the number of targets needs to be known in advance. To resolve this, an extension that allows track splitting was later proposed by Bar-Shalom et al. (1991). Although these probabilistic approaches bypass the exponential complexity of the MHT, they only provide suboptimal solutions to the problem because all association hypotheses are entirely summarized in the current time step, *i.e.* the temporal observation horizon is reduced to one frame. Considering all possible hypotheses to ensure global optimality would still lead to intractable inference.

One advantage of such probabilistic models is that their formulation allows one to include quantities like the false alarm rate of the detector or the true target density. It is thus possible to make predictions about the expected performance of a tracking system given a certain scenario. Although this information may seem quite valuable from a theoretical point of view, such specifications are often unreliable or can only be roughly estimated and are thus not always applicable in practice.

PARTICLE FILTERS. In most real world situations, the posterior probability distribution in tracking problems exhibits a strongly multi-modal behavior. To accurately approximate its shape and propagate it through time, Isard and Blake (1998) use a set of weighted samples, which are updated upon new evidence. This conditional density propagation, or *condensation*, was one of the first algorithms for visual tracking based on particle filters. Later, Vermaak et al. (2003) proposed a mixture model, where the multi-modal distribution over all targets is approximated by a certain number of components. This prevents undesired merging of neighboring modes due to ambiguity. However, special care must be taken to correctly estimate the number of mixture components that are assumed to correspond to individual targets. The work of Okuma et al. (2004) builds on the same idea of mixture particle filters, but additionally employs AdaBoost to learn target specific appearance models. The resulting boosted particle filter (BPF) is successfully applied to track ice hockey players in a short clip.

Breitenstein et al. (2009) set aside the complex association problem, concentrating on carefully designing a robust particle filter that learns the appearance of each target online. To solve data association they use a simple greedy strategy to select which detection should guide which filter. In addition to a classic tracking-by-detection approach they also exploit the intermediate detector output before applying non-maxima suppression. A basic occlusion reasoning handles situations when a detector fails due to close proximity of nearby targets by appropriately adjusting the likelihood.

2.4 BATCH PROCESSING TECHNIQUES

Note that global *need not mean globally optimal but rather refers to the observation horizon that goes beyond one frame.*

As already briefly discussed in Section 1.1, one way to classify multi-target tracking approaches is to distinguish between online methods, where the state is estimated at each time step (see previous section), and offline or batch techniques, which consider an entire temporal sequence at once. We will next review the literature that belongs to the latter class, often referred to as *global data association*.

2.4.1 *Measurement-based state representation*

All methods described in this section have in common that the number of possible paths is limited by forcing the targets to follow the detection responses. Even though special care is taken that the same detector response is never associated with two different targets, such approaches still lack proper exclusion modeling. Firstly, the actual trajectories are usually computed in a post-processing step using the Kalman filter, which may lead to overlapping target locations in the final result. More important is the fact that trajectories are simply

interpolated in case of missing detections which may directly result in collisions between targets.

INTEGER PROGRAMMING. Early work by Morefield (1977) suggests a 0-1 integer program (IP)-based formulation, which is closely related to the set packing problem. A binary vector whose dimension corresponds to the number of feasible tracks indicates which tracks are to be selected and which measurements are discarded as false alarms. Linear constraints ensure that only feasible solutions are possible. The optimization of the 0-1 problem is then carried out by implicit enumeration, a back-tracking type algorithm. A similar idea was later pursued by Storms and Spieksma (2000), but a more efficient Lagrangian relaxation is proposed for minimizing the objective function. This leads to a linear program (LP) relaxation of the problem and can be solved by any available LP technique such as the Simplex algorithm (Dantzig, 1998) or the interior point methods (Karmarkar, 1984).

HIERARCHICAL DATA ASSOCIATION. Many algorithms that analyze a batch of frames at a time follow a similar strategy. Starting from a set of short, yet confident tracks, or *tracklets*, longer trajectories are built based on global information. This technique is also sometimes referred to as *tracklet association* in literature.

Tracklets are short tracks, usually only few frames long.

Kaucic et al. (2005) link short tracklets across sensor gaps in aerial traffic scenes by optimizing the matching matrix using the Hungarian algorithm. Wu and Nevatia (2007) present a framework for detecting and tracking partially occluded humans in surveillance scenarios. They employ their previously developed edgelet detector (Wu and Nevatia, 2005) to identify visible parts and apply a greedy strategy to maximize their joint likelihood in each frame. Subsequently, data association between adjacent frames forms track hypotheses that are then grown to longer tracks taking their dynamic behavior and the learned appearance into account. Similarly, Huang et al. (2008) perform data association on a three-level hierarchy starting from conservative two-frame linking of overlapping detector responses. These tracklets are then grown to longer ones based on a dynamic model and a refined appearance computation. Finally, high-level data association infers scene information such as scene occluders and entry or exit areas and produces final tracks using an alternating optimization algorithm. Along the same line of thought, Li et al. (2009) employ machine learning techniques such as RankBoost and AdaBoost to automatically learn the similarity score between tracklets in a supervised manner instead of defining them heuristically. In a related way, Yang and Nevatia (2012a) learn a CRF energy online while specifically concentrating on the difficulty of resolving ambiguities between similar pairs of tracklets.

Andriluka et al. (2008) are able to reliably detect and estimate the pose of humans based on the pictorial structures model (Fischler and Elschlager, 1973; Felzenszwalb and Huttenlocher, 2005) and generate short tracklets using Gaussian process-based models. This allows tracking through full, long-term occlusions by matching the walking cycle of side view pedestrians. The same approach is also shown to work well in 3D and in sequences with moving cameras (Andriluka et al., 2010). In both cases, the Viterbi algorithm is used to infer the optimal sequence of the underlying HMM.

NETWORK MODELS. A network-based global optimization scheme was proposed by Jiang et al. (2007) who apply an integer linear program (ILP) formulation to visual tracking of multiple targets instead of only dealing with radar or sonar applications (Morefield, 1977). To allow for occlusion, a special node is introduced where targets can linger as long as no adequate detection is available. Besides temporal constraints that enforce plausible data interpretation, additional layout constraints are used, motivated by the assumption that objects maintain their relative distance to each other across time. The relaxed objective is convex and in most cases yields an integer solution that corresponds to the global optimum. Another network-related approach was formulated by Zhang et al. (2008). Two types of arcs between detections describe the observation likelihood and the putative motion of a target and each node is connected to a source and to a sink node. A globally optimal solution is then found in polynomial time by a min-cost flow algorithm. Occlusion is again handled explicitly in a post-processing step by introducing new nodes into the graph and reiterating the optimization. Their approach was later refined by Pirsiavash et al. (2011) who could achieve linear complexity by successively solving the shortest-path problem. Surprisingly, the global optimum can still be reached by this apparently greedy approach. A multi-pass dynamic programming (DP) algorithm is used to obtain a suboptimal solution even faster. This method is somewhat reminiscent of the DP approach of Berclaz et al. (2006). However in the latter case, the paths were constructed through a predefined grid and not based on detection responses.

This DP approach is probably the fastest currently available multi-target tracker.

Brendel et al. (2011) pose the data association task as a maximum-weight independent set (MWIS) problem. Temporally adjacent detection pairs form nodes in a graph and are then grown to longer trajectories by iteratively choosing the best independent set of vertices, *i.e.* a set of nodes that are not connected by any edge. The edges represent constraints that ensure plausible data interpretation and physical consistency. The locally optimal solution is obtained in polynomial time by a customized MWIS algorithm. Zamir et al. (2012) form fully connected graphs between all detections of the same person within a temporal window. Each frame corresponds to a disconnected cluster of

nodes representing targets or false alarms, respectively. The weights between the nodes in different clusters carry information about the appearance and motion of the targets. To obtain tracklets of the same person, the generalized minimum clique problem (GMCP) is solved locally for short time spans. The same procedure is then repeated to generate long trajectories. Short- and long-term occlusions are handled by introducing hypothetical nodes, similar to Zhang et al. (2008).

2.4.2 *Explicit state representation*

The methods described above entirely rely on detection responses to derive the locations of targets. Even though reducing the state space in this way is attractive from the optimization perspective, it has one major drawback: The targets' actual locations are not considered during optimization if no detections are present. In other words, if the object detector fails due to occlusion or due to any other reason, trajectories are usually interpolated, which may result in physically impossible solutions with overlapping paths.

To remedy this shortcoming, all three approaches presented in this dissertation explicitly represent the state (*i.e.* the location) of each target, regardless of the fact whether the target is visible or occluded. This section deals with previous approaches that follow a similar strategy and explicitly incorporate the layout of the trajectories into the optimization problem. Leibe et al. (2007) propose to couple the two related tasks of object detection and trajectory estimation. Following the minimum description length (MDL) paradigm, an over-complete set of detections and trajectories is first generated and a model selection mechanism that includes pairwise constraints between both detection and trajectory hypotheses is then formalized as a unified quadratic boolean program (QBP) optimization framework. The resulting problem contains non-submodular terms, which leads to NP-hard optimization. To obtain a locally optimal solution the authors therefore resort to an iterative procedure and heuristic pruning techniques. Wu et al. (2012) also tackle both problems jointly, but succeed in formulating a linear objective function that is easier to optimize. In addition, they adjust the behavior of the person detector according to current data association. Mitzel et al. (2010) extend the classic tracking-by-detection approach and apply level-set-based segmentation to find the silhouette of a detected target. This information is propagated through time by space warping, which results in an additional low-level tracker. Besides keeping the trajectories alive in case of detector failure, the level-set tracker can significantly speed up the computation by only requesting new detections if necessary and not at every frame. Horbert et al. (2011) further improve the level-set tracking framework by using more sophisticated hierarchical segmentation and accurately inferring foreground/background probabilities.

However, their approach entirely forgoes interactions between targets, an important aspect in multiple target tracking scenarios. Mitzel and Leibe (2012) take a step beyond tracking only one single target class and additionally identify carried objects. They use depth information from stereo to match the point cloud of a target against a learned pedestrian model using the iterative closest point (ICP) algorithm and detect further objects like bags or strollers in regions where the models do not align.

MARKOV CHAIN MONTE CARLO SAMPLING. We now turn to a different class of algorithms that rely on stochastic sampling techniques. Particle filters offer a powerful tool for approximating a complex probability distribution by sequential sampling (see Section 2.3). However, they rely on the first-order Markov assumption, which may impose a serious limitation on the model. Here we will look at some multi-target tracking methods that use a more general sampling technique.

Khan et al. (2005) introduce a Markov random field (MRF) motion prior to model the pairwise interaction between targets. This helps to maintain the identities in complex situations with crossing trajectories. The exponential complexity is approached by Markov chain Monte Carlo (MCMC) sampling. Using a set of predefined moves, the sampling scheme is able to handle a variable number of targets by jumping between various dimensionalities. In their related work, Khan et al. (2006) introduce the notion of merged and multiple measurements that arise in real world visual tracking scenarios either when several proximal objects give rise to one single measurement or when one target produces two separate detections. Under the assumption of a linear motion model and that the posterior can be approximated by a mixture of Gaussians it is possible to marginalize out the continuous target space and to apply the MCMC sampling to resolve data association. Both works show encouraging results on videos of ant colonies. Although the number of targets is quite high, no occlusion takes place in such data because the bird's-eye view shows the entire target space.

A surveillance scenario showing several people is approached by Markov chain Monte Carlo data association (MCMCDA) scheme proposed by Oh et al. (2004). It accurately models probabilistic data association and also employs stochastic sampling for inference. The authors also prove that their algorithm converges to the full Bayesian solution if given enough computational resources. Yu et al. (2007) exploit the spatio-temporal smoothness of motion and appearance and allow for more complex dependencies between targets and measurements than a simple one-to-one mapping. A data driven Markov chain Monte Carlo (DDMCMC) sampling is employed to search the non-trivial solution space of possible trajectories and data association.

More recently, Benfold and Reid (2011) rely on stable and accurate head detections of pedestrians in an urban environment. The inference also follows the MCMCDA and the multi-threaded design of their system allows leveraging parallel computing power to achieve real-time performance in high-definition videos. Wojek et al. (2010) significantly improve the robustness of object detection in traffic scenes from a moving on-board camera by aggregating evidence over multiple frames. Under the assumption that all objects reside on a common ground plane and by exploiting semantic labeling information, the entire 3D scene, *i.e.* the location of pedestrians and cars as well as the camera pose are inferred by reversible jump Markov chain Monte Carlo (RJMCMC). Choi and Savarese (2010) follow a similar approach but additionally model interaction between targets by using repulsive and attractive forces, respectively. While they are able to reconstruct long tracks of people maintaining their identities, the complex graph structure significantly handicaps inference, leading to long computation times of few minutes per frame.

All of the above techniques use some form of jumps or moves, such as adding or removing an object from the solution or changing the current configuration to explore different discrete states. A similar technique is also employed in our continuous minimization framework presented in Chapter 5 of this dissertation.

DISCRETE GRID. A different way to reduce the search space of all possible trajectories is to discretize the physical space to a regular grid and force the targets to move along the grid cells. One of the main advantages of such an approach is that the location of each target is always represented explicitly and does not entirely depend on the presence of detections. Occlusions can thus be handled implicitly within the global optimization without the need of adding hypothetical or special occlusion nodes.

A general approach to finding the best set of k paths through a trellis is described by Wolf et al. (1989) who employ the Viterbi algorithm for inference but make rather strong assumptions about the absence of missing or merged detections. Berclaz et al. (2006) use a dynamic programming (DP) approach to track several individuals in a relatively tight indoor environment. The evidence is accumulated from multiple cameras. Although the trajectories are optimized one at a time, a heuristically defined processing order ensures that the optimization is hardly ever trapped in local minima. In their later work, Berclaz et al. (2009) still stick to the same ground plane discretization but pose the optimization as an integer linear program (ILP). Binary variables indicate the presence or absence of flow between two neighboring cells while linear constraints ascertain that flow can neither appear nor vanish, except at certain locations, *e.g.* along the border. This allows the joint optimization for all trajectories within a time

window. The LP relaxation yields an integer solution (except in rare degenerate cases), thus the global optimum can be found efficiently. Berclaz et al. (2011) reformulate the same system and solve it by the k-shortest paths (KSP) algorithm by exploiting the special structure of the problem. This novel formulation is much more efficient than the general LP relaxation technique and can achieve real-time performance in real-world scenarios. Finally, Ben Shitrit et al. (2011) enrich the grid tracking idea with a long-range appearance model. To this end, they design a multi-layered graph where each layer represents a certain predefined identity group. To keep the inference tractable, a KSP pass first significantly prunes the graph before the multi-layer network optimization is approached. Although global optimality is forfeited, the work shows remarkable results in long basketball sequences, where players, whose t-shirt number serves as a unique person identifier, are tracked correctly during the course of very long time spans without switching their identity.

Note that Part I of this dissertation is inspired by the ILP formulation of Berclaz et al. (2009). In contrast to their original work, we extend the formulation to allow including a constant heading dynamic model and a simple appearance model while still holding similar optimality conditions (Andriyenko and Schindler, 2010). The dynamic model is achieved by extending the functionality of the binary indicator variables to three consecutive frames allowing for measurement of the target's change of heading direction.

2.4.3 *Merging and splitting*

The classical task of multiple target tracking requires the exact trajectory of each target to be reconstructed. Obviously this task becomes difficult or even infeasible when targets come close to each other causing complete occlusions. This effect becomes more severe when background subtraction is used for detecting moving objects (*cf*. Section 3.2). In this case, blobs will eventually merge into one since such techniques are unable to disambiguate between partially occluded targets. Some work, as described below, therefore forfeits the precise solution and resorts to a coarser approximation where targets are merged into groups when they are close to each other and split later.

Nillius et al. (2006) construct a Bayesian network called a track graph, where nodes represent either single-target trajectories or merged multi-people tracks. To solve the final problem, a message propagation algorithm finds the track of each person through the graph. To keep the inference tractable the graph complexity is reduced by removing dependencies that are distant in time. Perera et al. (2006) follow up on the work of Kaucic et al. (2005) and link tracklets across long occlusion gaps. However, they additionally model track merg-

ing and splitting to allow for merged measurements. Henriques et al. (2011) also build on a graph with merge/split nodes but solve the complete problem in polynomial time where the matching is performed by the Hungarian algorithm and the plausible number of objects in each group is ensured by a polynomial time min-cost flow circulation algorithm. Although the presented results are quite impressive, it is impossible to directly compare the performance to standard multi-target tracking methods since the individual tracks of merged targets remain undefined until the targets split apart.

2.5 RELATED AREAS OF APPLICATION

In this section, we will review selected work in areas related to multi-target tracking.

2.5.1 *Multi-camera networks and handover*

A surveillance system typically consists of more than just one camera. A whole network of pan-tilt-zoom (PTZ) cameras may be deployed to observe most corners of an area of interest. In such cases it may be desirable to not only track the targets in one view, but to also identify the same target across different, possibly non-overlapping fields of view. This task is referred to as *handover*, since one camera 'passes' a target to a different camera. Depending on the exact camera layout, a simple dynamic model or a learned appearance model of the target may be used as cues to reconstruct the global trajectories.

Given a set of trajectories that are obtained by a particle filter independently in each view, Meden et al. (2012) pose the handover task as a global optimization problem and solve it by MCMC sampling. Idrees et al. (2012) go beyond the standard re-identification problem and actually reconstruct the hidden trajectories that lie outside the field of view of any camera. To this end, a global cost function that incorporates certain constraints such as collision avoidance, smoothness or target following is minimized. The inferred scene structure and the targets' dynamics clearly outperform a simple constant velocity assumption.

2.5.2 *Social behavior and crowd analysis*

A substantial amount of work exists that deals with the analysis of individual trajectories or the combined solution. The applications range from discovering social roles between individuals to activity recognition and prediction of single objects and crowd behavior.

Rodriguez et al. (2011) use a per pixel confidence value of a discriminatively trained head detector in conjunction with a crowd density estimator of Lempitsky and Zisserman (2010). A binary energy

that forces the number of objects to agree in both cases is then minimized in a greedy fashion. Tracking is based on a robust association scheme proposed by Everingham et al. (2006), where correspondence between detections is established by counting relevant trajectories generated by a Kanade-Lucas-Tomasi (KLT) feature point tracker (Shi and Tomasi, 1994). Inspired by the advances in crowd simulation and based on the fact that people usually head towards a determined destination, Pellegrini et al. (2009) propose a more sophisticated formulation of the targets' dynamics that goes beyond a simple first-order Markov model. Taking into account scene structure and collision avoidance, social behavior of individuals as well as between different pedestrians can be predicted reasonably well for several future frames. Ge et al. (2012) concentrate on identifying groups of pedestrians that are socially related. To this end, the geometric constellation between individually obtained trajectories is analyzed and groups are constructed by a bottom-up clustering approach. A somewhat similar problem is posed by Choi and Savarese (2012). However, they approach both problems, multi-person tracking and social analysis, jointly. The rather complex discrete optimization is handled by a combination of belief propagation and branch-and-bound methods. Amer et al. (2012) aim at detecting and classifying actions and activities of individuals as well as groups of people. Using high definition video footage they are able to digitally zoom in to recognize fine details and zoom out for a more general overview of the scene. Activity forecasting is a task recently introduced by Kitani et al. (2012). By leveraging most recent advances in semantic scene understanding (Munoz et al., 2010) and reinforcement learning (Abbeel and Ng, 2004), it is possible to predict entire trajectories far into the future from as little as a single detection.

3

PRELIMINARIES AND NOTATION

In the beginning the Universe was created. This has made a lot of people very angry and has been widely regarded as a bad move.

The Restaurant at the End of the Universe
Douglas Adams

CONTENTS

This chapter serves as an introduction to the technical part of this dissertation. First, all necessary notation used throughout the document is outlined. We then briefly review the pedestrian detector that is employed for obtaining object hypotheses. Finally, we will discuss the experimental setup that is used for measuring the performance of the newly developed methods. In particular, we will consider the datasets and have a closer look at the metrics that are employed for quantitative evaluation.

3.1 NOTATION

Although each method will require its own set of symbols and notation, a general notation is introduced here to avoid repetition in each chapter. For quick reference, all notation is summarized in Table 3.1.

Each one of the three methods presented in this dissertation performs energy minimization to estimate the state of all objects within a certain time window. Depending on the exact formulation and the application, this window can either stretch through the entire available video sequence or contain only a small subset of consecutive frames. In either case, we will refer to the length of the temporal window, *i.e.* the number of frames under consideration, as F. To refer to a specific frame we use the superscript $t \in \{1, \ldots, F\}$, while a specific target is denoted with the subscript $i \in \{1, \ldots, N\}$, where N is the total number of targets. Since N varies over time, we specify

Symbol	Description
$\mathbf{X}$	world coordinates of all targets in all frames
$\mathbf{X}_i^t$	world coordinates of target i in frame t
$\mathbf{x}_i^t$	image coordinates of target i in frame t
(X, Y)	world coordinates on the ground plane
(x, y)	image coordinates
F	total number of frames
N	total number of targets
s_i, e_i	first, respectively last frame of trajectory i
$F(i)$	number of frames where target i is present
$N(t)$	number of targets in frame t
$D(t)$	number of detections in frame t
$\mathbf{D}_g^t$	world coordinates of detection g in frame t

Table 3.1: Notation.

the number of targets that are present in frame t with $N(t)$. The state containing the locations of all targets in all frames is denoted $\mathbf{X}$. Depending on the experimental setup, the state can either be inferred in image space, or on the ground plane in world coordinates. To avoid confusion, we use lowercase to refer to image locations and capital letters for 3D positions. The temporal limits, *i.e.* the first and final frame of trajectory i are denoted s_i and e_i, respectively, which in turn means that the temporal span where a target is present can be written as $F(i) = e_i - s_i + 1$. Finally, $D(t)$ denotes the number of detector responses in frame t and $\mathbf{D}_g^t$ is the location of one specific detections.

3.2 OBJECT DETECTION

The main focus of this dissertation is to investigate various approaches to multi-target tracking. Like most state-of-the-art methods, we follow the tracking-by-detection paradigm, meaning that we rely on a pre-processing step that generates an independent set of object candidates. Although this detection procedure is entirely independent from all three tracking approaches that are presented in this work, this section provides a brief overview of several useful detector choices, including the HOG-based detector that is used throughout this dissertation.

BACKGROUND SUBTRACTION. Early tracking approaches, (*e.g.*, Kaucic et al., 2005), employed background subtraction methods (Stauffer and Grimson, 1999) to obtain regions of moving objects. After learn-

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

Figure 3.1: Pedestrian detectors may fail for various reasons including *(a-b)* low contrast, *(c)* strong pose variation, *(d)* occlusion or *(e-f)* clutter. Note that in the last case it is not even clear whether a mannequin in a display window should be regarded as a false positive.

ing a background model for each pixel, it is possible to determine the presence of an object by comparing the intensity value of the current input to the background model. To reduce the effect of noise, a heuristic smoothing procedure is usually carried out to obtain connected regions or blobs. This has several critical drawbacks. The first assumption is that the background remains more or less fixed over time. On the one hand this prohibits the use of moving cameras, for instance in cars or on robots. But even in a typical surveillance setting, the environment can change rather abruptly due to dynamic lighting conditions leading to strong deviations in intensity values. A second assumption is that all targets move uninterruptedly all the time, which, of course, is quite restricting since still objects may blend with the background and thus remain undetected. Finally, if certain parts of a target and the background have similar colors, the target will inevitably produce several disconnected regions. A more complex tracking model is then required to deal with such ambiguous splitting and merging situations.

HISTOGRAMS OF ORIENTED GRADIENTS. Recent progress in object detection has enabled robust localization of pedestrians in more or less unconstrained environments. A common practice, which is simple, yet powerful, is a technique called *sliding window*, which is not much more than a brute force search over the entire image. Broadly speaking, the same question is asked about every rectangular area in an image: does this image patch contain the object or not? Pedestrians are assumed to be standing or walking upright and usually resemble a vertical structure. Therefore, the aspect ratio and the orientation of the bounding box is fixed and the search is only performed across all locations and all scales.

One of the most popular sliding window detectors to date remains the one introduced by Dalal and Triggs (2005). The image informa-

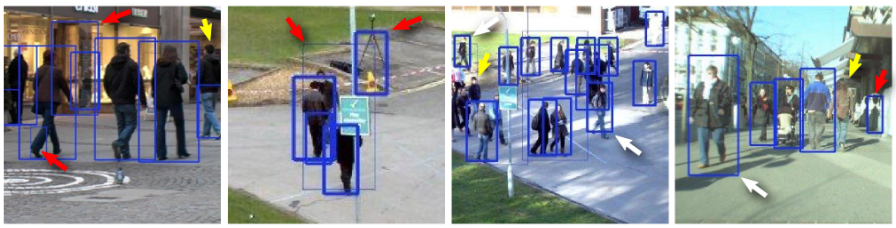


Figure 3.2: Exemplar detections used as input for our multi-target tracking. The detector's confidence is reflected in the line width of each box. Some of the detector failures such as false alarms, false negatives or poorly localized bounding boxes are indicated with red, yellow and white arrows, respectively.

tion extracted from each window forms a feature vector that is based on local gradient histograms. This approach is therefore known as histogram of oriented gradients (HOG). In a nutshell, the functionality of a HOG detector can be summarized as follows: Each window is subdivided into groups of neighboring pixels, called *cells*, from which local gradients are computed. Neighboring cells form larger units, called *blocks*, where the histograms are normalized to provide invariance to image contrast. Finally, a support vector machine (SVM) serves as a classifier to determine whether a feature vector represents a pedestrian or not. Although the general idea may seem rather simple, a complete implementation of this approach contains many important details such that best performance can only be achieved by carefully designing each one of the components. Some extensions have since been proposed that, generally speaking, concentrate on designing additional features including optic flow (Dalal et al., 2006), color self-similarity (Walk et al., 2010a) or pixel disparities computed from stereo images (Walk et al., 2010b).

DEFORMABLE PART-BASED MODELS. While HOG-based detectors are still widely used today for detecting pedestrians, they have several crucial limitations. One of them is the monolithic structure that only allows for small deformations up to a certain degree. If, however, one is interested in detecting less symmetric objects such as people in general, whose posture may significantly deviate from an upright walking pedestrian, this single-template-based approach is likely to fail. A further deficiency is sensitivity to partial occlusion, which may lead to detector failures.

The deformable part-based model (DPM) developed by Felzenszwalb et al. (2010) addresses both issues by treating the object as a constellation of individual parts that are spatially dependent. It builds on the pictorial structures idea, originally introduced by Fischler and Elschlager (1973). The entire object itself is treated as a root node that is connected to a certain number of smaller parts that are arranged in a certain star-shaped layout. The locations of the individual parts are

not known during training and treated as hidden variables within a latent SVM framework. Even though the DPM also builds on HOG features, it can better handle large deformation and partial occlusions. Recent tracking approaches that directly integrate occlusion handling (Shu et al., 2012; Izadinia et al., 2012) often employ the DPM as object detector.

IMPLICIT SHAPE MODEL. Another object detection approach that relies on object parts is the implicit shape model (ISM) developed by Leibe et al. (2008a). However, it follows a slightly different methodology. First, a visual vocabulary is learned for a specific object class. To this end, small image patches are sampled around interest points and clustered together to form a codebook, which stores information about appearance, location and scale of an object part. During detection, the learned visual words cast probabilistic votes in Hough space for the center of the object. The modes of these votes in the (x, y, scale)-space are then found by mean-shift clustering. Furthermore, these modes can be backprojected onto the image to produce a foreground segmentation. Since the implicit shape model solely relies on small patches to represent the object, it requires the object to be large enough which cannot always be guaranteed in typical surveillance settings. It is therefore rarely used nowadays in tracking-by-detection systems (Leibe et al., 2007; Breitenstein et al., 2009).

3.3 DATASETS

To demonstrate the applicability of a computer vision approach to real-world situations, it is indispensable to test the performance of any method on realistic data. Obviously, using data as diverse as possible to cover all potential scenarios is advantageous. This avoids over-fitting and shows robustness to various situations. In practice, however, this is challenging for a variety of reasons. First, supervised learning and quantitative evaluation both require the data to be manually annotated, which is tedious and costly. Second, the amount of resources is always limited such that only a subset can be used to evaluate the performance. Having these constraints, choosing the 'right' data is not always trivial. Some criteria for this choice should be:

- *Openness.* It is important that the data that is used for presenting the performance of any method is freely available so that it can be used by others for comparison.
- *Variability.* The dataset should be sufficiently versatile to show that a method is capable of handling various scenarios.
- *Complexity.* It may not make much sense to show near flawless performance on an easy dataset. Instead, it is beneficial to eval-

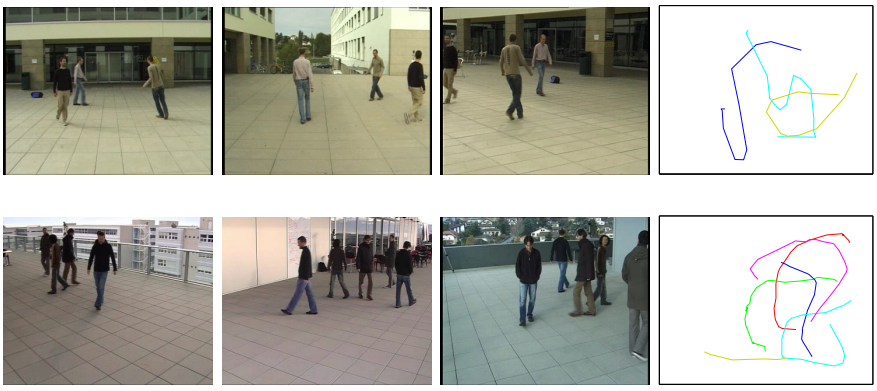


Figure 3.3: Three views of the *campus2 (top)* and *terrace1 (bottom)* sequences. On the right-hand side, 12 seconds of ground-truth trajectories are plotted from the bird's-eye view.

uate rather challenging datasets and point out the limitations of a method.

In this section, several video sequences are presented that will later be used to show the functionality of the proposed tracking approaches.

EPFL. The computer vision lab at *EPFL* in Lausanne, Switzerland, offers several people tracking datasets, all of which are filmed by several cameras.

The four sequences *campus1*, *campus2*, *terrace1* and *terrace2* (Berclaz et al., 2006; Fleuret et al., 2008) show up to six people walking around outdoors around an area of about 10 by 15 meters (see Figure 3.3). The cameras are positioned in three, respectively four different corners at a height of about two meters. Although people frequently become completely occluded in one view, every person is usually visible by one or more other cameras. A homography matrix is given as camera calibration. The ground truth provided by the authors is only given at discrete grid points and only every 25 frames. To quantitatively assess the performance more accurately, we provide continuous annotations both in the spatial and in the temporal domain. A short fragment is depicted on the right hand side of Figure 3.3.

Note that correctly aggregating evidence from multiple overlapping fields of view is not trivial. In our experiments in Chapter 4 we follow a simple strategy where the detections from individual views are accumulated independently in world coordinates and weighted according to the number of views they are visible in.

PETS. The First IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, better known as PETS, was held in the year 2000. Since then, several datasets in various surveillance settings have been recorded and published, each with the goal of

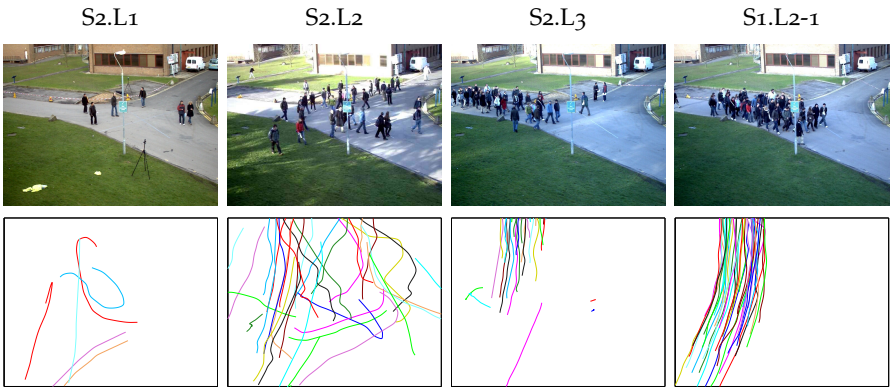


Figure 3.4: The PETS sequences exhibit strong variability in people count, formation, walking behavior and lighting. The cumulative trajectories in world coordinates *(bottom)* are plotted for a 12-second period.

creating a common benchmark to compare different approaches for various applications. The most recent one from 2009 (Ferryman and Shahrokni, 2009) is still widely used by the computer vision community. The data was recorded at the campus of the University of Reading, UK. A total of eight calibrated cameras were used to record pedestrians walking or running around an intersection. The data of one view is withheld by the organizers to be used solely for testing. This dissertation mainly concentrates on monocular people tracking. Hence, only the first view of the entire setup will be used.

The entire benchmark consists of 18 sequences that are divided in three sets: people count, people tracking and event recognition. The most popular sequence for people tracking, *S2.L1*, has been used extensively in the past. Although people walk close together causing occlusion and sometimes randomly change their moving direction and speed, recent approaches (Henriques et al., 2011; Andriyenko et al., 2012) achieve near flawless performance on this sequence.

To show the strength of our methods described in Chapters 5 and 6, we thus step up to the more challenging sequences. The crowded scenarios show up to 42 people simultaneously. The motion behavior of the pedestrians ranges between practically random (*S2.L2*) to quite regular (*S3.L2-1*). Some example frames from four sequences are shown in Figure 3.4. Using several different scenarios offers a good way to show the robustness of a tracking method.

The PETS organizers do not intend to release the annotations because the workshop is usually organized as a challenge. For everyone outside the challenge we have annotated many of the 18 sequences and make all ground truth data publicly available with the hope that the evaluation on the more challenging scenarios will become more commonplace in future research.

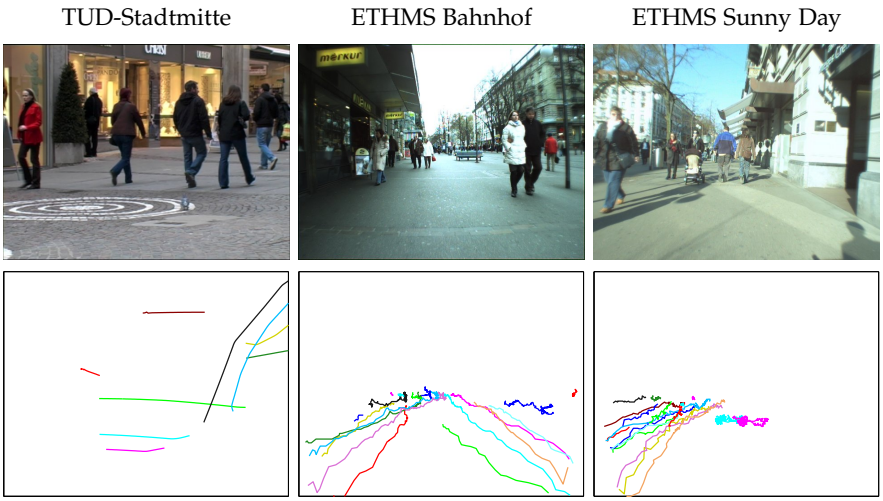


Figure 3.5: Example frames from the *TUD* and the *ETH Mobile Scene* datasets. The bottom row shows ground truth trajectories for a time span of about 12 seconds. Note that the ground truth for the *ETHMS* sequences is plotted in image space.

TUD. Although the two previous datasets, *EPFL* and *PETS*, show real video footage, the people are volunteers who were asked to walk in a certain way. In contrast, the three sequences of the *TUD* dataset show 'real' pedestrians filmed on the street. In this dissertation, only the *TUD-Stadtmitte* sequence (Andriluka et al., 2010) is used because it offers a camera calibration, which allows one to perform the tracking in world coordinates. *TUD-Stadtmitte* is a short video showing a busy pedestrian street in Darmstadt, recorded from a rather low viewpoint (see Figure 3.5). There are two main challenges. On the one hand, people occlude each other for longer periods leading to large detection gaps. On the other hand, the low perspective makes 3D estimation rather inaccurate, which poses a challenge for reconstructing exact trajectories. Interestingly, there exist several sets of publicly available annotations for this sequence that strongly deviate from each other. This issue will be discussed more thoroughly in Section 7.1.

ETH MOBILE SCENE. Finally, we test our discrete-continuous energy minimization scheme from Chapter 6 on two widely used sequences of the *ETH* dataset (Ess et al., 2008). These were recorded from a moving platform, where a stereo camera was placed at a height of about one meter from the ground (*cf*. Figure 3.5). Although a rough camera pose can be estimated using structure-from-motion, it is rather unreliable. Therefore, we will apply our tracker directly in image space on this dataset.

AN OVERVIEW OF PUBLIC DATASETS. For the sake of completeness, Table 3.2 presents an overview over popular, publicly available

Table 3.2: An overview of some of the most popular multi-target tracking datasets. Only sequences with publicly available identity preserving annotations are listed here.

Dataset	sequence	# frames	frame rate	resolution	crowd density	# views	camera motion	stereo	calibration
CAVIAR[1]	26 seq.	37224	25	low	l-m	2	-	-	✓
EPFL[2]	passageway	2500	25	low	low	4	-	-	✓
	lab1	3750	25	low	med	4	-	-	✓
	lab2	3750	25	low	med	4	-	-	✓
	campus1	2000	25	low	low	3	-	-	✓
	campus2	1400	25	low	low	3	-	-	✓
	terrace1	5010	25	low	med	4	-	-	✓
	terrace2	4480	25	low	med	4	-	-	✓
PETS'09[3]	S1.L1-1	221	7	med	hi	7	-	-	✓
	S1.L1-2	241	7	med	hi	7	-	-	✓
	S1.L2-1	201	7	med	hi	7	-	-	✓
	S1.L2-2	131	7	med	hi	7	-	-	✓
	S2.L1	794	7	med	med	7	-	-	✓
	S2.L2	436	7	med	hi	7	-	-	✓
	S2.L3	240	7	med	hi	7	-	-	✓
	S3.MF1	107	7	med	low	7	-	-	✓
TUD[4]	Campus	71	25	med	med	1	-	-	-
	Crossing	201	25	med	med	1	-	-	-
	Stadtmitte	179	25	med	med	1	-	-	✓
ETHMS[5]	Bahnhof	999	14	med	med	1	✓	✓	✓
	Sunny Day	354	14	med	med	1	✓	✓	✓
AVG[6]	TownCentre	4500	25	hi	med	1	-	-	✓
PNNL[7]	ParkingLot	1000	25	hi	med	1	-	-	-

benchmark datasets for multiple object tracking that are frequently used in literature to show the strength of state-of-the-art approaches.

1 http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1
2 http://cvlab.epfl.ch/data/pom
3 http://www.cvg.rdg.ac.uk/PETS2009
4 http://www.d2.mpi-inf.mpg.de/andriluka_cvpr08
http://www.d2.mpi-inf.mpg.de/node/428
5 http://www.vision.ee.ethz.ch/~aess/dataset
6 http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bbenfold_headpose/project.html
7 http://crcv.ucf.edu/data/ParkingLOT

3.4 METRICS FOR QUANTITATIVE EVALUATION

To quantitatively measure the performance of one method and optionally to compare it with others, a clearly defined protocol is required. Unfortunately, objectively assessing the quality of a multi-target tracking solution is not an easy task. Furthermore, the 'perfect' solution, or *ground truth*, is needed to serve as reference. We will discuss these issues and related challenges in Section 7.1. In this section, we will only present various protocols that are currently used for evaluating multi-target tracking.

3.4.1 *CLEAR MOT*

To evaluate the correctness of any tracker at least three entities need to be defined:

- the tracker output (or hypothesis) $\mathcal{H}$, which is the result of the tracking algorithm;
- the correct result, or ground truth $\mathcal{GT}$; and
- a distance measure $\bar{d}$ that measures the similarity between the true target and the prediction.

Note that these requirements are kept very general without any assumptions on the concrete representation or on the exact definition of the distance function.

Intuitively, one wishes to incorporate and grade every possible error that a solution may contain. One of the protocols that follow this goal is the *CLEAR MOT* evaluation (Bernardin and Stiefelhagen, 2008). It emerged from the Classification of Events, Activities and Relationships (CLEAR) Workshop[8] in 2006 and has since been widely accepted as a standard evaluation tool by the tracking community. The two proposed quantities, MOTA and MOTP on the one hand measure the number of errors that occur during tracking, and on the other hand assess the tracker's precision, *i.e.* its ability to localize the target in the image. Let us now take a closer look at the different components that give rise to these quantities.

MOT ACCURACY. As in object detection, the two most common errors in multi-target tracking are false positives *(FP)* and false negatives *(FN)*. The former correspond to spurious tracking results that do not match any ground truth trajectory, while the latter ones are annotated targets that are not identified by the tracker. To determine whether a target is being tracked, a correspondence between true targets and hypotheses must be established. This is usually done in a greedy manner, however, not independently in each frame but in

False positive detections are often called false alarms *and false negatives –* missed targets *in the literature.*

8 http://clear-evaluation.org

consideration of temporal matching. More precisely, if and only if a target is not tracked, it is assigned the closest unmatched hypothesis. Otherwise, the correspondence from the previous frame is maintained. To decide, whether a track is a potential candidate for a match, a distance between all hypotheses and all targets must be computed. If the distance between a track-object pair is small enough, they can potentially be matched. Note that this procedure to compute the correspondences is application and representation specific. If both the output and the annotations are described by bounding boxes, then usually the PASCAL criterion

$$\overline{d}(\mathcal{H}, \mathcal{GT}) = \frac{\text{bbox}(\mathcal{H}) \cap \text{bbox}(\mathcal{GT})}{\text{bbox}(\mathcal{H}) \cup \text{bbox}(\mathcal{GT})}, \tag{3.1}$$

i.e. the intersection over union (Jaccard index) or the relative overlap of the true and the predicted bounding boxes, determines the similarity between the two, where 0 means no overlap and 1 means that both bounding boxes are identical. The most common threshold for considering a pair correct is 0.5. For 3D tracking, it may be more reasonable to compute the correspondence directly in world coordinates (*cf*. Figure 3.6). In this case, the Euclidean distance between the centroids of two objects gives a suitable estimate. For people tracking, the foot position, *i.e.* the center of the bottom edge of the bounding box, is used as the target's centroid and a threshold of 1 meter is used.

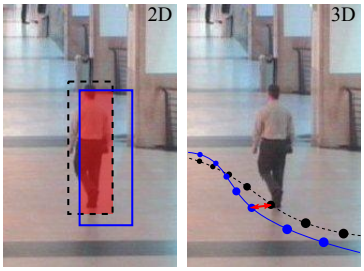


Fig. 3.6: Measuring correspondence as bounding box overlap (2D) or as distance on the ground plane (3D).

Recall that the goal of multi-target tracking is not only to find all objects and suppress all false alarms but also to correctly follow each object over time. In other words, the reconstructed trajectory should adhere to one specific object from the moment of entry until it exits the scene. Whenever there is a mismatch between a hypothesis and the corresponding ground truth trajectory, an identity switch *(ID)* occurs, which is counted as an error. A simple example illustrating these three error types is depicted on the left-hand side of Figure 3.7. Although temporally-aware target-to-tracker matching suppresses unnecessary identity switches, it may lead to undesirable artifacts, as illustrated in Figure 3.7 (right).

MOTA *combines all relevant errors into one number between* 0 *and* 1.

Let us now formally define the Multiple Object Tracking Accuracy (MOTA). Let $\text{FP}(t)$, $\text{FN}(t)$ and $\text{ID}(t)$ denote the number of false positives, missed targets and identity switches at time t, respectively. Further, let $N_{\text{GT}}(t)$ denote the number of annotated targets at time t. Then the MOTA score is computed as

$$\text{MOTA} = 1 - \frac{\sum_t \Big(\text{FP}(t) + \text{FN}(t) + \text{ID}(t)\Big)}{\sum_t N_{\text{GT}}(t)}. \tag{3.2}$$

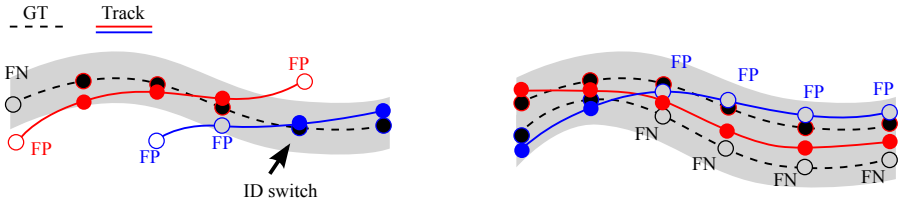


Figure 3.7: Illustration of the CLEAR MOT components. Events that are classified as correct are denoted with solid circles. Errors are indicated with empty circles. The influence of track to ground truth assignments is illustrated on the right: A 'wrong' decision at the beginning of a trajectory leads to persistent errors over the whole sequence.

Note that if a solution contains no errors, *i.e.* the numerator sums up to 0, then the accuracy equals 100%. This value decreases as the number of failures increases. The MOTA score can also result in negative values and is in fact unbounded (from below). Allowing for a negative accuracy may seem unnatural, but this can only occur when the number of errors is larger than the number of targets in the scene, which only rarely happens in practice.

Combining the quality of a tracking result into a single number has both positive and negative consequences. On the one hand, it enables a simple comparison. On the other hand, the strengths and weaknesses of a particular method may become concealed. It is therefore preferable to present all available numbers, as we will see, *e.g.*, in Table 5.7.

MOT PRECISION. The MOTA described above measures the discrete number of errors made by the tracker. On the contrary, the Multiple Object Tracking Precision (MOTP) avoids such hard decisions and instead estimates, how well a tracker localizes the targets. Again, in its general form it is defined as

MOTP *amounts to the average distance between the tracker and the annotation.*

$$\text{MOTP} = \frac{\sum_{t,i} \overline{d}\left(\mathcal{GT}_i^t, \mathcal{H}_{g(i)}^t\right)}{\sum_t m_t}, \tag{3.3}$$

where $\mathcal{GT}_i^t$ and $\mathcal{H}_{g(i)}^t$ are the target and its associated hypothesis, respectively, and m_t is the number of matches at time t. Intuitively, it provides the average distance over all matched pairs. In 2D, this number directly represents the average overlap of matched bounding boxes while for the evaluation in 3D we normalize it to the hit/miss threshold such that it provides a percentage value between 0 and 100%. We point out that MOTP is a rather rough estimate of the performance because it heavily relies on the quality of the annotations which are often inaccurate or even ambiguous.

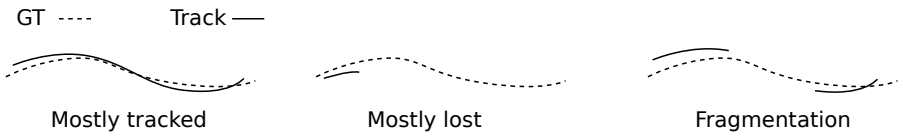


Figure 3.8: Trajectory-level measures defined by Li et al. (2009).

3.4.2 *Further metrics*

Next to the widely used CLEAR metrics, other performance measures have been introduced in the literature.

TRAJECTORY-BASED MEASURES. Wu and Nevatia (2006) describe a set of measures that assess the performance on entire trajectories rather than on a frame-by-frame basis. Their definition has later been refined (Li et al., 2009) to capture some ambiguous cases. For our evaluation, we follow the latter, more precise formulation.

A target is often correctly tracked only for a certain period and not for its entire presence in the scene. To quantify this property, a trajectory can be classified as mostly tracked *(MT)*, partially tracked *(PT)* and mostly lost *(ML)*. A target is considered mostly lost when it is found by the tracker during less than 20% of its presence. Similarly, a target is mostly tracked when at least 80% of its ground truth trajectory is found. Consequently, all other trajectories are partially tracked. Note that identity switches do not play any part in the computation of these figures.

Finally, track fragmentations count how many times a ground truth trajectory changes its status from 'tracked' to 'not tracked', *i.e.* each time it is lost by the current hypothesis. These three trajectory-based measures are illustrated in Figure 3.8.

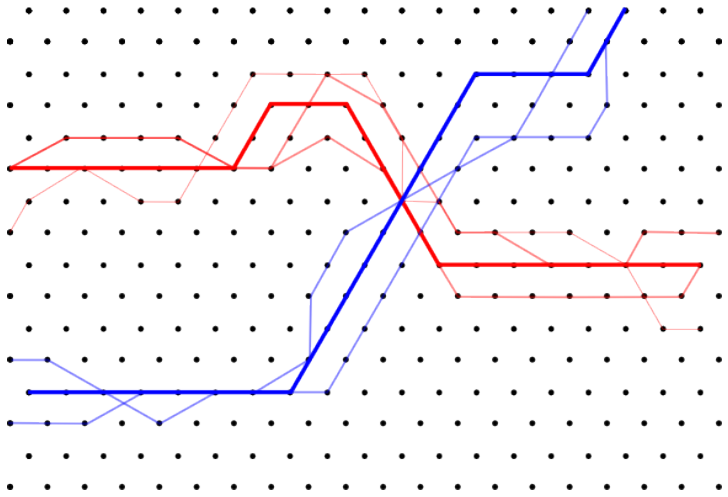
CONFIGURATION DISTANCE AND PURITY. Smith et al. (2005) also integrate standard errors and measures such as false positives, false negatives, precision or recall in their evaluation protocol. In addition, they propose a more detailed inspection of each tracker, each trajectory and the configuration state. In particular, they allow multiple tracker-to-target assignments but count these as multiple trackers or multiple objects errors. The configuration distance measures the difference between the number of predicted and true targets and indicates the bias towards more false alarms or towards missed targets. Further measures like tracker or object purity are somewhat related to the mostly tracked definition above, but provide a more detailed evaluation on the produced hypotheses and not only on the ground truth trajectories. Since these metrics are rarely used in the literature, we do not employ them in this dissertation.

SUMMARY. To summarize, there is no single objective measure for quantitative evaluation of a multi-target tracking algorithm that incorporates all possible cases. Many proposed protocols follow a similar intuition, but are somewhat ambiguous in their exact definitions. As a result, the computed numbers usually give a fair assessment of the overall performance, but may vary depending on the concrete implementation of the evaluation software. Please refer to Section 7.1 for a more detailed study on this subject.

Part I

TRACKING IN DISCRETE SPACE

Multi-target tracking is formalized as an integer linear program by discretizing the location space to a regular grid.



4

GLOBALLY OPTIMAL MULTI-OBJECT TRACKING ON A HEXAGONAL LATTICE

CONTENTS

As discussed in Section 1.3, many approaches to multi-target tracking address the problem at hand by reducing the search space to a finite set. In this chapter, we follow this line of thought and build on a recently proposed formulation by Berclaz et al. (2009). To that end, the search space is discretized to a regular grid, where each grid cell is connected to its neighbors in adjacent frames, thereby forming feasible paths for the targets. These connections are modeled by binary variables, where 1 indicates that a target moves along the corresponding edge, and 0 means that there is no motion along this edge. Additional constraints prevent multiple occupancy of any individual cell as well as abrupt interruptions of trajectories in the middle of the grid. The resulting integer linear program (ILP) is then relaxed to a linear program (LP) and solved to (near) global optimality by well established optimization techniques. This work has previously appeared as (Andriyenko and Schindler, 2010).

4.1 INTRODUCTION

Based on the observation that tracking performance can be hampered by local optima of the underlying objective, a recent key challenge in multi-target tracking research has been to develop schemes that are able to find (nearly) *global* maxima of the posterior over the set of

trajectories. To make this possible, these schemes restrict the set of permissible target locations to a finite set, in such a way that the problem becomes amenable to global optimization. A-priori, the set of possible locations is infinite, or at least very large: targets can move anywhere within the observed region. There are two main strategies to restrict possible target locations to a reasonably small set: either candidate locations are found by thresholding and/or NMS of the observation likelihood; or the tracking region is sampled on a regular grid.

The dominant strategy so far has been the first one: the image evidence – typically the output of object detection or background subtraction (see Section 3.2 for details) – is used to identify the most promising target locations per frame. These serve as input for the tracker, which links them to trajectories. A limitation of this strategy is that candidate locations are implicitly assumed to correspond perfectly with true target positions; there is no concept of localization uncertainty. Another problem is that the space is sampled *only* at promising locations, hence target locations are not even defined in case of missing evidence (*e.g.* if two targets were both missed by the observation model, it is no longer checked whether they would collide in that frame).

A regular discretization of the observation area is attractive because the state (*i.e.* target locations) is defined explicitly at each time step, even when no evidence from the object detector is available, thus allowing for principled probabilistic modeling. A disadvantage is that in order to keep tracking computationally tractable, the grid needs to be significantly coarser than typical image resolutions, which introduces aliasing. A particularly undesirable consequence of the discretization is that the space is no longer isotropic – the smoothness of a trajectory depends on its alignment with the grid, and jagged trajectories complicate the usage of reasonable dynamic models, which favor smooth motion.

This chapter describes a global optimization approach to multi-target tracking on a regular grid, with an a-priori *unknown* number of targets. In particular, the contributions compared to previous work are:

- A "re-introduction" of the dynamic model, which has traditionally been an integral part of tracking, but was dropped in previous work in order to achieve objective functions that can be solved to (near) global optimality. Specifically, we include a constant heading prior (*cf*. Section 4.2.4).

- To best utilize the dynamic model and achieve smoother, more accurate trajectories despite the discrete setting, the location space is sampled on a hexagonal lattice, rather than a rectangular one (see Section 4.2.5).

Symbol	Description
$\mathbf{X}_u$	discrete X, Y-location u
$\mathcal{S}(u)$	all neighbors of u
K^t_{uvw}	tracklet over $\{\mathbf{X}^{t-1}_u, \mathbf{X}^t_v, \mathbf{X}^{t+1}_w\}$
$\mathbf{K}$	set of all tracklets
$\mathbf{B}$	set of all indicator variables
c^t_{uvw}	log-likelihood ratio for B^t_{uvw}

Table 4.1: Additional notation used in this chapter.

- The non-maxima suppression (NMS) is performed during tracking rather than independently in every frame, allowing the tracker to recover the most likely locations in the light of *all* evidence, rather than the locally best guess per frame (*cf*. Section 4.2.3).

Despite the proposed extensions the resulting maximization of the posterior can still be written as an integer linear program (ILP), by an extension to the formulation of Berclaz et al. (2009). The ILP is solved efficiently through linear programming relaxation, in most cases to global optimality.

4.2 TRACKING ON A DISCRETE GRID

In the following we give a detailed description of the proposed multi-view tracking method. We start with the formulation of maximum a-posteriori trajectory estimation as an integer linear program (ILP). Next, we introduce the observation model, a probabilistic variant of *tracking-by-detection* designed for tracking targets observed from multiple viewpoints in world coordinates. Furthermore, we propose to include non-maxima suppression in the tracker, rather than viewing it as a preprocessing step. We then write the dynamic model as a local soft constraint, by penalizing the changes between consecutive motion vectors. In this form it can be re-introduced into the ILP-formulation of multi-target tracking. Finally, we move to an important technical issue: in the discrete setting the dynamic model suffers from grid aliasing, hence it is more effective to quantize locations to a hexagonal rather than a rectilinear grid.

4.2.1 *Tracking as integer linear program*

The proposed formulation extends the ILP-formulation of multi-target tracking introduced in recent work (Jiang et al., 2007; Zhang et al., 2008; Berclaz et al., 2009). The possible target locations are discretized to a finite set of sites $\mathbf{X}_u = (X_u, Y_u)$. Among those sites, a neighborhood system $\mathcal{S}$ is defined, where a site's neighbors $\{\mathbf{X}_v : v \in \mathcal{S}(u)\}$

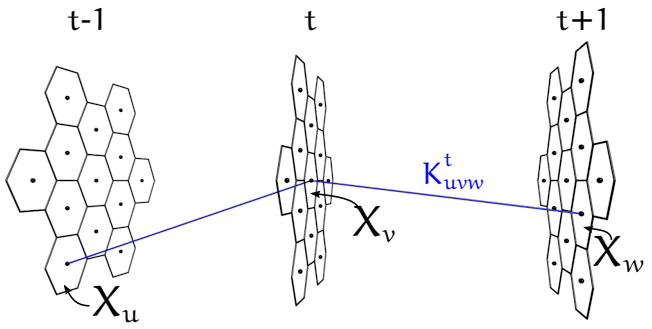


Figure 4.1: Illustration of a single tracklet B^t_{uvw} connecting neighboring cells in three consecutive frames.

are all sites than can be reached from $\mathbf{X}_u$ in a single time step (see Fig. 4.4).

Tracklets are triplets defining the target's locations in three consecutive frames.

Next, we define a *tracklet* K^t_{uvw} as an allowable path over 3 consecutive frames, *i.e.* a set of three sites

$$K^t_{uvw} = \{\mathbf{X}^{t-1}_u, \mathbf{X}^t_v, \mathbf{X}^{t+1}_w\} \tag{4.1}$$

such that $v \in \mathcal{S}(u)$ and $w \in \mathcal{S}(v)$. A single tracklet is illustrated in Figure 4.1. The set of all index triplets (uvw) for all frames that produce a valid tracklet is denoted $\mathbf{K}$. For each tracklet K^t_{uvw} there exists a corresponding indicator variable B^t_{uvw}. The set of all indicator variables is denoted $\mathbf{B}$. They are the variables of our optimization problem and take on values $B^t_{uvw} \in \{0, 1\}$, where $B^t_{uvw} = 1$ means that tracklet K^t_{uvw} is part of some trajectory, and $B^t_{uvw} = 0$ means that it is not part of any. The reason for introducing the tracklets is that the dynamic model cannot be included efficiently when operating directly on the sites $\mathbf{X}^t_u$, as will become clear in Section 4.2.4.

Based on the observed evidence $\mathbf{R}$, each indicator variable is assigned a goodness-of-fit

$$c^t_{uvw} = \log \frac{P(B^t_{uvw} = 1|\mathbf{R})}{P(B^t_{uvw} = 0|\mathbf{R})}, \tag{4.2}$$

which compares the hypotheses $B^t_{uvw} = 1$ and $B^t_{uvw} = 0$ in light of the observation model (Sec. 4.2.2) and the dynamic model (Sec. 4.2.4). Thus, multi-target tracking amounts to maximizing the log-odds-ratio of all indicator variables $\mathbf{B}$ under three additional constraints:

1. *continuity*: tracklets must form continuous trajectories – whenever a certain tracklet is used in a solution, *i.e.* $B^t_{uvw} = 1$, there must be exactly one tracklet K^{t+1}_{vwz} in the next time step, which is also used. Targets entering or leaving the tracking area are modeled by two virtual *source* and *sink* sites, which are neighbors of all boundary sites and can emit, respectively absorb, targets.

2. *collision avoidance*: no two tracklets can have the same midpoint $\mathbf{X}^t_v$; whenever a tracklet K^t_{uvw} is selected, all other tracklets K^t_{svz} must be discarded.

3. *extended collision avoidance*: no two tracklets can start in directly adjacent cells; whenever a tracklet K^t_{uvw} is selected, all other tracklets K^t_{svw} with $s \in \mathcal{S}(u)$ must be discarded.

This results in the following optimization problem with the vector $\mathbf{B}$ of all indicator variables B^t_{uvw} as argument:

$$\mathbf{B}^*_{\mathrm{ILP}} = \arg\max_{\mathbf{B}} \sum_{uvw \in \mathbf{K}, t} \left(c^t_{uvw} \cdot B^t_{uvw} \right) \tag{4.3}$$

subject to:

$$\sum_{s:svw \in \mathbf{K}} B^t_{svw} = \sum_{z:vwz \in \mathbf{K}} B^{t+1}_{vwz} \qquad \text{(continuity)} \tag{4.4}$$

$$\sum_{s,z:svz \in \mathbf{K}} B^t_{svz} \leqslant 1 \qquad \text{(collision avoidance)} \tag{4.5}$$

$$\sum_{s \in \mathcal{S}(u)} B^t_{svw} \leqslant 1 \qquad \text{(ext. collision avoidance)} \tag{4.6}$$

$$B^t_{uvw} \in \{0, 1\} \qquad \text{(domain of variables)} \tag{4.7}$$

$$\forall\, uvw \in \mathbf{K}, t.$$

We will discuss the motivation behind the additional exclusion constraints (Eq. (4.6)) later in Section 4.2.3.

OPTIMIZATION. Maximizing Eq. (4.3-4.7) belongs to the class of integer linear programs, which are hard to optimize in general. A common way to address the combinatorial complexity in practice is to relax it to a linear program by replacing the condition $B^t_{uvw} \in \{0, 1\}$ with $0 \leqslant B^t_{uvw} \leqslant 1$. The relaxed problem can be efficiently solved with the simplex algorithm or an interior-point method. Moreover, if all variables B^t_{uvw} at the relaxed optimum $\mathbf{B}^*_{\mathrm{LP}}$ take on integer values, then it is also a global optimum of the original problem, $\mathbf{B}^*_{\mathrm{LP}} = \mathbf{B}^*_{\mathrm{ILP}}$. Even if the solution is not completely integral, then in practice the optimality gap is small, and only a tiny fraction of non-integer variables remains (in our experiments $< 0.2\%$), and these are clustered in relatively small connected components of the neighborhood system. Hence, an optimum of the ILP can be found using a branch-and-cut method with the relaxation as bounding function ("mixed integer programming"), or by "probing", *i.e.* rounding some non-integer values and solving for the others while monitoring the objective value C (a similar strategy is known as QPBO-P in the graph cuts context (Rother et al., 2007)). The branch-and-cut algorithm that was used in our case is outlined in Section A.1.

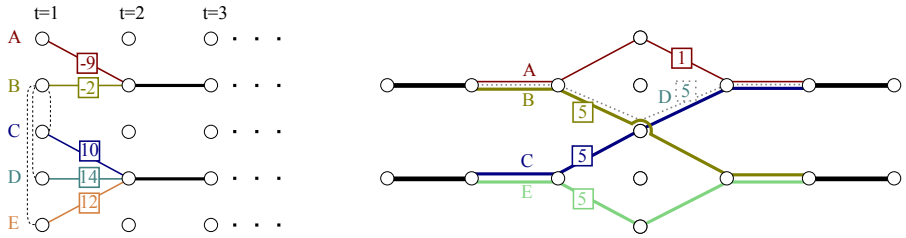


Figure 4.2: LP-relaxation can yield non-integral solutions due to the additional exclusion constraints *(left)* or due to graph pruning *(right)*. See text for a detailed discussion.

The maximum a-posteriori set of trajectories $\mathbf{K}^*_{\mathrm{ILP}}$ over the observed time window Φ corresponds to the set of variables from $\mathbf{B}^*_{\mathrm{ILP}}$ with the value of 1:

$$\mathbf{K}^*_{\mathrm{ILP}} = \left\{(uvw), t | B^t_{uvw} = 1\right\} \tag{4.8}$$

In practice, the time interval Φ is bounded by the available storage and computation power. The number M of variables and constraints to be stored grows linearly with Φ, and the average-case computational complexity of LP-solvers is polynomial (in practice even linear) in M, too (see *e.g.* Gondzio, 2012). A practical solution is to solve Eq. (4.3) for overlapping time intervals and constrain the solutions to be consistent by fixing the first frame. Empirically, intervals of $\Phi = 30$ frames are sufficient.

HALF-INTEGRAL SOLUTIONS. Our experience shows that the LP-solution $\mathbf{B}^*_{\mathrm{LP}}$ is integral in most cases. This is, perhaps, not surprising, since all indicator variables B_Q on a junction-free path Q have the same value, *i.e.* they are either all $B_Q = 0$ or $B_Q = 1$ because of the continuity constraint; B_Q will always be integral, because the total contribution of the path to the objective value is $B_Q \sum c_Q$, which attains its maximum at $B_Q = 0$ for $\sum c_Q \leqslant 0$, and at $B_Q = 1$ for $\sum c_Q > 0$. If a path were to split into two branches Q and R at any point and converge again at a later point, then one branch would get all the weight, whereas the other would be suppressed. Only in a pathological case, where the likelihood of both branches is identical, any linear combination of $B_Q \sum c_Q + (1 - B_Q) \sum c_R$ will have the exact same objective value. This is, however, highly unlikely in practice.

$\mathbf{B}^*$ *is half-integral if all values are* 0, $\frac{1}{2}$ *or* 1, i.e. $2\mathbf{B}^*$ *is integral.*

We observed two types of scenarios, where the LP-relaxation yields half-integral solution. The first one (*cf*. Figure 4.2 *(left)*) occurs at the beginning (respectively at the end) of the temporal window. In this example, the optimal integer solution for the first frame chooses the paths A and D, which yields the objective value 5. Note that choosing B and D, *i.e.* setting $B_B = B_D = 1$, is not feasible since it would violate the exclusion constraint from Eq. (4.6). The global optimum of the LP-relaxation is 5.5 and is attained at $B_A = B_B = B_C = B_E = \frac{1}{2}$.

The second case (*cf*. Figure 4.2 *(right)*) only happens if the set of candidate tracklets is heuristically pruned (see Sec. 4.3). In the depicted situation, much of the contribution to the total objective resides between two similar paths that run in parallel alongside each other over several frames. However, an integer solution that includes both paths D and E cannot be found because the necessary set of tracklets D (dotted line) has been pruned away. Therefore, the optimal solution is again half-integral with $B_A = B_B = B_C = B_E = \frac{1}{2}$

4.2.2 *Observation model*

Tracking is formulated in world coordinates for the general case of multiple cameras observing the scene from different viewpoints. Multi-camera setups greatly improve tracking accuracy when the camera positions are low over the ground, such that one has to accept inaccurate depth estimates as well as frequent occlusions. Our framework includes single-view tracking as a special case, by setting the number of cameras to 1. As usual, the posterior is split into an observation likelihood and a motion prior. Furthermore, the observation is decomposed into two parts, measuring object detection response, respectively color similarity:

$$P(B^t_{uvw}=1|\mathbf{R}) \propto P_O(\mathbf{R}|B^t_{uvw}=1) \cdot P_A(\mathbf{R}|B^t_{uvw}=1) \cdot P(B_{uvw}=1)\ . \tag{4.9}$$

OBJECT DETECTION. To measure the support of targets in the image data, the popular HOG detector (Dalal and Triggs, 2005), which is described in Section 3.2, is employed. The detector scans the images I^t_v (taken from viewpoints $\mathbf{c}_v$ at all three frames of the tracklet) over all positions $\mathbf{x}$ and scales s with a binary classifier trained to discriminate people from background, and returns for every location and scale a classification score R^t_v. The scores are mapped from image locations $(\mathbf{x}, s)$ to locations $\mathbf{X}$ and target heights h in the world coordinate system with appropriate projections, and aggregated over all views and the three frames to obtain the total evidence $\mathbf{R}$ for a tracklet.

The evidence at this point depends not only on B^t_{uvw}, but also on the person height h, via the detection scale s. In principle, one could track directly in the $(\mathbf{X}, h)$-space, with a constraint that the height of any given person should not change over time. To reduce the computational burden, a Gaussian prior is placed on the person height instead, and marginalized out,

$$P_O(\mathbf{R}|B^t_{uvw}=1) = \int \left(P_O(\mathbf{R}|B^t_{uvw}=1, h) \cdot P(h)\right)\ \mathrm{d}h. \tag{4.10}$$

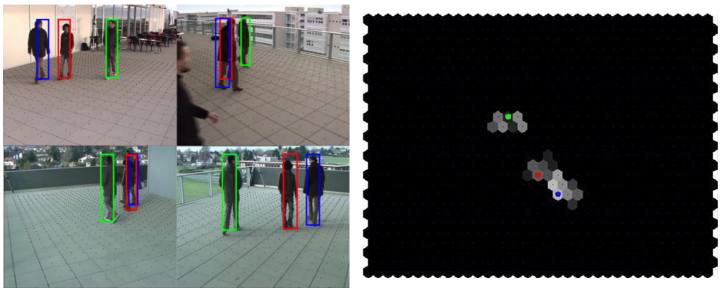


Figure 4.3: The evidence $P(B^t_{uvw}|\mathbf{R})$ has smooth peaks, which are not precisely localized. *(left)* tracking results in four views. *(right)* birds-eye view of the scene. Note that the correct position for the green subject is *not* the one with the highest score. The presented algorithm avoids per-frame decisions and chooses the best location during tracking.

The Bhattacharyya coefficient for two normalized histograms is defined as $\sum_i \sqrt{H^1_i \cdot H^2_i}$.

APPEARANCE. The generic object model is complemented with a target-specific appearance model to better distinguish different targets. To this end, we demand that the color distribution of a target varies slowly over short time spans. All sites of a tracklet K^t_{uvw} are projected back to the respective image locations $\mathbf{x}$, and at each location a color histogram is extracted. The histograms of consecutive sites in a tracklet are then compared with the Bhattacharyya coefficient d_B, and the results are combined over all pairs of sites and all viewpoints $\mathbf{c}_v$:

$$P_A(\mathbf{R}|B^t_{uvw}=1) \propto \prod_{\mathbf{c}_v} \exp\left(-\frac{d_B(\mathbf{x}^{t-1}_u, \mathbf{x}^t_v) + d_B(\mathbf{x}^t_v, \mathbf{x}^{t+1}_w)}{\sigma^2_B}\right). \quad (4.11)$$

4.2.3 *Exclusion constraints*

Exclusion constraints between different tracklets ensure plausible interactions between the targets. The simplest form of constraint, which has been widely used in multi-target tracking, is the *collision avoidance* implemented by Eq. (4.5). However, exclusion constraints can also be applied over larger neighborhoods, to incorporate NMS in the tracking framework rather than do it at the frame level, such that the retained location is the one which is optimal for the entire time interval, rather than for a single frame.

A main limitation of most tracking schemes is that non-maxima suppression is carried out on a per-frame basis. The evidence $P(\mathbf{R}|B^t_{uvw})$ measured by the observation model is in practice not a set of perfect spikes, but a smooth distribution with peaks, which are not well localized, see Figure 4.3. To remedy this, the distribution is replaced by its modes, found by some mode-seeking procedure like mean-shift or morphological erosion. Traditional non-maxima suppression thus commits to a location without taking into account the fact that target locations should be consistent over time. Instead, we propose to

integrate NMS into tracking, rather than detection: the detector output is left to be ambiguous around the modes, and the optimization can choose which location is most likely, given also evidence from neighboring frames and the dynamic model.

However, in this context an additional difficulty arises: some form of "sharpening" of the modes is required, otherwise targets with strong image evidence will have one or several "ghosts" following them along their trajectory. These false positives link the weaker, but nevertheless strong evidence belonging to the same mode. In other words, a prior is required that formalizes the intuition that plaits of intertwined trajectories are unlikely. To this end, a number of additional constraints (*cf*. Eq. (4.6)) are introduced, which prohibit not only collisions of targets at the *same* location, but also tracklets starting at immediately *neighboring* locations (which amounts to the assumption that the grid sampling distance is smaller than the minimal possible distance between two targets). These constraints prevent targets from moving too close to one another, and also avoid trajectories crossing in such a way that a collision would happen in the empty space between two grid locations.

It is important to note that the effect of the prior is not the same as single-frame NMS: under the exclusion constraints the optimization is free to choose a target location $\mathbf{X}^t$, which is *not* a maximum of the detection score in frame t, in order to achieve a smoother trajectory, or to avoid collisions with other targets.

4.2.4 *Dynamic model*

An important component of tracking is the dynamic model, which encodes prior knowledge about likely motion patterns of the tracked objects. Using such dynamic models – mostly assuming constant heading, constant velocity or constant acceleration – has a long and successful tradition, however such models have been dropped in grid-based tracking (Berclaz et al., 2006, 2009).

To overcome this, we extend the grid-based formulation to incorporate the *constant heading* model, *i.e*. we assume that objects tend not to change their motion *direction*. A prerequisite for the ILP formulation is that the objective function Eq. (4.3) be linear. To preserve the linearity, the motion prior $P(B^t_{uvw} = 1)$ must be formulated such that it can be computed *locally for each variable* (*i.e*. its contribution must be part of the unary terms). This is the reason why we have introduced the *tracklets*: checking for constant heading requires two consecutive motion vectors, and hence three consecutive sites, thus the variables

must cover at least three consecutive frames. Given the two motion vectors

$$\mathbf{m}_{uv} = \begin{pmatrix} X_v - X_u \\ Y_v - Y_u \\ 1 \end{pmatrix}, \qquad \mathbf{m}_{vw} = \begin{pmatrix} X_w - X_v \\ Y_w - Y_v \\ 1 \end{pmatrix}$$

in a tracklet K^t_{uvw}, one can model the prior by penalizing the heading change α between them, measured in (X, Y, t)-space. The tracklet is assigned a probability that grows inversely with α^2, such that deviations from the constant-heading assumption are penalized, as desired:

$$P(B^t_{uvw}=1) \propto \exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2}\right), \tag{4.12}$$

where

$$\alpha = \arccos \frac{\mathbf{m}_{uv}^\top \mathbf{m}_{vw}}{|\mathbf{m}_{uv}||\mathbf{m}_{vw}|}. \tag{4.13}$$

Note that the angle α is computed in (X, Y, t)-space. The method can be trivially extended to favor constant *velocity* by penalizing the difference between $\mathbf{m}_{uv}$ and $\mathbf{m}_{vw}$, however we found the angle to work better, probably because of the varying step length on a discrete grid.

The obvious effect of the dynamic model is that smoother, more accurate trajectories are estimated in the presence of inaccurate or weak evidence. Beyond its original purpose, the dynamic model also has a more subtle benefit on the optimization: by penalizing tracklets with strong heading changes, the motion prior sharpens the posterior, and thus the objective function C. As a consequence, the relaxation gap narrows, and fewer non-integer values occur. This effect is particularly strong in difficult circumstances, when the evidence $P(\mathbf{R}|B^t_{uvw} = 1)$ is rather flat, such that the potential target locations spread out over a large number of tracklets. Therefore the dynamic model drastically reduces computation time (in our experiments by at least a factor of 10). In some cases the number of non-integer values without motion prior even becomes so high that it is no longer tractable to find an integral solution with branch-and-cut or probing.

4.2.5 *Hexagonal discretization*

To make tracking amenable to global optimization with ILP, in the spirit of Jiang et al. (2007); Berclaz et al. (2009), the location space $\mathbf{X}$ must be discretized to a finite set of locations. As explained above, the presented method does not heuristically prune the per-frame likelihood $P(\mathbf{R}|B^t_{uvw})$ to a small set of permissible locations, but rather samples the ground plane in a regular lattice. A natural choice, which

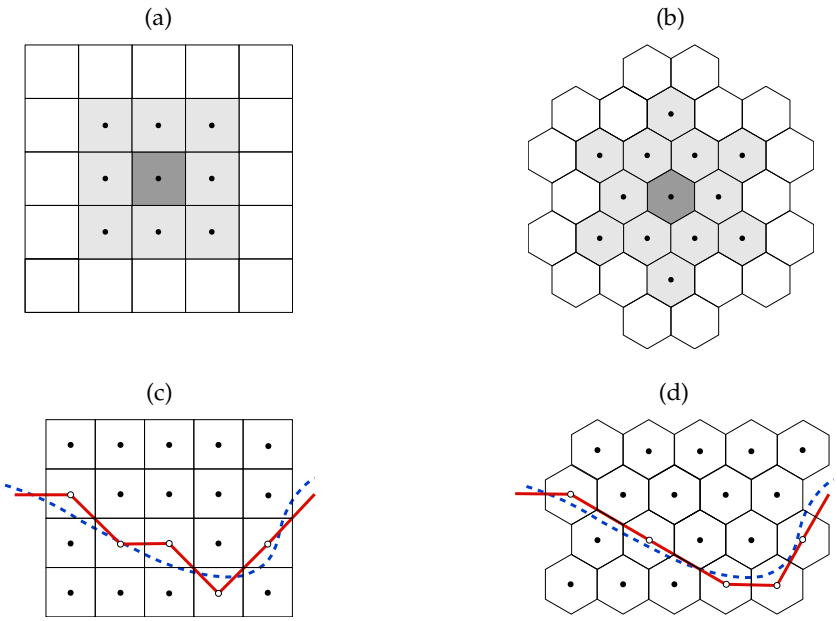


Figure 4.4: The 8-neighborhood *(a)* and the 12-neighborhood *(b)* in a rectilinear and in a hexagonal tiling, respectively. The bottom row shows the aliasing effect of an example trajectory on a rectangular grid *(c)* and a hexagonal grid *(d)* with the same sample density.

has been used in previous work, is a rectilinear grid, similar to the image grid. Unfortunately, such a grid has a strong preference for the two canonical directions along the x- and y-axes, whereas target trajectories in other directions exhibit severe aliasing.

Aliasing is not a big problem in the absence of a dynamic model, but together with the proposed motion model it creates difficulties: to check the deviation from constant heading *locally*, one needs to rely on the vectors between the grid locations, thereby penalizing trajectories which are not grid-aligned and hence continuously change directions. To alleviate this effect and boost the positive effect of the dynamic model, we propose to use instead a hexagonal tiling of the ground plane, inducing a tri-axial neighborhood system. In this grid, the 8-neighborhood is replaced by a 12-neighborhood, which reduces staircasing artifacts, and allows one to better enforce the constant heading assumption, see Figure 4.4. The hexagonal tiling has been used in other contexts in image processing and computer vision (Miller, 1999; Middleton and Sivaswamy, 2005), precisely because it has more preferred directions and reduces aliasing artifacts. Note that the change of sampling grid does not impair data quality: the transformation is performed when mapping the target probabilities from images to the world coordinate system, so there is no additional resampling step that would further blur the data.

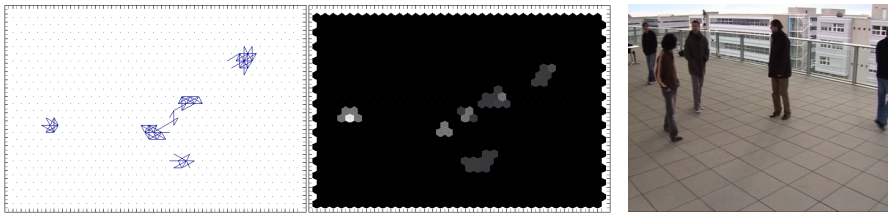


Figure 4.5: An example of graph pruning. All tracklets with low likelihood *(middle)* are removed from the solution space *(left)*. The corresponding frame is shown on the right.

4.3 IMPLEMENTATION

Before discussing the experimental setup, let us briefly turn to some implementation issues.

PRUNING AND CACHING. Recalling the MAP formulation from Eq. (4.3), the goal is to find a vector $\mathbf{B}^*$ that maximizes the objective function subject to certain constraints. The binary vector $\mathbf{B}$ corresponds to the set $\mathbf{K}$ of all possible triples between neighboring cells for each frame within a time window. Assuming a lattice size of $G_x \times G_y$ cells and a sequence length Φ, the dimensionality of $\mathbf{B}$ is approximately

$$|\mathbf{B}| \approx G_x \times G_y \times |\mathcal{S}|^2 \times \Phi, \tag{4.14}$$

where $|\mathcal{S}|$ is the number of neighbors of each cell. In a typical setting with a 50×50 grid, $\Phi = 30$ frames and a 12-neighborhood, the length of the parameter vector of the optimization problem amounts to $|\mathbf{B}| \approx 10^7$. The number of constraints is much lower since it is not dependent on the cardinality of the chosen neighborhood. Nonetheless, the problem is too large to be handled at once, even for modern LP-solvers.

To reduce the problem size, a pruning technique is employed. The main assumption is that low-likelihood tracklets that have no high detection score in their vicinity will not be part of the final solution and may be discarded a-priori. In our implementation, we remove a tracklet K^t_{uvw} from the solution space if

1. $c^t_{uvw} < \theta_a$ *and*
2. $\max_{\mathcal{S}(v)} P(\mathbf{R}|B^t_{uvw} = 1) < \theta_b$.

An example frame is shown in Figure 4.5. This strategy reduces the number of variables by over 90%, allowing for efficient optimization. Unfortunately, this may also lead to pruning too many tracklets in occluded areas leading to non-optimal solutions.

A large fraction of the entire optimization process is occupied by constructing the variables (*i.e.* tracklets) and the corresponding constraints. This step can be significantly sped up if the lattice remains unchanged over time, which is the case here. Therefore, we found it beneficial to save the tracklet indices and the constraint matrices once they are computed and to load them for all successive time windows.

LINEAR PROGRAMMING SOLVERS. We experimented with several software packages to solve the ILP problem from Eq. (4.3). Since the entire framework is developed in MATLAB, our first choice was its native linear programming routine `linprog`. Unfortunately it turned out to be rather slow for our problem size, even after excessive pruning. Moreover, it only solves the relaxed version of the problem such that further iterations are needed to resolve the non-integer values of the obtained solution. The binary integer programming solver `bintprog` showed even worse performance in terms of computation time and could not be used in our case.

The GNU Linear Programming Kit (GLPK)[1] offers an acceptable alternative to MATLAB's methods. This open source package is suitable for large-scale optimization problems and offers a MATLAB interface, which allows a fast and uncomplicated integration into existing projects. An even better option is the SCIP[2] software package (Achterberg, 2009). Using pre-processing heuristics, it is approximately four times faster than GLPK when solving (mixed) integer problems and can be used free of charge for academic purposes. Moreover, its interface allows one to replace the native linear solver by another one that may be more suitable for a specific problem.

The speed-up is stated according to the official website and our experience.

4.4 EXPERIMENTS

In this section, experiments on five different public multi-view video sequences are presented. Sequences *campus1* and *campus2* (Berclaz et al., 2006) were both recorded from 3 different camera viewpoints, and have 2000, respectively 1400 frames showing up to 6 people moving outdoors. Sequences *terrace1* and *terrace2* (Fleuret et al., 2008) were both recorded from 4 viewpoints, and have 2000 frames each with up to 6 people, also moving freely outdoors. Finally, for monocular tracking we use the first view of the sequence *PETS-S2L1* from the PETS 2009 benchmark. The sequence is better suited for single-view tracking because of the elevated viewpoint. The entire dataset contains 52 individual trajectories, which were manually annotated and used as ground truth. Please refer to Section 3.3 for more details on the chosen datasets.

1 http://www.gnu.org/software/glpk
2 http://scip.zib.de

We set the grid resolution to 35cm in world units. The size of the tracking area in the two terrace sequences is approximately 7.8×11.0 meters yielding a grid size of $23 \times 32 = 736$ cells for the hexagonal case. Both campus sequences have similar extents and require a discretization in $28 \times 28 = 784$ locations. The PETS sequence shows a much larger area of $\approx 19 \times 15.8$ meters. Consequently, the same resolution requires a grid size of $55 \times 53 = 2915$ individual cells per frame. Due to the low target speed, we processed only every other frame of *PETS-S2L1* and every 6^{th} frame in the remaining four sequences, such that targets move approximately one grid unit from one frame to the next.

All experiments have been carried out with the same set of parameters. The two free parameters of our method are the standard deviations σ_α and σ_B, which govern the relative influence of detection score, color similarity, and dynamic model (*cf*. Sections 4.2.2 and 4.2.4). To keep the optimization tractable for long sequences, we follow the usual strategy and process overlapping time windows. This adds two further parameters, the number of frames Φ per window, and the overlap Ω. We set $\Phi = 30$ (when processing every 6^{th} frame at 25 fps, this amounts to ≈ 7 seconds) and $\Omega = 10$.

4.4.1 *Qualitative Results*

Figure 4.6 shows example results from the sequences *terrace1* and *S2L1*. Targets are tracked successfully over many frames, new targets entering the scene are initialized automatically. Especially the middle column shows an example of many targets moving in a small space. People are often occluded simultaneously in several views. Long-term occlusion is a main cause of failure, such as for the person marked in cyan. Note that the corresponding trajectory is interrupted in the middle of the tracking area. The reason here is that the target steps outside the field of view of three out of four cameras, while at the same time being occluded in the remaining fourth view (*cf*. Frame 1748 in Figure 4.6).

Grid locations close to image borders allow for entering and exiting the tracking area.

In the *PETS-S2L1* sequence, up to 8 targets are tracked in *monocular* video over a large area of interest. Note the false positive on the tripod (light blue) near the image center: persistent false detections on background objects are the dominant cause of false positives, since they tend to appear frequently on the same structures and, being static, fulfill the constraints of the dynamic model.

4.4.2 *Comparison to previous work*

The trajectories estimated by the presented method are directly compared to those of Berclaz et al. (2009), which have been extracted from their published results. Their method is based on a similar ILP formu-

lation, but on a rectilinear grid. Moreover, the binary state variables in their formulation represent target motion that only extends over two consecutive frames, thereby precluding the use of any reasonable dynamic model.

Figure 4.7 shows sample trajectories from both methods, with similar grid resolutions. The middle row shows the results of Berclaz et al. (2009), while the plots on the bottom depict trajectories obtained with the proposed method. The examples illustrate how late non-maxima suppression, together with the dynamic model, avoids implausible jittering and produces trajectories that are more accurate with respect to annotated ground truth (top row). We emphasize that the improvement is due to the combination of all modeling choices: late non-maxima suppression preserves the necessary evidence for flexible target placement, while the dynamic model on a hexagonal grid supplies the constraints to handle the extra flexibility.

4.4.3 *Quantitative evaluation*

In the following, our tracker is quantitatively evaluated against the baseline ILP tracker without dynamic model and operating on a rectilinear grid with either the standard 9-neighborhood (8 neighbors and the central location itself) or a larger 21-neighborhood. We use our own implementation for all experiments. To quantify the performance of the presented tracker, three types of metrics are employed. The popular CLEAR MOT metrics are thoroughly discussed in Section 3.4. We also report the raw number of track fragmentations and identity switches. Moreover, we measure the smoothness of the estimated trajectories by the average angle between the segments of all tracklets. The evaluation results are summarized in Figure 4.8, where each metric type is presented in one row. The left column shows the average results on four multi-view sequences, while on the right the performance on the monocular PETS *S1L2* dataset is displayed. The number of neighbors for each method is indicated in parentheses.

The average angle between all trajectory segments measured in (X, Y, t)-space (*cf*. Eq. (4.13)) is plotted in the bottom row of Figure 4.8 for all three settings described above. As expected the discussed model *(dyn, hex)* greatly improves trajectory smoothness yielding an average angle of less than 10 degrees. Without the dynamic model the aliasing artifacts introduced by the discretization lead to many 90-degree or 135-degree turns, as illustrated in the middle row of Figure 4.7. Simply extending the neighborhood from 9 to 21 neighbors reduces the average angle by $\approx$ 30 to 50 percent because more tracklets with a finer granularity regarding their motion change are available. Unfortunately, this also leads to a higher computational cost. As we discussed in Section 4.1, the number of variables grows quadratically with the size of the chosen neighborhood. Our expe-

rience shows that going from a 9- to a 21-neighborhood takes ≈ 5 times longer to compute the solution due to the larger size of the ILP. The proposed hexagonal discretization hardly increases the neighborhood size, while at the same time providing a more flexible set of tracklets, which is favorable for a dynamic model. This combination of using a tri-axial grid and taking into account the targets' dynamics yields a five- to six-fold reduction in the average angle between consecutive trajectory segments. As discussed in the previous section, such smooth trajectories match the ground truth quite closely (*cf*. Figure 4.7).

The smoother trajectories also improve tracking accuracy as can bee seen in the top row of Figure 4.8. This can be explained by the fact that the smoothness prior mitigates the effect of inaccurate and uncertain evidence. If a target is partially occluded, the detector is likely to either produce a poorly localized response with respect to the actual target position, or fail completely providing only noise in the output. In this case, the baseline ILP formulation chooses the most likely target location for each cell that satisfies the linear constraints, but disregarding whether the target's motion remains physically plausible or not. In contrast, the proposed dynamic model favors smooth motion, which is more likely to occur in real-world situations, therefore improving the performance. In our experiments, Multiple Object Tracking Accuracy (MOTA) (measuring false negatives, false positives, and identity switches) increases by 10-20%. Using 21 instead of 9 neighbors also improves accuracy, but is still inferior to our result, while taking longer to compute, as already discussed above. Multiple Object Tracking Precision (MOTP) (measuring the localization error) improves insignificantly, because the metric is dominated by the alignment error due to the discrete location grid. One possible way to address this limitation is to describe trajectories in their natural continuous domain. Two approaches following this direction will be presented in Chapters 5 and 6.

Finally, there is a dramatic reduction of fragmented tracks and identity switches ($\approx 50\%$ for the monocular case, 80-90% for the multi-view case). The numbers for both scenarios are shown in Figure 4.7 *(middle row)*. Trajectory fragments are generated when the tracker drifts away from a target, which is less likely if late non-maxima suppression and the motion prior can correct inaccuracies of the evidence. Identity switches happen when data association fails for targets very close to one another. This is especially prominent when two persons with similar appearance walk alongside each other for an extended time interval. In such a case, the tracker of each target can switch to the other target and then back again several times, causing multiple errors. The motion prior improves correct data association, because it favors the option with more plausible dynamics.

4.5 DISCUSSION

This chapter presented an algorithm for tracking a varying number of targets on a discrete location grid. Multi-target tracking is cast as integer linear programming, and solved through LP-relaxation, in most cases to global optimality. The remaining non-integer values are resolved using branch-and-cut methods (see Section A.1). Compared to previous research in this direction, we argue that tracking should use the original target evidence as input and perform non-maxima suppression during trajectory estimation. Moreover, an approach to include standard dynamic models in the ILP-formulation is demonstrated. As expected, best results are achieved on a hexagonal rather than a rectilinear grid in this setting. The experimental comparison on public benchmark videos confirms that beyond its theoretical appeal the proposed formulation delivers better tracking results and achieves superior performance in quantitative comparisons.

However, the question remains whether the benefits of global optimization outweigh the rather strong restriction that is posed on the state space. The next chapter will address this very issue by dropping the discretization and shifting entirely to a continuous formulation. In particular, a more accurate and less restrictive objective function is developed at the cost of global optimality.

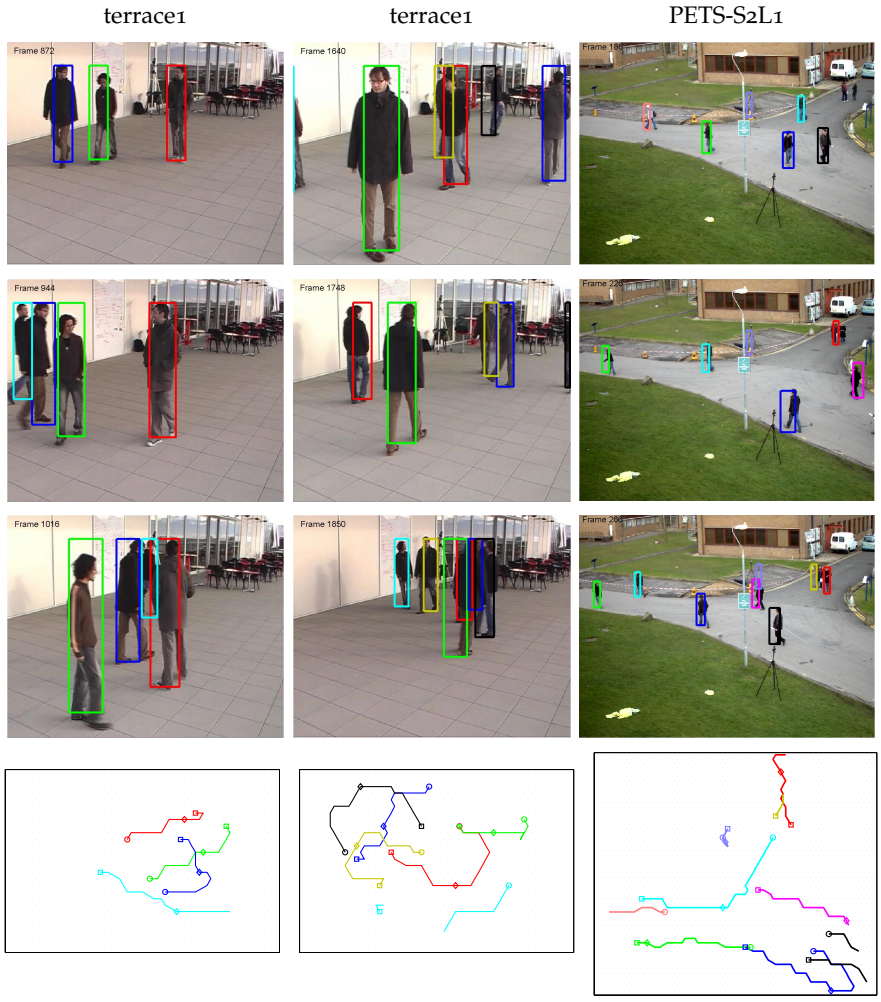


Figure 4.6: Tracking results obtained with the proposed ILP algorithm. The left and middle columns are from the *terrace1* sequence, the right column is from *PETS-S2L1*. Displayed are three sample frames (1^{st}-3^{rd} row), and a bird's-eye view of target trajectories (last row). The displayed frames are marked (top ◯, middle ♢, bottom □). See Section 4.4.1 for details.

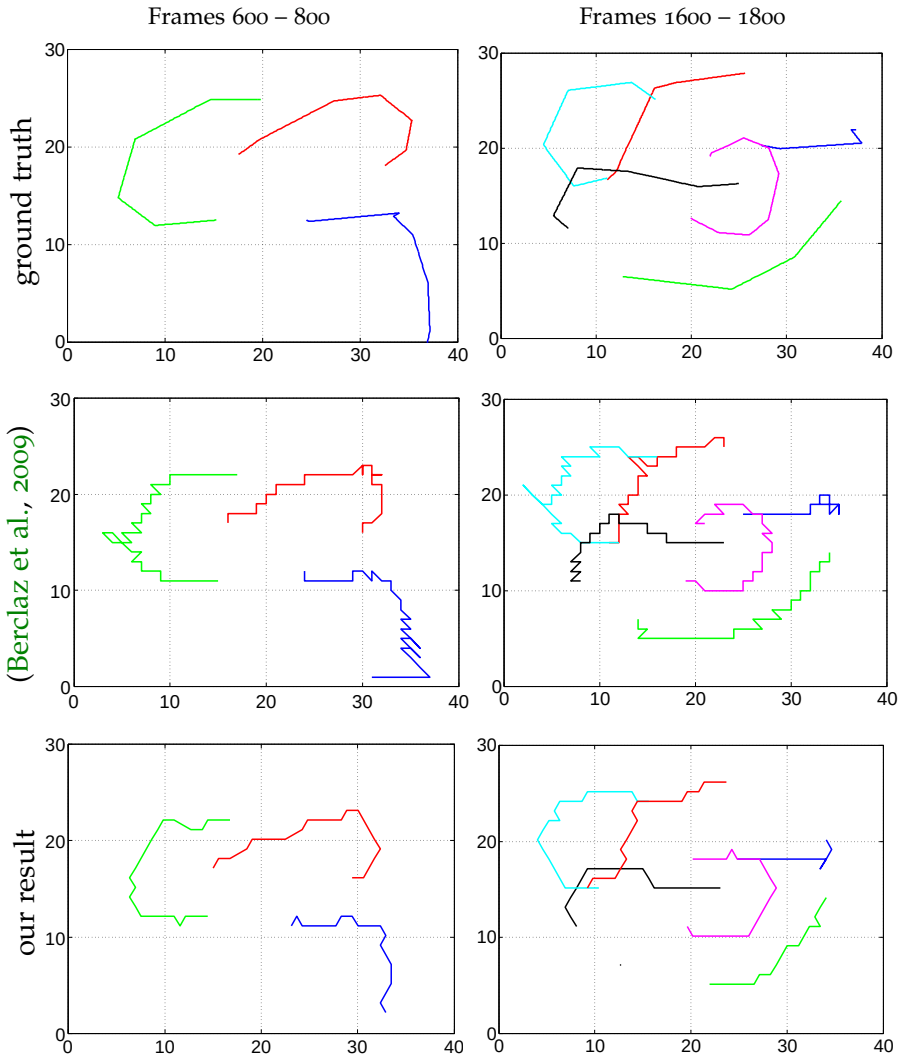


Figure 4.7: Improved trajectories with the proposed model. *(top)* manually annotated ground truth for 200 frames of sequence *terrace1*. *(center)* trajectories reconstructed by state-of-the-art tracking *without* dynamic model (Berclaz et al., 2009). *(bottom)* trajectories estimated by the presented system with dynamic model on a hexagonal grid.

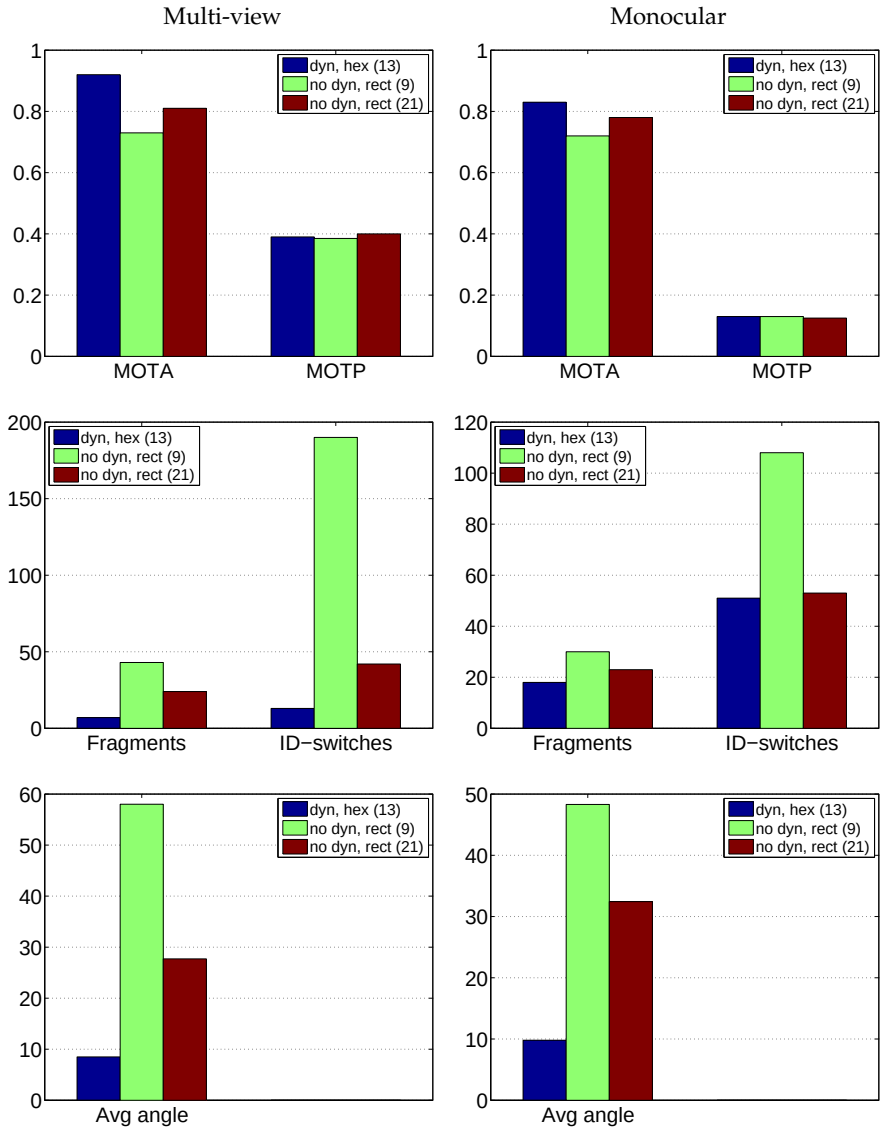
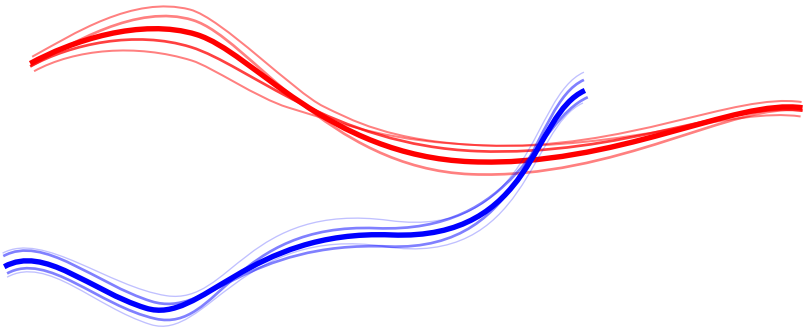


Figure 4.8: Tracking performance. *(top)* CLEAR metrics – higher is better. *(middle)* fragmentation and ID switches – lower is better. *(bottom)* smoothness – lower is better. Globally optimal tracking benefits significantly from dynamic models on the hexagonal grid, both in multi-view *(left column)* and in the monocular setting *(right column)*.

Part II

TRACKING IN CONTINUOUS SPACE

Each target is represented by its continuous coordinates and a high-dimensional non-convex energy function is designed to naturally describe the problem of multi-target tracking.



5

TRACKING MULTIPLE TARGETS BY CONTINUOUS ENERGY MINIMIZATION

The truth will set you free, but first it will make you miserable.

JAMES A. GARFIELD

CONTENTS

In the previous chapter we discussed a multi-target tracking approach based on minimization of a discrete objective function. More precisely, all targets were restricted to move across disjoint cells of a regular lattice. Short 3-frame tracklets were represented by binary variables and combined with linear constraints to formulate an integer linear program (ILP). Fortunately, due to the nature of the problem, the LP-relaxation led to globally optimal solutions in most cases.

This chapter presents a rather different approach. We pose the question whether global optimality of a necessarily approximate model is a suitable strategy to follow. In particular, a more complex energy function is defined in continuous space to explicitly address many crucial aspects of multi-target tracking, which have previously been

forgone to preserve (near) global optimality of the solution. Moreover, a fast minimization scheme based on gradient descent and greedy discontinuous jumps is presented to explore scattered areas of the solution space.

The presented formulation of a continuous energy function for multi-target tracking including a global occlusion model has previously been published in (Andriyenko and Schindler, 2011; Andriyenko et al., 2011). An extended version of this work has been submitted to the *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* for a second revision and is currently under review.

5.1 INTRODUCTION

Multi-target tracking is relevant for many important applications (*cf*. Section 1.2). Despite enormous progress in recent years, the task of robustly keeping track of targets over time to date remains challenging, particularly in many real-world scenarios. Noisy observations, occlusions and the combinatorial problem of *data association* are some of the many challenges to overcome. Moreover, motion, appearance, and visibility of objects are influenced by mutual dependencies that have to be taken into account. From a probabilistic point of view this entails inference – often maximum a-posteriori (MAP) – in a posterior distribution over several variables that have complex dependencies. As we already discussed in Section 1.1, the resulting optimization problem is highly non-convex (in case of a continuous domain) or non-submodular (in the discrete case), and thus cannot be optimized globally without major simplifying assumptions.

Yet, several recent multi-target tracking formulations aim to obtain a (nearly) globally optimal set of trajectories within a temporal window (Jiang et al., 2007; Zhang et al., 2008; Berclaz et al., 2009; Andriyenko and Schindler, 2010; Pirsiavash et al., 2011; Henriques et al., 2011; Ben Shitrit et al., 2011). We discussed one of these approaches in the previous chapter. In order to make (near) global optimization possible and efficient, the state space is reduced by restricting possible target locations to a finite set and the energy function is simplified. While global optimality undoubtedly has many benefits, we must also not lose sight of the actual purpose of formulating multi-target tracking as an energy minimization problem: the energy should adequately reflect the task at hand so that low-energy solutions are close to the true situation. Unfortunately, in the realm of multi-target tracking, typical specifications of the desirable aspects do not lead to models that can be globally optimized.

In this chapter, we investigate the question whether it is really beneficial for multi-object tracking to find the global minimum of an overly restricted energy function. In contrast to previous work, the goal of this chapter is to design the objective function such that it

offers a more complete representation of the various aspects of the problem. The energy is defined in continuous space and depends on the locations and motion of *all* targets in *all* frames, including cases where image evidence is missing, and explicitly includes physical constraints, such as smoothness of motion and mutual exclusion. It is beneficial to model these terms in the continuous domain, since this allows describing the true situation more closely than in a discrete setting. The price to pay is having to forgo global optimality, since such a complex model of multi-target tracking is unlikely to be convex or submodular. Nevertheless, local optima of our energy yield better results in practice, both visually and in terms of a quantitative evaluation with respect to ground truth.

To make the optimization efficient, all energy terms are formulated as functions that can be computed and differentiated in closed form. Hence, computationally efficient gradient-based optimization methods can be applied. To find strong local minima and to reduce the influence of the initialization, we run standard conjugate gradient descent from several starting points. Additionally, this purely continuous minimization is extended by a set of trans-dimensional jump moves, which enable the search to escape the initial basin of attraction and explore a larger region of the energy landscape (see Figure 5.1 for an illustration). To support the claim that accurate modeling might be more important than optimality guarantees for tracking performance, we run extensive experiments on various public datasets and show state-of-the-art results quantitatively measured by standard multi-target tracking metrics.

The main contribution of this chapter is an energy-based model of multi-target tracking that

- is defined over all target locations (in continuous space) and all video frames in a given time window;
- includes per-frame detection evidence, appearance, dynamics, persistence, and collision avoidance;
- explicitly handles partial as well as full inter-object occlusion; and
- can be computed and differentiated efficiently in closed form.

Furthermore, we provide an empirical study on the influence of all major parameters of the model, and an analysis of various optimization strategies for model inference, ranging from greedy search to more randomized and sampling-based algorithms.

5.2 MULTI-TARGET TRACKING IN CONTINUOUS SPACE

Although the notation was already formally introduced in Section 3.1 and summarized in Table 3.1, the main components are briefly re-

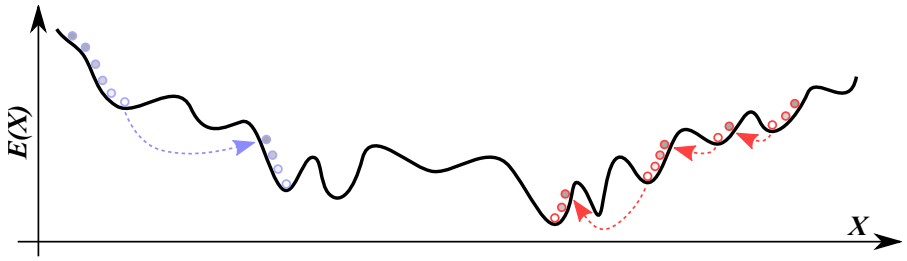


Figure 5.1: A schematic illustration of our continuous energy minimization. Starting from different initializations (red and blue), the energy is minimized by alternating between conjugate gradient descent (circles) and discontinuous jumps (dotted arrows).

viewed at this point for the reader's convenience. The state vector $\mathbf{X}$ consists of (X, Y) coordinates of all N targets in all F frames. Subscripts and superscripts ($\mathbf{X}_i^t$) are used to refer to a specific target at a certain time. Finally, to differentiate between world coordinates and image space coordinates, we use capital letters for the former, and lowercase for the latter.

5.2.1 *Continuous energy*

Although by no means the only way of performing inference, energy minimization methods – in one form or another – have become quite popular for multi-target tracking (Leibe et al., 2007; Zhang et al., 2008; Berclaz et al., 2009; Rodriguez et al., 2011). Their common goal is to set up a function that assigns every possible solution a cost (the "energy") and then (approximately) find the state with the lowest cost. An energy function for a certain application can be defined in many ways.

In computer vision one often faces two major problems: (1) The input data is "noisy" and requires robust models; (2) an accurate representation that captures all relevant nuances of the real situation quickly becomes very complex. Together, these two issues tend to produce complicated and highly non-convex objective functions (*cf*. Section 5.3). One is thus faced with a dilemma: Should the energy function be simplified until it is easily optimizable, or should it rather have the power to capture the complex situation, at the cost of less graceful mathematical properties? In the current chapter, we investigate the latter alternative for the case of tracking multiple objects in video. The energy we propose has been developed with an emphasis on precisely describing multi-object tracking. Algorithmic considerations were limited to keeping the function differentiable in closed form and thus efficient for gradient-based optimization. It turns out that for the case of multi-target tracking such an approach is rather successful.

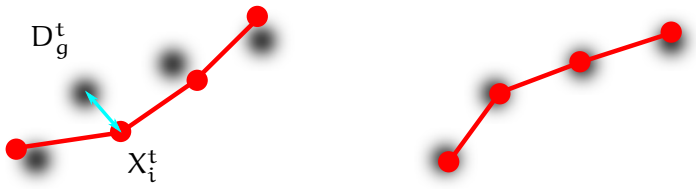


Figure 5.2: The observation model minimizes the distance between the detections (gray blobs) and the estimated trajectories.

Our energy function is a linear combination of six individual terms:

$$E = E_{det} + \alpha E_{app} + \beta E_{dyn} + \gamma E_{exc} + \delta E_{per} + \epsilon E_{reg}. \tag{5.1}$$

The data term E_{det} keeps the solution close to the observations; the term E_{app} captures the appearance of different objects to disambiguate data association; the three priors E_{dyn}, E_{exc} and E_{per} promote plausible motion and enforce physical constraints; the regularizer E_{reg} keeps the solution simple and prevents over-fitting. The aim is then to find the state $\mathbf{X}^*$ that minimizes the high dimensional continuous energy from Eq. (5.1):

The state vector $\mathbf{X}$ of the energy E consists of all (X, Y) coordinates of all targets over the entire video sequence.

$$\mathbf{X}^* = \arg\min_{\mathbf{X} \in \mathbb{R}^d} E(\mathbf{X}). \tag{5.2}$$

Depending on the length of the sequence and the number of targets, the dimension of the search space d normally takes on values between 10^3 and 10^4. In the remainder of this section we explain each individual term and its functionality in more depth. The influence of the individual terms is examined in Section 5.5.1 by adjusting their respective weights or discarding them entirely. The separate contribution of each term to the energy is visualized as heat maps in Figure 5.12 at the end of Section 5.2.3.

5.2.1.1 *Observation model*

In this work we concentrate on people as tracking targets, and follow the well established tracking-by-detection approach. Likely pedestrian locations are found with a sliding-window linear SVM detector (Dalal and Triggs, 2005; Walk et al., 2010a). More details can be found in Section 3.2. Detection peaks are determined by non-maxima suppression (NMS) and projected onto the ground plane of the world coordinate system, where they form the image evidence for tracking. We limit ourselves to using non-maxima suppressed detections to reduce the computational cost, but note that this is not a major limitation of our approach; it could easily be extended to use a per-pixel target likelihood instead (*cf*. Breitenstein et al., 2009). The intrinsic and extrinsic camera parameters required for the projection are constant for static cameras and can be inferred by structure-from-motion

for moving cameras (as done, *e.g.*, in Ess et al. (2009) for multi-target tracking). Hence, the requirement of a calibrated camera does not pose a major limitation and enables more accurate modeling of target dynamics and interaction.

The main purpose of the data term is to keep the trajectories close to the observations (*cf*. Figure 5.2). In other words, the energy should be minimal when the location of each target precisely matches a detection. To capture the localization uncertainty of the object detector, the energy smoothly increases with the distance between the estimated object location $\mathbf{X}_i^t$ and a detection location $\mathbf{D}_g^t$. This behavior is modeled by an isotropic (inverse) bell-shaped function centered at the detector output,

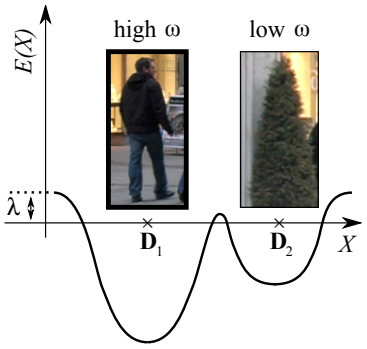


Figure 5.3: The term E_{det} in one dimension.

$$E^*_{det}(\mathbf{X}) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i} \left[\lambda - \sum_{g=1}^{D(t)} \omega_g^t \frac{s^2}{\|\mathbf{X}_i^t - \mathbf{D}_g^t\|^2 + s^2} \right]. \tag{5.3}$$

Each detection $\mathbf{D}_g^t$ is weighted by its confidence ω_g^t and the scalar s accounts for the object size, *i.e.* the area on the ground plane occupied by that object. It is set to 35cm for pedestrian tracking. The offset λ is added uniformly to all existing targets to penalize all those with no image evidence. This penalty, however, must not be applied if a target is occluded and consequently cannot possibly be "seen" by the detector. It is therefore scaled by the fraction of the visibility v_i^t of that target:

$$E_{det}(\mathbf{X}) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i} \left[v_i^t \cdot \lambda - \sum_{g=1}^{D(t)} \omega_g^t \frac{s^2}{\|\mathbf{X}_i^t - \mathbf{D}_g^t\|^2 + s^2} \right]. \tag{5.4}$$

One slice of the energy term E_{det} is illustrated schematically in Figure 5.3. The global occlusion reasoning including the computation of v is explained in detail in Section 5.2.2. We also defer the discussion of the appearance term to Section 5.2.3, as it relies on the visibility fraction of individual targets.

5.2.1.2 *Dynamic model*

A defining property of tracking (as opposed to independent object detection per frame) is that objects move slowly relative to the frame rate, and in most cases also smoothly. This gives rise to constraints on the target motion, captured by a dynamic model. A simple constant velocity model that minimizes the distance between consecutive

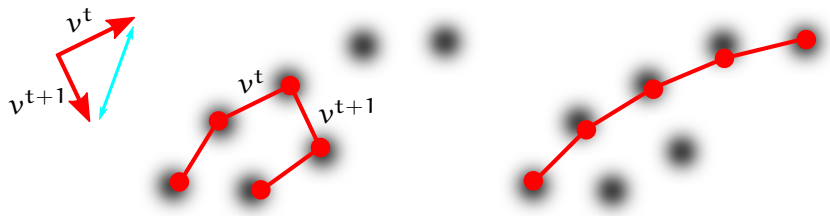


Figure 5.4: The dynamic model penalizes strong deviations between adjacent motion vectors and enforces constant velocity.

velocity vectors (*cf*. Figure 5.4) is powerful enough to capture the motion of objects in many real scenarios:

$$E_{dyn}(\mathbf{X}) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i-2} \left\| \mathbf{X}_i^t - 2\mathbf{X}_i^{t+1} + \mathbf{X}_i^{t+2} \right\|^2. \tag{5.5}$$

On one hand, the dynamic model helps reducing identity switches by favoring straight paths. On the other hand, the detections are often misaligned and do not form smooth curves. Naive smoothing might yield visually pleasing results, but is not appropriate to achieve high data fidelity and thus high tracking precision. The dynamic model as part of a global energy function can be seen as a form of "intelligent smoothing", yielding trajectories that are natural and smooth, while at the same time avoiding collisions and not drifting too far away from the actual observations.

5.2.1.3 *Mutual exclusion*

Collision avoidance is a crucial aspect when tracking multiple targets (*cf*. Section 5.5.1 and Fig. 5.17). In our model a continuous penalty is applied to configurations in which two targets come too close to each other:

$$E_{exc}(\mathbf{X}) = \sum_{t=1}^{F} \sum_{i,j \neq i}^{N(t)} \frac{s}{\|\mathbf{X}_i^t - \mathbf{X}_j^t\|^2}. \tag{5.6}$$

Note that the penalty is closely related to the intersection of the target volumes, which is also used by some authors (Ess et al., 2009), but our variant goes to infinity in the impossible case when both objects occupy the same 3D space. Besides enforcing the obvious physical constraint, a mutual exclusion term also ensures that one piece of image evidence can be explained by at most one target. This is especially important when dealing with soft likelihoods that exhibit a smooth falloff around the detection peaks (*i.e.*, target locations are not clamped to the exact location of the detector output), since otherwise the same peak could give rise to multiple trajectories (see Figure 5.5).

Our formulation of the exclusion model can handle two notoriously difficult problems: On one hand, the pairwise distance between all

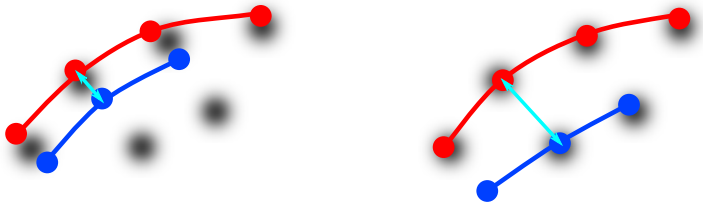


Figure 5.5: The exclusion model prevents collisions between targets by penalizing their mutual proximity.

targets is taken into account at all frames. Hence, two targets cannot occupy the same space, even if both are occluded. On the other hand, if one detection of two neighboring targets is missing, the targets will be pushed apart just as much as needed to avoid a physically impossible situation. Tracking on a discrete grid does not allow intermediate steps and the entire trajectory may be discarded.

Note that our approach does not perform an explicit assignment between target hypotheses and measurements (detections). Data association is indirectly achieved, mainly by two continuous terms – observation and mutual exclusion. Such soft assignments produce visually more pleasing trajectories due to their continuous representation. However, the concrete measurement-to-target assignment problem remains unsolved. We will see in Chapter 6 how both tasks can be approached simultaneously by energy minimization.

5.2.1.4 *Trajectory persistence*

Missing evidence can lead to track fragmentation or abrupt track termination in the middle of the tracking area. To encourage trajectories to start and end along image borders or along a predefined perimeter, tracks that do not obey this requirement are penalized. To keep the term both robust and smooth, we use a sigmoid centered on the border of the tracking area:

$$E_{\text{per}}(\mathbf{X}) = \sum_{\substack{i=1,\dots,N \\ t\in\{s_i,e_i\}}} \frac{1}{1+\exp\left(-q\cdot b(\mathbf{X}_i^t)+1\right)}, \tag{5.7}$$

where $b(\mathbf{X}_i^{s_i})$ and $b(\mathbf{X}_i^{e_i})$ measure the distance of the first, respectively last known location of target i to the closest border of the tracking area and the parameter q represents the soft entry margin and is set to $q = 1/s$ in all experiments.

5.2.1.5 *Regularizer*

Finally, a regularizer is needed to prevent the number of targets from growing arbitrarily large so as to better fit the data. To that end, we simply penalize the number of existing targets. It turns out that including the trajectory length into the regularization term leads to

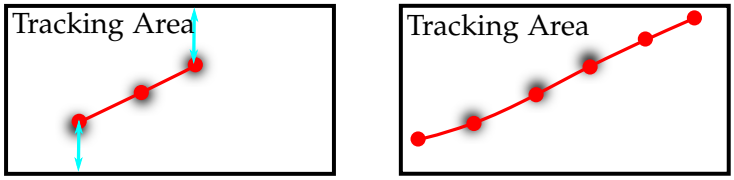


Figure 5.6: To suppress abrupt termination of trajectories in case of missed detections, the persistence term raises the energy value in cases where the start, respectively end point of a trajectories lies far from the border of the tracking area.

better performance, because solutions with many short tracks are less likely. These two terms are combined to form

$$E_{\text{reg}}(\mathbf{X}) = N + \sum_{i=1}^{N} \frac{1}{F(i)}. \tag{5.8}$$

Note that the second term can be weighted individually to adjust the importance of the lengths of the trajectories. Although empirically this leads to slightly better performance on some test sequences, we prefer to set it uniformly to 1 in all our experiments. Having fewer parameters facilitates the search for a good parameter set and avoids over-fitting.

5.2.2 *Global occlusion reasoning*

Occlusion reasoning plays an important role in many areas of computer vision, including pose estimation (Sigal and Black, 2006; Eichner and Ferrari, 2010), and object detection (Wu and Nevatia, 2005; Enzweiler et al., 2010; Wojek et al., 2011). The reason why occlusion modeling improves results is consistent in all cases: the knowledge that the observed object is only partially (or not at all) visible predicts that less evidence will be found in the image, and the appraisal of the evidence can be adapted accordingly.

Having introduced the basic tracking framework, we now turn to our explicit occlusion reasoning scheme. In typical real-world scenarios three different types of occlusion take place: *(1)* in crowded scenes, targets frequently occlude each other causing *inter-object occlusion*; *(2)* a target may move behind static objects like trees, pillars, or road signs, which are all examples of common *scene occluders*; *(3)* depending on the object type, extensive articulations, deformations, or orientation changes may cause *self-occlusion*. All three types of occlusion reduce – or completely suppress – the image evidence for a target's presence, and consequently incur penalties in the observation model. Specifically, in our tracking-by-detection setting they cause the object detector to fail and thereby increase E_{det}. However, simply treating occlusion as missing data, *i.e. ignoring the fact that the observed*

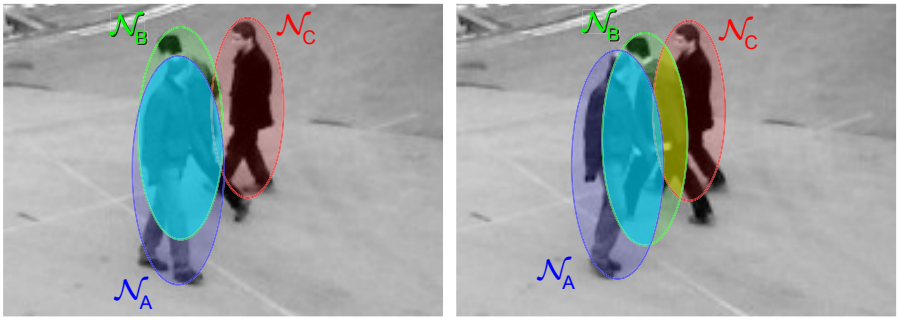


Figure 5.7: A typical example of inter-object occlusion. In the proposed occlusion model targets are represented as anisotropic Gaussians in image space (red, green, blue), whereas pairwise occlusions between all targets (cyan, yellow) are approximated by products of Gaussians.

occluder actually predicts the lack of evidence, can seriously impair tracking performance.

Consequently, explicit occlusion handling is important for successful multi-target tracking. Unfortunately, principled modeling of occlusion dependencies is rather tricky as the following example illustrates (see Fig. 5.7):

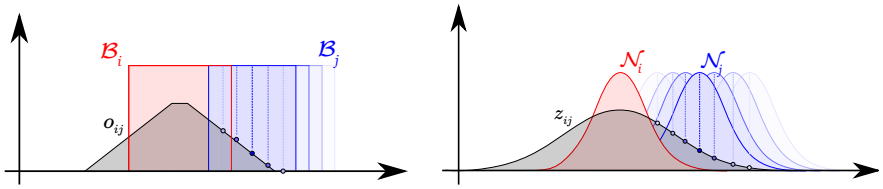
> *If target* A *is at location* $\mathbf{x}_A$*, then target* B *at* $\mathbf{x}_B$ *is occluded; but if* A *is a bit further to the left and* B *slightly further to the right, then* B *is partially visible; however then it would partially occlude target* C*;* etc.

In order to deal with situations where dynamic targets occlude each other, the main task is to overcome the difficulties that arise from the complex dependence between a target's visibility and the trajectories of several other targets, which could potentially block the line of sight. An explicit occlusion model thus leads to complicated objective functions, which tend to be difficult and inefficient to optimize. Therefore, most previous approaches either ignore the issue altogether, or resort to some form of greedy heuristic, usually separating target localization from occlusion reasoning.

We present a method that tightly couples both trajectory estimation and explicit inter-object occlusion reasoning. Note that it can be trivially extended to handle scene occluders. Not surprisingly, taking into account occlusions directly during trajectory estimation significantly reduces the number of missed targets and lost tracks – especially in highly crowded environments.

5.2.2.1 *Analytical global occlusion model*

Our approach handles mutual occlusion between all targets with a closed-form, continuously differentiable formulation. Since this procedure is identical for each frame, the superscript t is omitted for better readability.



(a) (b)

Figure 5.8: Schematic illustration (in 1D) of targets' overlap as a function of the occluder's position. In case of bounding boxes *(a)*, the overlap o_{ij} is non-differentiable on the borders. In contrast, our occlusion term z_{ij} is a Gaussian.

RELATIVE OVERLAP. Let us for now assume that each target i is associated with a binary indicator function

$$\mathbf{o}_i(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \mathcal{B}(\mathbf{X}_i) \\ 0, & \text{otherwise,} \end{cases} \tag{5.9}$$

which is 1 on the bounding box $\mathcal{B}(\mathbf{X}_i)$ of target i. The total image area of target i is thus given as $\int \mathbf{o}_i(\mathbf{x})\ d\mathbf{x}$. To compute the relative area of target i that is occluded by target j, we simply have to calculate the (normalized) integral of the product of both indicator functions:

$$\frac{1}{\int \mathbf{o}_i(\mathbf{x})\ d\mathbf{x}} \int \mathbf{o}_i(\mathbf{x})\mathbf{o}_j(\mathbf{x})\ d\mathbf{x} \tag{5.10}$$

Note that we assume here that target j is in front of target i; we will address the more general case below. If we define the target visibility using the relative area as given in Eq. (5.10), then the visibility is not differentiable w.r.t. the object positions of $\mathbf{X}_i$ or $\mathbf{X}_j$, which precludes gradient-based optimization methods (*cf*. Figure 5.8(a)).

To address this issue we here propose to use a Gaussian "indicator" function $\mathcal{N}_i(\mathbf{x}) := \mathcal{N}(\mathbf{x}; \mathbf{c}_i, \mathbf{C}_i)$. Besides achieving differentiability, this is further motivated by the fact that a Gaussian blob is a crude, but reasonable approximation for the shapes of many objects (see Fig. 5.7 for an illustration). In our case, each person in image space is represented by an anisotropic Gaussian with mean $\mathbf{c}_i = \mathbf{x}_i$ and covariance

$$\mathbf{C}_i = \begin{pmatrix} \frac{1}{2}\left(\frac{s_i}{2}\right)^2 & 0 \\ 0 & \left(\frac{s_i}{2}\right)^2 \end{pmatrix}$$

with s_i being the target's height on the image plane. As before, we compute the area of overlap by integrating the product of the two "indicator" functions, here Gaussians:

$$z_{ij} = \int \mathcal{N}_i(\mathbf{x}) \cdot \mathcal{N}_j(\mathbf{x})\ d\mathbf{x}. \tag{5.11}$$

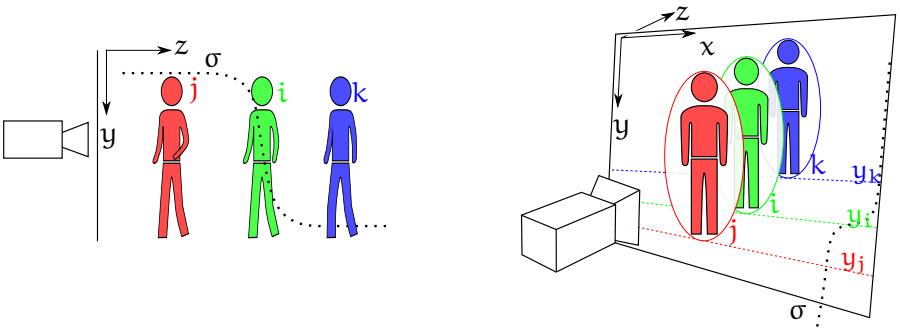


Figure 5.9: Target i has a non-zero overlap with j and with k. However, it is only occluded by j. Hence, the overlap is weighted with a sigmoid σ (dotted line) centered on y_i.

Besides differentiability, the choice of Gaussians allows this integral to be computed in closed form. Conveniently, the integral is another Gaussian (Brookes, 2005): $z_{ij} = \mathcal{N}(\mathbf{c}_i; \mathbf{c}_j, \mathbf{C}_{ij})$ with $\mathbf{C}_{ij} = \mathbf{C}_i + \mathbf{C}_j$ (see Fig. 5.8(b) for a schematic illustration). Since we are interested in the *relative* overlap that corresponds to the fraction of occlusion between two targets, we compute it using the unnormalized Gaussian

$$V_{ij} = \exp\left(-\tfrac{1}{2}[\mathbf{c}_i - \mathbf{c}_j]^\top \mathbf{C}_{ij}^{-1}[\mathbf{c}_i - \mathbf{c}_j]\right), \tag{5.12}$$

which is differentiable w.r.t. $\mathbf{c}_i$ and $\mathbf{c}_j$ and has the desired property that $V_{ij} = 1$ when $\mathbf{c}_i = \mathbf{c}_j$. Moreover, due to the symmetry of Gaussians we have $V_{ij} = V_{ji}$.

DEPTH ORDERING. To also take into account the depth ordering of potentially overlapping targets, we could make use of a binary indicator variable, which once again has the issue of making the energy function non-differentiable. We once more replace it with a continuous, differentiable version and use a sigmoid along the vertical image dimension centered on y_i (*cf*. Figure 5.9):

$$\sigma_{ij} = \frac{1}{1 + \exp(y_i - y_j)}. \tag{5.13}$$

Note that this definition assumes a common ground plane as well as a camera at a rather low viewpoint and in standard landscape or portrait orientation, such that the depth order corresponds to the order of the targets' y-coordinates (it is however straightforward to extend the idea to more general setups). Also note that if we assume small variation in target size, then the occluder will always appear larger than the occluded object on the image plane and hence will entirely cover the farther target if their center points coincide.

VISIBILITY. To define the overall visibility of each target, we first define an occlusion matrix $\mathcal{O} = (\mathcal{O}_{ij})_{i,j}$ with $\mathcal{O}_{ij} = \sigma_{ij} \cdot V_{ij}, i \neq j$ and $\mathcal{O}_{ii} = 0$. The entry in row i and column j of $\mathcal{O}$ thus corresponds to (a

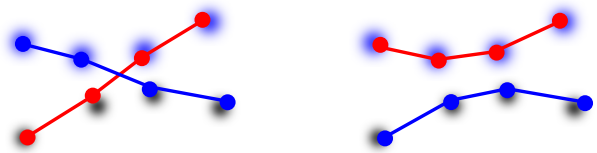


Figure 5.10: The appearance model takes color information into account and prefers similar regions to belong to the same trajectory.

differentiable approximation of) the fraction of i that is occluded by j. Disregarding cases where multiple occluders cover the same limited fraction of a target, we can now approximate the total occlusion of i as $\sum_j o_{ij}$. A straightforward definition of the visible fraction of i would thus be

$$\max\left(0, 1 - \sum_j o_{ij}\right). \tag{5.14}$$

However, to avoid the non-differentiable max function, we prefer an exponential function and define the visibility for target i as

$$v_i(\mathbf{X}) = \exp\left(-\sum_j o_{ij}\right). \tag{5.15}$$

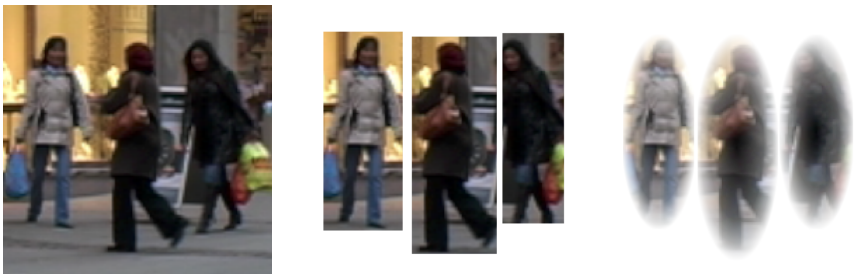
This definition allows us to efficiently approximate the visible area by taking into account mutual occlusion for each pair of targets. Furthermore, by consistently using appropriate differentiable functions the entire energy has a closed form and remains continuously differentiable.

LIMITATIONS. The main limitation of this approach is that targets are represented with a simple oval shape. However, our experiments show that the actual fraction of occlusion can be estimated quite reliably even for pedestrians, despite their non-rigid, articulated motion.

5.2.3 *Appearance model*

The appearance of an object may provide important cues for disambiguating it from the background and from other objects. This aspect has previously either been ignored (Berclaz et al., 2009; Andriyenko and Schindler, 2011; Andriyenko et al., 2011), or addressed only in the discrete setting (Zhang et al., 2008; Kuo et al., 2010; Ben Shitrit et al., 2011). Here, we present a novel appearance term that is continuously differentiable in closed form, thus admitting gradient-based optimization.

Assuming that an object's color remains constant over time and that lighting changes slowly, our appearance model imposes a higher penalty in cases of abrupt changes. To maintain the benefits of the



(a) (b) (c)

Figure 5.11: Instead of extracting and comparing full bounding boxes *(b)*, we propose to weigh the area using anisotropic Gaussians *(c)*. The energy remains differentiable, and the influence of undesired background pixels is reduced.

continuous formulation, it is desirable to describe the appearance of an object analytically. To ensure that the energy remains smooth without costly interpolation, we propose to use Gaussian weighted regions (*cf*. Figure 5.11). This not only ensures differentiability, but a closed-form gradient. This is also motivated by the fact that the object of interest typically occupies the central area inside the bounding box. The background pixels along the borders and in the corners are therefore naturally downweighted, while the pixels closer to the center receive higher weights.

Formally, the Gaussian weighted histogram count of the image region occupied by target i in frame t is defined as

$$h_n(\bar{\mathbf{x}}_i^t) = \sum_{\mathbf{x}} \left[\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_i^t, \Sigma_i^t) \cdot H_n(\mathbf{x}) \right], \tag{5.16}$$

where H_n is a binning function

$$H_n(\mathbf{x}) = \begin{cases} 1, & \text{if } I(\mathbf{x}) \text{ falls into bin } n \\ 0, & \text{otherwise,} \end{cases} \tag{5.17}$$

and $\bar{\mathbf{x}}$ is the center of the target's bounding box. We employ the widely used Bhattacharyya coefficient

$$BC(\mathbf{X}_i^t) := \sum_{n}^{\#\,\text{bins}} \sqrt{h_n(\bar{\mathbf{x}}_i^t) * h_n(\bar{\mathbf{x}}_i^{t+1})}. \tag{5.18}$$

for histogram comparison. In our experiments a standard RGB color histogram with 16 bins per channel yields the best results. Obviously, the appearance of a target will change if it becomes occluded and thus should not be taken into account. We therefore reduce the influence of the appearance term in such cases by weighting the histogram

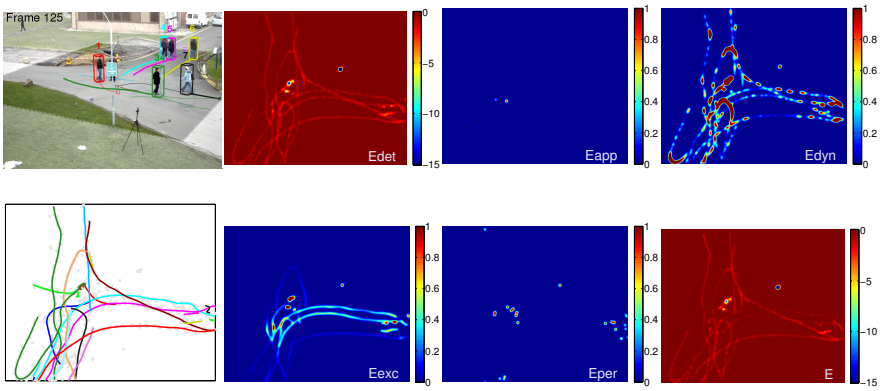


Figure 5.12: Per target contributions to the individual energy components rendered as heat maps, accumulated across 150 consecutive frames. The value range for each component and combined entire energy is indicated by a color bar. Note the large penalty of the exclusion term E_{exc} for the two targets moving alongside each other, and the higher energy values induced by E_{dyn} in sections with high curvature.

deviation with the geometric mean of the visibilities (*cf*. Section 5.2.2) of the two bounding boxes:

$$AC(\mathbf{X}_i^t) = \bar{v}_i^t(\mathbf{X}) \cdot (1 - BC(\mathbf{X}_i^t)) \tag{5.19}$$

with

$$\bar{v}_i^t(\mathbf{X}) := \sqrt{\left(v_i^t(\mathbf{X}) \cdot v_i^{t+1}(\mathbf{X})\right)}. \tag{5.20}$$

Instead of simply adding this penalty to the energy, we found it to be beneficial in practice to use a soft threshold to better discriminate between true matches with a high similarity, *i.e.* low energy value, and identity switches. To that end, the final appearance term uses a sigmoid:

$$E_{app}(\mathbf{X}) = \sum_{i=1}^{N} \sum_{t=s_i}^{e_i-1} \frac{1}{1 + \exp(a_1 - a_2 * AC(\mathbf{X}_i^t))} \tag{5.21}$$

The parameters a_1, a_2 are determined by a least squares fit to a small subsample of the available data.

Our appearance model is designed to fit gradient based optimization methods. As we show in Section 5.5, including appearance significantly reduces the number of identity switches and track fragmentations, though not increasing the average accuracy on the chosen datasets. Moreover, it forces the tracker to follow the targets more closely thereby increasing the tracking precision. We believe that appearance will be even more helpful in high resolution videos – where targets usually provide more color information – or in situations with stronger appearance variation than in existing benchmarks.

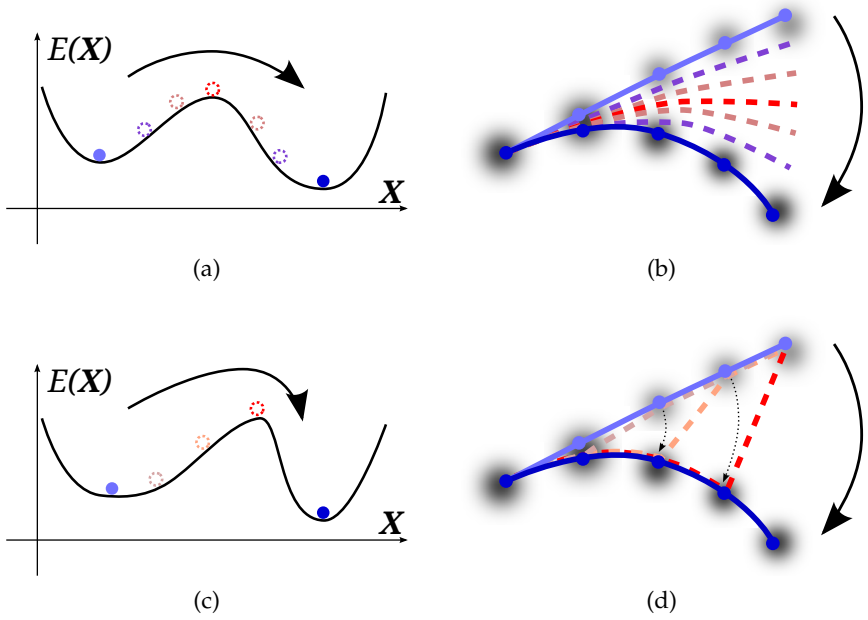


Figure 5.13: Illustration of the non-convexity of the continuous tracking formulation. To get from the light blue solution (weaker optimum) to the dark blue one (stronger optimum) in a continuous state space one has to overcome a ridge of high energy. *(a,b)* Keeping E_{dyn} low incurs high penalties in E_{obs} as one moves away from the observations. *(c,d)* Keeping E_{obs} low incurs high penalties in E_{dyn} as the paths gets distorted to fit the observations. With a peaked likelihood intermediate cases are even worse.

5.3 OPTIMIZATION

The energy in Eq. (5.1) described in Section 5.2.1 is clearly not convex. In fact, it is not unlikely that a realistic model of multi-target tracking cannot be convex: It is easy to construct examples that have two virtually equal minima separated by a ridge of high energy (*cf*. Figure 5.13). The main reason for this behavior is the high-order dependence between variables caused by physical constraints.

To minimize the energy function in Eq. (5.1) locally, we use the standard conjugate gradient method. Upon convergence, a jump move is executed (unless it would increase the energy), which may change the dimensionality of the model (see Figure 5.1 for a schematic illustration). The jumps give the optimization a high degree of flexibility – the initial solution need not even have the correct number of targets. To speed up the optimization process, all trajectories are given the chance to execute a certain jump at the same time. Based on our experience, the order in which the jumps are executed is not crucial, because the optimization may choose to perform an inverse move to find the way towards a lower energy. Please refer to Section 5.5.2 for an empirical study on various optimization strategies.

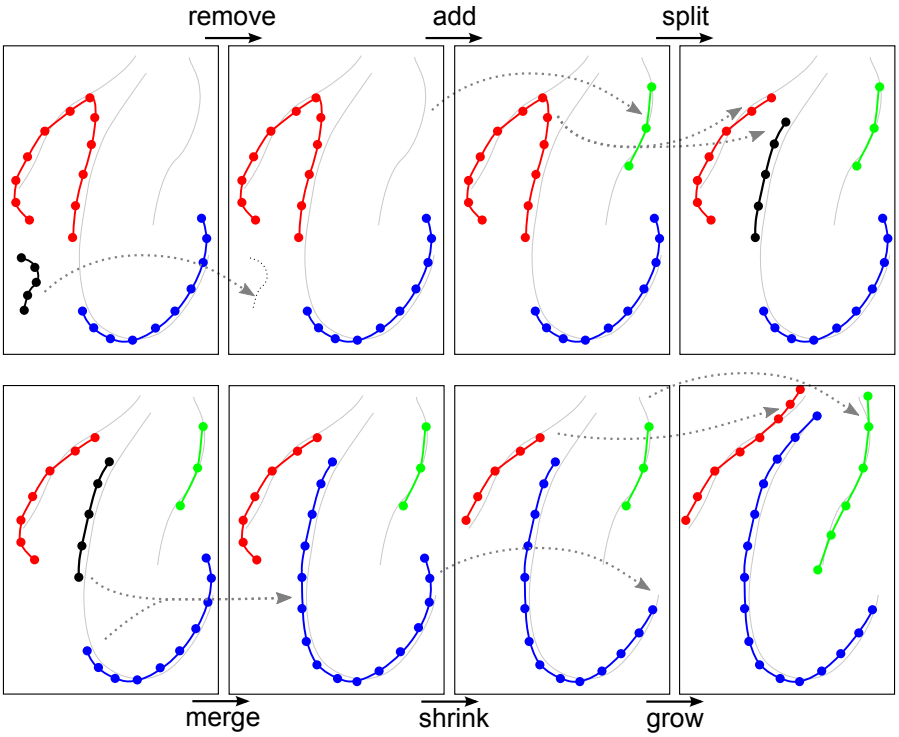


Figure 5.14: The proposed jump moves make continuous optimization more flexible, allowing a variable number of targets. Even a poor initial configuration can be used to recover the true trajectories. The ground truth is rendered in gray.

Our data-driven strategy for changing the dimension of the state vector is reminiscent of reversible jump Markov chain Monte Carlo methods (Green, 1995), which have been applied to multi-target tracking in various ways (Ge and Collins, 2008; Benfold and Reid, 2011; Wu et al., 2011). A crucial difference to traditional Monte Carlo sampling is that our method is deterministic: It exploits the advantages of gradient descent over sampling within one mode, and performs jumps according to a prescribed schedule, only if they decrease the energy. The energy minimization algorithm is summarized in Algorithm 1.

5.3.1 *Transdimensional jumps*

To escape weak local minima we introduce six types of jump moves, which change the configuration of the current solution, thereby altering the dimension of the current state $\mathbf{X}_{\text{curr}}$. By jumping to different regions of the search space while always lowering the energy, the optimization is able to find much stronger local minima. An example of an optimization run with jumps is shown in Figure 5.14. Here, a weak trajectory (black) is removed entirely while a new one (green) is initialized. Note that each jump leads to a configuration with a lower energy.

GROWING AND SHRINKING. The time span during which a target is visible in the target area can be changed by growing or shrinking its trajectory. To extend the trajectory's length, it is simply linearly extrapolated in space-time (both forward and backward).

Let $\mathbf{X}_i = \mathbf{X}_i^{s_i:e_i}$ denote the current state of the i^{th} trajectory defined between frames s_i and e_i. To evaluate the energy $E_{new} = E(\mathbf{X}_{new})$, the trajectory is extrapolated backwards for t frames resulting in

$$\tilde{\mathbf{X}}_i = (\mathbf{X}_i^{s_i-t:s_i-1}, \mathbf{X}_i) \tag{5.22}$$

leading to the new state

$$\mathbf{X}_{new} = \Big(\bigcup_{\substack{j=1\ldots N \\ j\neq i}} \mathbf{X}_j \Big) \cup \tilde{\mathbf{X}}_i. \tag{5.23}$$

The procedure for forward extrapolation is analogous with

$$\tilde{\mathbf{X}}_i = (\mathbf{X}_i, \mathbf{X}_i^{e_i+1:e_i+t}). \tag{5.24}$$

Shortening is achieved by simply discarding t past or future positions of a target: $\tilde{\mathbf{X}}_i = \mathbf{X}_i^{s_i+t:e_i}$, respectively $\tilde{\mathbf{X}}_i = \mathbf{X}_i^{s_i:e_i-t}$. Such growing and shrinking steps help to pick up lost tracks and weed out spurious trajectories.

MERGING AND SPLITTING. Allowing merging and splitting of trajectories can effectively improve data association, *i.e.* eliminate identity switches and track fragmentations. Splitting at time t is implemented by breaking a path $\mathbf{X}_k$ in two:

$$\tilde{\mathbf{X}}_i = \mathbf{X}_k^{s_k:t}, \quad \tilde{\mathbf{X}}_j = \mathbf{X}_k^{t+1:e_k} \tag{5.25}$$

if the split yields lower energy. Merging is executed if two paths can be smoothly connected into one with lower energy, preserving physically plausible motion:

$$\tilde{\mathbf{X}}_k = (\mathbf{X}_i, \mathbf{X}_{con}^{e_i+1:s_j-1}, \mathbf{X}_j). \tag{5.26}$$

Especially merging is a powerful tool to overcome temporary tracker failure due to weak evidence or occlusion.

ADDING AND REMOVING. New trajectories can be generated at locations with strong detections, which are not yet assigned to any trajectory. To that end, the detection with the maximum confidence ω_g^t that does not have a trajectory nearby is found

$$(t, g) = \arg\max_{r,j} \left\{ \omega_j^r \,\middle|\, \|\mathbf{D}_j^r - \mathbf{X}_i^r\| \geqslant 2s \quad \forall i \right\} \tag{5.27}$$

and a new track is started conservatively with three consecutive frames at the same location:

$$\tilde{\mathbf{X}}_i^{t-1:t+1} = (\mathbf{D}_g^t, \mathbf{D}_g^t, \mathbf{D}_g^t). \tag{5.28}$$

Algorithmus 1 : Continuous energy minimization

```
input  : S initial solutions, detections D
output : Best of ⩽ S solutions

for s ← 1 to S do
    while ¬ converged do
        for m ∈ {grow, shrink, add, remove, merge, split} do
            for i ← 1 to N do
                try jump move m on trajectory i
                (greedy parameter selection)
                if E_new < E_old then
                    perform jump move m
                end
            end
            perform conjugate gradient descent
        end
    end
end
return arg min_{X_s} E(X_s)
```

Note that such short 3-frame tracklets can grow or merge with existing ones at a later stage.

An entire trajectory is removed if its total contribution to the energy is above a certain threshold, meaning that it reduces the overall likelihood of the current state, rather than increasing it. Adding helps to find missing trajectories not picked by the original tracking solution, whereas removal discards trajectories which have been pushed to a state with little evidence, unreasonable dynamics, and/or overlap with other trajectories.

We repeatedly iterate through the six different move types in a fixed, prescribed order (see Alg. 1). For each move type, the move parameters – *e.g.* the number of frames a trajectory is grown or the time step at which a trajectory is split – are optimized independently for each trajectory in a greedy fashion. It is important to note, however, that the optimization is not entirely greedy, since the move type order is fixed; thus it is not guaranteed that every step leads to the largest energy decrease. Please see Section 5.5.2 for a study on various optimization techniques.

5.3.2 *Initialization*

Like any non-convex optimization, the result depends on the initial value from which the iteration is started. However, the described exploration strategy greatly weakens this dependency compared to a pure gradient method. By allowing jumps to low-energy regions of the search space, even if they are far away from the current state, the

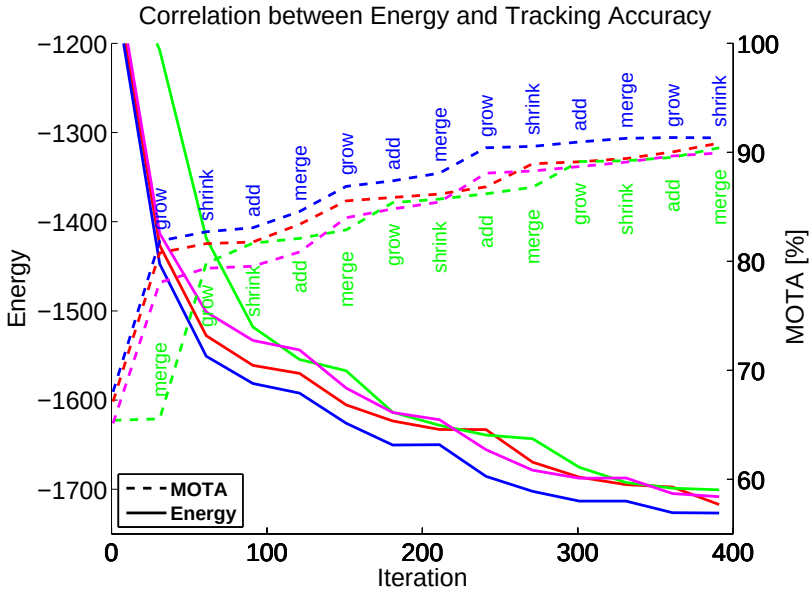


Figure 5.15: Four optimization runs started from four different initializations on the sequence *S2L1*. The proposed energy (solid) correlates well with tracking performance w.r.t. ground truth (dashed). Energy values have been scaled uniformly for ease of visualization.

attraction to local minima is reduced: the weaker a minimum is, the more likely it gets to find a jump out of its basin of attraction that lowers the energy.

Empirically, even a trivial initialization with no targets works reasonably well, however takes many iterations to converge. Instead, we propose to rather use the output of an arbitrary, simpler tracker as a more qualified initial value. In our experiments we used per-target extended Kalman filters (EKFs), where the data association is performed in a greedy manner using a maximum overlap criterion. Note that the EKF trackers are not intended to achieve the best possible performance but rather to quickly generate a variety of starting value. This is accomplished by running the trackers several times with different parameters. Naturally, if time does not play a crucial role, other, possibly more sophisticated solutions, such as the ILP-tracker from Chapter 4, can be used as initialization.

Usually, different starting values converge to similar, albeit not identical solutions (see Fig. 5.15).

5.3.3 *Goodness of local minima*

In the beginning of Section 5.3 we outlined why an accurate continuous energy for multi-target tracking can hardly be convex. In our case it is not only impossible to guarantee global optimality, but it is

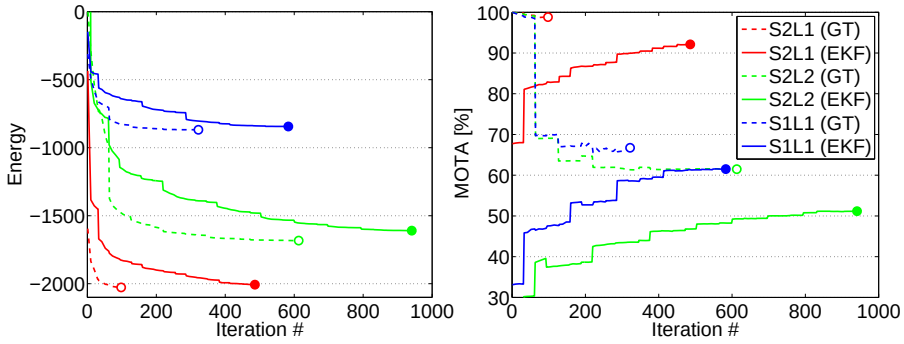


Figure 5.16: Initialization with ground truth (dashed) and with EKF (solid).

also intrinsically difficult to assess how close or how far the obtained solution is from the global minimum. Still, it is useful to gain at least some insights on the goodness of the local minima found.

Previous work on stereo revealed some interesting insights concerning the same issue. One of them was that the energy value of the ground truth annotation is always larger than solutions found by optimization techniques like belief propagation (BP) or graph cuts (Tappen and Freeman, 2003; Kolmogorov and Rother, 2006). Intuitively, we can expect similar behavior for the problem of multi-target tracking for the following reasons.

First, we have to deal with missing data. Trajectories of undetected targets will increase the energy, which will be explained below in more detail. Second, annotations are created manually and thus will never be perfectly aligned with the detector output. Consequently, ground truth tracks will in general have a higher data energy than fitted trajectories. Finally, ground truth annotations are not always smoothed temporally and therefore induce extremely high energy values for the dynamic model.

One way to determine the global minimum is to densely sample the continuous state space on a minimalistic toy example. This is, however, impractical for realistic scenarios, even for the relatively simple ones. Instead, we compare the results of our standard approach with those obtained when the optimization is initialized with the ground truth 'solution'. The plots on three exemplar sequences are illustrated in Figure 5.16. Obviously, it is not guaranteed that a ground-truth-initialized optimization converges to the global optimum of the energy. But at least one can expect to find good local minima in this way, which can be employed to quantitatively estimate the goodness of the found solution.

Meltzer et al. (2005) pursued a similar idea and determined the global optimum of a discrete energy on simple stereo examples.

We make several notable observations. As hypothesized above, the energy value of the ground truth (dashed line at iteration 0) is far away from any energy minimum. Moreover, the relative gap between the solutions from the two initializations roughly corresponds to the

Seq. (init)	E init	E end	MOTA init	MOTA end	FP end	FN end	ID end
S2L1 (GT)	-1594.9	-2027.1	100.0	98.8	23	24	0
(EKF)	-435.1	-2006.9	67.7	92.1	33	272	7
S2L2 (GT)	20.8	-1682.8	100.0	61.5	87	3123	8
(EKF)	8906.6	-1609.5	28.7	51.2	118	3896	65
S1L1 (GT)	-311.4	-868.1	100.0	66.7	10	848	2
(EKF)	-148.3	-843.9	33.1	61.5	16	972	7

Table 5.1: Per sequence results with different initial values.

difference in performance as measured by MOTA. This shows that the energy appears to give a fairly accurate representation of the problem. Finally, starting from the 'correct' solution, the optimization requires many fewer iterations to converge and leads to a better final performance in all three cases. However, starting from a ground-truth initialization the optimization stays within the same basin of attraction only on the relatively easy sequence *(S2L1)*. The large amount of missing data in the other two drive the optimization towards a less accurate solution. This may seem somewhat surprising and contradictory to the claim that the proposed energy presumably describes the tracking problem more accurately.

Let us take a closer look at the reason for this behavior. The proposed energy combines a-priory knowledge with input data to compute the goodness of a particular solution. Ignoring one of these two components would result in strong overfitting or in biased solutions, respectively. It is reasonable to assume that in general, enough data is available to support the hypotheses and overrule the priors. This is not the case for the crowded sequences *S2L2* and *S1L2*, where the majority of pedestrians remain undetected due to occlusion. Here, exclusion, dynamics, persistence and regularization, all impose a penalty, while no data is present to explain the trajectory hypotheses. Therefore, most of the existing targets are removed from the solution, which better meets the a-priory assumptions.

Overall, the presented experiment confirms that better performance can be achieved by both providing better starting values and designing more powerful optimization techniques. The detailed results for each sequence are listed in Table 5.1.

5.4 IMPLEMENTATION

Before presenting the experiments we would like to point out some implementation details.

CONTINUOUS OPTIMIZATION. We use Carl Rasmussen's implementation[1] with its default parameters to perform conjugate gradient descent on the energy. In this setting, the Polack-Ribiere formula is employed for determining the search directions, while the Wolfe-Powell conditions and the slope ratio method are used for estimating the step size. For each search direction, a line search with a maximum of 20 function evaluations is performed using second- and third-order polynomial approximations of the objective function. We limit the gradient descent to a maximum of 30 iterations because it is usually sufficient to get close enough to a local optimum such that meaningful discrete jumps can be executed.

TRACKING AREA. In order to compute the distance to valid entry and exit points to enforce persistent trajectories (*cf*. Eq. 5.7), the boundary of the tracking area needs to be known. For our purposes we define a rectangular area on the ground so as to facilitate the computation of the distances. Targets outside its limits are excluded from the solution. This is, however, not a major limitation because the quadrilateral formed by the forward-projected image borders can easily be used instead as tracking area.

RUN TIME. Given the detections, our current MATLAB/MEX implementation takes approximately 1s/frame to obtain one solution using explicit occlusion reasoning. Without the expensive occlusion computation, the optimization runs an order of magnitude faster, achieving near real-time performance. Unfortunately, computing color information and its derivatives for all pixels significantly slows down the optimization. While this can still be improved, the use of the appearance term is thus only recommended if computation time does not play a crucial role.

CONVERGENCE. As stated in Alg. 1, the energy is minimized until there is no jump that leads to a lower energy. Convergence is usually reached quickly (after 5 to 10 iterations). We set a maximum of 15 iterations because of timing constraints. Note that in some cases the results may still improve with more computational resources.

PARAMETERS. Although the precise parameter values are highly dependent on the implementation at hand, we state them here for completeness. The weights α through ϵ are set to $\{.1, .02, .5, .7, .7\}$ and λ=.1 in all our experiments including the appearance term. Turning it off (*i.e*. setting $\alpha = 0$) also requires both δ and ϵ to be decreased slightly to a value of .6 to achieve best results. Finally, the setting for the basic energy without occlusion handling (no OM) benefits from the values $\beta = .03, \gamma = \delta = \epsilon = .6, \lambda = .075$. Note that these parame-

1 http://www.gatsby.ucl.ac.uk/~edward/code/minimize/minimize.m

Parameter	Description	no OM	OM	FULL	2D
α	appearance	0	0	0.1	0
β	dynamics	0.03	0.02	0.02	2
γ	exclusion	0.6	0.5	0.5	1
δ	persistence	0.6	0.6	0.7	1
ϵ	regularizer	0.6	0.6	0.7	0.5
λ	offset	0.075	0.1	0.1	0.1
s	target size		35 [cm]		20 [px]

Table 5.2: Typical parameter settings for running the continuous energy-based multi-target tracking.

ter settings have been chosen conservatively and are not necessarily optimal for any particular dataset (*cf*. Figure 5.17). An overview of typical settings for all three variants is provided in Table 5.2. The rightmost column also states the default parameter set for tracking on the image plane (2D).

Our complete implementation together with all the necessary additional data, including detector output and ground truth, can be freely obtained online.[2]

5.5 EXPERIMENTS

In Section 5.2 we introduced an energy function that has been conceived with the primary goal of accurately reflecting the actual behavior of multiple interacting targets (*cf*. Figure 5.15). As a consequence, the energy minimization can only be solved to local optimality, and there are no theoretical guarantees about the goodness of the solution. Our claim is that minimizing this function will nevertheless on average yield higher tracking accuracy. To empirically support this claim we performed an extensive experimental evaluation on various datasets.

Before presenting detailed quantitative results, we first analyze our approach in two regards: First, we examine the influence of the individual energy terms on the tracking performance and the robustness of the chosen parameters to variations of their respective values. (*cf*. Section 5.5.1 and Figure 5.17). Next, we compare different optimization strategies and their influence on the convergence rate and the final result. (*cf*. Section 5.5.2).

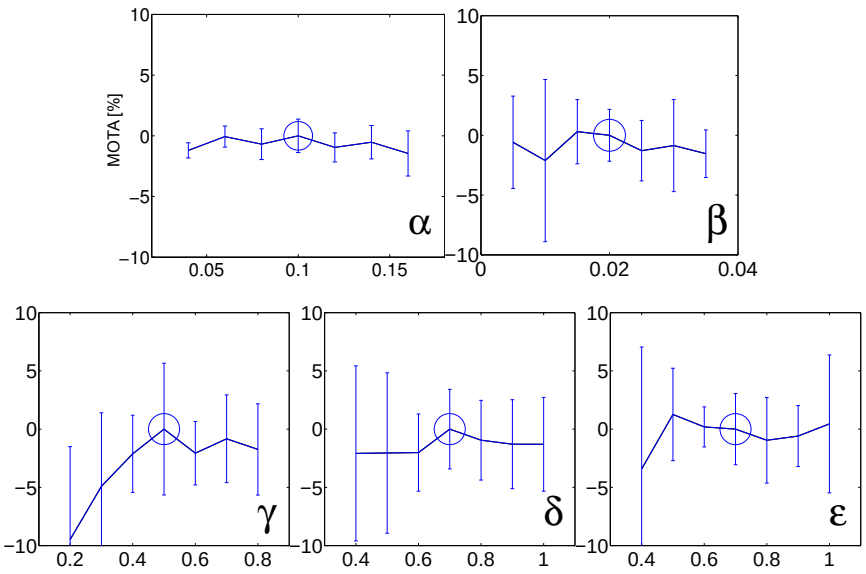


Figure 5.17: Influence of individual parameters on tracking performance. Each plot shows the relative change in performance (measured by MOTA) by changing the weight of a single energy component while keeping the other ones fixed. The results shown here are averaged over all datasets and normalized for better readability. The error bars indicate the standard deviation around the mean. The parameter value used in our experiments is marked with a circle. As can be seen our choice of parameters is rather conservative and does not correspond to the best set. This is an indication that the model has not been over-tuned on the given test data.

5.5.1 *Parameter study*

Manually tweaking several parameters is both tiresome and time-consuming. Ideally, model parameters should be learned automatically from example data, however that would require a large amount of annotated ground truth. We thus had to resort to determining the model parameters manually, which is not only tedious, but carries the danger of over-fitting. To mitigate this, we use only one parameter set for all test sequences, even though they exhibit strong variations both visually and in terms of target behavior.

To examine the influence of each individual weight of the energy in Eq. (5.1), we run our tracking algorithm and modify the corresponding parameter while keeping all the other ones fixed. In Figure 5.17, for each term, the relative change in performance, as measured by MOTA, is plotted against the parameter value. For illustration, the average mean-normalized value is shown along with error bars, depicting the variation between various sequences. Note that even a relatively drastic scaling of the weights (*e.g.*, by a factor of 1/2 or 2)

2 http://www.gris.tu-darmstadt.de/~aandriye/contracking

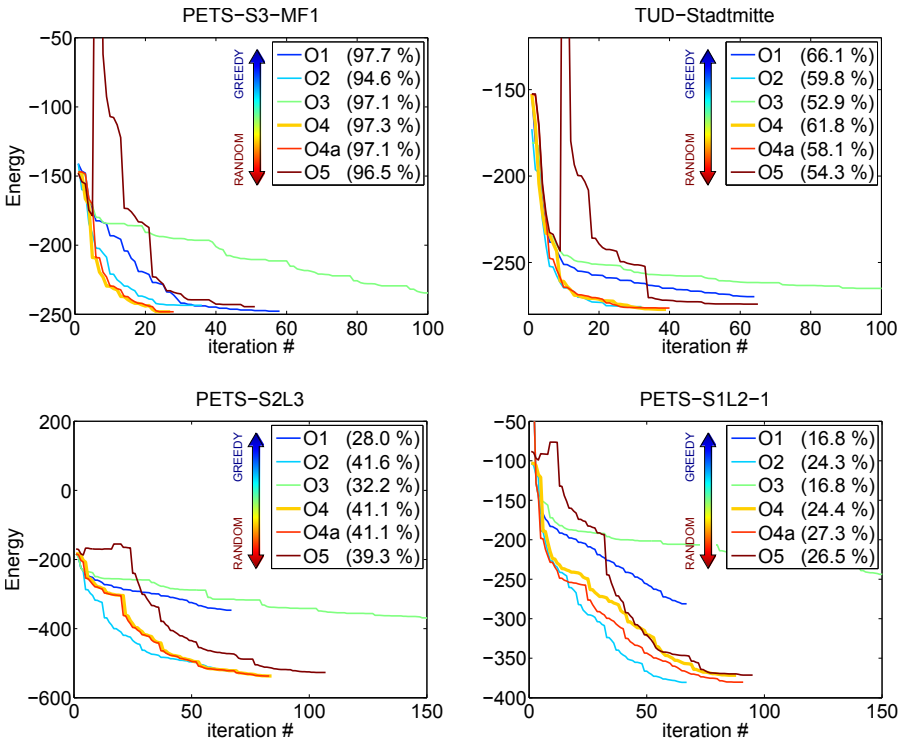


Figure 5.18: Energy minimization with different optimization techniques on four exemplar sequences (see text for a detailed explanation of the individual strategies). Our proposed optimization scheme described in Section 5.3 corresponds to *O4*. The final tracking accuracy w.r.t. ground truth is reported in parentheses for each case.

hardly affects the overall performance. The strongest decline can be observed when γ – the weight for target exclusion – is set too low. This once again demonstrates the importance of explicitly modeling the spatial dependencies to avoid situations with overlapping targets. Moreover, we can conclude that the results are stable over a range of settings and tracking performance is only slightly affected by parameter changes within a reasonable range.

5.5.2 *Optimization strategies*

There are many possible ways of integrating discontinuous jump moves into an optimization scheme. To understand this choice, we conduct a set of experiments that vary in the way the jumps are selected and applied. They show that the exact choice is not critical, and that the optimization scheme described in Section 5.3 is a reasonable compromise between fast convergence and low energy. To this end, we compare our results to those obtained with five modified energy minimization algorithms ranging from greedy to random (*cf*. Figure 5.18).

To better understand their differences, let us first recall our originally proposed scheme (Section 5.3). We alternate between two dis-

tinct algorithms: (1) Purely continuous conjugate gradient descent, which runs until convergence or to a maximal number of iterations (here set to 30, which suffices to get close to a local minimum), and (2) discontinuous jump moves that are executed for all trajectories at once. Note that the move parameters (*e.g.* the number of frames to extend) are determined independently for each target $i \in \{1 \dots N\}$ such that the jump leads to the largest decrease of the energy value. We now examine the influence of five alternative jump move strategies; the gradient descent is left unchanged. Figure 5.18 shows the results.

1. Out of all possible move types and trajectories, the most greedy strategy *O1* always chooses the best possible modification of the current configuration, *i.e.* the one that yields the largest decrease of the energy value. Both the trajectory to be modified as well as the move type and its parameters are determined anew after each iteration. Note that only one trajectory is modified between two continuous optimization runs, which generally leads to slower convergence.

2. The less greedy *O2* chooses the move type that maximally reduces the energy as applied to all trajectories, rather than only one as for *O1*. This is similar to our proposed algorithm, however, here the move order is not predefined but chosen in a greedy manner after each iteration. This often leads to a fast energy drop within the first few iterations. However, the reached minimum is usually not as strong as the one found with a more random strategy, such as *O4*.

3. To evaluate the effect of greedily choosing trajectories, *O3* uses a predefined move order. The difference to our method (*O4*) is that instead of modifying all trajectories at once, the best one is picked greedily. This severely limits the possible state space changes. Consequently, the search largely stays within one region of the solution space and continuous minimization is not able to descend much further. As a result, this optimization leads to extremely slow convergence.

4. *O4a* also uses a prescribed move order, but modifies all trajectories at each iteration, which significantly speeds up the optimization process. The only difference between our proposed scheme (*O4*) and *O4a* is that a different prescribed order of the jump moves is used. As expected, these two strategies are very close in terms of convergence rate and the achieved results. This shows that the move order does not play a crucial role on average.

5. Finally, *O5* represents the most random strategy. First, the move type is picked randomly each time. Moreover, a 'bad' jump

Method	MOTA	MOTP	MT	ML	FM	ID	Rcl	Prc
FULL	61.4	67.8	11	11	17	24	65.7	95.1
discr.	52.2	62.4	9	11	34	42	62.2	89.3
cont.	41.4	68.2	3	17	14	15	44.7	94.5
EKF	39.8	66.3	3	18	16	13	43.1	94.5

Table 5.3: Average results of a purely discrete *(discr.)* vs. purely continuous *(cont.)* optimization.

that increases the energy is accepted with probability p, which is in turn decreased with time: $p = e^{-0.05 \cdot \text{iter}}$. This strategy is reminiscent of simulated annealing methods. We find that allowing jumps towards higher energy regions delays the search and does not lead to stronger minima. A more conservative strategy, such as O_4, finds its way towards regions of a lower energy more quickly and more reliably.

From our results (Figure 5.18) we can thus conclude that different optimization schedules lead to minima with a comparable energy. The crucial aspect is to include jump moves to escape weak local minima, since a purely continuous optimization is only able to search within a small local neighborhood of the state space in case of non-convex energies. However, the exact order, frequency, and selection of jumps is of minor importance.

Finally, Table 5.3 shows two further experiments where we either turn off the gradient descent based optimization and only perform the proposed discontinuous jumps *(discr.)* or vice versa *(cont.)*. As expected, a purely continuous optimization only slightly improves the accuracy over the EKF initialization by quickly terminating in the same local minimum. On the other hand, the purely discrete optimization does a good job by sampling varying configurations of the solution space but is at the same time rather constrained to the present shape of trajectories, since only non-moving targets can be created, while the linear extrapolation disallows reconstructing curved trajectories. Only by combining the two schemes *(full)* is it possible to reach good optima of the proposed energy.

5.5.3 *Number of targets*

Due to the discrete jump moves (see Section 5.3.1), the optimization is able to automatically infer the number of trajectories in a given sequence, independent from the initial solution. In this section, we investigate the question whether estimating the crowd density could help to guide the minimization procedure to a better result. A similar strategy was proposed by Rodriguez et al. (2011).

weight E_{cnt}	MOTA	MOTP	MT	ML	FM	ID	Rcl	Prc
$\eta = 0.0$	55.6	63.9	12	12	21	25	61.3	93.7
$\eta = 1.0$	46.0	62.6	12	10	35	43	63.2	78.3
$\eta = 0.1$	58.7	63.1	13	10	24	33	64.8	93.5

Table 5.4: Average results on four sequences. Our basic method (top row) compared to the extended energy with the constraint (5.29) on the number of targets.

To this end, an additional energy term

$$E_{cnt} = \eta \cdot \sum_{t=1}^{F} |N_{GT}(t) - N(t)| \tag{5.29}$$

is introduced, that penalizes per-frame absolute differences between the number of targets in the current state $N(t)$ and the actual number of targets present in the scene at each time $N_{GT}(t)$. Note that for this experiment, we rely on the true person count extracted from the ground truth. In a more realistic scenario, an object density estimation method can be used instead (Lempitsky and Zisserman, 2010).

Table 5.4 lists average results on four sequences (*TUD-Stadtmitte* and PETS *S2.L1*, *S2.L2*, *S1.L2-1*). Setting $\eta = 0$ amounts to the continuous energy from Eq. (5.1) with the default parameter set without occlusion modeling. Surprisingly, simply adding E_{cnt} significantly degrades the overall performance (≈ 10 percentage points). This can be explained by the fact that the optimization is forced to insert additional targets into crowded scenes, where the majority of targets is missed by the object detector due to severe occlusion. Since any information on target locations is entirely discarded in such cases, spurious trajectories are spawned across the entire area, which leads to extremely low precision.

Carefully tuning the weight of E_{cnt} slightly improves the average recall, mostly because the isolated detection responses – that are otherwise considered as false alarms – are explained by additional trajectory snippets. This is why the number of track fragmentations and identity switches increases.

In summary, this experiment shows that the proposed model does not benefit from the information about the number of targets present in the scene. Presumably, a formulation that relies on a dense likelihood (*cf*. Breitenstein et al., 2009), rather than on non-maxima suppressed NMS detections, could profit by this additional knowledge.

5.5.4 *Comparison to ILP*

Before presenting an exhaustive quantitative evaluation of our method, we first compare it to the discrete integer linear program (ILP) ap-

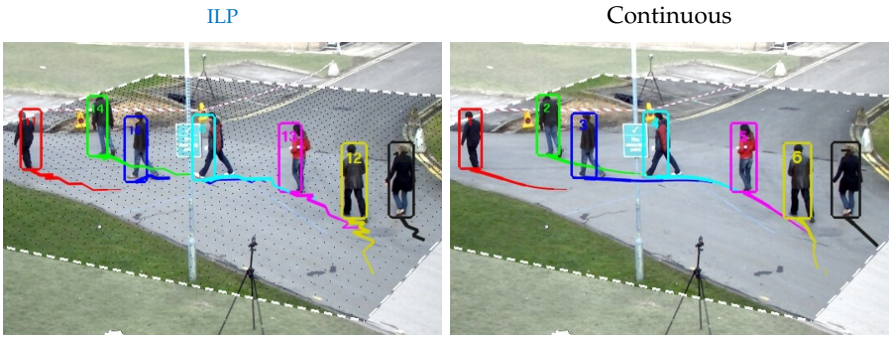


Figure 5.19: Tracking results of the integer linear program (ILP)-formulation *(left)* and continuous energy minimization *(right)*. Trajectories appear much smoother in the latter case because they are not constrained to discrete grid locations.

proach from Chapter 4. To that end, we run both methods using the same pedestrian detector (Walk et al., 2010a) on three sequences: *TUD-Stadtmitte*, PETS-*S2L1* and PETS-*S3-MF1*. In more crowded scenarios, the number of variables in the discrete formulation increases dramatically such that the resulting large-scale optimization cannot be solved efficiently by non-commercial ILP-solvers.

An example frame from both approaches is shown in Figure 5.19. Note the accurate alignment of bounding boxes on targets 1 and 5 (red and magenta) when the state is represented in continuous space. The reconstructed trajectories remain smooth over time because they do not rely on discretization. Note that, despite using a dynamic model within the ILP-formulation, a trajectory can suffer from undesirable jittering when a target moves parallel to one of the grid axes but in between two rows of cells, such as target number 12 (yellow).

Average quantitative results are summarized in Table 5.5. The drop in recall for the ILP approach can be explained by a too strict pruning, such that too many tracklets are removed from the solution space when a target is occluded for several frames in a row. A more conservative pruning strategy may recover more targets but is not feasible in practice due to the increased size of the integer program. As expected, MOTP (measuring the alignment error between ground truth and tracker output) rises by almost 10 percentage points without discretizing the state space. The continuous method can thus clearly show its advantages.

Method	MOTA	MOTP	MT	ML	Recall	Precision
ILP	71.2%	68.5%	8.7	0	78.3%	93.0%
Cont.	82.2%	77.1%	10.7	0	89.7%	92.8%

Table 5.5: Average results on three sequences.

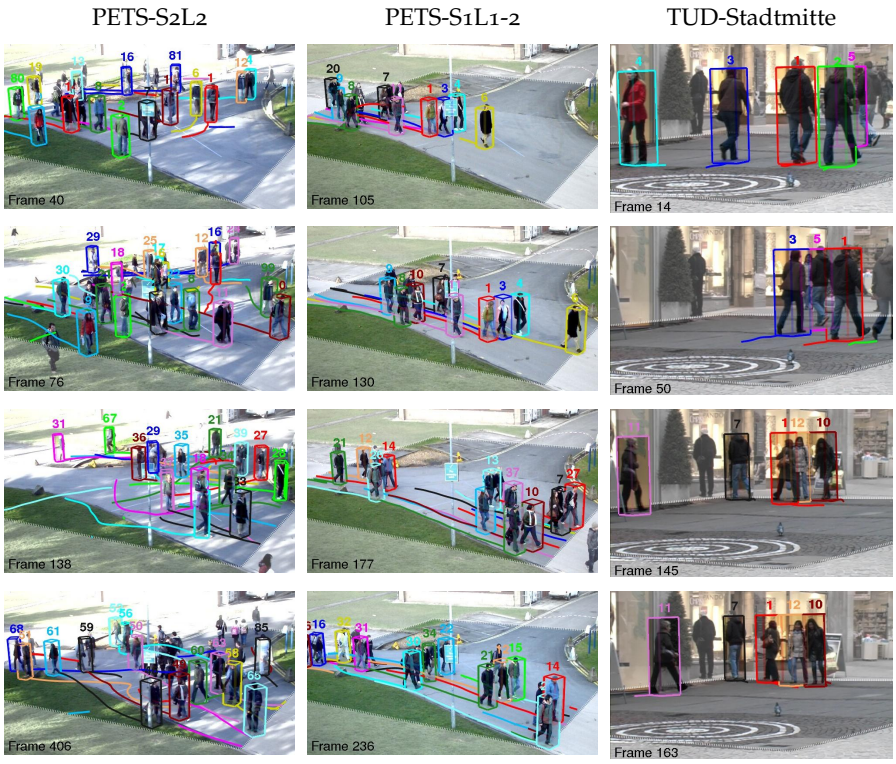


Figure 5.20: Qualitative tracking results of the presented method (FULL).

5.5.5 *Qualitative results*

Figure 5.20 shows example frames of our full model on three datasets. The region outside the tracking area is grayed out. The presented continuous energy minimization approach is able to accurately reconstruct full trajectories in rather challenging scenarios. For example, in *S1L1-2 (middle column)*, targets 1, 3 and 7 (red, blue and black) are correctly tracked throughout the entire sequence. In the more challenging case *(S2L2)* targets move more randomly inside the tracking area causing frequent occlusions. Moreover, more targets are missed by the detector due to poor lighting conditions. Nonetheless, the reconstructed trajectories provide a reasonable estimate for the overall situation. Finally, the rightmost column shows a scene filmed from a low view point, which leads to rather inaccurate target localization on the ground plane. This causes track number 1 (red) to jump from one person to a more distant one (between rows 2 and 3). Note that, although in world units they may be several meters from one another, they are only separated by few pixels on the image.

5.5.6 *Quantitative evaluation*

Our method is validated on seven challenging, publicly available video sequences. Six of them are part of the PETS 2009/2010 bench-

Sequence	Method	MOTA	MOTP	MT	ML	FP	FN	ID	FM	Rcll	Prcsn	Fa/F
mean (G1)	Det	-	-	-	-	900.7	158.0	-	-	89.1	60.6	2.7
(low density)	FULL	**86.1**	76.1	**11.7**	0.3	52.0	140.7	**5.0**	**3.0**	91.6	94.7	0.2
(13 GT tracks)	OM	85.6	74.8	**11.7**	**0.0**	118.0	**95.0**	8.3	5.3	**93.0**	93.1	0.3
	no OM	85.6	**76.6**	11.0	0.3	**35.0**	148.7	7.0	5.0	88.9	**96.5**	**0.1**
	KSP	69.9	68.8	8.0	1.3	88.3	321.3	6.0	12.3	78.3	90.3	0.3
	BPF	45.4	68.2	8.7	0.3	566.7	317.0	34.0	43.7	81.1	70.6	1.5
	EKF	64.3	72.2	4.7	0.3	60.0	504.7	9.0	12.3	70.7	93.2	0.2
mean (G2)	Det	-	-	-	-	1331.8	1919.5	-	-	**56.5**	66.4	4.4
(high density)	FULL	**47.8**	58.2	15.8	15.2	291.5	1919.8	54.8	37.0	55.1	89.7	1.0
(49 GT tracks)	OM	47.6	59.1	**16.8**	**13.8**	337.8	**1873.0**	56.5	43.5	55.7	88.9	1.1
	no OM	45.3	59.7	12.5	15.0	203.0	2154.5	59.8	40.0	50.8	91.9	0.7
	KSP	31.0	62.0	8.0	28.5	100.0	3121.8	**10.0**	**17.2**	33.5	93.5	0.3
	BPF	30.1	**62.7**	6.2	21.5	257.0	2773.8	91.8	143.5	36.9	88.4	0.8
	EKF	23.3	60.0	1.5	29.5	**85.2**	3274.5	29.0	53.2	25.1	**94.9**	**0.2**

Table 5.6: Average quantitative results on all datasets. Due to the large variability in the number of targets, we report averages over the easier *(G1)*, first three datasets in Table 5.7) and the four more challenging sequences *(G2)* separately. We additionally report the average performance of the underlying people detector. See Section 5.5.6 for more details.

mark (Ferryman and Shahrokni, 2009; Ferryman and Ellis, 2010) and one is from the *TUD* dataset. All datasets and the evaluation metrics are presented and described in detail in Section 3.3 and in Section 3.4, respectively.

For clarity, the quantitative results for all metrics are presented in two separate tables. Table 5.6 shows the average performance of all methods (including the average detector performance *(Det)*), while Table 5.7 reveals a detailed breakdown on each individual sequence. Since the data exhibits a strong variability in person count, we compute the average performance for two separate groups of sequences: An easier set *(G1)*, containing less than 10 individuals per frame, and a more challenging group *(G2)*, where up to 42 pedestrians are present simultaneously. Please note that we use the same set of parameters for each approach on all seven sequences.

We report the results of six methods: The full model including occlusion reasoning and the appearance model, denoted as FULL (see also Figure 5.20 for a visual illustration). For comparison, we also report results of our method without appearance term, both without (no OM, Andriyenko and Schindler (2011)) and with occlusion modeling (OM, Andriyenko et al. (2011)). Note that the results for these two previous methods improve upon those presented in the respective previous publication. The results are compared to those of a state-of-the-art discrete tracker (Berclaz et al., 2011), based on

the k-shortest paths (KSP) algorithm on a regular grid as well as to a well-known boosted particle filter (BPF) method (Okuma et al., 2004). Finally, the tracking results of an extended Kalman filter (EKF) serve as a baseline.

OCCLUSION (OM). As expected, explicitly taking occlusion into account increases the overall tracking accuracy (MOTA) over the most basic continuous formulation. In all cases, the number of missed targets is reduced significantly. This is most prominent in the difficult case *S2L2*, where approximately 800 more targets are found through explicit occlusion reasoning, which amounts to about two targets per frame (*cf*. *FN* in Table 5.7). However, in less dense tracking scenarios *(G1)* occlusion computation cannot show its benefits, because pedestrians are fully visible most of the time. In fact, the accuracy only improves on the *TUD-Stadtmitte* sequence, where pedestrians are frequently fully occluded due to the low viewing angle (see Figure 5.20). In the other two cases (*S2L1* and S3-MF1), the higher number of false positives causes the overall accuracy to drop around three percentage points such that, on average, the MOTA stays at 85.6% both with and without occlusion modeling. On the other hand, in crowded environments the accuracy increases by over 2 percentage points on average (from 45.3% to 47.6%), and over 5 percentage points in the most difficult case (*PETS-S2L2*), achieving a MOTA of 57.2%. There, the number of identity switches (120) may seem rather high. However, given the complexity of this sequence with a per-frame average of 19 targets being inside the tracking area, one identity switch every 3 to 4 frames is tolerable. Within the same group *(G2)*, the number of mostly tracked targets rises by 35%, while having almost 10% fewer trajectories that are mostly lost without modeling occlusions. In less dense sequences, no ground truth target is tracked for less than 20% of its length, *i.e.* all trajectories are either fully or partially recovered.

APPEARANCE (FULL). Compared to our full tracking system including occlusion reasoning (OM), the appearance model forces some parts of the tracks to be removed, thereby raising the amount of missed targets by $\approx$ 5% on average. At the same time, the number of ID swaps is almost halved for the low density group *(G1)* and still reduced by $\approx$ 3% in the difficult cases. A similar trend can be observed for track fragmentations *(FM)*. Only three interruptions of ground truth trajectories are counted on average for *G1*. More prominent is the effect on false alarms. The use of the appearance model weeds out 56% of all false positive detections in less dense scenarios, yielding a false alarm rate *(Fa/F)* of only 0.2 targets per frame. The number of false positives still remains higher than in the most basic case *(no OM)* because taking occlusions into account drives the op-

timization towards more "hallucinated" targets without any image evidence.

Even though including the appearance model does not lead to higher combined accuracy score in every single case, it turns out to improve the performance on average and must not be ignored when the correct identification of targets is crucial.

K-SHORTEST PATHS (KSP). For a comparison to tracking on a discrete grid (Berclaz et al., 2011), the detections are projected onto the ground plane and the target evidence is distributed to all neighboring cells according to a normal distribution. The corresponding parameters have been manually determined to yield the best possible results. Discrete global optimization clearly outperforms the recursive tracker (EKF) in terms of accuracy, by recovering more trajectories while better keeping track of the target identities. However, the proposed continuous scheme outperforms the discrete tracker on all sequences. Moreover, the spatial discretization limits the achievable precision. This becomes most apparent in the low density setting *(G1)* where targets can be localized more precisely by the detector. Here, the MOTP score is 3.4% lower than that of a Kalman filter and 6.5% lower than our best result (FULL).

BOOSTED PARTICLE FILTER (BPF). To compare our method to another baseline we use a recent implementation[3] of the boosted particle filter (BPF) (Okuma et al., 2004), where we tuned the parameters to achieve the best possible results. While this method recovers substantially more tracks than the Kalman filter, it struggles to suppress persistent false alarm detections, which in turn leads to a low precision value. Note, however, that this approach operates entirely in image space and does not require any camera calibration.

5.6 DISCUSSION

In this chapter we have presented a continuous energy minimization framework for multi-target tracking, which includes explicit occlusion reasoning and appearance modeling. Contrary to many previous non-recursive tracking methods, the aim here was to forgo the goal of achieving (near) globally optimal solution of the objective and instead model (most of) the crucial aspects of tracking multiple targets as closely as possible. All components are modeled by closed-form, continuously differentiable functions, which allow for an efficient evaluation of the gradient in closed form. The resulting non-convex energy is minimized by both, a local gradient descent search and a set of discontinuous jump moves. Although the energy can only be minimized locally, an extensive experimental evaluation

3 http://www.cs.ubc.ca/~okumak/research.html

on several challenging datasets showed that our approach leads to very competitive results, both visually and in terms of quantitative evaluation with respect to ground truth. Although the novel, differentiable appearance model does not lead to a consistent accuracy improvement across all sequences, it significantly reduces the number of false positives and identity switches, which are an important factor in a number of applications, such as surveillance and video analysis.

One of the main drawbacks of the approach described in this chapter remains the highly non-convex optimization problem that can only be solved locally. Although the proposed jump moves offer a great flexibility, they are executed greedily and only one jump at a time. In some situations, a local minimum can only be escaped by performing two (or more) discrete jumps of different types simultaneously, *e.g.* entirely removing one trajectory and instantaneously connecting two other ones to fill in the resulting gap. Such 'high-order' moves seem conceivable, but a naive implementation would lead to complex combinatorial problems that cannot be solved fast enough for practical applications. A further limitation is the absence of explicit data association. For some applications it may be useful to cluster the detector responses into groups that belong to the same individuals, for example to learn their appearance.

Both these issues are addressed in the third and final part of this dissertation. Chapter 6 will introduce a combined discrete-continuous energy formulation for multi-target tracking.

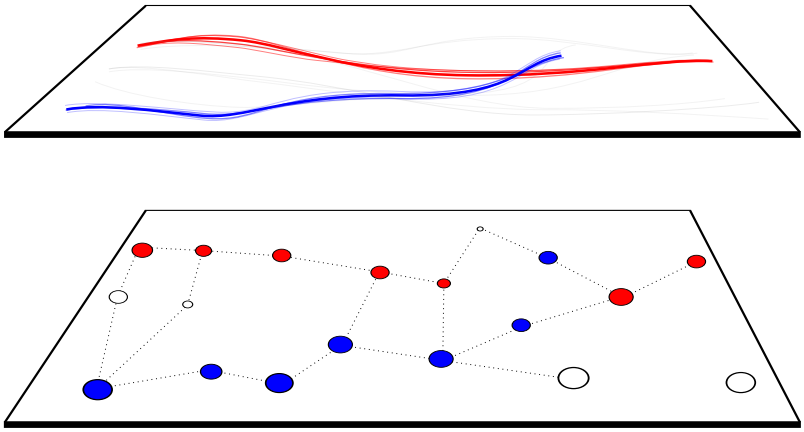
Sequence	Method	MOTA	MOTP	MT	ML	FP	FN	ID	FM	Rcll	Prcsn	Fa/F
PETS-S2L1	FULL	90.6	**80.2**	**21**	1	59	302	**11**	**6**	92.4	98.4	0.07
(795 frames)	OM	88.6	76.9	**21**	**0**	259	**171**	19	12	**95.7**	93.6	0.33
(⩽8 targets)	no OM	**91.6**	79.3	**21**	**0**	**53**	262	16	11	93.4	**98.6**	**0.07**
(23 GT tracks)	KSP	80.3	72.0	17	2	126	641	13	22	83.8	96.3	0.16
	BPF	48.8	73.3	16	1	1257	689	78	81	82.8	72.2	1.58
	EKF	68.0	76.5	9	1	65	1173	25	30	70.3	97.7	0.08
Stadtmitte	FULL	71.1	65.5	**7**	**0**	92	108	4	**3**	84.7	86.7	0.51
(179 frames)	OM	**73.4**	65.0	**7**	**0**	83	**102**	3	**3**	**85.6**	87.9	0.46
(⩽5 targets)	no OM	68.0	**67.1**	5	1	**49**	172	5	4	75.7	**91.6**	**0.27**
(9 GT tracks)	KSP	45.8	56.7	1	1	117	261	5	15	63.1	79.2	0.65
	BPF	19.7	54.8	4	**0**	324	222	18	43	68.4	59.7	1.82
	EKF	58.2	58.3	3	**0**	115	172	**2**	6	75.1	81.9	0.65
PETS-S3-MF1	FULL	96.7	82.7	**7**	**0**	5	**12**	**0**	**0**	**97.7**	99.0	0.05
(107 frames)	OM	94.7	82.6	**7**	**0**	12	**12**	3	1	**97.7**	97.7	0.11
(⩽7 targets)	no OM	**97.1**	**83.4**	**7**	**0**	3	**12**	**0**	**0**	**97.7**	99.4	0.03
(7 GT tracks)	KSP	83.7	77.8	6	1	22	62	**0**	**0**	87.9	95.4	0.21
	BPF	67.9	76.5	6	**0**	119	40	6	7	92.2	79.9	1.12
	EKF	66.7	81.9	2	**0**	**0**	169	**0**	1	66.7	**100.0**	**0.00**
PETS-S2L2	FULL	56.9	59.4	28	12	622	2881	99	73	65.5	89.8	1.43
(436 frames)	OM	**57.2**	59.7	**31**	**8**	772	**2684**	120	87	**67.9**	88.0	1.77
(⩽33 targets)	no OM	51.9	60.1	18	11	434	3473	115	86	58.4	91.8	1.00
(74 GT tracks)	KSP	24.2	**60.9**	7	40	**193**	6117	**22**	**38**	26.8	**92.1**	**0.44**
	BPF	33.1	59.8	8	17	657	4690	236	393	43.8	84.7	1.51
	EKF	28.6	60.3	2	32	280	5565	74	116	32.9	90.7	0.64
PETS-S2L3	FULL	**45.4**	64.6	9	**18**	169	1572	38	27	51.8	90.9	0.70
(240 frames)	OM	43.9	61.4	**11**	20	214	1586	28	22	51.3	88.7	0.89
(⩽42 targets)	no OM	44.1	**65.8**	9	22	89	1694	38	22	48.0	94.6	0.37
(44 GT tracks)	KSP	28.8	61.8	5	31	45	2269	**7**	**12**	30.4	95.7	0.19
	BPF	31.5	65.8	4	27	71	2110	51	72	35.2	94.2	0.30
	EKF	20.4	63.3	1	35	**13**	2543	8	33	21.1	**98.1**	**0.05**
PETS-S1L1-2	FULL	57.9	59.7	**19**	11	148	918	21	13	64.5	91.8	0.61
(241 frames)	OM	57.8	61.9	18	8	188	875	27	20	66.2	90.1	0.78
(⩽20 targets)	no OM	**59.0**	59.2	16	**4**	118	921	22	16	64.4	93.4	0.49
(36 GT tracks)	KSP	51.5	64.8	16	14	98	1151	**4**	**8**	55.5	93.6	0.41
	BPF	37.6	**66.7**	10	14	185	1407	21	32	45.5	86.4	0.77
	EKF	34.6	63.2	3	17	**10**	1664	6	18	35.2	**98.9**	**0.04**
PETS-S1L2-1	FULL	30.8	49.0	**7**	20	227	2308	61	35	38.5	86.4	1.13
(201 frames)	OM	**31.4**	53.2	**7**	**19**	177	2347	51	45	37.4	88.8	0.88
(⩽42 targets)	no OM	26.3	53.5	**7**	23	171	2530	64	36	32.6	87.7	0.85
(43 GT tracks)	KSP	19.5	**60.6**	4	29	64	2950	**7**	**11**	21.4	**92.6**	0.32
	BPF	18.4	58.6	3	28	115	2888	59	77	23.0	88.2	0.58
	EKF	9.5	53.1	0	34	**38**	3326	28	46	11.3	91.8	**0.19**

Table 5.7: Quantitative results on all datasets. The number of frames, the crowd density (maximal number of simultaneous targets) and the number of ground truth trajectories are stated for each sequence. The first three sequences show moderately crowded scenes, while the last four are more challenging, showing up to 42 targets simultaneously. See Section 5.5.6 for a detailed discussion.

Part III

TRACKING IN DISCRETE-CONTINUOUS SPACE

Parametric trajectory models are fitted to the observations in continuous space, while data association is approached as a multi-labeling problem that is solved via discrete optimization.



6

DISCRETE-CONTINUOUS OPTIMIZATION FOR MULTI-TARGET TRACKING

When you combine ignorance and leverage, you get some pretty interesting results.

WARREN BUFFETT

CONTENTS

THE two previous chapters approached the multi-target tracking task from two orthogonal directions. The main idea described in Chapter 4 was to reduce the state space to a countable finite set. To this end, the tracking area is subdivided into an array of identical disjunctive cells and all feasible paths are formed by edges between spatially neighboring cells in adjacent frames. Tracking is then formulated as an integer linear program (ILP), where binary variables indicate targets' motion between grid cells. Even though the relaxation of the resulting ILP as defined in Eq. (4.3-4.7) forms a convex objective

function that can efficiently be solved to (near) global optimality, the objective itself remains a rather crude approximation. For instance, collision between targets can only be modeled as a binary decision and the dynamics term only incorporates a few discrete values that depend on the chosen neighborhood.

The continuous optimization approach presented in Chapter 5 follows a rather opposite strategy. Here, no constraints are imposed on the state space and all trajectories are represented entirely in continuous space. The plausibility of a certain solution is modeled by several terms that are combined to form a high-dimensional, highly non-convex energy function (Eq. (5.1)). To still enable the optimization to find good optima of the energy, the classical conjugate gradient descent is augmented with heuristic discontinuous jumps that are executed in a greedy manner.

In this chapter we again address the multi-target tracking task by minimizing a global energy. However, it is conceptually different from the two aforementioned formulations. The state now contains both discrete *and* continuous variables. The discrete part of the energy is designed as a graphical model and handles the data association, while the continuous part assesses the goodness of the actual trajectories without constraining them to a finite set of locations. The joint estimation of all variables is formulated as the *minimization of a consistent discrete-continuous energy*, which treats each aspect in its natural domain. Moreover, we conduct a statistical analysis of ground-truth data to develop a better intuition of the underlying distributions. The results of this analysis then serve as basis for the choice of individual components of the energy.

Two alternatives of the discrete-continuous formulation are presented in this chapter. The first one places emphasis on maintaining a well-behaved optimization problem, where the discrete part of the energy remains submodular and the (simplified) continuous part is a smooth convex function. Therefore, the latter can be optimized globally in closed form while the multi-labeling solution provides theoretical optimality bounds. Parts of this work have appeared in (Andriyenko et al., 2012). The second variant describes a more accurate model that aims to correctly handle target exclusion in both the discrete domain at the level of data association and in the continuous domain at the level of trajectories. To that end, we develop a pairwise label cost that penalizes co-existence of mutually excluding targets. The proposed formulation is generic and can be applied to other multi-labeling problems beyond multi-target tracking. Furthermore, we propose an iterative optimization scheme based on expansion moves and message passing to locally minimize the resulting energy. Despite the fact that the optimization becomes more complex, experiments on challenging sequences confirm the benefits of this formulation. This approach has been accepted for publication and will

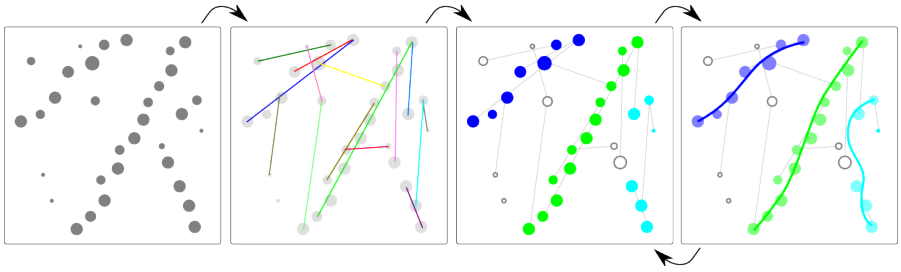


Figure 6.1: Given a number of unlabeled object detections and a number of possible trajectory hypotheses, the method presented in this chapter labels all detections and re-estimates the trajectories using an alternating discrete-continuous optimization scheme.

appear in the Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2013 (Milan et al., 2013b).

6.1 INTRODUCTION

The task of tracking multiple targets is comprised of two separate, but closely linked challenges. Intuitively speaking, one has to establish a unique identity for each target, and then simultaneously estimate the motion patterns of all targets and the assignment of detections to the targets. In realistic conditions, both tasks are complicated by an unknown number of targets, missing or spurious detections due to occlusion or clutter, as well as physical phenomena that impose complex constraints between different variables (targets). Addressing these challenges requires coping with two distinct, but tightly coupled modeling issues. The task of data association, *i.e.* labeling each detection as either belonging to a certain target or being a false alarm, is intrinsically a *discrete problem* with unordered labels. However, trajectory estimation, *i.e.* reconstructing the target locations over time, is a problem that is naturally described in a *continuous state space*.

Besides location, the state may also include further properties such as size, velocity, etc.

Existing techniques strike the balance between the two tasks in different ways. An extensive body of recent work focuses on data association and uses powerful discrete optimization algorithms to approach this NP-hard problem. However, the continuous aspect of trajectory estimation suffers, either because trajectories have to be precomputed in absence of any data association (Zhang et al., 2008; Wu et al., 2011), or the trajectories are spatially discretized (see Chapter 4). Other techniques focus on trajectory estimation in a continuous state space, but limit the data association to a choice from a pre-computed set of potential labelings (Leibe et al., 2007). The energy minimization approach discussed in Chapter 5 also deals with continuous trajectory estimation, but sidesteps the classical data association problem.

In this chapter we aim to unify data association and trajectory estimation in a single model that formulates each aspect in its natural

domain through the *minimization of a consistent discrete-continuous energy*. To that end we build on recent advances in multi-model fitting introduced by Delong et al. (2012). We show how to formulate multi-target tracking in that framework and extend the inference algorithm accordingly. Even though two different versions of this formulation will be introduced, the underlying methodology remains unchanged: Trajectories are modeled by piecewise polynomials, which are fitted to a set of target hypotheses. Given these trajectories, the data association is updated by solving a multi-labeling problem, taking into account global trajectory properties such as the dynamics and persistence of moving objects, as well as mutual exclusion between trajectories through individual, respectively pairwise label costs. The two steps are alternated to minimize a single discrete-continuous objective, such that trajectory estimation can take advantage of data association and vice versa. The principle of the algorithm is illustrated in Figure 6.1.

This chapter makes the following contributions:

- We formulate a unified discrete-continuous energy for multi-target tracking;
- we demonstrate the applicability of the label-cost framework to the tracking problem;
- we introduce a pairwise label cost to handle mutual dependencies in the model selection;
- we introduce an energy minimization algorithm for pairwise label costs; and
- we provide a systematic analysis of ground-truth data to extract the underlying statistics of multi-target tracking.

6.2 DISCRETE-CONTINUOUS MULTI-OBJECT TRACKING

To formally describe our method, we rely on the notation introduced in Section 3.1, and already employed in Chapter 5. However, the following formulation requires a larger set of symbols since it deals with two types of functions. We therefore need to extend the notation, respectively undertake some minor modifications. In particular, to easily distinguish between discrete and continuous variables, a sans-serif font ($\mathsf{a}, \mathsf{b}, \mathsf{A}, \mathsf{B}, \ldots$) is used to accentuate the discrete ones. Bold letters ($\mathbf{A}, \mathbf{B}, \ldots$) denote discrete sets, while calligraphic letters ($\mathcal{A}, \mathcal{B}, \ldots$) represent continuous sets. With these modifications, N and F correspond to the number of all trajectory hypotheses and the length of the sequence (in frames), respectively. Trajectories, which are represented by splines (see below), are denoted with $\mathcal{T}$ and their temporal limits are s and e. As before, $\mathbf{D}_{\mathsf{i}}^{\mathsf{t}}$ is the location of one particular detection

Symbol	Description
$\mathcal{T}$	set of all N models (trajectory hypotheses)
$\mathcal{T}^* \subseteq \mathcal{T}$	set of all active trajectories
$\mathcal{T}_i$	trajectory i
$\mathcal{T}_i(t) = \mathbf{X}_i^t$	(X, Y)-position of trajectory i in frame t
$\mathbf{D}$	set of all detections
d_g^t	detection g in frame t
$\mathbf{D}_g^t$	position of detection g in frame t
$\mathbf{L}$	set of all possible labels (trajectory hypotheses)
$f_{d_g^t}$	label of detection g in frame t
$\mathbf{f}$	labeling of all detections
$\varnothing$	outlier label
$\mathbf{E}$	set of all edges of the CRF
$\mathbf{E}_S$	temporal smoothness edges
$\mathbf{E}_X$	detection-level exclusion edges
$\Gamma(\cdot, \cdot)$	distance function (data term)
ϕ, ψ	unary and pairwise potentials of the CRF
$h_f(\mathcal{T}_i)$	label cost for trajectory i
$h_f^X(\mathcal{T}_i, \mathcal{T}_j)$	pairwise label cost for trajectories i and j

Table 6.1: Additional notation used in this chapter.

response. To refer to the random variable of that detection, we use the lowercase d_i^t. Each random variable d is assigned a label $f_d \in \mathbf{L}$ from the label set $\mathbf{L} = \{1, \ldots, N, \varnothing\}$ of all trajectory hypotheses, where $\varnothing$ denotes an outlier label, or equivalently a false alarm. The notation is once again listed in Table 6.1.

6.2.1 *Continuous trajectory model*

In contrast to the purely discrete approach to multi-target tracking that was discussed in Chapter 4, individual trajectories are represented in continuous space. However, unlike the explicit sequence of per-frame coordinates as in Chapter 5, we choose a parametric model, in particular we use cubic B-splines for that purpose. This turns out to be a suitable representation for target motion in real-world scenarios, as it avoids discretization artifacts and of-

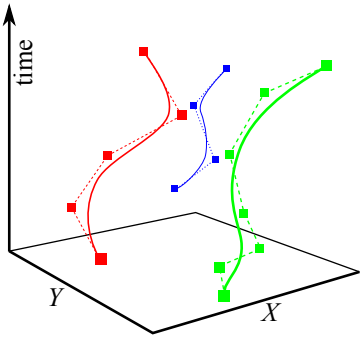


Fig. 6.2: Trajectories are represented by 2D cubic B-splines.

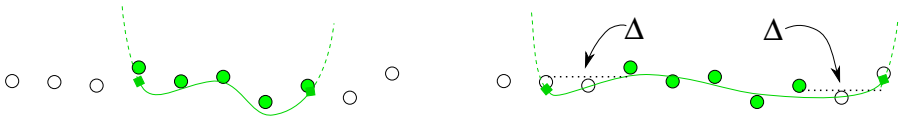


Figure 6.3: A safety margin is added to each side of the trajectory (right) to avoid extreme values of the spline immediately beyond its support (left).

fers a good trade-off between model flexibility and intrinsic motion smoothness. More specifically, the spline for each trajectory

$$\mathcal{T}_i : t \in \mathbb{R}_0^+ \rightarrow (X, Y)^T \in \mathbb{R}^2 \tag{6.1}$$

describes the target location $(X, Y)^T$ for each point in time t, as illustrated in Figure 6.2. We assume that the spline has a varying number c_i of control points and is parametrized by a coefficient matrix $C_i \in \mathbb{R}^{2c_i \times 4}$. The number of control points depends on the length of each trajectory and is set to $\max(4, \lfloor F(i)/2.5 \rceil)$, where $\lfloor \cdot \rceil$ is the rounding operator. We found that it is advantageous to explicitly model the temporal starting points s_i and end points e_i of each trajectory ($t \in [s_i - \Delta, e_i + \Delta]$), because the splines tend to take on extreme values outside their support otherwise, which results in highly unlikely motion patterns. To ensure that the spline does not take on extreme values immediately outside of $[s, e]$, which would prevent other detections in adjacent frames from being assigned to the trajectory later, we add a safety margin of Δ on either side (*cf*. Figure 6.3).

If we for now suppose that we are already given a data association $\mathbf{f}$, we can formulate the trajectory estimation problem as minimization of the energy

$$E_{\mathbf{f}}^{te}(\mathcal{T}) = \sum_{i=1}^{N} \left(E_{\mathbf{f}}^{te}(\mathcal{T}_i) + \hat{E}_{\nu}^{te}(\mathcal{T}_i) \right), \tag{6.2}$$

where $E_{\mathbf{f}}^{te}(\mathcal{T}_i)$ models how well trajectory $\mathcal{T}_i$ fits to the hypotheses assigned by $\mathbf{f}$ and $\hat{E}_{\nu}^{te}(\mathcal{T}_i)$ models the smoothness of $\mathcal{T}_i$ on the safety margin. For each trajectory we aim to minimize the weighted distance to each assigned target hypothesis – which is computed by the distance function Γ – in all valid frames:

$$E_{\mathbf{f}}^{te}(\mathcal{T}_i) = \sum_{t=s_i}^{e_i} \sum_{j=1}^{D(t)} \delta\left[i - f_{d_j^t}\right] \cdot \omega_j^t \cdot \Gamma\left(\mathbf{D}_j^t, \mathbf{X}_i^t\right), \tag{6.3}$$

where $D(t)$ is the number of detections in frame t and ω_j^t is the confidence value of the j^{th} detection response. The Kronecker delta

$$\delta[a - b] = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise,} \end{cases} \tag{6.4}$$

ensures that only target hypotheses d_j^t are counted that are assigned to trajectory i. On the safety margin the spline is fit to virtual locations v_i^t obtained by linear extrapolation (dotted lines in Figure 6.3):

$$\hat{E}_v^{te}(\mathcal{T}_i) = \sum_{\substack{s_i-\Delta \leqslant t < s_i \\ e_i < t \leqslant e_i+\Delta}} \Gamma\left(v_i^t, \mathbf{X}_i^t\right). \tag{6.5}$$

In all our experiments we use $\Delta = 2$ frames. The difficulty of minimizing Eq. (6.2) depends on the exact definition of the distance function Γ. We will see in Section 6.3 that it can be solved in closed form if the squared Euclidean distance is used for that purpose.

6.2.2 *Discrete data association*

Data association is often the most challenging aspect of tracking multiple targets. We formulate it explicitly as a multi-labeling problem, which has the advantage that powerful discrete optimization approaches can be leveraged. Recalling the notation from above, our goal is to estimate a labeling $\mathbf{f}$ that uniquely assigns each detection $d \in \mathbf{D}$ to one of the N trajectory hypotheses $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$, or identifies it as a false alarm using the outlier label $\varnothing$.

A large class of labeling problems in computer vision are formulated in terms of the minimization of an energy of a discrete, pairwise conditional random field (CRF). This also serves as the starting point here. To that end, we identify each individual detection $d \in \mathbf{D}$ with a vertex of the graph $\mathbf{G} = \{\mathbf{D}, \mathbf{E}\}$. The energy is then decomposed into unary and pairwise potentials:

$$E_{\mathcal{T}}^{da}(\mathbf{f}) = \sum_{d \in \mathbf{D}} \phi_d(f_d, \mathcal{T}) + \sum_{(d,d') \in \mathbf{E}} \psi_{d,d'}(f_d, f_{d'}). \tag{6.6}$$

As usual, the data term is responsible for keeping the solution close to the observed data. To stay consistent with Eq. (6.3), we use the same general formulation of the distance function Γ between the detection location $\mathbf{D}_j^t$ and its associated trajectory $\mathcal{T}_l$, weighted by the detection confidence ω_j^t:

$$\phi_{d_j^t}(l, \mathcal{T}) = \omega_j^t \cdot \Gamma\left(\mathbf{D}_j^t, \mathbf{X}_l^t\right). \tag{6.7}$$

If the detection is labeled as an outlier, it is penalized with a constant outlier cost $\Gamma_\varnothing$, again modulated by ω_j^t:

$$\phi_{d_j^t}(\varnothing, \mathcal{T}) = \omega_j^t \cdot \Gamma_\varnothing. \tag{6.8}$$

A low confidence score of the object detector usually means one of two things: either the output is a false alarm, or the bounding box is not properly aligned with the object. The data term incorporates this by penalizing a larger distance to a weak detection less than to

a confident one (Eq. (6.7)). The weight of the outliers is similarly reduced (Eq. (6.8)), so as to promote false detections being labeled as outliers.

While the exact form of ϕ is less important for optimization, the pairwise connections ψ often lead to complex combinatorial problems that are hard to optimize in general. A notable exception are energies that belong to a certain class of functions, for which polynomial-time inference algorithms exist. In particular, if the label space is binary and ψ is a metric, then the global minimum of Eq. (6.6) can be found efficiently (Kolmogorov and Zabih, 2004). Especially the first one is a rather strong restriction that, unfortunately, cannot be met by the proposed multi-target tracking formulation. Nonetheless, efficient approximate algorithms exist to handle the multi-label case that we describe here (Boykov et al., 2001). Moreover, such expansion-based optimization provides optimality bounds on the obtained solution. In particular, for a fixed set of trajectories $\mathcal{T}$, any local minimum $\hat{\mathbf{f}}$ of the energy from Eq. (6.11) is guaranteed to be within a factor of 2 from the globally optimal solution $\mathbf{f}^*$:

The additional constant factor vanishes because ψ_S is identical for all $f_d \neq f_{d'}$.

$$E(\mathcal{T}, \hat{\mathbf{f}}) \leqslant 2 \cdot E(\mathcal{T}, \mathbf{f}^*). \tag{6.9}$$

As for the definition of ψ, we will discuss two alternatives – one with only submodular terms in Section 6.3 and one with arbitrary pairwise connections in Section 6.5.

6.2.3 *Discrete-continuous tracking with label costs*

Due to the choice of formulations for both trajectory estimation and data association, it is now possible to unify them in a single, consistent energy function:

$$\begin{aligned} E(\mathcal{T}, \mathbf{f}) = &\sum_{d \in \mathbf{D}} \phi_d(f_d, \mathcal{T}) + \sum_{(d,d') \in \mathbf{E}} \psi_{d,d'}(f_d, f_{d'}) \\ &+ \sum_{i=1}^{N} \hat{E}_v^{te}(\mathcal{T}_i) + \lambda_{h_f} \cdot h_{\mathbf{f}}(\mathcal{T}). \end{aligned} \tag{6.10}$$

To understand this formulation, it is instructive to first consider the case when the last term is not active (*i.e.* $\lambda_{h_f} = 0$). In this case we can make the following observations:

- minimizing Eq. (6.10) w.r.t. the trajectories $\mathcal{T}$ given a fixed labeling $\mathbf{f}$ is equivalent to trajectory estimation, *i.e.* minimizing Eq. (6.2), and
- minimizing it w.r.t. the labeling $\mathbf{f}$ given fixed trajectories $\mathcal{T}$ is equivalent to data association, *i.e.* minimizing Eq. (6.6).

However, alternating minimization of such an objective will not lead to the desired result. The most obvious problem (but not the only

one) is that neither of the two parts includes a model selection term to regularize the number of trajectories. Given the variable number of targets, the alternation would thus overfit by instantiating more trajectories to reduce the fitting error.

To overcome the problem we follow the recent work of Delong et al. (2012) and rely on a so-called label cost term $h_{\mathbf{f}}(\mathcal{T})$, which specifies a cost that is applied to each label and takes effect as long as the labeling $\mathbf{f}$ contains this label at least once. More specifically, our label cost term $h_{\mathbf{f}}(\mathcal{T})$

- integrates a dynamic model that includes both linear and angular velocities and keeps trajectories within physical limits,
- encourages long, persistent trajectories, by penalizing long sections of missing evidence, as well as tracks that start or end far from the image border, and finally
- penalizes the total number of current targets.

Although the label cost $h_{\mathbf{f}}(\mathcal{T})$ induces high-order cliques, it can be decomposed into pairwise potentials that are submodular to enable minimization by move-making algorithms (Boykov et al., 2001; Delong et al., 2012). Thus, a strong local optimum of the energy in Eq. (6.10) with respect to $\mathbf{f}$ can still be found efficiently as long as the pairwise potentials ψ also remain submodular. However, as we will see in Section 6.5, a more complex label cost that does not meet the requirements for efficient inference is needed to deal with more challenging sequences.

6.3 SUBMODULAR-CONVEX ENERGY

We will now describe the first variant of our discrete-continuous energy

$$E(\mathbf{f}, \mathcal{T}) = \sum_{d \in \mathbf{D}} \phi_d(f_d, \mathcal{T}) + \sum_{(d,d') \in \mathbf{E}_S} \psi_S(f_d, f_{d'}) + h_{\mathbf{f}}(\mathcal{T}), \tag{6.11}$$

where the discrete part is submodular and the (simplified) continuous part is globally optimizable in closed form. To that end, we need to define both types of potentials, the underlying graph structure and the label cost.

UNARY TERMS. Recalling Equations (6.3), (6.5) and (6.7), the data term measures the distance between the models (in our case trajectories) and the observed data (detections) and is computed by the distance function Γ. This is true for both the discrete and the continuous component of the energy. To enable efficient continuous optimization

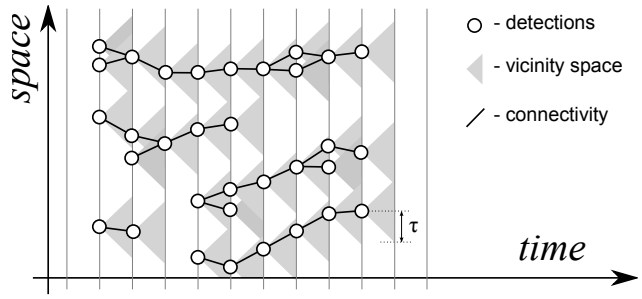


Figure 6.4: Neighborhood structure of the underlying pairwise conditional random field (CRF). Detections in adjacent frames are connected if their distance is below a certain threshold.

that computes the global minimum of $\mathcal{T}$ given $\mathbf{f}$ in closed form, we employ the squared Euclidean distance:

$$\Gamma\left(\mathbf{D}_j^t, \mathbf{X}_i^t\right) = \lambda_\phi \cdot \left\|\mathbf{D}_j^t - \mathbf{X}_i^t\right\|^2, \tag{6.12}$$

such that the cost for a detection d_j^t to belong to trajectory i becomes

$$\phi_{d_j^t}(i, \mathcal{T}) = \lambda_\phi \cdot \omega_j^t \cdot \left\|\mathbf{D}_j^t - \mathbf{X}_i^t\right\|^2. \tag{6.13}$$

Optimizing the continuous part of $E(\mathbf{f}, \mathcal{T})$ without the label cost then corresponds to solving a system of linear equations in a least squares sense.

PAIRWISE TERMS. To construct pairwise cliques, all pairs of detections in adjacent frames whose distance is below a threshold τ are connected by an edge (*cf*. Figure 6.4):

$$\mathbf{E}_S = \left\{ \left(d_j^t, d_k^{t+1}\right) \middle| \left\|\mathbf{D}_j^t - \mathbf{D}_k^{t+1}\right\| < \tau, t = 1, \ldots, F-1 \right\}. \tag{6.14}$$

The motivation for this is that nearby detections in adjacent frames should be encouraged to have the same trajectory label. We refrain from longer-range temporal connections, as a large threshold τ would be needed to allow for sufficient target dynamics, coming at the cost of a dense graph and potentially inappropriate label smoothing.

The pairwise terms connect spatio-temporal neighbors and favor consistent labelings between them based on a simple generalized Potts potential:

$$\psi_S(f_{d_j^t}, f_{d_k^{t+1}}) = \lambda_{\psi_S} \cdot \delta\left[f_{d_j^t} - f_{d_k^{t+1}}\right]. \tag{6.15}$$

LABEL COST. The main purpose of the label cost is to include a parsimony prior to keep the number of selected trajectories low. Here, we exploit its ability to also assess the goodness of each individual trajectory. To that end, the cost of trajectory i being part of the solution is defined as

$$h_f(\mathcal{T}_i) = h_{ang}(\mathcal{T}_i) + h_{lin}(\mathcal{T}_i) + h_{occ}(\mathcal{T}_i) + h_{per}(\mathcal{T}_i), \tag{6.16}$$

which is a combination of terms assessing the angular and linear velocities h_{ang} and h_{lin}, a high-order data fidelity h_{occ} to handle occlusion gaps and weed out false positives, as well as a persistence term h_{per} to avoid interrupted trajectories. The individual components are derived from a statistical analysis of annotated data and will be defined later in Section 6.4.

The full label cost is computed as the sum over all individual label costs of active labels:

$$h_{\mathbf{f}}(\mathcal{T}) = \sum_{\substack{i=1 \\ \exists d: f_d = i}}^{N} \left(\lambda_{reg} + h_{\mathbf{f}}(\mathcal{T}_i)\right), \tag{6.17}$$

where λ_{reg} is a constant penalty term that is added uniformly to all active trajectories and acts as a regularizer.

6.3.1 *Optimization*

While optimization with label costs is challenging due to the fact that they are global terms, it can be approached using the integrated energy minimization framework of Isack and Boykov (2012); Delong et al. (2012). To that end, we alternate between minimizing Eq. (6.11) w.r.t. $\mathbf{f}$ and $\mathcal{T}$. Data association, *i.e.* minimization w.r.t. $\mathbf{f}$, thereby benefits from a seamless integration of the label costs into the well studied α-expansion framework with graph cuts, because the energy function remains submodular. This not only leads to strong local optima in practice, but also guarantees a bounded optimality gap (see Delong et al. (2012) and Section 6.2.2 for details regarding the theoretical properties). Trajectory estimation, *i.e.* minimization w.r.t. $\mathcal{T}$, is somewhat more challenging because the label cost is difficult to optimize w.r.t. the trajectories $\mathcal{T}_i$. To cope with this, we temporarily disregard the label cost, perform least squares minimization of the remaining terms for each individual $\mathcal{T}_i$ and verify that this actually reduces the overall energy, including the label cost. If the overall energy with label cost is not reduced, the previous trajectory is retained. The energy from Eq. (6.11) can thus only decrease or stay the same in every iteration.

Note that the idea of an instance cost was introduced earlier by Hoiem et al. (2007).

The motivation is the following: on one hand, the simplified minimization is convex and can be carried out efficiently in closed form, yet is guaranteed to never increase the energy. On the other hand, the simplification should have only a small effect in the context of the complete optimization scheme: near good minima of the energy the gradient of $h_{\mathbf{f}}(\mathcal{T})$ will be small, because the solution already obeys the physical constraints of Sec. 6.2.3; far from the minima, a large $\frac{\partial}{\partial \mathcal{T}} h_{\mathbf{f}}(\mathcal{T})$ would mean that a different path of the trajectories would be physically a lot more plausible while still staying close to the evidence, in which case it is likely to be picked up by the hypothesis

expansion (see below). We thus prefer to defer the difficult aspects of the energy to subsequent iterations of the discrete optimization.

GENERATING INITIAL TRAJECTORY HYPOTHESES. The optimization is bootstrapped with an initial set of trajectory hypotheses obtained in two ways:

1. We use a variant of RANSAC to fit trajectories to small randomly chosen subsets of detections (two in our case). To maximize the number of useful trajectory hypotheses, the random sampler prefers detections that are close in space and time. More specifically, two randomly chosen candidate detections d_i^s and d_j^t are discarded if a linear interpolation between them would result in a target velocity greater than $s = 35$cm per frame. Otherwise, the acceptance probability is $\exp(-\max(0, |s-t|-4))$. In other words, if the temporal gap between the two candidate detections is four frames or less, and if the linear interpolation results in physically plausible velocity, a new trajectory hypothesis is generated through linear interpolation. If the temporal gap is larger, the acceptance probability is decreased.

2. Additionally, we generate candidate trajectories using two further tracking methods. We employ an extended Kalman filter (EKF) initialized at all detections and using a variety of parameters. This set of initial hypotheses corresponds to the one we used as starting values for the continuous optimization in Chapter 5 (*cf*. Section 5.3.2). Moreover, we use the output of a different multi-object tracker based on dynamic programming (Pirsiavash et al., 2011).

It is a common observation that α-expansion is largely independent of the initialization, unless the unaries are very weak.

Although different sets of initial trajectory hypotheses may in general lead to slightly different results, we found that the variations of the final solution are marginal.

EXPANDING THE HYPOTHESIS SPACE. Depending on the initial number of trajectories, a hypothesis space with a fixed number of candidates may be too restrictive to obtain a strong minimum of the energy. To give the optimization more flexibility, we therefore expand the search space after each iteration, based on the current solution. Note that additional hypotheses do not change the nature of the energy; solutions in the expanded space can only have equal or lower energy.

New hypotheses are generated in a variety of ways:

- new trajectories are randomly fitted to all detections, as well as specifically to those labeled as outliers using the same strategy as above;

- existing trajectories are expanded in time or split in regions with no detections;

- pairs of existing trajectories are merged into new ones as long as their combination results in a physically plausible motion;
- splines with a higher number of control points are added on top of currently active ones.

Note that in all cases existing trajectories are retained to ensure that the energy does not increase. To nonetheless keep the number of possible trajectories from growing arbitrarily, all hypotheses that remained disabled during the past few iterations or those that have a higher label cost than the current value of the energy are removed from the hypothesis space, which guarantees that active hypotheses are never removed.

IMPLEMENTATION DETAILS. The model parameters are found by an iterative random search. Starting from a set $\mathbf{p}$ of default parameter values (see Table 6.3 *(basic)*), several optimization runs are performed. For each trial, a new set $\hat{\mathbf{p}}$ is generated by sampling either uniformly from $[\mathbf{0}, \mathbf{p}]$ or according to a normal distribution $\mathcal{N}_{\mathbf{p};\mathbf{p}/10}$. The parameter vector that yields the best solution (measured by MOTA) is then taken as the new mean and the procedure is iterated until convergence. This strategy is advantageous in practice because it samples the parameter space more efficiently than grid search, but is still largely unsupervised unlike manual search. The interested reader is referred to (Bergstra and Bengio, 2012) for a thorough discussion.

To reduce the effect of random sampling, we run the optimization with two different random seeds and pick the result with the lowest energy. Our current MATLAB code takes ~ 0.5s per frame to converge (excluding the object detector). With an optimized implementation real-time performance is within reach.

6.3.2 *Experiments*

We show the applicability of the discrete-continuous energy from Eq. (6.11) to multi-target tracking on three video clips: *S2L1* and S3MF1 from PETS as well as the *TUD-Stadtmitte* sequence. In Table 6.2, we compare the performance to our purely continuous energy from Chapter 5, Eq. (5.1). In both cases, the parameters were determined by a random search (see above). Although the accuracy (MOTA) is slightly better in the discrete-continuous case, the precision (MOTP) decreases a little, probably because the splines are less flexible in representing the exact trajectory shape as opposed to the non-parametric trajectories in Chapter 5.

The comparison at this point serves mainly to demonstrate that the discrete-continuous formulation works reasonably well and in fact outperforms the purely continuous approach with greedy jumps on the chosen dataset in terms of accuracy. However, the model pre-

Method	MOTA	MOTP	MT	ML	FM	ID	Rcll	Prcn
Cont. easy	78.4%	76.1%	10	0	3	2	84.6%	93.1%
DCO easy	80.9%	74.1%	10	0	3	8	87.3%	93.6%

Table 6.2: Average quantitative results of the continuous optimization *(Cont.)* and the discrete-continuous energy formulation *(DCO)* on three simple sequences.

sented in this section is not powerful enough to deal with more challenging situations where targets move in close proximity over longer time spans. To remedy this shortcoming we present a more sophisticated solution in Section 6.5. But first, let us first turn to a slightly different issue. In the following section we will present a method for choosing a suitable representation for ϕ and $h_f(\mathcal{T})$ based on real-world data statistics.

6.4 STATISTICAL DATA ANALYSIS

Energy minimization offers a flexible framework for modeling in vision, and CRF energies additionally give insight into the dependency structure. But aside from the structure, the potentials also need to be specified appropriately. In many cases the potentials (or energy components) are handcrafted, guided by intuition or mathematical convenience. Arguably, it is beneficial to instead derive their functional form from the statistics of the modeled quantities.

We use all S1 *and* S2 *sequences from Table 3.2.*

Here, we systematically analyze the distribution of various trajectory properties based on eight video sequences (PETS (Ferryman and Shahrokni, 2009) and TUD-Stadtmitte (Andriluka et al., 2010)) with ground truth annotations. It is clear that this comparably small amount of data does not cover all possible tracking scenarios. Rather the goal here is to allow adapting the tracker to a specific application scenario at hand. With the proposed methodology, other researchers or practitioners can easily adjust the approach to their specific application case.

To construct more realistic energies we analyze the empirical frequencies of the trajectory properties that we model in our CRF. Note that due to the limited amount of available ground truth data for multi-target tracking, full CRF learning is not the goal here. Instead we derive a suitable functional form of the potentials. To that end we study the negative logarithm of the empirical histograms of each property, following the definition of the Boltzmann distribution. Our analysis is carried out for the following properties:

LOCALIZATION ACCURACY OF THE DETECTOR. While it is safe to assume that an object detector will not always localize objects perfectly, the question remains what pattern the deviations follow. Fig-

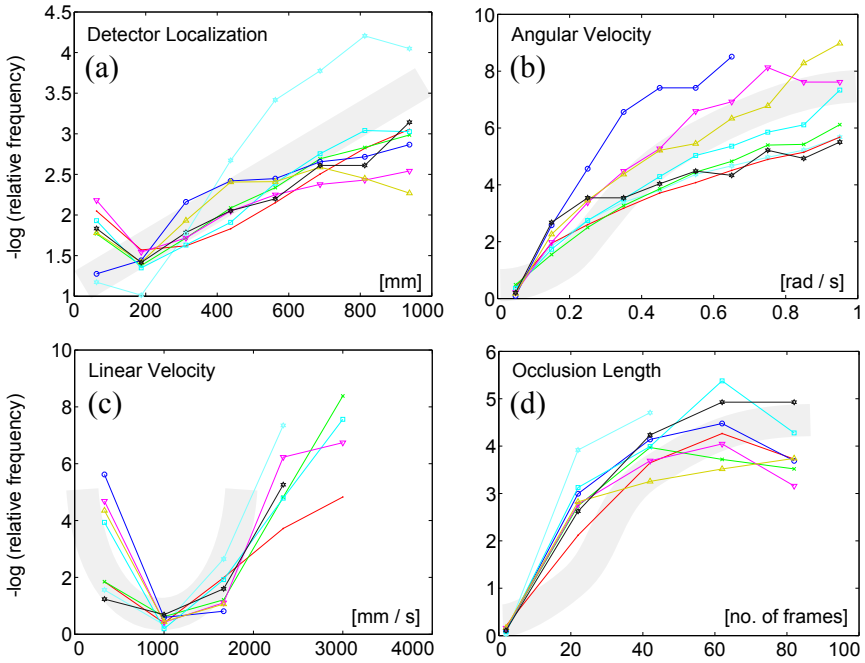


Figure 6.5: Empirical analysis of various trajectory properties in multiple people tracking, using ground truth data. Thick gray curves denote our suggested models, motivated by their empirical distributions (negative log-frequency shown).

ure 6.5(a) shows the (negative logarithm of the) empirical distribution of distances between the detector output and the closest ground truth object on the ground plane. To robustify the estimate, only nearest neighbors within 1m are considered. We observe that the energy grows linearly with the distance, suggesting a linear penalty for the data term (respectively an exponential distribution on the distance)

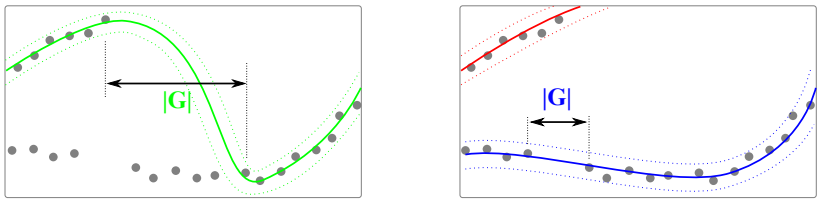
$$\Lambda(\mathbf{D}_i^t, \mathbf{X}_j^t) = \lambda_\phi \cdot \left\| \mathbf{D}_i^t - \mathbf{X}_j^t \right\| . \tag{6.18}$$

ANGULAR DYNAMICS. Real objects can only move within physical limits. Here we examine the angular velocity of people from their trajectory. Let $x = x(t)$ and $y = y(t)$ be the coordinates of a parametric planar curve and $\dot{x}, \dot{y}$ and $\ddot{x}, \ddot{y}$ its first and second temporal derivatives, respectively. The angular velocity at time t is then given as

$$\dot{\theta}(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{\dot{x}(t)^2 + \dot{y}(t)^2}. \tag{6.19}$$

Note that the definition only applies to regular curves, *i.e.* for

$$\dot{x}(t)^2 + \dot{y}(t)^2 \neq 0 \quad \forall t.$$



(a) (b)

Figure 6.6: The high-order data fidelity h_{occ} addresses the problem of long time spans during which a trajectory has no nearby detections *(a)*. The blue trajectory in *(b)* has a much lower label cost.

This is not a major limitation, since a realistic trajectory will usually have a positive velocity. The distribution of $\dot{\theta}$ in Figure 6.5(b) suggests representing the penalty with a Cauchy-Lorentz distribution:

$$h_{ang}(\mathcal{T}_i) = \lambda_{ang} \sum_t \log\left(1 + \dot{\theta}(t)^2\right) \tag{6.20}$$

LINEAR DYNAMICS. In addition to the angular velocity we also examine the linear velocity. Figure 6.5(c) shows that people mostly move at a speed of about one meter per second. Deviations from that speed are rare such that a quadratic penalty is appropriate:

$$h_{lin}(\mathcal{T}_i) = \lambda_{lin} \sum_t \left(\sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} - 1000\right)^2 . \tag{6.21}$$

OCCLUSION LENGTH. In many applications, *e.g.* robotics and some surveillance scenarios, targets are observed from a relatively low camera viewpoint. Hence they are periodically occluded, causing the detector (or any other observation model) to fail temporarily. A tracker should nevertheless be able to bridge such short occlusion gaps without spawning false new trajectories. To determine the expected length of such occlusions, we analyze the frequencies of different durations of occlusion (in frames) as shown in Figure 6.5(d). Although most occlusions last less than 20 frames, longer ones do occur. We therefore model the penalty for trajectories that are not supported by detections through multiple consecutive frames as a Cauchy-Lorentz distribution:

$$h_{occ}(\mathcal{T}_i) = \lambda_{occ} \sum_{j \in \text{gaps}(\mathcal{T}_i)} \log\left(1 + |G|_j^2\right), \tag{6.22}$$

Here, $|G|_j$ is the number of frames in which trajectory i has no detections close by (*cf*. Figure 6.6).

PERSISTENCE AND LENGTH. Assuming that the scene does not contain doors or other openings where objects might disappear, a trajectory will always start and terminate close to the border of the

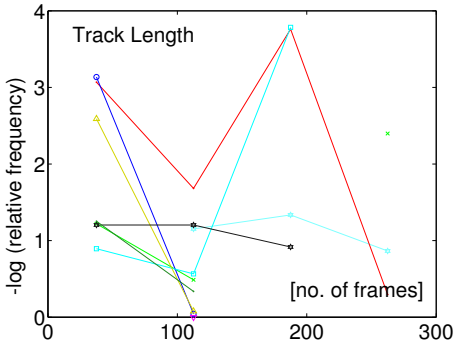


Figure 6.7: Trajectory length does not resemble any particular distribution.

image (or the tracking area). An extensive data analysis of this property is thus not necessary. To prevent fragmented trajectories and allow a buffer entry zone τ we impose a soft threshold

$$h_{\text{per}}(\mathcal{T}_i) = \lambda_{\text{per}} \cdot \min\Big(\tau, \text{dist}(\mathcal{T}_i^{t^*}, \text{border})\Big), \tag{6.23}$$

where $t^* \in \{s_i, e_i\}$ stands for birth or death time of a trajectory.

The temporal length of trajectories varies significantly across sequences and does not exhibit a consistent behavior as can be seen in Figure 6.7. We therefore do not make any assumptions about it.

6.5 MODELING MUTUAL EXCLUSION

When tracking multiple targets in crowded scenarios, modeling mutual exclusion between distinct targets becomes important at two levels:

1. in data association, each target observation should support at most one trajectory and each trajectory should be assigned at most one observation per frame;

2. in trajectory estimation, two trajectories should remain spatially separated at all times to avoid collisions.

Yet, the formulation in Section 6.3 sidesteps these important constraints to enable efficient optimization.

In this section, we address them using a similar mixed discrete-continuous conditional random field (CRF) as before, but explicitly model both types of constraints: Exclusion between conflicting observations with supermodular pairwise terms, and exclusion between trajectories by generalizing global label costs to suppress the co-occurrence of incompatible labels (trajectories). Mutual exclusion is thus addressed both at the data-association and at the trajectory level. Typical failure cases of the energy in Eq. (6.11) and the solution proposed in the following sections are illustrated in Figure 6.8.

(a) (b)

Figure 6.8: Typical failure cases *(top)* are addressed with the proposed discrete-continuous CRF *(bottom)*: Detections are forced to take on different labels *(a)* and physically overlapping trajectories are suppressed even if they do not share detections *(b)*.

6.5.1 *Detection-level exclusion*

We first describe how we integrate mutual exclusion at the detection level. Assuming a target size s, it is impossible that two detections originating from the same frame and being at least the distance $s/2$ apart are caused by the same object. Therefore, following the same notation as before, we introduce an exclusion term

$$\psi_X(f_d, f_{d'}) = \begin{cases} \overline{\psi}_X, & f_d = f_{d'} \\ 0, & \text{otherwise} \end{cases} \tag{6.24}$$

to all edges between simultaneous detector responses

$$(d, d') \in \mathbf{E}_{\mathcal{X}} = \left\{ \left(d_i^t, d_j^t\right) \middle| i \neq j, \left\|\mathbf{D}_i^t - \mathbf{D}_j^t\right\| > \frac{s}{2} \right\}. \tag{6.25}$$

The penalty $\overline{\psi}_X$ is thus incurred if two distant detections are assigned the same trajectory label. For detections that are very close to one another, on the other hand, it is reasonable to accept multiple assignments, since state-of-the-art object detectors sometimes erroneously produce multiple outputs from the same object. This can occur even after non-maxima suppression. The exclusion factors are illustrated in Figure 6.9(b).

Note that only considering exclusion at the detection level is not enough in order to prevent collisions between targets. In fact the optimization may otherwise be forced to pick two almost identical trajectories in order to satisfy these inter-object constraints. It is thus crucial not to disregard the path of the actual trajectories.

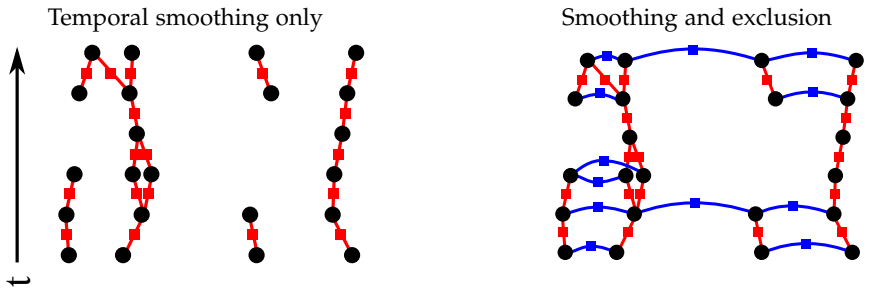


Figure 6.9: Factor graph of the underlying CRF with black circular nodes representing the random variables (trajectory hypothesis for each detection) and square nodes representing the pairwise potentials. For clarity, all unary and high-order potentials are omitted. In addition to simple temporal smoothing factors (shown in red on the left-hand side), we model pairwise exclusion between detections within the same time step (blue, subset shown) to prevent implausible data association *(right)*.

6.5.2 *Trajectory-level exclusion*

Let us now turn to the more challenging task of enforcing exclusion at the level of continuous trajectories. It is obvious that multi-target tracking should take care to prevent situations where two or more targets occupy the same physical space at the same time. Unfortunately, such constraints lead to hard optimization problems. We will describe our algorithm later in Section 6.5.4. Let us first define the pairwise label cost and its application to multi-target tracking.

During the discrete optimization step (see Eq. (6.6)) the trajectories remain fixed. To avoid collisions it is therefore necessary to select only those targets (or labels) with no significant spatio-temporal overlap. To that end, we introduce a *pairwise label cost*. Its general purpose is to impose a penalty on the energy if there exist two labels that are unlikely to appear simultaneously. In the present case of multi-target tracking such unlikely events occur when two trajectories come too close to each other, causing physically implausible situations. It is therefore reasonable to apply a suitable penalty ζ if and only if two mutually exclusive trajectories are active:

$$h_f^X(\mathcal{T}_i, \mathcal{T}_j) = \begin{cases} \zeta(\mathcal{T}_i, \mathcal{T}_j), & \exists d, d' : f_d = i \wedge f_{d'} = j \\ 0, & \text{otherwise.} \end{cases} \tag{6.26}$$

In our case, the co-occurrence penalty is proportional to the spatio-temporal overlap between two trajectories:

$$\zeta(\mathcal{T}_i, \mathcal{T}_j) = \sum_{t \in \mathbf{O}(\mathcal{T}_i, \mathcal{T}_j)} \zeta_{i,j}^t, \tag{6.27}$$

which is computed by summing the mutual overlap over all frames during the common lifespan $\mathbf{O}$ of the trajectories. The overlap is

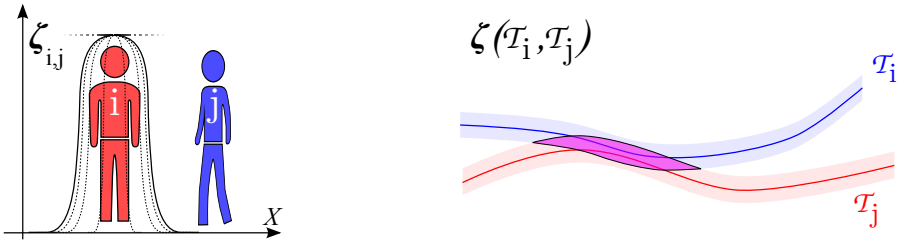


Figure 6.10: The distance between two targets is modeled by an isotropic sigmoid *(left)*, while the spatio-temporal overlap between two trajectories is computed by accumulating the overlap at each time step.

approximated with an isotropic sigmoidal function around the center of the target (*cf*. Figure 6.10 *(left)*):

$$\zeta_{i,j}^{t} = \lambda_X \cdot \left(1 - \frac{1}{\exp(-o_a \|\mathbf{X}_i^t - \mathbf{X}_j^t\| + o_b)}\right). \tag{6.28}$$

The two parameters o_a and o_b control the size and the falloff of the sigmoid and are directly related to the application-specific shape of the targets. For our experiments we set $o_a = 0.05$ and $o_b = s \cdot o_a/2$, where s is the target size as in Chapter 5 (Eq. (5.6)). The spatio-temporal overlap is illustrated in Figure 6.10.

It is important to note that this formulation of a co-occurrence label cost is general and not restricted to multi-target tracking. It can trivially be transferred to other applications that involve multi-model fitting, such as semantic segmentation or motion estimation. Note that Ladicky et al. (2010), for example, use a co-occurrence cost to prevent unlikely labeling configurations in the context of semantic segmentation. There, however, the cost is overestimated to keep inference tractable. We prefer to model the cost exactly, but can no longer guarantee global optimality of each expansion step.

6.5.3 *Advanced discrete-continuous energy*

It remains to define the complete CRF energy with mutual target exclusion:

$$\begin{aligned} E(\mathbf{f}, \mathcal{T}) = & \sum_{d} \phi(f_d, \mathcal{T}) + && \text{(unaries)} \\ & \sum_{(d,d')\in\mathbf{E}_S} \psi_S(f_d, f_{d'}) + && \text{(temporal smoothness)} \\ & \sum_{(d,d')\in\mathbf{E}_X} \psi_X(f_d, f_{d'}) + && \text{(detection-level exclusion)} \\ & \sum_{i} h_{\mathbf{f}}(\mathcal{T}_i) + && \text{(single label cost)} \\ & \sum_{i,j\neq i} h^X_{\mathbf{f}}(\mathcal{T}_i, \mathcal{T}_j), && \text{(pairwise label cost)} \end{aligned} \tag{6.29}$$

with the following components:

UNARY TERMS. The unaries ϕ measure how well the trajectories follow the detector evidence. Here, we use the distribution derived from the statistical analysis suggesting a linear penalty term (see also Eq. (6.18)):

$$\phi_{d_j^t}(l, \mathcal{T}) = \omega_j^t \cdot \lambda_\phi \cdot \left\| \mathbf{D}_j^t - \mathbf{X}_l^t \right\| . \tag{6.30}$$

PAIRWISE TERMS. The first pairwise term ψ_S is the same as before (*cf*. Eq. (6.15)) and encourages temporally smooth data association with a standard generalized Potts model (see Figure 6.9(a)). The second pairwise term ψ_X models the detection-level exclusion constraints from Eq. (6.24).

LABEL COST. The first higher-order term (label cost) $h_{\mathbf{f}}(\mathcal{T})$ was defined in Eq. (6.17) and models the plausibility of each trajectory in terms of its dynamics, data fidelity and persistence. The second higher-order term $h^X_{\mathbf{f}}(\mathcal{T}_i, \mathcal{T}_j)$ is the pairwise co-occurrence label cost from Eq. (6.26) for trajectory-level exclusion.

6.5.4 *Optimization*

Like in Section 6.3 we again perform MAP estimation by alternatingly minimizing the energy of the discrete and the continuous variables. The minimization scheme is summarized in Algorithm 2. Let us now discuss each step in more detail.

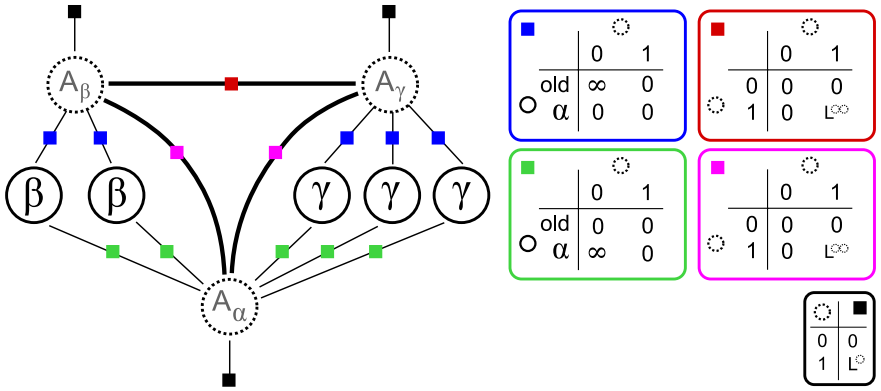


Figure 6.11: Factor graph encoding of the unary and pairwise label cost before expanding on α. Random variables and their current labels are represented by solid circles, while auxiliary variables are outlined with dashed circles. Solid squares represent unary (black) and pairwise (colored) terms, respectively. The corresponding potentials are depicted on the right with L° and L^∞ being the respective label cost for a single label and a pair of labels. Note that all factors that are unrelated to the label cost are omitted for clarity.

DISCRETE ENERGY MINIMIZATION. As before, we follow an expansion move-based strategy to minimize the discrete part of the energy (6.29). However, the current situation is more difficult because the proposed pairwise label cost introduces non-submodular components. Here, we develop a way to seamlessly integrate this co-occurrence potential between two labels into the CRF. To simplify the treatment, we will describe the potential in the context of our expansion move-based MAP estimation approach. In particular, we describe the corresponding factor graph for a single α-expansion step, where 0 corresponds to no label change and 1 means a variable is switched to label α. An illustration of the factor graph (without unary and pairwise terms) is depicted in Figure 6.11.

Let us first look at the standard per-label cost h_f that we used in Section 6.3. Similar to Delong et al. (2012), one auxiliary node for each existing label is added (dotted circle) and connected to each variable

Algorithmus 2 : Discrete-continuous energy minimization

input : Initial trajectory hypotheses, detections **D**
output : Labeling **f** for **D**, final trajectories $\mathcal{T}^*$
while $\neg$ *converged* **do**
  Obtain labeling **f** by minimizing Eq. (6.6)
  Refit trajectories $\mathcal{T}$ by minimizing Eq. (6.2)
  Modify hypothesis set
end
return **f**, $\mathcal{T}^*$

that carries the corresponding label. However, here we use a different encoding for the auxiliary variables, which act as indicator switches for each label: The auxiliary variable contributes the cost L° of having a certain label only once if it is switched on, otherwise its associated cost is 0 (black factors). An infinite pairwise cost prevents the indicator from being off when there is at least one node with the corresponding label (blue and green factors). While this encoding yields supermodular costs (the overall cost is already non-submodular due to Eq. (6.24)), its purpose will become apparent shortly.

In practice, a sufficiently large value is used instead to avoid instability.

We now turn to the pairwise label cost. Having the same graph structure as above, it is possible to insert a connecting factor between each pair of auxiliary variables (red and magenta). To increase the energy value in case when both labels exist simultaneously, a penalty L^∞ is applied if and only if both corresponding auxiliary variables are switched on.

Since the energy is non-submodular, we use sequential tree-reweighted message passing (TRW-S) (Kolmogorov, 2006) for each binary expansion step. As it is not guaranteed that each expansion step finds a global minimum of the binary sub-problem, we found it beneficial to add a greedy search step in each expansion move: for each label in turn we check whether the energy can be decreased further by entirely removing that label from the current solution (*i.e.* replacing the trajectory by the outlier model). The discrete optimization is implemented using OpenGM[1] (Andres et al., 2012). We have also experimented with other inference methods such as iterated conditional modes (ICM) or quadratic pseudo-boolean optimization (QPBO), however in our experience, TRW-S showed superior performance both in terms of computational time as well as the obtained solution.

It may seem unnatural to use message passing within α-expansion instead of an st-cut, since message passing algorithms are generally capable of performing inference in multi-label problems. The motivation is that directly running message passing on the multi-label problem is prohibitively slow even for very small graphs due to the global factor in the energy. The factor graph for each expansion move on the other hand is much smaller.

Note that during inference both higher-order terms are transformed in each α-expansion step to pairwise ones using auxiliary variables, as outlined in Figure 6.11. Also note that the energy can only be minimized approximately: finding a global optimum of an energy of the general form from Eq. (6.29) in polynomial time is only possible for binary energies with $|\mathbf{L}| \leqslant 2$, and only if the discrete part of the energy is submodular and the continuous part is convex.

CONTINUOUS ENERGY MINIMIZATION. The continuous part of the energy function (6.29) is not convex and cannot be minimized in

1 http://hci.iwr.uni-heidelberg.de/opengm2

closed form. We therefore perform a simplex-based search (Nelder and Mead, 1965) over the continuous parameters of all active trajectories $\mathcal{T}^*$, starting from a least squares approximation of the objective to find a better minimum. In practice, we employ MATLAB's implementation `fminsearch` of the Nelder-Mead simplex algorithm (Lagarias et al., 1998).

We minimize a simplified energy including only the unary terms ϕ and the continuous label costs h_{ang} and h_{lin}. The solution of this simplified energy minimization step is discarded if it does not decrease the full CRF energy from Eq. (6.29). Since each iteration the hypothesis space $\mathcal{T}$ is updated, the optimization is nevertheless able to escape poor local minima.

PRUNING. To speed up inference, we prune the graph in two different ways. We reduce the connectivity by disregarding neighbors from $\mathbf{E}_{\mathcal{X}}$ that lie too far apart. In our experiments, we prune all edges between detections that are more than 2 meters away from each other. This does not change the CRF energy in the relevant portion of the solution space (*i.e.* near a sensible minimum), because the data term already ensures that such detections will never be assigned the same label. Moreover, the label space of each random variable is reduced to only those (few) trajectory hypotheses that lie within reasonable reach of a detection (in our case 1 meter). Again, this will not change the energy of any remotely plausible solution, for the same reason as above.

PARAMETERS. Our advanced model has ten parameters that can be tuned individually, eight for the basic energy and two to control the weight of detection- and trajectory-level exclusion, respectively. To find a good set we follow the same random search strategy as discussed in Section 6.3.1. A set of values for each variant that was used throughout the experiments in the following section is listed in Table 6.3. The values have been rounded for better readability.

It is worth pointing out that the weight λ_{ψ_S}, governing the spatio-temporal smoothness between detections in adjacent frames, is set rather low in the case of the full model. This is not surprising because situations with overlapping trajectories, such as in the example illustrated in Figure 6.8 *(top right)*, are resolved by explicitly handling exclusion at the level of trajectories. Note that tracking is performed in world coordinates with millimeters as units of length. Therefore, the values for λ_{lin} controlling the weight for linear velocity are naturally rather small.

SLIDING WINDOW. Even though the presented method can be applied to entire sequences, we found it beneficial, both in terms of speed and in terms of accuracy, to perform the optimization on smaller

Parameter	Description	basic	FULL
λ_{ϕ}	unary weight ϕ	201.24	125.96
λ_{ψ_S}	smoothness weight ψ_S	1.89	0.08
$\Gamma_{\varnothing}$	outlier cost	334.90	111.63
λ_{lin}	weight h_{lin}	5.9e-05	1.7e-05
λ_{ang}	weight h_{ang}	0.07	1.11
λ_{occ}	weight h_{occ}	0.74	0.38
λ_{per}	weight h_{per}	0.13	0.14
λ_{reg}	weight h_{reg}	0.48	1.11
$\overline{\psi}_X$	det. exclusion penalty	0.00	27.02
λ_X	traj. exclusion penalty	0.00	19.62
τ	entry buffer / threshold for $\mathbf{E}_S$	20 [cm]	
s	target's size	35 [cm]	

Table 6.3: Typical parameter settings for running the discrete-continuous energy-based multi-target tracking. The table shows parameters for both the basic submodular energy (Section 6.3, Eq. (6.11)) and the advanced energy with exclusion modeling (Section 6.5, Eq. (6.29)).

temporal windows. The length of each window is set to 50 frames and successive windows overlap by 15 frames. To ensure seamless correspondence between adjacent windows, all trajectories with a significant spatio-temporal overlap within the overlapping time interval are merged.

6.6 EXPERIMENTS

We evaluate our tracker on eight video sequences. Besides the widely used *PETS S2.L1* sequence, we also include four more challenging scenarios from the PETS dataset as well as the *TUD-Stadtmitte* sequence. Finally, we also test our method on the sequences *Bahnhof* and *Sunny Day* from the ETH Mobile Scene *(ETHMS)* dataset. Note that we do not use the available camera calibration and depth maps for these sequences, but rather track the pedestrians in image space. For further details on the chosen datasets please refer to Section 3.3.

As usual, we report the widely accepted CLEAR MOT metrics evaluated in 3D with a 1m hit/miss threshold. To better assess the quality we additionally report the numbers of mostly tracked *(MT)* and mostly lost *(ML)* trajectories, along with the numbers of track fragmentations *(FM)* and identity switches *(ID)*. The exact definitions of these metrics are thoroughly discussed in Section 3.4. Table 6.7 also shows the false alarm rate per frame *(FAF)*. All figures in this table

Method	MOTA	MOTP	MT	ML	FM	ID	Rcll	Prcn
Cont.	54.9%	66.7%	13	14	22	29	59.1%	95.5%
DCO (no exc.)	53.6%	63.8%	13	11	33	43	61.3%	90.9%
DCO (FULL)	57.2%	65.1%	14	13	24	30	63.3%	92.9%

Table 6.4: Average quantitative results on six datasets of the continuous energy *(Cont.)* and of our discrete-continuous optimization without *(no exc.)* and with proper exclusion modeling *(FULL)*.

are produced with a 2D evaluation protocol using a publicly available implementation[2]. The evaluation is furthermore based on the detector output and the ground truth of Yang and Nevatia (2012a). As we will see in Section 7.1 this is essential for a fair comparison.

6.6.1 *Comparison to the continuous energy*

Before discussing the contributions regarding the discrete-continuous formulation, let us first compare its performance with respect to the continuous energy from Chapter 5. To enable a fair comparison, we restrict the state space to the same tracking area as was used in the previous chapter by discarding all detections that lie outside that area. The parameters of both methods were determined in a similar manner by performing a random search. Since our discrete-continuous approach does not model appearance nor explicitly handles occlusions, we compare it to the basic continuous formulation *(no OM)*.

See Section 6.6.5 for a discussion.

Quantitative results, averaged over six sequences are presented in Table 6.4. We can see that without taking mutual target exclusion into account *(no exc.)*, the discrete-continuous optimization cannot quite reach the performance of the continuous energy minimization. Low precision as well as the high number of identity switches are the result of overlapping and poorly localized trajectories (*cf*. Figure 6.8 *(top)*). However, the advanced discrete-continuous optimization *(FULL)* with explicit exclusion modeling is able to outperform the continuous formulation on average in terms of accuracy (MOTA). Although the spline models in the discrete-continuous case are able to achieve higher recall by bridging longer occlusions, they also produce more spurious trajectories leading to a lower precision value, as we will discuss in Section 6.6.5. Multiple Object Tracking Precision (MOTP) decreases by almost 2 percentage points, perhaps because the splines with only few control points are less flexible than a per-frame representation. A lower MOTP score may also be caused by the higher number false positive trajectories when they drift away from the target leading to a larger misalignment error.

2 http://iris.usc.edu/people/yangbo

Figure 6.12: Exemplar frames of multi-target tracking by discrete-continuous energy minimization.

6.6.2 *Qualitative results*

Example frames from three sequences overlaid with the output of our discrete-continuous multi-target tracker are shown in Figure 6.12. Most targets are successfully recovered, even in such crowded scenarios. However, false positives, such as the yellow box on the sign in the top row (middle), remain a frequent source of errors. Interestingly, the proposed approach also recovers true targets that are counted as false positives. In particular, the two bottom frames of the Sunny day sequence contain a mannequin in the shop window on the right (no. 32) and a man inside a phone booth on the left (no. 30) that are both missing in the ground truth. Identity switches and target losses cannot be completely avoided either. Note how track number 5 in the right column switches from one target to the one right beside it and later disappears entirely due to missing detections.

More issues on ground truth and evaluation will be discussed in Sec. 7.1.

Method	MOTA	MOTP	MT	ML	FM	ID	Rcll	Prcn
basic	41.6%	61.1%	12	10	53	82	62.7	74.2
det. exclusion	46.7%	63.0%	11	12	38	48	58.4	86.5
traj. exclusion	46.6%	62.7%	10	12	49	69	57.8	86.3
FULL	51.5%	64.4%	11	13	43	54	57.0	93.7

Table 6.5: Cross-validation results on six sequences.

Sequence	MOTA	MOTP	MT	ML	FM	ID
S2.L1	90.5 %	76.3 %	17	0	32	43
S2.L2	46.3 %	60.3 %	10	15	114	132
S2.L3	37.3 %	65.6 %	9	23	18	22
S1.L1-2	57.1 %	66.4 %	17	13	19	21
S1.L2-1	26.2 %	58.1 %	7	25	18	20
Stadtmitte	55.4 %	64.2 %	4	0	2	2

Table 6.6: Results of our full method on each test sequence.

6.6.3 *Comparison to the basic energy*

We systematically compare the individual contributions presented in this section against the basic discrete-continuous energy from Section 6.3. To make this comparison as fair as possible we use the same ground truth data and detector evidence throughout our experimentation. Moreover, we determine all required parameters by a random search over the parameter space via leave-one-out cross validation (*cf*. Section 6.5.4 for a discussion on random search).

Table 6.5 shows the cross-validation results averaged over all test sequences. The top row *(basic)* is our proposed discrete-continuous method without proper inter-object exclusion modeling, *i.e.* both $\overline{\psi}_X$ and λ_X are set to 0. The next two lines present two intermediate results: only adding the detection-level exclusion factors, *i.e.* $\lambda_X = 0$ (*det. exclusion*), and only adding the co-occurrence label cost, *i.e.* $\overline{\psi}_X = 0$ (*traj. exclusion*). Finally, the last row shows the average cross-validation results of our full model *(FULL)* from Eq. (6.29).

We observe that modeling exclusion on either level boosts the tracker performance, but it is crucial to handle mutual exclusion on the detection and trajectory level simultaneously to achieve best possible results. MOTA rises by ten percentage points while the number of identity switches is almost halved. To ease comparison with other approaches, we also give per-sequence results (Table 6.6) using a single set of parameters.

Method	Rcll	Prcn	GT	MT	PT	ML	Frag	ID
Our method	77.3%	87.2%	124	66.4%	25.4%	8.2%	69	57
DP	67.4%	91.4%	124	50.2%	39.9%	9.9%	143	4
PIRMPT	76.8%	86.6%	125	58.4%	33.6%	8.0%	23	11
Online CRF	79.0%	90.4%	125	68.0%	24.8%	7.2%	19	11

Table 6.7: Quantitative comparison to three state-of-the-art methods on the *ETHMS* dataset: The dynamic programming (DP) approach of Pirsiavash et al. (2011), Person Identity Recognition based Multi-Person Tracking (PIRMPT) of (Kuo and Nevatia, 2011) and a tracklet-based CRF tracker (Yang and Nevatia, 2012a).

6.6.4 *Further quantitative results*

We also evaluate our method on video sequences *Bahnhof* and *Sunny day* from the *ETHMS* dataset (Ess et al., 2008). Both sequences are filmed from a moving platform in a busy urban environment. Note that the dataset was captured by a stereo camera and provides an image pair for each frame. However, we do not rely on additional depth information and thus only use the left images throughout the experiments. Also note that we perform tracking directly in image space since the available camera calibration is rather unreliable for this dataset.

We use the detector output from Kuo and Nevatia (2011); Yang and Nevatia (2012a) and run their publicly available evaluation script to produce the results summarized in Table 6.7. State-of-the-art methods for these sequences heavily rely on tracklet linking through significant periods of occlusion, based on appearance, scale and other cues. Although it is conceivable to include occlusion reasoning and an appearance model in our CRF formulation, such steps lie beyond the scope of this dissertation and are left for future work. We therefore postprocess our tracker output with a simple extrapolation-based track linking scheme to explore the capabilities of our method when combined with such track linking. More precisely, we compute a similarity matrix combining spatio-temporal distance, scale and appearance to compute a score for each pair of tracks. All trajectory pairs with a similarity score above a certain threshold are then merged in a greedy fashion. Both the threshold and the weighting parameters for the individual cues are determined via cross-validation.

While our simplistic linking scheme leads to comparatively many ID switches, the high recall and precision numbers indicate that our discrete-continuous CRF yields a competitive basis for appearance-based occlusion handling.

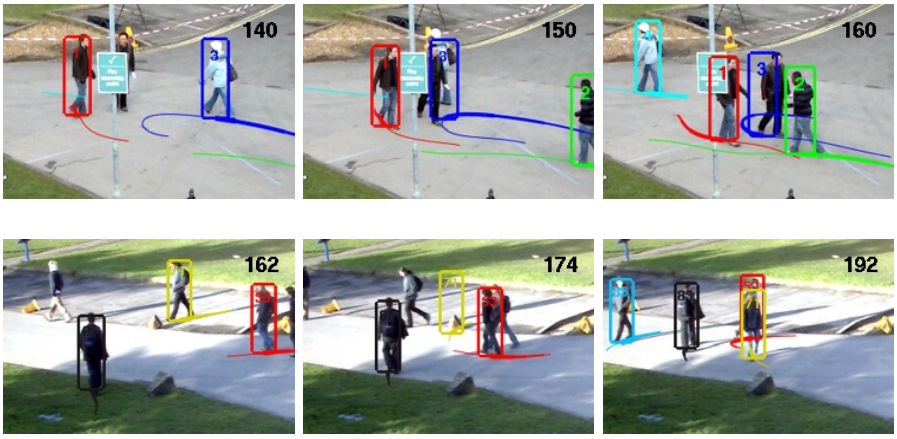


Figure 6.13: Limitations of the current approach. Occlusion causes a trajectory to switch from one target another in both cases. The frame number for each image is shown in the upper right corner.

6.6.5 *Limitations*

Although the proposed method shows encouraging results, an average accuracy of about 50% (*cf*. Table 6.5) is an indicator that there is still room for improvement. We would like to point out two examples for typical failure cases, which are illustrated in Figure 6.13. The top row shows three images being 10 frames apart, where the person in light blue hides behind a scene occluder (the lamp post in the foreground), while a different person is simultaneously revealed in the same region. The tracker confuses the two as being the same person and reconstructs one single trajectory (blue) causing two track fragmentations and two identity switches. Note that, despite having distinct colors, a naive frame-wise appearance comparison would probably not suffice to resolve this situation due to occlusion. A long-range connection that spans over frames 140 and 160 may provide a more discriminative cue to prevent such failures.

A second example in the bottom row shows a somewhat related issue. Again, a target is occluded (by the person with the black bounding box), while another one reappears after occlusion (the couple on the right). However, in this case, a time span of about 20 frames lies between the two events. During that time, trajectory number 71 is able to 'survive' although no detections are present to support it. This situation is particularly challenging for any appearance model. First, both targets wear similar clothes, making it hard to distinguish one from the other. Second, targets appear rather small on the image. Due to low video resolution, these pedestrians are only about 50 pixels high, which complicates extracting enough valuable color or texture information. Finally, there is a large variation in lighting causing drastic changes in the targets' appearance. One possible way

to nevertheless prevent such tracking failures is to evaluate the detector score in all frames along the trajectory hypothesis to determine its data fidelity more accurately. This may help weeding out more false positive trajectories thereby improving the overall performance.

6.7 DISCUSSION

In this chapter we presented a global multi-target tracking approach that unifies data association and trajectory estimation in a consistent discrete-continuous energy. This formulation simultaneously addresses several drawbacks of previous methods. In contrast to the continuous energy approach from Chapter 5, the discrete problem of data association is handled explicitly within a graphical model framework, which enables leveraging powerful discrete optimization techniques. Moreover, as opposed to greedy local jump moves, the multi-labeling problem is able to make larger steps in search for a low-energy configuration. At the same time, the actual target trajectories are defined in their natural, continuous domain. This allows one to avoid discretization artifacts that arise when tracking is performed on a discrete grid as in Chapter 4. Moreover, trajectories directly approximate the true target motion such that smoothing in a separate post-processing step is not necessary (Zhang et al., 2008; Yan et al., 2012).

We presented two variants of the discrete-continuous energy. The first one includes a smoothness prior and employs a quadratic distance term such that both parts of the energy are easily optimizable. The complete energy is minimized iteratively by solving data association to (near) global optimality by α-expansion with label costs, and analytically fitting continuous trajectories to the assigned detections. The second variant handles inter-object exclusions on two levels:

- at the data association level with non-submodular constraints, such that each detection may only explain one target and vice versa;
- at the trajectory level, where a novel co-occurrence label cost penalizes solutions with overlapping or colliding trajectories. Note that the proposed pairwise label cost formulation is generic and therefore also applicable to other problems that involve model-fitting.

We suggested an expansion move-based optimization scheme to handle the non-submodular energy with global co-occurrence label costs.

Our experiments show state-of-the-art results on public benchmarks, with clear improvements from the simultaneous exclusion constraints. In both cases, a statistical data analysis was used to derive appropriate CRF potentials for the label cost. Future work may consider incorporating appearance cues into the CRF to better disambiguate targets

after long-term occlusions. A further consideration may be to introduce long-range connections to the factor graph to explicitly handle temporally distant detections. Another promising avenue to follow would be to refine the continuous optimization step to include more aspects of the complete energy.

7 FURTHER CONSIDERATIONS

We're not here to score benchmarks but to ask fundamental questions.

DAVID FORSYTH at *ECCV 2012*

CONTENTS

WE HAVE now seen three different energy-based approaches to multiple target tracking. All three present technical contributions to this field of research, which is supported by state-of-the-art results. Nonetheless, some of the more practical issues have so far been left untouched. In this chapter we will discuss several important facets of this work. The first one is pragmatic and deals with the problem of objectively evaluating and comparing different multi-target tracking methods. We will point out some of the ambiguities in evaluation protocols and also discuss some of the problems related to obtaining, or even defining ground truth. The second part discusses some pitfalls that arise with iterative optimization methods, such as the ones presented in this dissertation. Finally, the third part of this chapter addresses the ethical aspect of multi-target tracking. We will turn to the issue of privacy when tracking people and briefly discuss the benefits, but also the potential dangers that a fully developed technology may carry.

7.1 ON EVALUATION AND GROUND TRUTH

Quantitatively evaluating computer vision algorithms is not a straight forward task. The reasons for that are varied. On one hand it is not always obvious what the 'correct' solution should look like. Arguably, for some application this question is easier to answer than for others. In low-level tasks such as image restoration or deblurring, the

ultimate goal is usually to precisely reconstruct the original, unmodified image. Even in this seemingly clear case the ground truth might be either unavailable or contain some level of noise itself (Zoran and Weiss, 2009). For a more high-level problem like image classification it should be easy to answer the question of whether a certain object is present in the image or not. However, the answer becomes ambiguous if the object is only partially visible, either due to occlusion or due to cropping. Looking at tasks like segmentation, the situation becomes even worse. When five people are asked to draw the outline of the same object in the same image, one will probably get five different contours.

Let us look at multi-target tracking. Here, the ground truth is not always well-defined either. Although most human annotators would agree on the presence or absence of a person in a certain image region, pinpointing the precise location poses a more difficult task. As a matter of fact, we will see in the following section how large the spatial displacement between independent annotations can be.

The second challenge of evaluation is measuring the similarity between the obtained solution and the ground truth. To this end, several protocols and metrics that we discussed earlier in Section 3.4 have been proposed and have in fact become widely accepted. Nonetheless, their definition remains somewhat ambiguous and involves meta-parameters, such as the overlap threshold.

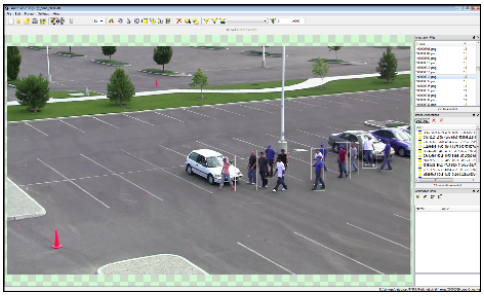
Another important issue specifically concerns tracking-by-detection methods. These methods heavily rely on the output of an object detector. As a consequence, a better detector will most likely yield better tracking results. Therefore it is essential that the same input, *i.e.* the same set of detections, is used if one is interested in only comparing the merits of different tracking algorithms themselves.

In summary, all the aspects mentioned above contribute to the challenge of obtaining an objective performance evaluation of a particular tracking method. We will now discuss these issues in more detail.

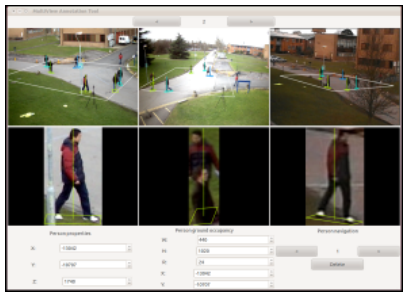
7.1.1 *Obtaining ground truth*

Annotating images is a tedious task. The most naive way, which was frequently followed in the past, is to draw rectangles around the objects of interest to define their bounding boxes frame by frame. There are, however, several software packages that assist the user in one way or another to facilitate the annotation process.

ANNOTATION TOOLS. The annotations for the *TUD-Stadtmitte* and all the PETS sequences that were used throughout this work were created using the AnnoTool by Oliver Schwahn. It allows one to linearly interpolate the location and the size of the bounding box between key frames, which leads to a significant speed up. The tracks were



AnnoTool



MultiView

VATIC

Figure 7.1: User interfaces of three different annotation tools (see text for details).

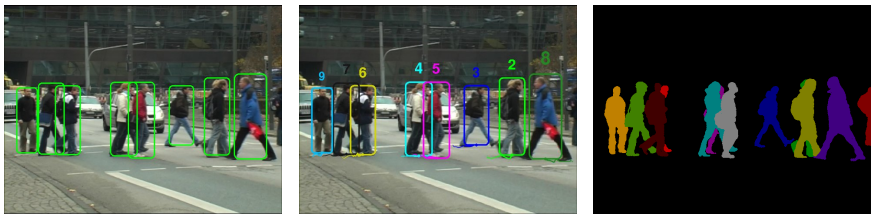
smoothed afterwards to reflect natural people motion. While key-frame interpolation facilitates the annotation process, one must bear in mind that it also leads to an approximation since the 'true' target motion hardly ever precisely follows a linear (or a higher order polynomial) pattern.

The Video Annotation Tool from Irvine, California (VATIC)[1] is a more recent annotation tool presented by Vondrick et al. (2013). It offers an integrated interface to Amazon's Mechanical Turk such that one can leverage the power of crowdsourcing for the annotation task.

Finally, Utasi and Benedek (2012) provide an annotation software specifically designed for a multi-camera setup[2]. Interestingly, they define a target by its actual height in world units and the rectangular area that it occupies on the ground plane instead of the usual bounding box representation. Screenshots of all three annotation tools are shown in Figure 7.1.

ANNOTATION QUALITY. As we already briefly discussed above, different annotations of the same video sequence may vary quite severely, both in terms of quality and in terms of the actual information that is provided (see Figure 7.2). Many of the widely used tracking datasets including the *ETHMS* and the *TUD* sequences were

1 http://mit.edu/vondrick/vatic
2 http://web.eee.sztaki.hu/~ucu/mvatool



(a) (b) (c)

Figure 7.2: Different level-of-detail. Next to unordered bounding boxes *(a)*, annotations for multi-target tracking should also provide the corresponding ID of each box *(b)*. In some cases even a pixel-level segmentation mask is available *(c)*.

originally annotated for the purpose of evaluating person detection. The annotations that were originally provided by the authors of these datasets only included bounding boxes of people without their corresponding IDs. Moreover, partially occluded pedestrians (approximately 50% and more) are ignored by the annotators since they are not expected to be found by the detector. An important ability of a multi-target tracker, however, is to keep track of individuals over time, even through complete occlusions. Therefore, performance results reported on these sequences either ignored the number of identity switches (Choi and Savarese, 2010) or resorted to manual counting (Mitzel et al., 2010), which is both tiresome and inaccurate.

Annotation data can also be provided on different levels-of-detail, both spatially and in terms of temporal resolution. For example, Horbert et al. (2011) provide pixel-level segmentation masks for each person in the *TUD-Crossing* sequence. But due to the time required to obtain such detailed information it is only available every 10^{th} frame. The authors of the *ParkingLot* sequence (Shu et al., 2012) annotate every 3^{rd} frame but the quality is rather poor. As illustrated in Figure 7.3, some trajectories are interrupted (*e.g.* the yellow one) and even slightly occluded people are not marked. This can in fact lead to a good detector or tracker that is able to find and identify all persons including the ones that are only partially visible to be penalized. Berclaz et al. (2006); Fleuret et al. (2008) discretize their ground truth both spatially and temporally. Their annotations include the cell occupancy of a ground plane grid only every 25^{th} frame, *i.e.* once every second.

To analyze how much different annotations affect the measured performance we conduct two experiments on the *TUD-Stadtmitte* sequence:

1. We evaluate the identical tracker output (obtained with our continuous tracker from Chapter 5) on three different sets of ground truth data.

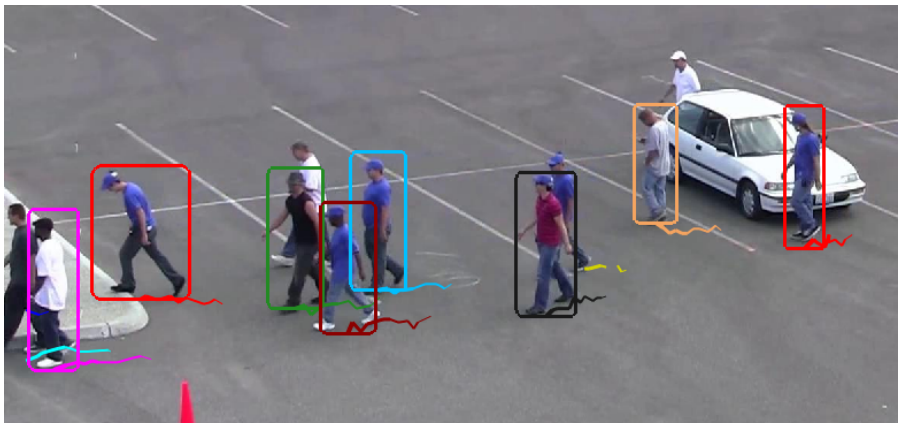


Figure 7.3: A sample frame from the *PNNL ParkingLot* sequence showing some deficits in the provided annotation. Besides non-smooth trajectories, several people are not marked in this ground truth.

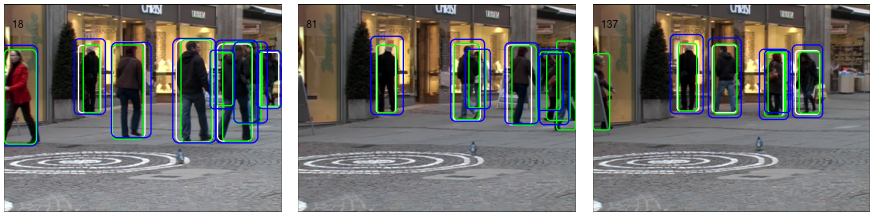


Figure 7.4: Three different publicly available annotations on the *TUD-Stadtmitte* sequence. The original annotations provided by the authors of the dataset (Andriluka et al., 2010) (plotted in white) do not contain any occluded pedestrians. Our annotations are shown in green and Yang's annotations (Yang and Nevatia, 2012a) in blue. Note the large difference in the size of the bounding boxes.

2. We evaluate the accuracy of one ground truth annotation with respect to the other ones for all three combinations.

The *TUD-Stadtmitte* sequence (Andriluka et al., 2010) has become quite popular and is frequently used for evaluating detection as well as tracking quality. Somewhat surprisingly, several 'ground truths' are publicly available for this short sequence, which differ significantly from one another. The reasons for this may be that the original annotations do not contain target IDs and that occluded pedestrians are not annotated. For the following experiment we obtained the IDs by greedy nearest neighbor linking but did not connect trajectories across occlusion gaps. The other two sets were annotated independently by two different groups. One is our own annotation and the other one is available for download on Bo Yang's website[3] . Bounding boxes from all three ground truth sets are overlaid and shown in

3 http://iris.usc.edu/people/yangbo/downloads.html

Gr. truth	Rcll	Prcn	GT	MT	ML	ID	FM	MOTA	MOTP
white	90.1	97.1	18	11	4	3	3	87.1	83.3
green	69.3	99.5	10	4	0	7	6	68.3	76.6
blue	72.1	99.1	10	4	0	7	6	70.8	71.9

Table 7.1: Evaluating the same tracking result with respect to different ground truth annotations.

"Sol."	Gr. tr.	Rcll	Prcn	GT	MT	ML	ID	FM	MOTA	MOTP
white	green	75.1	100.0	10	6	0	8	288	74.4	81.1
	blue	77.2	98.5	10	6	0	10	252	75.2	68.9
green	white	100.0	75.1	18	18	0	0	0	66.8	81.1
	blue	85.1	81.5	10	9	1	0	165	65.8	66.7
blue	white	98.5	77.2	18	18	0	2	13	69.2	68.9
	green	81.5	85.1	10	8	1	0	214	67.2	66.7

Table 7.2: A quantitative comparison of various ground truth annotations with respect to one another.

Figure 7.4. A coarse qualitative assessment reveals that the boxes in the latter dataset (blue) are much larger than those in the other two.

A quantitative performance of the same result but with respect to the three different ground truths is listed in Table 7.1. The numbers are computed in 2D with an overlap threshold of 0.5. As expected, the recall is much higher on a ground truth with fewer annotated bounding boxes (white). But there is still a noticeable gap in tracking accuracy MOTA and an even larger one in tracking precision MOTP between the two other annotation sets that were created specifically for multi-target tracking evaluation. This observation clearly demonstrates that the computed figures may vary greatly depending on what ground truth annotation is used.

In our second experiment we use one of the three sets of annotations as the "solution" and evaluate it with respect to the other two. Obviously, one cannot expect that the bounding boxes are always perfectly aligned to each other across various sets. However, it is reasonable to assume that at least different annotations would agree on the presence or absence of targets in the image. The figures shown in Table 7.2 are quite disillusioning. For instance, the top two rows show how the *white* ground truth scores when evaluated on the the *green* and on the *blue* one. Obviously, the recall stays low since occluded people are not present in this annotation. But even when comparing the more complete annotations to each other (rows 4 and 6), the overall accuracy (MOTA) remains below 70%. The reason here is that the difference in bounding box sizes leads to an overlap that is less than 50% in many cases, hence the annotations are counted as false positives. Note that the output of the tracker in Table 7.1

Eval.	Rcll	Prcn	FP	FN	MT	ML	ID	FM	MOTA	MOTP
ours	69.3	99.5	4	355	4	0	7	6	68.3	76.6
Yang	67.6	98.0	16	373	2	1	2	3	(66.0)	-
Masi & Lisanti	67.9	99.7	4	355	-	-	16	-	67.6	77.0
ours	59.4	85.3	118	469	2	0	9	9	48.4	59.8
CLEAR	(59.4)	(85.3)	118	469	-	-	10	-	48.4	(59.8)

Table 7.3: Evaluating the same result with respect to the same ground truth but with different evaluation scripts. The first part states evaluation in 2D while the bottom one is computed on the ground plane.

actually produces better quantitative results than a different ground truth. This once again shows that bounding box annotations are in fact quite ambiguous.

To conclude, both the quality and the level-of-detail can vary significantly across annotations, even for the same video sequence. A misalignment of bounding boxes in different annotation sets may not only lead to a lower tracking precision, but can severely impair the overall performance due to wrongly counted errors. It is therefore always important to state which ground truth data was used for measuring performance of a certain output.

7.1.2 *Evaluation software*

In the previous section we analyzed the quality of ground truth annotations and their impact on the reported numbers. We will now investigate whether a particular implementation of the evaluation protocol has an impact on the computed measures. To that end we evaluate the same tracking result as above on our own ground truth, but with different evaluation scripts. All tested scripts provide the raw number of false alarms and missed targets, such that precision and recall can easily be computed. Bo Yang's software, which operates on bounding boxes in 2D, additionally computes the number of mostly tracked and mostly lost trajectories, but unfortunately does not provide the average overlap. A second evaluation script, written by Iacopo Masi and Giuseppe Lisanti,[4] computes the CLEAR MOT metrics but not the trajectory-based ones (Bagdanov et al., 2012). Finally, we also compare our own implementation to the one provided for the original CLEAR challenge, written by Keni Bernardin (Bernardin and Stiefelhagen, 2008). All available numbers are listed in Table 7.3. The values in parentheses are computed based on the provided number of false positives, false negatives and identity switches. Note the extremely high number of detected mismatches in Masi & Lisanti's implementation.

Yang's software can be downloaded from the same website as the ground truth.

4 http://www.micc.unifi.it/masi/code/clear-mot

This number is probably not very reliable because the authors state that *"ID switches should be carefully counted by visual inspection"* in their documentation. Other than that, the figures in Tables 7.1 and 7.2 do not deviate substantially. Nonetheless, for a meaningful comparison it is crucial to use exactly the same evaluation software.

7.1.3 *Metrics ambiguity*

Having analyzed the impact of different ground truth annotations as well as various implementations of the same evaluation protocol on the resulting performance, we now take a closer look at the used protocols themselves. In Section 3.4, we formally defined several methods for measuring the performance of a tracking system, where some of the problems related to the quantitative evaluation were already mentioned. Here, we will follow up on this issue and point out concrete deficits of the existing definitions.

Throughout this dissertation, we employed two sets of evaluations metrics, CLEAR MOT and the trajectory-based measures of Li et al. (2009). As we can see in Table 7.3, computing the same error measure is not clearly defined since various evaluation scripts do not produce identical numbers. Besides possible implementation discrepancies, the metrics' definitions themselves carry ambiguities.

DISTANCE. To establish correspondences between the true objects and the produced results, a distance measure is required to assess how similar or how close the hypothesis is to the ground truth object. One possible choice is the PASCAL VOC criterion, which measures the overlap between two bounding boxes (*cf*. Eq. (3.1)). When tracking is performed directly in the world coordinate system, the standard Euclidean distance between the objects' centers can be employed. In both cases, a threshold is required that determines whether a target-hypothesis pair constitutes a potential match or not. In other words, the evaluation procedure itself is dependent on at least one parameter that should always be stated. For the overlap criterion, a threshold of 0.5 has been widely accepted. For measuring distances in world coordinates, Stiefelhagen et al. (2006) propose 500mm. However, the main application there is to track multiple people in meetings in a rather small area. We found that such a threshold is too conservative for outdoor scenes for two reasons: First, in surveillance settings cameras are usually far away from the scene showing a much larger area of interest, such that targets only occupy a small image region. Second, the camera calibration may be unreliable, *e.g*. due to a low view point. In both cases targets that are only slightly misplaced on the image induce a large 3D error. Consequently, a threshold that is too small will lead to an undesirable behavior when correct results are counted as false alarms, while the true target remains untracked.

We therefore use a 1 meter hit/miss threshold throughout all experiments.

ASSIGNMENT. One further ambiguity of tracking metrics lies in the exact procedure how the output hypotheses are assigned to the ground truth objects, which is not specified explicitly. A greedy assignment strategy is arguably the simplest choice, albeit not the one that leads to the best matching. A typical case of non-optimal assignment is illustrated in Figure 3.7 (right). One way to avoid such cases is to perform a two-pass matching with the Hungarian algorithm, as is done, *e.g.*, by Yang and Nevatia (2012a).

ERROR WEIGHTING. Recalling the definition of MOTA from Eq. (3.2), all three types of errors (FP, FN and ID) are weighted equally as suggested by Stiefelhagen et al. (2006); Bernardin and Stiefelhagen (2008). Naturally, each error type can be weighted individually according to its importance for the respective application. For offline motion analysis it may be important to reconstruct correct, identity preserving trajectories, while finding absolutely all present targets is less crucial. A higher weight for identity switches may therefore be more desirable. On the contrary, a driver assistance system should detect every single pedestrian and at the same time maintain a low number of false positives to avoid unnecessary warnings. On the other hand it is less relevant to keep the identity of each person over time. In such case, the aim is to achieve the highest possible precision and recall while less attention is paid to the number of ID switches. This may also be the motivation of Ellis and Ferryman (2010), who impose a logarithmic weight on the number of mismatch errors when computing the MOTA score.

7.1.4 *Benchmarking multi-target tracking*

Many tasks in computer vision are approached by designing models that need to be trained or tuned, *i.e.* fitted to the annotated training data, to make predictions about unseen data. To enable a fair comparison between various methods, some areas offer well-established benchmarks with pre-defined training and test sets. To name a few, there is the PASCAL challenge for object detection or segmentation (Everingham et al., 2012), the Middlebury benchmark for multi-view stereo (Seitz et al., 2006), or KITTI for stereo or optical flow (Geiger et al., 2012). Although several multi-target tracking datasets are frequently used in the literature (see Section 3.3), there is no established consensus of how to separate the data into training and testing sets. The common strategy to present the performance of a tracking method is to tune the parameters to a fixed set of sequences, thereby treating them as training and test data at the same time. Obviously,

this is not ideal since the model is overfitted to the chosen data and will usually perform considerably worse on unseen data. To nonetheless reduce the effect of overfitting, it is considered good practice to choose several datasets that exhibit strong variations in person count, view point and resolution. In our experiments in Chapter 6 we further address this issue by performing leave-one-out cross validation on all six sequences to show the robustness of our method.

There is another issue that complicates elementary comparison. Most current multi-target trackers perform tracking-by-detection, *i.e.* the actual input data are not the raw images but a set of independently precomputed detections. Clearly, the performance of both the data association and the reconstruction of trajectories will greatly depend on the quality of the detector. One way to evaluate various trackers independently might be to provide a standard detection set for each method. However, this is not straightforward to implement in practice, since different methods require different types of input. Some rely on plain bounding boxes (Pirsiavash et al., 2011), others also consider the confidence value of each detection (*cf*. Chapters 5 and 6) – which is non-trivial to calibrate in general – while other approaches work on contours of pedestrians (Henriques et al., 2011).

Nevertheless, we believe that a standardized multiple target tracking benchmark consisting of a variety of diverse video sequences is needed to facilitate comparison between state-of-the-art methods. Similar to the examples above, it should include a clear training and test set, a reasonably accurate ground truth and a centralized evaluation tool. If possible, participants should also use the same detector results as input for their tracker. The only currently existing method (that we are aware of) to objectively measure the performance of a tracking algorithm is to send the results on the *S2L1* sequence (represented by bounding boxes) to the PETS organizers (Ellis and Ferryman, 2010). The computed CLEAR MOT metrics, evaluated with respect to unpublished ground truth, are then sent back to the authors. Provided that current methods achieve near perfect results on that particular sequence, it is time to move towards more challenging datasets. Clearly, such benchmarks entail the risk of shifting the research goals from developing innovative techniques to pushing the numbers higher on that particular data. However, previous benchmarks, such as Middlebury[5] (Baker et al., 2011) or PASCAL (Everingham et al., 2012) for example, show that the raw ranking is not the only criterion how a specific method is valued in the community. In fact, despite caveats of benchmarks both projects considerably boosted research in their respective area of computer vision.

5 http://vision.middlebury.edu

7.2 NUMERICAL INSTABILITY

Most modern processors use 64 bits to represent numeric values.

Digital computers operate on numbers of a fixed length. While this simply means a limited available range for integers, real numbers can be approximated only up to a certain precision. The most common way that allows for a wide range of values is to use the floating-point representation, where decimal numbers are stored in two blocks: a fixed number of significant digits and an exponent. Each time a non-integer value is computed, the CPU has to decide how to store it in memory. This procedure may lead to undesirable effects. Even though certain standards that clearly define a correct handling exist, compilers do not always follow such rules in order to optimize the code for a specific processor. Depending on the exact representation of a real number, identical code may thus produce different results on a different hardware. Although the deviations may at first seem negligible, in practice the errors may accumulate over time and lead to significantly varying outcomes.

Here, we would like to point out that our proposed algorithms are also affected by the described issue. Most of the code for all three tracking approaches is written in MATLAB. While this language offers a wealth functionality and boosts development speed, the software itself is not open source and the documentation only provides a high-level description of the provided methods. It is therefore not possible to pinpoint to the exact source of numerical inconsistencies.

In our experiments, the largest deviations across various machines arose during the continuous optimization from Chapter 5. Figure 7.5 illustrates this behavior on the *TUD-Stadtmitte* sequence, where identical code with identical input data is executed on five different CPUs and leads to significant inconsistencies in the final results. The main reason is the iterative nature of the optimization algorithm. A slight inaccuracy in the high-dimensional gradient will lead the continuous minimization to a similar, albeit not identical state. This, in turn, may have a much larger impact when a discrete jump move is made. Depending on the exact shape of the trajectories, a different jump will be chosen such that the optimization will take an alternative path through the energy landscape. Note that small deviations in the computed energy (here $\approx 2\%$) may in fact cause rather large gaps in performance (≈ 5 percentage points in MOTA) due to hard discrete decisions during optimization. To enable a meaningful comparison of different variations of the continuous method, all results in Table 5.7 were computed on machines with the same hardware architecture.

We ran into similar problems using the discrete-continuous optimization (*cf*. Chapter 6). Here we found, that certain MATLAB routines, such as the backslash operator '\' for solving linear equations or even the computations of a simple dot product, are not consistent across various platforms. To give an intuition, the following numbers

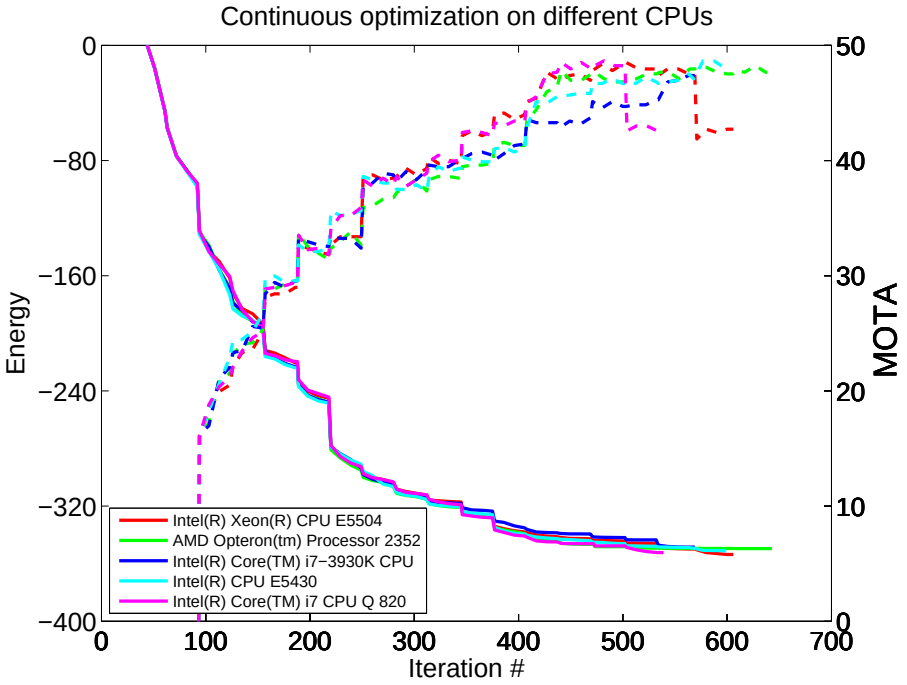


Figure 7.5: Five runs of the same continuous optimization on five different CPU types lead to slightly different final results.

are computed by the MATLAB built-in dot product $a^\mathsf{T} b$ on two different CPUs:

95608.406663221758208237588405 (Intel i7-3930K CPU)
95608.406663221743656322360038 (Intel i7 CPU Q 820).

In both cases, the input is identical and consists of a binary vector $a \in \{0,1\}^{105}$ and a real vector $b \in \mathbb{R}^{105}$, respectively. Again, the absolute difference in the order of 10^{-10} may seem insignificant. In practice, however, discrete optimization may be guided towards a different data association solution, which in turn may lead to a different overall result. We found that the problem of inconsistent computations can be mitigated by replacing MATLAB routines by their counterparts written in C, where all compiler optimizations can be turned off during compilation. Of course, developing code that is closer to machine level defeats the purpose of using a high-level programming language, but at least offers a possible workaround. For our experiments in Chapter 6 we only used identical CPUs as discussed above. Interestingly, we did not encounter any numerical inconsistencies with the ILP formulation in Chapter 4. A likely explanation is that the main optimization is outsourced to external solvers that are likely compiled following certain standards.

7.3 PRIVACY ISSUES AND FURTHER CONCERNS

Fig. 7.6: CCTV cameras in a metro station.

In modern cities, it is almost impossible to move through public space without being filmed. Surveillance (or closed-circuit television (CCTV)) cameras are installed in many places such as squares, buses, train stations, banks, *etc*. Falling prices of electronic equipment, corporate policies and political decisions all have their influence on this development. These cameras capture video footage to prevent crimes before they happen, or, more frequently to help the investigation afterwards (Welsh and Farrington, 2008). Interestingly, the majority of the population feels more secure by the sheer presence of cameras (Phillips, 1999), although the effect on crime reduction to date remains debatable (Welsh and Farrington, 2008). On the other hand, many people, primarily the younger generation, are concerned how this development may affect the privacy of individuals (Hempel and Töpfer, 2004). In this section we will elaborate on this subject and state how multi-target tracking relates to this issue.

Although theft offenses decrease in monitored areas, there is no significant reduction in violence related crimes.

First of all, it is important to emphasize that research on multiple object tracking concentrates on reconstructing trajectories of *unknown* individuals. In other words, there is no link to any personal data of a tracked person. In the presented work, each target is assigned a unique ID that only exists as long as the target resides inside the field of view of the camera. Whenever a person exits the scene and re-enters at a later point, a new ID is assigned. Even though there is work that primarily addresses the problem of re-identifying targets (Gheissari et al., 2006; Hirzer et al., 2012), its main purpose is to facilitate the 'handover' problem across multiple cameras. In such cases, the data is again always handled in a depersonalized way, *i.e*. the identities are represented by unique numbers, unrelated to any personal data. It is true, however, that a connection to a specific person may be made through recognition and identification, but this task lies beyond the goal of multi-target tracking.

Nevertheless, there is a general concern that CCTV footage can be exploited for criminal purposes. In particular, there is evidence for deliberate spying, voyeurism and discrimination. Norris and Armstrong (1999) show in their study that people of certain age and race are observed disproportionately often for 'no obvious reason' by the surveillance operators. This and similar findings have spawned a novel research branch that addresses the task of privacy preserving surveillance.

As a preliminary point, it is important to emphasize that personal privacy is not well defined. Some may consider that masking their face is enough to remain anonymous while others would feel an intrusion in their private life in the presence of any recording device that can reveal their age, gender, race, *etc*. According to a psychological study by Babaguchi et al. (2009), the decision about the amount of information that a monitored person is willing to reveal strongly depends on the relationship to the observer. It is therefore desirable to automatically pre-process the captured scene and only make as much data available as needed for a particular situation.

One of the simpler methods to hide the identity is to perform surveillance outside the visible light spectrum. Tao et al. (2012) employ a network of passive infrared sensors – also known as motion detectors – for activity recognition and fall detection in private or semi-private indoor environments. However, the rich visual information that may be crucial for investigation is not preserved in such setting. Another way to ensure anonymity of tracked individuals is to decompose the input stream into several components and to scramble the targets such that they appear unrecognizable to the observer (Qureshi, 2009). Various ways of disguising the targets have been proposed. Simply obscuring the face or the entire person by blurring or pixelation may hide the identity (Spindler et al., 2006), but at the same time it may discard valuable information about the activity or the facial expression. Newton et al. (2005) and Gross et al. (2009) propose face de-identification, a method that preserves much of the facial nuances but makes faces unrecognizable by altering their general appearance. Chen et al. (2009) develop an edge-based representation that reveals the activity while entirely hiding the identity. Another strategy is to selectively apply the scrambling algorithms only to a certain set of targets. To this end, Schiff et al. (2009) present *Respectful Cameras* that are able to recognize specified markers (*e.g*. helmets or vests of a certain color) and obscure only those identities that wear an 'invisible cloak'. This method may be applicable within a certain area, such as construction site or laboratory, but is hard to put into practice in public space.

As with many other applications, these computer vision-related methods cannot guarantee perfect performance when deployed in real-world scenarios. However, they at least succeed at complicating potential abuse by unauthorized personnel. With increasing public awareness, the pressure on authorities and manufacturers may grow and lead to a certification and registration of surveillance technology (Senior et al., 2005), similar to network security.

Every technology can be used with both good and harmful intentions. The benefits of omnipresent video surveillance remain doubtful, although most people tolerate or even welcome it. However, the multi-target tracking models presented in this dissertation are rather

generic and can be employed in numerous applications including road safety, life sciences, or accident prevention in crowds (*cf*. Section 1.2). Therefore, we believe that the gains outweigh potential dangers and privacy concerns.

8 CONCLUSION AND OUTLOOK

Once we accept our limits, we go beyond them.

ALBERT EINSTEIN

CONTENTS

ENERGY minimization methods provide a suitable tool to approach the task of multiple target tracking. The main challenge is to design an objective function that accurately describes the problem at hand while at the same time remains feasible to optimize in practice. This dissertation investigated three different approaches that address the problem from three different sides, each having its benefits and drawbacks. Moreover, common pitfalls and challenges that arise with quantitative evaluation of various methods were presented. In this final chapter we will summarize and discuss both the contributions and the limitations of this work and indicate possible future directions to further improve automated multi-target tracking.

8.1 CONTRIBUTIONS

8.1.1 *Discrete tracking with a dynamic model*

In Chapter 4, we formulated multi-target tracking on a discrete grid, inspired by the work of Berclaz et al. (2009). The main motivation behind this formulation was to reduce the (potentially infinite) solution space of all possible trajectories to a finite set of feasible paths. Note that a significant amount of the previous work also regarded

multi-target tracking as a discrete combinatorial problem (Morefield, 1977; Reid, 1979; Storms and Spieksma, 2000; Zhang et al., 2008), but performed discretization at the level of detections. In contrast, grid-based partitioning provides a more natural way of implicitly (or explicitly) handling missing image evidence.

We made several contributions to this methodology. Firstly, we introduced a dynamic model into a integer linear program (ILP) formulation by appropriately extending the underlying state space. To encode the targets' dynamic behavior into the binary variables, the pairwise relations between two neighboring frames were extended to triples in three consecutive time steps. The benefit is that this allows encouraging smooth target motion, which can be an important cue for reliably keeping track of multiple targets over time. Furthermore, the lattice structure was changed from rectilinear to hexagonal. The resulting tri-axial grid reduces the effect of aliasing and allows a more accurate measurement of motion change without considerably enlarging the neighborhood. Additionally, we proposed an extended set of constraints that perform non-maxima suppression (NMS) on the trajectories rather than independently frame by frame. These constraints go beyond simple collision avoidance and additionally suppress the existence of targets in all neighboring cells, which avoids multiple intertwined trajectories. Our extended formulation produces smoother trajectories without unnatural jittering artifacts. Moreover, taking the dynamic model into account also improves the average tracking accuracy by reducing the number of track fragmentations and identity switches.

8.1.2 *Continuous energy minimization*

While the ILP formulation above achieves (near) global optimality, the trajectories are restricted to pass through a discrete set of locations, which is a strong limitation. To remedy this shortcoming, Chapter 5 introduced a novel approach to multi-target tracking that followed a different strategy. In contrast to a restrictive objective that can easily be optimized, we turned to the opposite side of the spectrum and focused on a formulation with as few simplifications as possible, without aiming at achieving global optimality. In particular, we designed a continuous energy function with the primary goal to capture the important aspects of multi-target tracking as completely as possible. To the best of our knowledge, this had not been done before. The energy is composed of several individual components, including the observation term, a first-order dynamic model, physically motivated exclusion and persistence constraints and a regularizer. Moreover, we developed a global occlusion formulation that seamlessly fits into the energy minimization approach. Each term is modeled by a differentiable function, such that standard continuous minimization

techniques can be applied. However, the resulting high-dimensional objective function is highly non-convex, which prohibits global optimization. Nevertheless, we developed custom discrete jump moves that provide enough flexibility to ensure that strong local minima can be found efficiently. The continuous energy minimization for multi-target tracking yields state-of-the-art results on particularly challenging video sequences, both visually and in terms of a quantitative evaluation.

Yet, there are two drawbacks of this formulation. One is that the proposed discrete jumps are executed in a greedy fashion. Although the achieved local minima provide some of the best tracking results obtained so far, we could show in Section 5.3.3 that solutions with a lower energy yield even better performance. The second shortcoming is that, similar to the ILP approach from Chapter 4, the data association is bypassed since the variables of the energy only describe the locations of the reconstructed (or potentially feasible) trajectories, but give no information about the detector responses used, which form the basis of the observation model.

8.1.3 *Unified data association and trajectory estimation*

In Chapter 6, we presented a discrete-continuous energy that combines both challenges – the combinatorial problem of data association and the continuous problem of trajectory estimation – in a single objective function. The advantage is that probing various permutations of the data assignment can be approached by powerful discrete optimization techniques while all trajectories are represented in their natural continuous domain. The energy is minimized by alternating between the discrete and the continuous part.

Two alternative strategies were developed under this framework:

- an energy function that is amenable to standard optimization algorithms and works reasonably well on moderately crowded scenarios, and
- a more complex energy that takes into account inter-object exclusion at the level of both detections and trajectories.

For the second variant, we proposed a general formulation of a pairwise label cost that penalizes co-occurrence of certain labels within a CRF framework. This approach enabled performing mutual exclusion at the level of trajectories in discrete-continuous multi-target tracking, achieving superior performance over the basic formulation. Moreover we proposed an expansion move-based optimization scheme to efficiently minimize the resulting non-submodular energy. This method slightly outperforms the continuous energy formulation on average with respect to standard metrics, while at the same time providing a

solution to data association by labeling each detection with the corresponding target ID.

Furthermore, we provided a methodology for deriving the shape of the energy potentials from real-world data. By analyzing ground truth statistics of numerous sequences, we inferred the functional form for various terms, such as velocity or occlusion length, directly from the present data instead of guessing based on intuition or by trial and error.

8.1.4 *Evaluation challenges*

Finally, Chapter 7 addressed a number of important practical aspects that should not be ignored in multi-target tracking research. Notably, it was shown that evaluating multiple object tracking is far from trivial. Besides ambiguous or non-standardized definitions of quality metrics, both the quality of human-annotated ground truth and the implementation of a specific evaluation procedure can deviate by a large margin from one another. To quantify these statements, we carried out several experiments on various ground truth annotations and with different evaluation tools.

8.2 FUTURE PERSPECTIVES

Although much progress has been made in the last several years, the task of robustly tracking multiple targets in video sequences is far from solved. Each of the three proposed approaches furthered the state of the art in its own way. To conclude, we point out the remaining limitations of the presented models and discuss how they can be improved to further advance the performance of automated tracking systems.

8.2.1 *Object detector*

While it does not concern a particular tracking method, the performance of the object detector has a substantial influence on the final result of any tracking-by-detection approach. In our experience, working with synthetic data (*e.g.* considering all manually annotated detections) produces near perfect tracking results ($> 95\%$ MOTA) in all cases. In fact, even naive nearest neighbor data association, which we employed to reconstruct identity-preserving annotations (*cf.* Section 7.1.1), yields acceptable performance. Tracking-by-detection will therefore benefit from further advances in object detection, which is largely an independent research area.

We will discuss a joint strategy in Section 8.2.5.

8.2.2 *Extracting more image features*

One of the most promising directions towards better tracking performance is to utilize additional image information. The presented methods rely only on spatio-temporal target locations and their corresponding confidence values of the detector. Arguably, a video sequence provides more cues that can potentially help to reconstruct individual trajectories more accurately.

APPEARANCE. One of the most obvious features that has not been fully exploited in the proposed methods is the visual appearance of the objects. In Chapters 4 and 5, a basic appearance model is employed to measure the evolution of a target's color distribution in neighboring frames. The similarity is measured by comparing the color histograms inside the bounding boxes between adjacent frames. One drawback is that the considered region always contains background pixels and sometimes also other objects that block the line of sight, as illustrated in Figure 6.13. Therefore, care should be taken to only extract the relevant portions of the image that contain the desired target (see below). Moreover, further features like texture or shape may be considered as well to construct a richer visual representation for each target (Yang and Nevatia, 2012b). Furthermore, long-range connections that go beyond consecutive frames but spanning several seconds may be more helpful to establish track correspondences through long-term occlusions (see Section 8.2.3 below).

POSE AND ORIENTATION. One possibility to only extract the foreground pixels is to estimate the pose of pedestrians (Andriluka et al., 2010; Sun et al., 2012). This may help in two cases. On one hand, the person inside the bounding box can be localized more precisely, which may facilitate discarding irrelevant background information for a more robust appearance model. On the other hand, one could match the walking cycle of a person to enforce more robust correspondence (Andriluka et al., 2008). This may, in fact, avoid identity switches in cases when the appearance information alone is unreliable, as depicted on the bottom of Figure 6.13. It is important to note that pose estimation becomes rather difficult and unreliable for objects that appear small in image space. Therefore, it can only help in high-definition video sequences or in applications where people are close enough to the observer.

The object's orientation offers one further cue. Continuing the example of people tracking, it is safe to assume that pedestrians mostly walk facing forwards, or in rare cases backwards, but rarely sideways. This information is ignored by current tracking-by-detection approaches. Both methods presented in Chapters 5 and 6 only consider the distance between a detection and the reconstructed trajec-

tory in their data term. This means that the lowest energy is achieved when a target precisely passes through a detection, independent of its heading direction. As a result, spurious trajectories running perpendicular to the actual moving direction are sometimes produced to explain the evidence. Most frequently, such behavior is observed during the discrete-continuous optimization. To address this issue, one could thus either explicitly or implicitly represent the facing direction of each target and enforce temporal smoothness or penalize deviations between the observation and the current target motion. Of course, in both cases a reliable view estimation of each detection is essential.

TEMPORAL INFORMATION. Somewhat similar to the considerations above, currently employed object detectors are based on processing individual frames and therefore discard valuable information about the targets' motion. Tracklet-based approaches (Huang et al., 2008; Andriluka et al., 2008) combine close detections into short tracklets and take those as a starting point for long-term data association. It is conceivable to follow a similar strategy for the presented energy minimization methods. One the one hand it can reduce the computational burden by fixing the tracks in those regions where data association is unambiguous. On the other hand, analyzing several subsequent frames could provide further information about the target's motion directly at the detection level. This can be achieved either by computing low-level features like optical flow, or by applying an image-based tracker, such as mean-shift or KLT, on the image region inside the bounding box. The estimated motion can then provide additional cues about the heading direction of a target and prevent discrepancies between the estimated and the true trajectory.

8.2.3 *Towards more expressive models*

Most components of the presented methods are rather simple. For example, the exclusion constraint in the case of continuous energy minimization only models pairwise interactions between targets on a frame-by-frame basis. Consequently, when several targets move close to one another over a longer period, a penalty is applied at each time step (*cf*. Figure 5.12 *(Eexc)*). Using a more sophisticated social model (see Section 2.5.2) it is possible to adjust this penalty accordingly for certain target groups (Qin and Shelton, 2012). Furthermore, the constant velocity model only captures a first-order relationship between successive motion vectors. Sometimes this leads to unnatural trajectories that do not exhibit abrupt turns but rather wiggle from side to side over a long time period, changing their direction several times. This is especially visible for the spline representation from Chapter 6.

A penalty based on the count of turns may enforce more plausible trajectories in such situations.

Another way for making the discrete model more expressive without resorting to high-order factors is to include long-term pairwise connections. Edges that span several time steps can help to re-identify a person after an occlusion, provided that robust dynamic and appearance models are available. While both extensions certainly seem attractive for increasing modeling accuracy, they inevitably lead to more complex inference. A possible solution is to keep an adaptive graph structure and to prune away connections in unambiguous areas.

8.2.4 *Parameter estimation*

To avoid the effect of overfitting, a single parameter set for each method was used for the entire dataset. Even though a pre-defined set of parameters may show good average performance on a particular dataset, a more careful analysis of individual sequences may significantly boost the results. Our experience showed that the found parameter set rarely corresponded to the top result on each sequence separately. In particular, properties like camera angle or motion, crowd density or target size may vary substantially across different video sequences and influence the behavior of the model. All these aspects can, in principle, be automatically inferred from input data and used to guide the parameter search towards a more promising region. Moreover, it may be possible to make a more high-level decision similar to Cifuentes et al. (2012), on which algorithm is best suited for the present data.

Another way to learn parameters is to employ synthetically generated video sequences (Flagg and Rehg, 2012). This way, a large amount of training data can be obtained at little cost. However, synthetic training examples should be handled with care since their appearance and also the behavioral statistics may deviate from real video footage.

8.2.5 *Joint detector-tracker optimization*

Although several notable exceptions exist (Leibe et al., 2008b; Wu et al., 2012; Yan et al., 2012), most tracking-by-detection approaches, including the ones presented in this dissertation, regard detection and tracking as two entirely separate tasks. Detections are usually filtered with non-maxima suppression (NMS), which significantly reduces the computational burden, but also means that most image evidence is entirely discarded. Our proposed methods, like most other approaches, allow tracks to survive without detections for a certain time span to bridge occlusions. However, this sometimes leads to

trajectories that weave through empty background areas that would be confidently classified as 'non-target' by the detector. Therefore, one can expect a performance improvement by combining object detection and tracking within an energy minimization framework (see also Section 6.6.5). A further approach may be to specifically use false negatives of a tracker output to bootstrap the detector thereby closing the gap between the two components.

As we have discussed, there are many possible ways in which the proposed multi-target tracking approaches can be extended and improved. With the current rate of development, we can expect much progress in this area in the near future both in research and in real-world applications.

Part IV

APPENDIX

A

APPENDIX

A.1 SOLVING MIXED INTEGER LINEAR PROGRAMS

In Chapter 4 we formulated multiple target tracking as an integer linear program (ILP) where the aim is to minimize (or equivalently maximize) a linear objective function with linear constraints with an additional requirement that all variables remain integers. Formally, an ILP is an optimization problem of the form:

The problem is called a mixed integer linear program (MILP) if only a subset of the variables is required to be integer.

$$\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x} \tag{A.1}$$

subject to:

$$A\mathbf{x} \leqslant \mathbf{b} \tag{A.2}$$

$$\mathbf{x} \geqslant 0 \tag{A.3}$$

$$\mathbf{x} \in \mathbb{Z}^n \tag{A.4}$$

While linear programs can be solved to global optimality in polynomial time, the integer constraints lead to much more complex problems that are NP-hard in general. In this section we will briefly outline the idea behind the *branch-and-cut* algorithm, which is essentially a combination of *branch-and-bound* and the *cutting planes* methods and is perhaps the most popular approach for addressing this class of combinatorial problems.

The most common initial step for (mixed) integer linear program MILP solvers is LP-relaxation. The general idea behind it is to simply discard the integer constraints (A.4) in order to efficiently obtain the globally optimal solution to a simplified problem. If all variables of the resulting solution are integers then it is also the globally optimal solution to the original problem. Although for some particular formulations this is indeed always the case (*e.g.*, Berclaz et al., 2011), unfortunately, this is not true in general, where the obtained solution may contain fractional components that violate the integer constraints. The question that then remains is how to proceed with these fractional values.

A naive, yet effective approach is to branch out on one of the fractional values. To illustrate this procedure, let us assume that we have found a globally optimal solution $\mathbf{x}^{\text{LP}}$ where x_i is fractional:

$$a < x_i < a + 1, \quad a \in \mathbb{Z}. \tag{A.5}$$

By adding two linear constraints

$$x_i \leqslant a \tag{A.6}$$

and

$$x_i \geqslant a + 1 \tag{A.7}$$

we obtain two additional problems that can be addressed by LP-relaxation as before. By recursively applying the same procedure we can eliminate all non-integer values to achieve the desired solution (or to determine that no feasible solution exists). Note that it is not necessary to branch further if both branches yield objective values that are worse than the currently known best feasible solution $\bar{\mathbf{z}}$. Its value serves as an upper bound on the optimal solution. Such pruning techniques can significantly speed up the search of this *branch-and-bound* technique.

An alternative to adding two constraints to enforce integrality of one specific variable is to add a so-called *cutting plane*. Intuitively, a cutting plane is another linear constraint that "cuts out" a large portion of the solution space, thereby reducing the complexity, while retaining all feasible solutions. A cutting plane can be constructed by *integral rounding*, where a combination of several linear constraints serves as a starting point and rounding is performed to maintain the integrality properties (Chvátal, 1973; Gomory, 1963).

The popular *branch-and-cut* approach is a combination of the two strategies described above. After each relaxation step, a decision is made whether to branch out by adding inequalities, or whether to add a cutting plane. This decision is largely influenced by heuristics and a particular implementation. An interested reader is referred to (Mitchell, 2002) for a comprehensive overview of various methods and some in-depth discussions.

BIBLIOGRAPHY

IRTAD road safety annual report 2011. Technical report, International Transport Forum, Paris, April 2012. URL http://www.internationaltransportforum.org/Pub/pdf/11IrtadReport.pdf. (Cited on page 5.)

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, July 2004. ISBN 1-58113-838-5. URL http://ai.stanford.edu/~ang/papers/icml04-apprentice.ps. (Cited on page 28.)

Tobias Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, July 2009. URL http://scip.zib.de/. (Cited on page 59.)

Sarmad Al-Bassam, Min Xu, Thomas J. Wandless, and Don B. Arnold. Differential trafficking of transport vesicles contributes to the localization of dendritic proteins. *Cell Reports*, 2(1):89–100, July 2012. ISSN 2211-1247. (Cited on page 7.)

George A. Alvarez and Steven L. Franconeri. How many objects can you track?: Evidence for a resource-limited attentive tracking mechanism. *Journal of Vision*, 7(13), October 2007. ISSN , 1534-7362. URL http://viscog.psych.northwestern.edu/publications/AlvarezFranconeri_MOT_inpress.pdf. (Cited on page 16.)

Mohamed R. Amer, Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song-Chun Zhu. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7575 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2012. ISBN 978-3-642-33764-2. URL http://web.engr.oregonstate.edu/~amerm/Website/eccv12_multiscale_activities.pdf. (Cited on page 28.)

Björn Andres, Thorsten Beier, and Jörg Kappes. OpenGM: A C++ library for discrete graphical models. *arXiv:1206.0111*, June 2012. URL http://arxiv.org/abs/1206.0111. (Cited on page 129.)

Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, June 2008. (Cited on pages 21, 159, and 160.)

Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3D pose estimation and tracking by detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, June 2010. (Cited on pages 22, 36, 120, 143, and 159.)

Anton Andriyenko and Konrad Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, volume 6311, pages 466–479, Lecture Notes in Computer Science, 2010. Springer. (Cited on pages 3, 12, 26, 47, and 70.)

Anton Andriyenko and Konrad Schindler. Multi-target tracking by continuous energy minimization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, Colorado, June 2011. (Cited on pages 4, 12, 70, 81, and 100.)

Anton Andriyenko, Stefan Roth, and Konrad Schindler. An analytical formulation of global occlusion reasoning for multi-target tracking. In *11th International IEEE Workshop on Visual Surveillance*, Barcelona, Spain, November 2011. (Cited on pages 12, 70, 81, and 100.)

Anton Andriyenko, Konrad Schindler, and Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, June 2012. (Cited on pages 4, 12, 16, 35, and 108.)

Shai Avidan. Ensemble tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 494–501, San Diego, California, June 2005. (Cited on page 2.)

Noboru Babaguchi, Takashi Koshimizu, Ichiro Umata, and Tomoji Toriyama. Psychological study for designing privacy protected video surveillance system: PriSurv. In Andrew Senior, editor, *Protecting Privacy in Video Surveillance*, pages 147–164. Springer London, January 2009. ISBN 978-1-84882-300-6, 978-1-84882-301-3. URL http://link.springer.com/chapter/10.1007/978-1-84882-301-3_9. (Cited on page 152.)

Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, 2009. URL http://vision.ucsd.edu/~bbabenko/data/miltrack_cvpr09.pdf. (Cited on page 2.)

Andrew D. Bagdanov, Alberto Del Bimbo, Fabrizio Dini, Giuseppe Lisanti, and Iacopo Masi. Compact and efficient posterity logging

of face imagery for video surveillance. *IEEE Multimedia*, 19(4):48–59, 2012. (Cited on page 145.)

Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 92(1):1–31, March 2011. ISSN 0920-5691, 1573-1405. URL http://vision.middlebury.edu/flow/floweval-ijcv2011.pdf. (Cited on page 148.)

Yaakov Bar-Shalom, Kuo-Chu Chu Chang, and Henk A. P. Blom. Tracking of splitting targets in clutter using an interacting multiple model joint probabilistic data association filter. In *Proceedings of the 30th IEEE Conference on Decision and Control*, volume 2, pages 2043–2048, 1991. (Cited on page 19.)

Yakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association*. Academic Press, 1988. (Cited on page 18.)

Yakov Bar-Shalom and A.G. Jaffer. Adaptive nonlinear filtering for tracking with measurements of uncertain origin. In *Proceedings of the IEEE Conference on Decision and Control and 11th Symposium on Adaptive Processes*, volume 11, pages 243 –247, December 1972. (Cited on page 19.)

Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua. Tracking multiple people under global appearance constraints. In *Proceedings of the Thirteenth IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011. URL http://infoscience.epfl.ch/record/167889/files/Tracking_Multiple_People_under_Global_Appearance_Constraints.pdf. (Cited on pages 26, 70, and 81.)

Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, Colorado, 2011. URL http://www.robots.ox.ac.uk/ActiveVision/Publications/benfold_reid_cvpr2011/benfold_reid_cvpr2011.pdf. (Cited on pages 25 and 85.)

Jérôme Berclaz, François Fleuret, and Pascal Fua. Robust people tracking with global trajectory optimization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, New York, 2006. URL http://www.idiap.ch/~fleuret/papers/berclaz-fleuret-fua-cvpr2006.pdf. (Cited on pages 22, 25, 34, 55, 59, and 142.)

Jérôme Berclaz, François Fleuret, and Pascal Fua. Multiple object tracking using flow linear programming. In *12th IEEE*

International Workshop on Performance Evaluation of Tracking and Surveillance (Winter-PETS), December 2009. URL http://www.idiap.ch/~fleuret/papers/berclaz-et-al-pets2009.pdf. (Cited on pages iii, v, 3, 10, 25, 26, 47, 49, 55, 56, 60, 61, 65, 70, 72, 81, and 155.)

Jérôme Berclaz, François Fleuret, Engin Türetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(9):1806–1819, September 2011. URL http://www.idiap.ch/~fleuret/papers/berclaz-et-al-tpami2011.pdf. (Cited on pages 26, 100, 102, and 165.)

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13: 281–305, March 2012. ISSN 1532-4435. URL http://jmlr.csail.mit.edu/papers/volume13/bergstra12a/bergstra12a.pdf. (Cited on page 119.)

Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Image and Video Processing*, 2008(1):1–10, May 2008. ISSN 1687-5281. URL http://jivp.eurasipjournals.com/content/2008/1/246309. (Cited on pages xv, 39, 145, and 147.)

Margrit Betke, Diane E. Hirsh, Angshuman Bagchi, Nickolay I. Hristov, Nicholas C. Makris, and Thomas H. Kunz. Tracking large variable numbers of objects in clutter. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, 2007. URL http://vision.cse.psu.edu/courses/Tracking/vlpr12/Betke-etal-cvpr07.pdf. (Cited on page 6.)

Samuel S. Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999. ISBN 9781580530064. (Cited on page 18.)

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, November 2001. ISSN 0162-8828. URL http://cirl.lcsr.jhu.edu/main/images/b/b9/BVZ-pami01-final.pdf. (Cited on pages 10, 114, and 115.)

Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Proceedings of the Twelfth IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, October 2009. URL https://epreuve-test.

googlecode.com/svn-history/r6/trunk/Bibliographie/breitenstein-detectorconfidencefilter-iccv09.pdf. (Cited on pages 8, 20, 33, 73, and 97.)

William Brendel, Mohamed R. Amer, and Sinisa Todorovic. Multiobject tracking as maximum weight independent set. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, Colorado, 2011. (Cited on page 22.)

Mike Brookes. The matrix reference manual. [online], Imperial College, 2005. URL http://www.psi.toronto.edu/matrix/special.html. (Cited on page 80.)

Yizheng Cai, Nando de Freitas, and James J. Little. Robust visual tracking for multiple targets. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Proceedings of the Ninth European Conference on Computer Vision (ECCV)*, volume 3954 of *Lecture Notes in Computer Science*, pages 107–118. Springer, 2006. ISBN 3-540-33838-1. URL http://www.cs.ubc.ca/~nando/papers/eccv06.pdf. (Cited on page 7.)

Kuo-Chu Chang and Yakov Bar-Shalom. Joint probabilistic data association for multitarget tracking with possibly unresolved measurements and maneuvers. *IEEE Transactions on Automatic Control*, 29 (7):585–594, July 1984. ISSN 0018-9286. (Cited on page xv.)

Datong Chen, Yi Chang, Rong Yan, and Jie Yang. Protecting personal identification in video. In Andrew Senior, editor, *Protecting Privacy in Video Surveillance*, pages 115–128. Springer London, January 2009. ISBN 978-1-84882-300-6, 978-1-84882-301-3. (Cited on page 152.)

Wongun Choi and Silvio Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, volume 6314 of *Lecture Notes in Computer Science*, pages 553–567. Springer, 2010. ISBN 3-642-15560-X, 978-3-642-15560-4. URL http://www.eecs.umich.edu/vision/papers/mtt_wg_eccv2010.pdf. (Cited on pages 25 and 142.)

Wongun Choi and Silvio Savarese. A unified framework for multitarget tracking and collective activity recognition. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7575 of *Lecture Notes in Computer Science*, pages 215–230. Springer, January 2012. ISBN 978-3-642-33764-2, 978-3-642-33765-9. URL http://www-personal.umich.edu/~wgchoi/choi_eccv_12.pdf. (Cited on page 28.)

V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, April 1973. ISSN 0012-365X. URL http://www.sciencedirect.com/science/article/pii/0012365X73901672. (Cited on page 166.)

Cristina Garcia Cifuentes, Marc Sturzel, Frederic Jurie, and Gabriel Brostow. Motion models that only work sometimes. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 55.1–55.12, Surrey, UK, 2012. ISBN 1-901725-46-4. URL http://www.bmva.org/bmvc/2012/BMVC/paper055/index.html. (Cited on page 161.)

Ingemar J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision (IJCV)*, 10(1):53–66, 1993. URL http://webdocs.cs.ualberta.ca/~nray1/CMPUT615/Tracking/data_assoc.pdf. (Cited on page 18.)

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, San Diego, California, June 2005. URL http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf. (Cited on pages xv, 31, 53, and 73.)

Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of the Ninth European Conference on Computer Vision (ECCV)*, volume 3952 of *Lecture Notes in Computer Science*, pages 428–441. Springer, 2006. URL http://lear.inrialpes.fr/pubs/2006/DTS06/eccv2006.pdf. (Cited on page 32.)

George Dantzig. *Linear Programming and Extensions*. Princeton University Press, August 1998. ISBN 0691059136. (Cited on page 21.)

Olivier Debeir, Philippe Van Ham, Robert Kiss, and Christine Decaestecker. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE transactions on medical imaging*, 24(6):697–711, June 2005. ISSN 0278-0062. URL http://lisa.ulb.ac.be/publifiles/56/bscdb_debeir05.pdf. PMID: 15957594. (Cited on page 7.)

Andrew Delong, Anton Osokin, Hossam N. Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision (IJCV)*, 96(1):1–27, January 2012. ISSN 0920-5691. URL http://www.csd.uwo.ca/~yuri/Papers/ijcv10_labelcost.pdf. (Cited on pages iv, vi, 110, 115, 117, and 128.)

Rachid Deriche and Olivier D. Faugeras. Tracking line segments. In *Proceedings of the First European Conference on*

Computer Vision (ECCV), Lecture Notes in Computer Science, pages 259–268. Springer, 1990. ISBN 0-387-52522-X. URL ftp://ftp-sop.inria.fr/odyssee/Publications/1990/deriche-faugeras:90.pdf. (Cited on page 18.)

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000. URL ftp://ftp.idsa.prd.fr/local/aspi/legland/ref/doucet00b.pdf. (Cited on page 18.)

Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001 edition, June 2001. ISBN 0387951466. (Cited on page 8.)

Marcin Eichner and Vittorio Ferrari. We are family: Joint pose estimation of multiple persons. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, volume 6311 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 2010. URL http://www.vision.ee.ethz.ch/publications/papers/proceedings/eth_biwi_00746.pdf. (Cited on page 77.)

Anna Ellis and James Ferryman. PETS2010 and PETS2009 evaluation of results using individual ground truthed single views. In *Advanced Video and Signal Based Surveillance (AVSS)*, 2010. (Cited on pages 147 and 148.)

Markus Enzweiler, Angela Eigenstetter, Bernt Schiele, and Dariu M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, 2010. URL http://www.science.uva.nl/research/isla/downloads/pedestrians/cvpr10_occlusion.pdf. (Cited on page 77.)

Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, June 2008. URL ftp://ftp.vision.ee.ethz.ch/publications/proceedings/eth_biwi_00543.pdf. (Cited on pages 36 and 135.)

Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc van Gool. Robust multiperson tracking from a mobile platform. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(10):1831–1846, October 2009. ISSN 0162-8828. URL http://www.igp.ethz.ch/photogrammetry/publications/pdf_folder/ess09pami.pdf. (Cited on pages 74 and 75.)

Mark Everingham, Luc Van Gool, Chris Williams, John Winn, and Andrew Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. 2012. URL http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. (Cited on pages 147 and 148.)

Mark R. Everingham, Josef Sivic, and Andrew Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 92.1–92.10, Edinburgh, UK, 2006. BMVA Press. ISBN 1-901725-32-4. URL http://www.bmva.org/bmvc/2006/papers/340.pdf. (Cited on page 28.)

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, January 2005. ISSN 0920-5691. URL http://www.cs.cornell.edu/~dph/papers/pict-struct-ijcv.pdf. (Cited on page 22.)

Pedro F. Felzenszwalb, Ross. B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, 2010. URL http://www.ics.uci.edu/~dramanan/papers/tmp.pdf. (Cited on pages xv and 32.)

James Ferryman and Anna Ellis. PETS2010: Dataset and challenge. In *Advanced Video and Signal Based Surveillance (AVSS)*, 2010. (Cited on page 100.)

James Ferryman and Ali Shahrokni. PETS2009: Dataset and challenge. In *11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, December 2009. (Cited on pages 35, 100, and 120.)

Martin A. Fischler and Robert A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1):67–92, January 1973. ISSN 0018-9340. (Cited on pages 22 and 32.)

Matthew Flagg and Jim Rehg. Video-based crowd synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 99(1):1, 2012. ISSN 1077-2626. (Cited on page 161.)

Mary Fletcher, Anna Dornhaus, and Min C. Shin. Multiple ant tracking with global foreground maximization and variable target proposal distribution. In *Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision (WACV)*, WACV '11, pages 570–576, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4244-9496-5. URL http://dx.doi.org/10.1109/WACV.2011.5711555. (Cited on pages 4 and 6.)

François Fleuret, Jérôme Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):1806–1819, February 2008. URL http://cvlab.epfl.ch/publications/publications/2008/FleuretBLF08.pdf. (Cited on pages 34, 59, and 142.)

Thomas E. Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Multi-target tracking using joint probabilistic data association. In *19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, volume 19, pages 807–812, December 1980. (Cited on pages 7, 17, and 18.)

Thomas E. Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983. ISSN 0364-9059. URL http://infoscience.epfl.ch/record/167889/files/Tracking_Multiple_People_under_Global_Appearance_Constraints.pdf. (Cited on page 19.)

Weina Ge and Robert T. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, Leeds, UK, 2008. BMVA Press. URL http://www.bmva.org/bmvc/2008/papers/262.pdf. (Cited on page 85.)

Weina Ge, Robert T. Collins, and Barry Ruback. Vision-based analysis of small groups in pedestrian crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(5):1003–1016, 2012. URL http://dblp.uni-trier.de/rec/bibtex/journals/pami/GeCR12. (Cited on page 28.)

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, June 2012. URL http://www.cvlibs.net/publications/cvpr12.pdf. (Cited on page 147.)

Niloofar Gheissari, Thomas B. Sebastian, and Richard Hartley. Person reidentification using spatiotemporal appearance. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1528–1535, New York, New York, June 2006. (Cited on page 151.)

Ralph Gomory. An algorithm for integer solutions to linear programs. In Robert L. Graves and Philip Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, 1963. (Cited on page 166.)

Jacek Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012. URL http://dblp.uni-trier.de/rec/bibtex/journals/eor/Gondzio12. (Cited on page 52.)

N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, April 1993. ISSN 0956-375X. (Cited on pages 8 and 18.)

Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995. URL http://biomet.oxfordjournals.org/content/82/4/711.abstract. (Cited on page 85.)

Ralph Gross, Latanya Sweeney, Jeffery F. Cohn, Fernando De la Torre, and Simon Baker. Face de-identification. In *Protecting Privacy in Video Surveillance*, pages 129–146. Springer London, 2009. ISBN 978-1-84882-300-6. (Cited on page 152.)

Leon Hempel and Eric Töpfer. CCTV in europe. Final report 15, Centre for Technology and Society, Berlin, August 2004. URL http://www.urbaneye.net/results/ue_wp15.pdf. (Cited on page 151.)

João Henriques, Rui Caseiro, and Jorge Batista. Globally optimal solution to multi-object tracking with merged measurements. In *Proceedings of the Thirteenth IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011. URL http://www.isr.uc.pt/~henriques/publications/henriques_iccv2011.pdf. (Cited on pages 16, 27, 35, 70, and 148.)

Martin Hirzer, Peter M. Roth, and Horst Bischof. Person re-identification by efficient impostor-based metric learning. *Advanced Video and Signal Based Surveillance, IEEE Conference on*, 0:203–208, 2012. URL http://lrs.icg.tugraz.at/pubs/hirzer_avss_2012.pdf. (Cited on page 151.)

Derek Hoiem, Carsten Rother, and John M. Winn. 3D LayoutCRF for multi-view object class recognition and segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, June 2007. URL http://research.microsoft.com/pubs/78792/3dLayoutCRF_Hoiem_Rother_Winn_CVPR2007.pdf. (Cited on page 117.)

Esther Horbert, Konstantinos Rematas, and Bastian Leibe. Level-set person segmentation and tracking with multi-region appearance models and top-down shape information. In *Proceedings of the Thirteenth IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011. URL http://www.vision.rwth-aachen.de/

publications/pdf/horbert-person-segm-iccv11.pdf. (Cited on pages 23 and 142.)

Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Proceedings of the Tenth European Conference on Computer Vision (ECCV)*, volume 5303 of *Lecture Notes in Computer Science*, pages 788–801. Springer, 2008. ISBN 978-3-540-88685-3. URL http://iris.usc.edu/Outlines/papers/2008/huang-wu-nevatia-eccv08.pdf. (Cited on pages 21 and 160.)

Haroon Idrees, Imran Saleemi, and Mubarak Shah. Statistical inference of motion in the invisible. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7575 of *Lecture Notes in Computer Science*, pages 544–557. Springer, 2012. ISBN 978-3-642-33764-2. URL http://vision.eecs.ucf.edu/papers/eccv2012/hiddenV11r.pdf. (Cited on page 27.)

Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision (IJCV)*, 97(2):123–147, April 2012. ISSN 0920-5691, 1573-1405. URL http://www.csd.uwo.ca/~yuri/Papers/tr735.pdf. (Cited on page 117.)

Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28, 1998. ISSN 0920-5691. URL https://www.cs.duke.edu/courses/cps296.1/spring05/handouts/IsardBlake1998.pdf. (Cited on pages 18 and 20.)

Hamid Izadinia, Imran Saleemi, Wenhui Li, and Mubarak Shah. (MP)2T: Multiple people multiple parts tracker. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7577 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 2012. URL http://www.cs.ucf.edu/~izadinia/files/MPMPT-ECCV12.pdf. (Cited on page 33.)

E. T. Jaynes. Information theory and statistical mechanics. *The Physical Review*, 106(4):620–630, 1957. URL http://bayes.wustl.edu/etj/articles/theory.1.pdf. (Cited on page 10.)

Hao Jiang, Sidney Fels, and James J. Little. A linear programming approach for multiple object tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, June 2007. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.1741&rep=rep1&type=pdf. (Cited on pages 3, 22, 49, 56, and 70.)

Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *International Symposium on Aerospace andDefense Sensing, Simulation and Controls*, pages 182–193, 1997. URL http://www.control.auc.dk/~tb/ESIF/julier97new.pdf. (Cited on page 17.)

Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. P-N Learning: Bootstrapping binary classifiers from unlabeled data by structural constraint. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, 2010. URL http://eprints.pascal-network.org/archive/00006951/01/cvpr2010.pdf. (Cited on page 2.)

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. URL http://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf. (Cited on pages 8 and 17.)

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, STOC '84, pages 302–311, New York, NY, USA, 1984. ACM. ISBN 0-89791-133-4. URL http://retis.sssup.it/~bini/teaching/optim2010/karmarkar.pdf. (Cited on page 21.)

Robert Kaucic, A. G. Amitha Perera, Glen Brooksby, John P. Kaufhold, and Anthony Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 990–997, San Diego, California, June 2005. (Cited on pages 21, 26, and 30.)

Bernhard. X. Kausler, Martin Schiegg, Bjoern Andres, Martin Lindner, Heike Leitte, Lars Hufnagel, Ulrich Koethe, and Fred. A. Hamprecht. A discrete chain graph model for 3d+t cell tracking with high misdetection robustness. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7574 of *Lecture Notes in Computer Science*, pages 144–157, 2012. URL http://hci.iwr.uni-heidelberg.de/publications/mip/techrep/kausler_12_discrete.pdf. (Cited on page 7.)

Philipp J. Keller, Annette D. Schmidt, Joachim Wittbrodt, and Ernst H. K. Stelzer. Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. *Science*, 322(5904): 1065–1069, November 2008. ISSN 0036-8075, 1095-9203. URL http://www.sciencemag.org/content/322/5904/1065. (Cited on page 4.)

Zia Khan, Tucker Balch, and Frank Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence (PAMI), 27 (11):1805–1918, November 2005. ISSN 0162-8828. URL http://home.uchicago.edu/~zia/papers/pami05.pdf. (Cited on page 24.)

Zia Khan, Tucker Balch, and Frank Dellaert. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(12):1960–1972, December 2006. ISSN 0162-8828. URL http://home.uchicago.edu/~zia/papers/pami05cvpr05.pdf. PMID: 17108370. (Cited on pages 4, 6, and 24.)

Kris M. Kitani, Brian D. Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7575 of *Lecture Notes in Computer Science*, pages 201–214. Springer, 2012. ISBN 978-3-642-33764-2, 978-3-642-33765-9. URL http://www.cs.cmu.edu/~kkitani/ActivityForecasting_files/Kitani-ECCV2012.pdf. (Cited on page 28.)

Daniel A. Kluepfel. The behavior and tracking of bacteria in the rhizosphere. *Annual Review of Phytopathology*, 31(1):441–472, 1993. URL http://www.annualreviews.org/doi/abs/10.1146/annurev.py.31.090193.002301. (Cited on page 7.)

Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(10):1568–1583, October 2006. ISSN 0162-8828. URL http://pub.ist.ac.at/~vnk/papers/TRW-S-PAMI.pdf. (Cited on page 129.)

Vladimir Kolmogorov and Carsten Rother. Comparison of energy minimization algorithms for highly connected graphs. In *Proceedings of the Ninth European Conference on Computer Vision (ECCV)*, volume 3952 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2006. ISBN 978-3-540-33834-5, 978-3-540-33835-2. URL http://pub.ist.ac.at/~vnk/papers/KR-ECCV06.pdf. (Cited on page 89.)

Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(2):147–159, February 2004. ISSN 0162-8828. URL http://research.microsoft.com/pubs/67377/kolmogorov-energy-func-pami-04.pdf. (Cited on pages 10 and 114.)

Cheng-Hao Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, Colorado, June

2011. URL http://iris.usc.edu/people/chenghak/download/Kuo_Nevatia_CVPR2011.pdf. (Cited on pages xvi and 135.)

Cheng-Hao Kuo, Chang Huang, and Ram Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, 2010. (Cited on page 81.)

Junseok Kwon and Kyoung Mu Lee. Tracking by sampling trackers. In *Proceedings of the Thirteenth IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011. URL http://cv.snu.ac.kr/research/~vts/. (Cited on page 2.)

Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H. S. Torr. Graph cut based inference with co-occurrence statistics. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, volume 6315, pages 239–253. Springer, 2010. ISBN 3-642-15554-5, 978-3-642-15554-3. URL http://www.robots.ox.ac.uk/~lubor/eccv10co.pdf. (Cited on page 126.)

Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, May 1998. ISSN 1052-6234. (Cited on page 130.)

L. D. Landau and E. M. Lifshitz. *Statistical Physics: 5*, volume 5 of *Course of Theoretical Physics*. Pergamon Press, Oxford, 2 edition, 1969. ISBN 0-08-009103-2. (Cited on page 10.)

Bastian Leibe, Konrad Schindler, and Luc Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Proceedings of the Eleventh IEEE International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, October 2007. (Cited on pages 23, 33, 72, and 109.)

Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision (IJCV)*, 77(1-3):259–289, May 2008a. ISSN 0920-5691. URL http://dx.doi.org/10.1007/s11263-007-0095-3. (Cited on pages xv and 33.)

Bastian Leibe, Konrad Schindler, Nico Cornelis, and Luc Van Gool. Coupled detection and tracking from static cameras and moving vehicles. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(10):1683–1698, October 2008b. URL http://www.igp.ethz.ch/photogrammetry/publications/pdf_folder/leibe08pami.pdf. (Cited on page 161.)

Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1324–1332, 2010. URL http://www.robots.ox.ac.uk/~vilem/NIPS2010.pdf. (Cited on pages 27 and 97.)

Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, June 2009. URL http://iris.usc.edu/Vision-Users/Oldusers/yli8/papers/tracking_cvpr09.pdf. (Cited on pages 21, 42, and 146.)

Jia Liu, Xiaofeng Tong, Wenlong Li, Tao Wang, Yimin Zhang, and Hongqi Wang. Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, 30(2):103–113, January 2009. ISSN 0167-8655. URL http://www.comp.leeds.ac.uk/bmvc2008/proceedings/2007/papers/paper-70.pdf. (Cited on page 7.)

Ye Liu, Hui Li, and Yan Qiu Chen. Automatic tracking of a large number of moving targets in 3D. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7575 of *Lecture Notes in Computer Science*, pages 730–742. Springer, 2012. ISBN 978-3-642-33764-2, 978-3-642-33765-9. (Cited on page 6.)

Xinghua Lou and Fred A. Hamprecht. Structured learning for cell tracking. In *Advances in Neural Information Processing Systems (NIPS)*, volume 24, pages 1296–1304, 2011. URL http://hci.iwr.uni-heidelberg.de/Staff/xlou/publications/lou_11_structured.pdf. (Cited on page 7.)

Boris Meden, Frédéric Lerasle, and Patrick Sayd. Mcmc supervision for people re-identification in nonoverlapping cameras. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 66.1–66.11. BMVA Press, 2012. ISBN 1-901725-46-4. (Cited on page 27.)

Talya Meltzer, Chen Yanover, and Yair Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV)*, pages 428–435, Beijing, China, October 2005. ISBN 0-7695-2334-X-01. URL http://dx.doi.org/10.1109/ICCV.2005.110. (Cited on page 89.)

Lee Middleton and Jayanthi Sivaswamy. *Hexagonal Image Processing: A Practical Approach*. Springer, 2005 edition, July 2005. ISBN 1852339144. (Cited on page 57.)

Anton Milan, Konrad Schindler, and Stefan Roth. Challenges of ground truth evaluation of multi-target tracking. In *Proceedings of*

the CVPR 2013 Workshop on Ground Truth - What is a good dataset?, Portland, Oregon, June 2013a. (Cited on page 13.)

Anton Milan, Konrad Schindler, and Stefan Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, Oregon, June 2013b. (Cited on pages 13 and 109.)

Erik G. Miller. Alternative tilings for improved surface area estimates by local counting algorithms. *Computer Vision and Image Understanding (CVIU)*, 74(3):193–211, June 1999. ISSN 1077-3142. URL http://dx.doi.org/10.1006/cviu.1999.0754. (Cited on page 57.)

John E. Mitchell. Branch-and-cut algorithms for combinatorial optimization problems. In *Handbook of Applied Optimization*, pages 65–77. Oxford University Press, 2002. ISBN 0-19-512594-0. URL http://homepages.rpi.edu/~mitchj/papers/bc_hao.pdf. (Cited on page 166.)

Dennis Mitzel and Bastian Leibe. Taking mobile multi-object tracking to the next level: People, unknown objects, and carried items. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7576 of *Lecture Notes in Computer Science*, pages 566–579. Springer, 2012. ISBN 978-3-642-33714-7. URL http://www.mmp.rwth-aachen.de/publications/pdf/mitzel-eccv12.pdf. (Cited on page 24.)

Dennis Mitzel, Esther Horbert, Andreas Ess, and Bastian Leibe. Multi-person tracking with sparse detection and continuous segmentation. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, volume 6311, pages 397–410. Springer, 2010. URL http://www.mmp.rwth-aachen.de/publications/pdf/eccv2010-dennis.pdf. (Cited on pages 23 and 142.)

Charles Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, 22(3):302–312, June 1977. ISSN 0018-9286. (Cited on pages 7, 17, 21, 22, and 156.)

David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989. (Cited on page 10.)

Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. Stacked hierarchical labeling. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, volume 6316 of *Lecture Notes in Computer*

Science, pages 57–70. Springer, 2010. ISBN 3-642-15566-9, 978-3-642-15566-6. URL http://www.ri.cmu.edu/pub_files/2010/9/munoz_eccv_10.pdf. (Cited on page 28.)

J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965. (Cited on page 130.)

Elaine M. Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):232–243, 2005. ISSN 1041-4347. URL http://arbor.ee.ntu.edu.tw/archive/ppdm/Heuristics/NewtonPP05.pdf. (Cited on page 152.)

Peter Nillius, Josephine Sullivan, and Stefan Carlsson. Multi-target tracking - linking identities using Bayesian network inference. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2187–2194, June 2006. URL http://www.nada.kth.se/~sullivan/Papers/406_sullivan_j.pdf. (Cited on page 26.)

Clive Norris and Gary Armstrong. CCTV and the social structuring of surveillance. In *Surveillance of Public Space*, volume 10 of *Crime Prevention Studies*, pages 157–178. Criminal Justice Press, 1999. ISBN 978-1-881798-22-4. (Cited on page 151.)

Songhwai Oh, Stuart Russell, and Shuai Sastry. Markov chain Monte Carlo data association for general multiple-target tracking problems. In *43rd IEEE Conference on Decision and Control (CDC)*, volume 1, pages 735–742, December 2004. (Cited on page 24.)

Kenji Okuma, Ali Taleghani, O. De Freitas, James J. Little, and David G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proceedings of the Eighth European Conference on Computer Vision (ECCV)*, volume 3021 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2004. ISBN 3-540-21984-6. URL www.cs.ubc.ca/~little/links/linked-papers/kenji-eccv2004.pdf. (Cited on pages xv, 20, 101, and 102.)

Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 261–268, June 2009. URL http://vision.cse.psu.edu/courses/Tracking/vlpr12/PellegriniNeverWalkAlone.pdf. (Cited on page 28.)

A. G. Amitha Perera, Chukka Srinivas, Anthony Hoogs, Glen Brooksby, and Wensheng Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Proceedings*

of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 666–673, 2006. (Cited on page 26.)

Coretta Phillips. A review of CCVT evaluations: crime reduction effects and attitudes towards its use. In *Surveillance of Public Space*, Crime Prevention Studies, pages 123–155. Criminal Justice Press, 1999. ISBN 978-1-881798-22-4. (Cited on page 151.)

Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, Colorado, 2011. URL http://www.ics.uci.edu/~hpirsiav/papers/tracking_cvpr11. (Cited on pages 3, 22, 70, 118, 135, and 148.)

Zenon W. Pylyshyn and Ron W. Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial vision*, 3(3):179–197, 1988. URL http://www.ingentaconnect.com/content/vsp/spv/1988/00000003/00000003/art00003. (Cited on page 16.)

Zhen Qin and Christian R. Shelton. Improving multi-target tracking via social grouping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, June 2012. (Cited on page 160.)

Faisal Z. Qureshi. Object-video streams for preserving privacy in video surveillance. In *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 442–447, September 2009. URL http://faculty.uoit.ca/qureshi/pubs/13-object-video-streams-ch.pdf. (Cited on page 152.)

Donald B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979. ISSN 0018-9286. (Cited on pages 7, 17, 18, and 156.)

Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *Proceedings of the Thirteenth IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011. URL http://www.di.ens.fr/willow/pdfscurrent/rodriguez11b.pdf. (Cited on pages 27, 72, and 96.)

Carsten Rother, Vladimir Kolmogorov, Victor Lempitsky, and Martin Szummer. Optimizing binary MRFs via extended roof duality. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007. URL http://www.robots.ox.ac.uk/~vilem/QPBOPI.pdf. (Cited on page 51.)

Jeremy Schiff, Marci Meingast, Deirdre K. Mulligan, Shankar Sastry, and Ken Goldberg. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In Andrew Senior, editor, *Protecting Privacy in Video Surveillance*, pages 65–89. Springer London, January 2009. ISBN 978-1-84882-300-6, 978-1-84882-301-3. URL http://goldberg.berkeley.edu/pubs/respectful-cameras-book-chapter-F08.pdf. (Cited on page 152.)

Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 519–528, New York, New York, June 2006. ISBN 0-7695-2597-0. (Cited on page 147.)

Andrew Senior, Sharath Pankanti, Arun Hampapur, Lisa Brown, Ying-Li Tian, Ahmet Ekin, Jonathan Connell, Chiao Fe Shu, and Max Lu. Enabling video privacy through computer vision. *IEEE Security and Privacy*, 3(3):50–57, May 2005. ISSN 1540-7993. (Cited on page 152.)

Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, Seattle, Washington, June 1994. (Cited on page 28.)

Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. Part-based multiple-person tracking with partial occlusion handling. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1815–1821, Providence, Rhode Island, June 2012. URL http://crcv.ucf.edu/papers/1439.pdf. (Cited on pages 33 and 142.)

Leonid Sigal and Michael J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2048, New York, New York, 2006. (Cited on page 77.)

Kevin Smith, Daniel Gatica-Perez, Jean-Marc Odobez, and Sileye Ba. Evaluating multi-object tracking. In *Workshop on Empirical Evaluation Methods in Computer Vision (EEMCV)*, San Diego, California, June 2005. (Cited on page 42.)

Torsten Spindler, Christoph Wartmann, Daniel Roth, Andreas Steffen, Ludger Hovestadt, and Luc van Gool. Privacy in video surveilled areas. In *International Conference on Privacy, Security and Trust (PST 2006)*, October 2006. (Cited on page 152.)

Chris Stauffer and W. Eric L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2246–2252, Fort Collins, Colorado, 1999. URL http://www.ai.mit.edu/projects/vsam/Publications/stauffer_cvpr98_track.pdf. (Cited on page 30.)

Rainer Stiefelhagen, Keni Bernardin, Rachel Bowers, John S. Garofolo, Djamel Mostefa, and Padmanabhan Soundararajan. The clear 2006 evaluation. In *CLEAR*, 2006. (Cited on pages 146 and 147.)

Patrick Storms and Frits Spieksma. An LP-based algorithm for the data association problem in multitarget tracking. In *Proceedings of the Third International Conference on Information Fusion*, volume 1, pages TUD2/10 – TUD2/16, July 2000. URL http://www.isif.org/fusion/proceedings/fusion00CD/fusion2000/papers/TuD2-2-PatrickStorms038.pdf. (Cited on pages 21 and 156.)

Andrew D. Straw, Kristin Branson, Titus R. Neumann, and Michael H. Dickinson. Multi-camera real-time three-dimensional tracking of multiple flying animals. *Journal of the Royal Society Interface the Royal Society*, 8(56):395–409, 2011. ISSN 17425662. (Cited on page 6.)

Min Sun, Murali Telaprolu, Honglak Lee, and Silvio Savarese. An efficient branch-and-bound algorithm for optimal human pose estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, June 2012. (Cited on page 159.)

Shuai Tao, Mineichi Kudo, and Hidetoshi Nonaka. Privacy-preserved behavior analysis and fall detection by an infrared ceiling sensor network. *Sensors*, 12(12):16920–16936, December 2012. ISSN 1424-8220. URL http://www.mdpi.com/1424-8220/12/12/16920. (Cited on page 152.)

Marshall F. Tappen and William T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)*, Nice, France, October 2003. ISBN 0-7695-1950-4. URL http://dl.acm.org/citation.cfm?id=946247.946707. (Cited on page 89.)

Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991. (Cited on page xv.)

Ákos Utasi and Csaba Benedek. A multi-view annotation tool for people detection evaluation. In *Proceedings of the First International*

Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications, Capri, Italy, May 2012. (Cited on page 141.)

Jaco Vermaak, Arnaud Doucet, and Patrick Pérez. Maintaining multimodality through mixture tracking. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)*, pages 1110–1116, Nice, France, October 2003. ISBN 0-7695-1950-4. URL http://dl.acm.org/citation.cfm?id=946247.946684. (Cited on page 20.)

Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision (IJCV)*, 101(1):184–204, January 2013. ISSN 0920-5691. (Cited on page 141.)

Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. New features and insights for pedestrian detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, June 2010a. (Cited on pages 32, 73, and 98.)

Stefan Walk, Konrad Schindler, and Bernt Schiele. Disparity statistics for pedestrian detection: Combining appearance, motion and stereo. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, volume 6316 of *Lecture Notes in Computer Science*, pages 182–195. Springer, 2010b. (Cited on page 32.)

Brandon P. Welsh and David P. Farrington. Effects of closed circuit television surveillance on crime. Campbell systematic reviews, The Campbell Collaboration, December 2008. (Cited on page 151.)

Christian Wojek, Stefan Roth, Konrad Schindler, and Bernt Schiele. Monocular 3D scene modeling and inference: Understanding multi-object traffic scenes. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, volume 6314 of *Lecture Notes in Computer Science*, pages 467–481. Springer, 2010. (Cited on page 25.)

Christian Wojek, Stefan Walk, Stefan Roth, and Bernt Schiele. Monocular 3D scene understanding with explicit occlusion reasoning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1993–2000, Colorado Springs, Colorado, June 2011. (Cited on page 77.)

J.K. Wolf, A.M. Viterbi, and G.S. Dixon. Finding the best set of k paths through a trellis with application to multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 25(2):287–296, March 1989. ISSN 0018-9251. (Cited on page 25.)

Bo Wu and Ram Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of

edgelet part detectors. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, October 2005. URL http://iris.usc.edu/outlines/papers/2005/wu-nevatia-iccv.pdf. (Cited on pages 21 and 77.)

Bo Wu and Ram Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958, New York, New York, 2006. ISBN 0-7695-2597-0. URL http://iris.usc.edu/outlines/papers/2006/wu-nevatia-cvpr06.pdf. (Cited on page 42.)

Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision (IJCV)*, 75(2):247–266, November 2007. ISSN 0920-5691, 1573-1405. URL http://imageprocessinggroup.googlecode.com/svn/trunk/STICKTEMP/faceDetection/wu-nevatia-ijcv07.pdf. (Cited on page 21.)

Zheng Wu, Thomas H. Kunz, and Margrit Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, Colorado, 2011. URL http://cs-people.bu.edu/wuzheng/research/publication/CVPR2011.pdf. (Cited on pages 85 and 109.)

Zheng Wu, Ashwin Thangali, Stan Sclaroff, and Margrit Betke. Coupling detection and data association for multiple object tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, 2012. URL http://cs-people.bu.edu/wuzheng/research/publication/CVPR2012.pdf. (Cited on pages 23 and 161.)

Jun Xie, Shahid Khan, and Mubarak Shah. Automatic tracking of Escherichia coli bacteria. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 11(Pt 1): 824–832, 2008. PMID: 18979822. (Cited on page 7.)

Xu Yan, Xuqing Wu, Ioannis A. Kakadiaris, and Shishir K. Shah. To track or to detect? An ensemble framework for optimal selection. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7576 of *Lecture Notes in Computer Science*, pages 594–607. Springer, 2012. ISBN 978-3-642-33714-7, 978-3-642-33715-4. URL http://link.springer.com/chapter/10.1007/978-3-642-33715-4_43. (Cited on pages 137 and 161.)

Bo Yang and Ram Nevatia. An online learned CRF model for multi-target tracking. In *Proceedings of the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 2034–2041, Providence, Rhode Island, 2012a. URL http://iris.usc.edu/outlines/papers/2012/yang-nevatia-cvpr-2-2012.pdf. (Cited on pages 21, 132, 135, 143, and 147.)

Bo Yang and Ram Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7572 of *Lecture Notes in Computer Science*, pages 484–498. Springer, 2012b. ISBN 978-3-642-33717-8. URL http://iris.usc.edu/Outlines/papers/2012/Yang_Nevatia_ECCV12.pdf. (Cited on page 159.)

Qian Yu, Gérard G. Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, June 2007. URL http://crue.isi.edu/muri/papers/USC/Medioni/trackingcvpr07.pdf. (Cited on page 24.)

Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, volume 7573 of *Lecture Notes in Computer Science*, pages 343–356. Springer, 2012. ISBN 978-3-642-33708-6, 978-3-642-33709-3. URL http://crcv.ucf.edu/papers/eccv2012/GMCP-Tracker_ECCV12.pdf. (Cited on page 22.)

Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, June 2008. URL http://www.cs.bu.edu/fac/betke/papers/WuKunzBetke-CVPR2011.pdf. (Cited on pages 3, 22, 23, 49, 70, 72, 81, 109, 137, and 156.)

Daniel Zoran and Yair Weiss. Scale invariance and noise in natural images. In *Proceedings of the Twelfth IEEE International Conference on Computer Vision (ICCV)*, pages 2209–2216, Kyoto, Japan, October 2009. ISBN 978-1-4244-4420-5. URL http://www.cs.huji.ac.il/~yweiss/zoranweiss09.pdf. (Cited on page 140.)

CURRICULUM VITÆ

ANTON MILAN

Date of birth: May 1st, 1983
Place of birth: Kiev, Ukraine

Education	2009 – 2013	*TU Darmstadt, Germany*
		PhD student in computer science
	2007 – 2008	*Universitat Politècnica de València, Spain*
		Visiting student
	2003 – 2008	*Universität Bonn, Germany*
		Diplom in computer science
	2000 – 2001	*Edward R. Murrow High School, Brooklyn, NY, USA*
		Visiting student
	1995 – 2003	*Heinrich-Mann-Gymnasium, Cologne, Germany*

Positions	2010 – 2013	*TU Darmstadt, Germany*
		Visual Inference group of Prof. Stefan Roth
		Research and teaching assistant
	2009 – 2010	*TU Darmstadt, Germany*
		Image Understanding group of Prof. Konrad Schindler
		Research assistant
	2008 – 2009	*Luminova Technologies, Melbourne, Australia*
		Shader developer
	2005 – 2008	*Universität Bonn, Germany*
		Student assistant

PUBLICATIONS

Anton Milan, Konrad Schindler, and Stefan Roth
Detection- and Trajectory-Level Exclusion in Multiple Object Tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, Oregon, June 2013.

Anton Milan, Konrad Schindler, and Stefan Roth
Challenges of Ground Truth Evaluation of Multi-Target Tracking. In *Proceedings of the IEEE CVPR 2013 Workshop on Ground Truth - What is a good dataset?*, Portland, Oregon, June 2013.

Anton Milan, Stefan Roth, and Konrad Schindler
Continuous Energy Minimization for Multi-Target Tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, accepted.

Anton Andriyenko[1] , Konrad Schindler, and Stefan Roth
Discrete-Continuous Optimization for Multi-Target Tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, June 2012.

Anton Andriyenko, Stefan Roth, and Konrad Schindler
An Analytical Formulation of Global Occlusion Reasoning for Multi-Target Tracking.
In *11th International IEEE Workshop on Visual Surveillance (ICCV Workshops)*, Barcelona, Spain, November 2011.

Anton Andriyenko and Konrad Schindler
Multi-target Tracking by Continuous Energy Minimization.
In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, Colorado, June 2011.

Anton Andriyenko and Konrad Schindler
Globally Optimal Multi-target Tracking on a Hexagonal Lattice.
In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, **6311**, pages 466–479, Lecture Notes in Computer Science, 2010. Springer.

1 My surname changed to *Milan* in May 2013.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/