Diss. ETH No. 20340

Modeling and Tracking Social Walkers

A dissertation submitted to the
ETH ZURICH

for the degree of
Doctor of Science ETH

presented by

Stefano Pellegrini
M.Eng. Sapienza University of Rome
born 17. December 1981
citizen of Italy

Examination Committee

Prof. Dr. Luc Van Gool, ETH Zurich and KU Leuven, examiner
Dr. Julien Pettré, INRIA-Rennes, co-examiner

2012

To my grandparents.

Abstract

Human motion behavior has been a popular topic in the social sciences since the 1960ies. Application such as evacuation simulators and autonomous character animation, soon raised the interest of other communities for the topic. As a consequence, the focus was not anymore only on the analysis of human motion, but on the synthesis as well. In recent years other fields, such as computer vision and robotics, have been increasingly more interested in exploiting human motion models for applications such as tracking, scene understanding and robot path-planning. In these days, visual trackers and motion capture technology have offered the possibility of both calibrating and validating the motion model with real data.

This dissertation investigates the importance of motion priors for applications such as tracking, trajectory prediction and crowd simulation.
We first introduce a steering model that accounts for the anticipation with which humans avoid obstacles. The model is goal driven and, together with the avoidance component, it comprises of a grouping component to represent people walking together. The model is then used in a scene dependent crowd simulation application. Here we propose to use computer vision techniques to reduce the amount of work necessary to simulate a specific scene. Furthermore, in order to reproduce the motion patterns of the scene, we extend the steering model with a probabilistic goal selection layer. We finally propose two strategies for mixing real and virtual pedestrians in our simulations. We validate our model both on synthetic data and through a user study.

The steering model is then added in a multi-target tracking application, to provide it with a more informative motion prior. The benefit of this combined system becomes evident when the observation model is less reliable, such as during periods of long target occlusion. We further extend the model in a multiple hypotheses way, and we apply it to the task of trajectory prediction. Here we show that the resulting system achieves robust results in predicting human trajectories within a time horizon of about five seconds in a busy scene.

The training and validation of the motion model of this dissertation requires the use of real trajectories. In order to extract them, we propose a novel multi-target tracker that exploits several motion priors, such as avoidance and grouping. We build the model within the framework of Conditional Random Fields, and we develop an efficient inference strategy to cope with the resulting model complexity.

The algorithms that we present in this dissertations have been evaluated on challenging sequences. The experiments show that we can reproduce naturally behaving motion and we can furthermore effectively exploit motion priors for tracking and trajectory prediction.

Sommario

I modelli del movimento umano sono stati un argomento di particolare interesse nelle scienze sociali fin dagli anni 60. Ben presto, l'interesse si diffuse in altre comunità, anche a causa di applicazioni come i simulatori delle situazioni di emergenza o l'animazione di personaggi virtuali. Di conseguenza l'attenzione non si concentrava più unicamente sull'analisi del movimento, ma anche sulla sua sintesi. Di recente altre comunità scientifiche si sono interessate all'argomento. L'area di visione computazionale e quella di robotica sono due esempi, dove questi modelli sono stati utilizzati in applicazioni quali il tracciamento di persone, l'analisi di alcune proprietà delle scene o per la pianificazione delle traiettorie dei robot. La disponibilità di metodi, come i dispositivi di motion capture e soprattutto i metodi di tracciamento basati su videocamere, ha reso possibile la calibrazione e la validazione di questi modelli con l'uso di dati reali.

Questa tesi si propone di studiare l'importanza di questi modelli del movimento umano per applicazioni quali il tracciamento, la predizione di traiettorie e la simulazione di persone.
Per prima cosa descriviamo un modello del movimento che simula come le persone anticipano la collisione con gli ostacoli. Il modello guida il soggetto verso il suo obbiettivo, considerando gli ostacoli da evitare e l'interazione con gli altri, eventuali, soggetti appartenenti allo stesso gruppo. Il modello è quindi utilizzato per simulare le persone nel contesto di alcune specifiche scene. Uno degli obbiettivi è quello di ridurre al minimo il lavoro manuale necessario per simulare una particolare scena. A tale scopo, il modello è esteso per permettere di rappresentare la scelta delle azioni di movimento da parte del soggetto. Nell'applicazione che presentiamo, siamo in grado di simulare allo stesso tempo soggetti reali e soggetti virtuali. I risultati sono utilizzati in uno studio condotto su un gruppo di volontari, che ne valuta il realismo.

Il modello del movimento è quindi utilizzato per il tracciamento delle persone. In particolare, il modello è utilizzato nella fase di predizione del sistema di tracciamento. Qui mostriamo come i benefici più significativi si hanno quando i dati osservati non sono abbastanza affidabili, come ad esempio accade quando il soggetto tracciato è temporaneamente occluso. Inoltre mostriamo come sia possibile estendere il modello probabilisticamente per ottenere una robusta predizione delle traiettorie. Il sistema è in grado di predire diverse possibili traiettorie per ogni soggetto. Questo rende possibile predire le traiettorie con una certa affidabilità all'interno di un orizzonte temporale di circa cinque secondi.

La calibrazione e la validazione del modello del movimento richiede l'utilizzo di dati reali. Per estrarre questi dati, proponiamo un nuovo sistema di tracciamento che sfrutta caratteristiche del movimento dei soggetti. Il modello del tracciamento utilizza i Conditional Random Field. Sviluppiamo a tal proposito una strategia specifica per la risoluzione del problema in questo contesto.

Gli algoritmi che presentiamo in questa tesi sono testati su delle sequenze di immagini che offrono diverse difficoltà per l'analisi. Gli esperimenti mostrano che siamo in grado di analizzare il movimento delle persone, come mostriamo nelle applicazioni di tracciamento e predizione delle traiettorie, e sintetizzare lo stesso, come mostriamo nelle simulazioni di alcune scene.

Acknowledgements

First of all, I would like to thank Prof. Dr. Luc Van Gool for giving me the opportunity of being a Ph.D. student here at the ETH and for inspiring and supporting my work during these years.

Most of the work contained in this thesis has been possible only because of the help of the great people that I had the luck to work with. I want to thank all of them, and in particular Dr. Andreas Ess, Prof. Dr. Konrad Schindler and Dr. Jürgen Gall. I also want to thank Prof. Dr. Peter Gehler, Dr. Thomas Deselaers and Prof. Dr. Vittorio Ferrari for their extremely precious advice.

In these years I have been supported by several projects. I would like therefore to thank all the project partners at Toyota, IM2 and Tango. In particular I would like to thank Dr. Julien Pettré for the research insights and for agreeing to co-examine this thesis.

But the days at ETH would have not been that fun without the BIWI crew. Table soccer meetings, lunch and coffee breaks, reading club sessions and all the after work time spent in these years has been great because of them. In particular I want to thank Gabriele Fanelli, Alex Mansfield, Tobias Gass, Angela Yao, Wicher Visser, Fabian Nater, Severin Stalder, Mukta Prasad, Andrea Fossati, Orçun Göksel, Alan Lehmann, Michael Breitenstein, Nima Razavi, Golnoosh Samei, Stephan Gammeter, Valeria De Luca, Bryn Lloyd, Henning Hamer, Thibaut Weise. Many thanks also the the staff of the Sekretariat BIWI: Barara Widmer, Christina Krueger and Fiona Matthews.

As for everything else I do, I am truly and deeply thankful to my family for all the support they give me and for recharging me at every stay with them.

Many other people helped me through this wonderful experience. My friends, the many interesting people that I met at conferences, the people I met and worked with at Google during my internship. All of them contributed, in a way or another, to this work. I want to express my gratitude to each of them.

Last, but certainly not least, I owe a big, very big, “thank you” to Jasmin, bearing with me from the very first day of this experience to these last days.

Contents

1 Introduction 1
1.1 Contributions . . . 3
1.2 Organization . . . 4

2 Linear Trajectory Avoidance 7
2.1 Related Work . . . 7
2.2 Steering Model . . . 10
2.2.1 Model Variables . . . 10
2.2.2 Visibility . . . 11
2.2.3 Avoidance Behavior . . . 11
2.2.4 Groups . . . 14
2.2.5 Destinations . . . 16
2.2.6 Model Summary . . . 17
2.3 Static Obstacles . . . 18
2.4 Conclusions . . . 19

3 Scene-dependent Crowd Simulation 21
3.1 Introduction . . . 21
3.2 Related Work . . . 22
3.3 Scene Transitions . . . 24
3.3.1 Segmenting the Trajectories . . . 25
3.3.2 Destination Flow . . . 27
3.3.3 A Clustering Variant . . . 28
3.3.4 Simulation . . . 29
3.4 Mixed Reality . . . 30
3.5 Experiments . . . 32
3.5.1 The Circle Experiment . . . 32
3.5.2 User Study . . . 34
3.5.3 Mixing Real and Simulated People . . . 38

3.5.4 Timings and Limitations . . . 40
3.6 Conclusions . . . 43

4 Tracking with LTA 45
4.1 Introduction . . . 45
4.2 Related Work . . . 46
4.3 Model Setup . . . 53
4.4 Training . . . 53
4.5 Results . . . 55
4.5.1 Prediction . . . 55
4.5.2 Patch-based Tracking . . . 57
4.5.3 Tracking with a Moving Observer . . . 60
4.6 Conclusion . . . 63

5 Stochastic LTA for Prediction 65
5.1 Introduction . . . 65
5.2 Stochastic LTA . . . 66
5.2.1 Why not a Particle Filter Framework? . . . 71
5.2.2 Training . . . 72
5.3 Experiments . . . 73
5.3.1 Prediction . . . 73
5.3.2 Tracking . . . 76
5.4 Discussion and Conclusions . . . 79

6 Towards Joint Grouping and Tracking 81
6.1 Introduction . . . 81
6.2 Group Classification . . . 82
6.3 Group CRF . . . 83
6.4 Learning the parameters . . . 85
6.4.1 Dataset . . . 86
6.4.2 Independent Motion and Appearance . . . 86
6.4.3 Grouping Relations . . . 88
6.4.4 Transitivity Constraints . . . 89
6.5 Inference . . . 89
6.6 Experiments . . . 91
6.6.1 Model Construction . . . 91
6.6.2 Ground-truth . . . 92
6.6.3 Detector . . . 93

6.7 Conclusions 94

7 Tracking with Interactions 97

7.1 Introduction 97
7.2 Model 99
  7.2.1 Model Variables 100
  7.2.2 Motion Model 101
  7.2.3 Observation Model 102
  7.2.4 Interaction Model 102
  7.2.5 Connectivity 104
7.3 Inference 105
  7.3.1 Belief Propagation with Continuous Variables . . . 105
  7.3.2 Sampling Strategies 109
  7.3.3 Splitting the Inference 113
  7.3.4 External Modules 114
  7.3.5 Implementation Remarks 116
7.4 Experiments 117
  7.4.1 Datasets and Setup 117
  7.4.2 Grouping Influence 119
  7.4.3 Group Formation and Splitting 120
  7.4.4 Avoidance 122
  7.4.5 Quantitative Results 123
7.5 Conclusions 127

8 Conclusions 129

8.1 Discussions 129
8.2 Outlook 131

A Derivation of the marginal probabilities 133

Bibliography 137

Curriculum Vitae 151

1

Introduction

People exhibit characteristic motion behaviors that we experience in everyday life:

> *I thought I saw, close to the Swanns house but going in the other direction, going away from it, Gilberte, who was walking slowly, though with a firm step, by the side of a young man with whom she was conversing [...]. The strolling couple were already some way away, and the parallel lines which their leisurely progress was quietly drawing were on the verge of disappearing in the Elysian gloom.*

In this passage of Proust's *Remembrance of Things Past* [Proust, 1919], Gilberte is walking toward a *destination*, with a chosen *speed* and while *interacting* with another person. Proust even describes how the trajectories of the two walking people look like, *i.e.* two parallel lines. This is indeed what we would expect from two people walking together, in absence of obstacles to avoid. Although each individual exhibits different behaviors, depending for instance on the cultural background, the scene and his emotional status, there is probably a common set of features that are able to synthesize the fundamental aspects of human motion behaviors. The goal of many researchers has therefore been that of isolating these features to build a model that could analyze, synthesize and predict human motion behaviors.

The topic has received particular attention in the social sciences already from the 1960ies. However, it became soon clear that a human motion model would be useful to a number of applications. The evacuation simulators and computer graphics animations are two of the first and most

important examples of such applications. Being able to simulate evacuation scenarios is of great help to engineers and architects for building and urban planning. The movie and game industry, on the other end, are searching for ways of reducing the human intervention in character animation. Similarly, virtual reality application would clearly benefit from human-like behaving interacting agents.
More recently interest rose also in the computer vision community. Here motion models are currently being used as prior information both for understanding and, more relevant to this dissertation, for predicting people motion. Another important application area is in robotics, where the ability of predicting human motion has considerable consequences in path-planning algorithms. For instance, car industries are investing a substantial effort in systems that warn the driver and possibly break in case a vehicle-pedestrian collision is expected.

We can distinguish three levels [Reynolds, 1999] in a human motion behavior. The highest level involves the decision of a destination. The middle level is the actual walking to the destination, possibly steering to avoid obstacles or interacting with other individuals. Finally the lowest level is that of locomotion, that takes care of the articulated motion necessary to walk to the desired destination. In this thesis we focus on the middle level and partly on the destination selection. One of our goals is indeed to investigate to what extent motion priors can help visual tracking. The problem of tracking multiple people in a crowded scene, despite the progress in people detection algorithms, is still largely unsolved. On one hand this is due to the variability of human appearance and the high amount of occlusions in such circumstances, that limit the reliability of the observation model. On the other hand there is the difficulty in effectively capturing the temporal correlation of human trajectories. Tracker applications usually employ a simple constant velocity model in order to do so. We propose here to use richer and more complex motion models. We show how different trackers can be adapted to employ such a model and we discuss the results with experiments in challenging sequences. Also, we adapt a motion model originally designed for a tracking application, to a crowd simulation application. In doing so we are able to validate the motion model from a different perspective.

1.1 Contributions

The contributions of this thesis can be summarized as follows:

1. We propose a motion model comprising a steering component and a goal selection component. The steering model, termed Linear Trajectory Avoidance (LTA), handles steering and interaction aspects of human motion. The goal selection component provides a probabilistic layer to model, in an event-dependent fashion, the scene motion patterns. The model has been designed with attention to efficiency and reflects the finding of researchers in other disciplines.

2. While trackers have traditionally used a constant velocity model to exploit the time correlation of human motion, we investigate to what extent more complex motion prior can affect the tracker performance.

3. We propose a stochastic model for predicting human trajectories within a short time horizon of about five seconds, still focusing on efficiency and real-time capabilities.

4. We develop a scene-dependent crowd simulation and use it to validate our findings in mixed reality experiments, where real and virtual agents are simulated in the same environment and interact with each other.

5. We propose a modular and extensible tracker where motion prior information can be integrated in the same way as any other component. We cope with the complexity of the resulting model with powerful and carefully designed inference strategies.

6. We recorded several sequences of naive walking people, using a single monocular camera. In order to capture large scenes, the recordings are carried out with a camera viewing the scene from above. The data has been annotated and is made available for further research.

1.2 Organization

This thesis is structured as follows.

In Chapter 2, *Linear Trajectory Avoidance*, we describe the steering model (LTA). The model is capable of reproducing the anticipative nature of the avoidance behavior that people exhibit while moving toward a destination. The attraction aspect of grouping relationship is also accounted for. LTA reflects findings from other fields, such as that people anticipate collisions and keep a safe elliptic personal area around them.

In Chapter 3, *Scene-dependent Crowd Simulation*, we build a crowd simulation application based on LTA. Here we extend the steering model with a goal selection layer capable of representing several scene motion patterns. The purpose is to study the effectiveness of the motion model in reproducing scene-dependent motion patterns and individual motion behaviors. In order to mimic a specific scene, the motion patterns of the target scene are extracted and reproduced in the simulation. Also the steering model is adapted to reflect the low level properties of motion that are characteristic of the simulated scene. We analyze the results of such application in a mixed reality experiment.

In Chapter 4, *Tracking with LTA*, we integrate the LTA steering model in a multi-target tracker with a negligible computational cost. We show that the main benefits of such a combined system can be observed when long occlusions occur, that is indeed when the tracker relies the most on the motion model.

People trajectories are affected by several, possibly contingent, factors. Reproducing more and more factors within the motion model would come at the expense of efficiency. An alternative approach is that of coping with the residual uncertainty of the model, rather than relying on its deterministic output. With the goal of trajectory prediction, in Chapter 5, *Stochastic LTA for Prediction*, we extend the LTA steering model to handle robust, efficient, multiple-hypothesis, trajectory prediction. We show the effectiveness of the prediction capabilities of the model and we study the implications of using such a model in a tracker.

In Chapter 6, *Towards Joint Grouping and Tracking*, we investigate the group classification task from two different perspectives. First we carry out group classification when the trajectories are given. Then we show some preliminary results of the joint grouping and tracking problem.

In Chapter 7, *Tracking with Interactions*, rather than extending an existing tracker with a more elaborate motion model, we build the motion and interaction priors into a new tracker and evaluate the results in challenging scenes. On the modeling side, we kept the focus on accuracy, extensibility and modularity. On the inference side, we exploited the structure of the tracker to develop an efficient inference strategy.

In Chapter 8, *Conclusions*, we conclude the dissertation and give an outlook on possible future research directions.

The related work is divided in three minimally overlapping parts and is described in Chapter 2, Chapter 3 and Chapter 4. With the exception of Chapter 6 and Chapter 7, the other chapters depend and require reading only of Chapter 2.

2

Linear Trajectory Avoidance

The modeling of human motion is of enormous interest for a multitude of applications, ranging from gaming, over movie effects, to evacuation simulators and urban planning. Humans exhibit a huge variety of motion behaviors that is not trivial to reproduce in virtual characters. Following the terminology of [Reynolds, 1999], we distinguish among three levels of motion behavior: goal setting, steering and locomotion. Goal setting is the higher level of motion behavior that deals with task selection. Steering is the middle level and deals with path selection while moving towards the goal. Locomotion, the lowest level, deals with the articulation that is needed to actually move along the trajectory. In this chapter we focus on the steering model, while the goal selection are delayed to the next Chapter.

2.1 Related Work

In this section we present work related to pedestrian steering model. A more detailed review can be found in [Pelechano *et al.*, 2008; Thalmann and Musse, 2007].

We distinguish two classes of pedestrian motion models [Pelechano *et al.*, 2008], i.e. *macroscopic* and *microscopic*. Macroscopic models look at pedestrians as a flow, focusing on global crowd properties like density or average velocity. An early example of this category is the work of [Henderson, 1971], where the crowd dynamics are described through an analogy with gas kinetics. Another example is the work of [Hoogendoorn, 2003], where subjects are assumed to jointly schedule activities and choose their route under uncertainty, while minimizing a subjective utility cost. Microscopic models describe instead the single agent motion, status and intentions. In this dissertation we are mostly interested in microscopic models. We review them below.

Social sciences. The way people use space and interact with each other has been argument of research in the social sciences since the 1960ies. [Hall, 1966], inspired by [Hediger, 1955], divides the space around each individual into concentric spheres. He describes an *intimate*, a *personal*, a *social* and a *public* space. One of the limitation of this static description, is that it does not account for density. In this regard, the work of [Freedman, 1975] proposes the *density-intensity* theory, according to which the intensity of the effects of the social interactions depends on the density of the crowd. Other studies have investigated the cultural differences found among cultures [Beaulieu, 2004] or the group behaviors [McPhail and Wohlstein, 1982]. More recently, the measurements from [Gérin-Lajoie *et al.*, 2005], suggest that individuals keep an elliptic, and not spherical, personal area around them. This had already been argued by [Goffman, 1971].

Simulation models. Autonomous agents are an active area of research since long time. Among the first efforts in this direction we mention the work of [Reynolds, 1987], that shows how three simple steering behaviors of the agents are sufficient to let more complex group behavior emerge. A different approach led to floor fields [Schadschneider, 2001], a cellular automaton model that offer the advantageous property of reducing the long range interaction to local ones with *memory*. The space and time discretization and the simplicity of the interactions make this model suitable for very efficient computation. Certainly one of the most popular work in this field is the Social Force model [Helbing and Molnár, 1995], a physically based steering model successfully applied in simulating evacuation scenarios. The model has been further extended in [Johansson *et al.*, 2007], where the model parameters are estimated from real data. Recently, [Kapadia *et al.*, 2009] combine sensory information into egocentric affordance fields to choose the next orientation and speed.
Few works have also modeled the group interactions within the motion model. [Reynolds, 1999] qualitatively describes behaviors like *separation*, *cohesion* and *alignment* that can be used to produce group-like effects. [Moussaïd *et al.*, 2010] analyze empirical data to describe several properties of group walking: size of group, average walking speed

as function of density and spatial group organization. Furthermore the authors extend the social force model with a term that is responsible of the attraction of each member of a group toward the center of mass of the group configuration.

Prediction models. Steering models have mostly been used for computer graphics animations and to simulate evacuation scenarios. Other fields, such as robotics and computer vision, have recently developed interest for such models and came out with their own solutions. The applications in this case are mostly path-planning, motion priors for tracking, but also unusual event detection (*e.g.* [Mehran *et al.*, 2009]). In robotics [Trautman and Krause, 2010] integrate a motion model to facilitate the robot navigation in crowded scenes. A solution based on geometric optimization is proposed by [van den Berg *et al.*, 2008; van den Berg *et al.*, 2009]. The choice of the next velocity is based on avoiding *velocity obstacle* [Fiorini and Shiller, 1998], *i.e.* a set of velocities that, by linear velocity prediction, are expected to lead to a collision. Also in computer vision novel motion models have been proposed, mainly with the goal of improving tracking applications [Pellegrini *et al.*, 2009; Antonini *et al.*, 2006; Scovanner and Tappen, 2009]. We review these works more in detail in Chapter 4.2.

Empirical models. Some models have been designed by careful analysis of human motion behaviors. [Pettré *et al.*, 2009] use an experimental study with real people to measure the motion adaptions occurring in pair interactions. They use the result of the study to build an anticipative model and to validate the results. [Guy *et al.*, 2010] extends the work of [van den Berg *et al.*, 2009] by introducing reaction and observation times, together with kinodynamic constraints. [Ondřej *et al.*, 2010] propose a vision-based, biologically inspired model and show that self-organized patterns can emerge in the crowd of walkers. [Paris *et al.*, 2007] describes the set of velocities that are expected to lead to a collision free trajectory, by discretizing the time horizon into several slots, and using a linear velocity predictor in each slot. The results are used to build a model that is calibrated with motion capture data. In [Brogan and Johnson, 2003] the authors focus on building realistic path. The work presents a simple walking model that reflects empirical finding extracted from real trajectories. A set of manually tuned equations and look-up table for heading direction are used at each timestep to update the simulated agent position. The model is however scene dependent and interactions among pedestrians are not taken into account.

In this chapter we introduce LTA, an energy minimization based steering model. Contrary to the Social Force model, this model uses a linear velocity predictor to anticipate collisions. Other models [Reynolds, 1999; Paris *et al.*, 2007; Pettré *et al.*, 2009; Ondřej *et al.*, 2010; Guy *et al.*, 2010] use a linear velocity prediction for the same purpose, albeit in different ways. Rather than specifying hard threshold to detect when a reaction is necessary in order to avoid collisions, LTA defines a smooth energy function that depends on the subject velocity. As a consequence *soft* boundaries exist between the collision/no-collision areas. Contrary to [Paris *et al.*, 2007], no time discretization is carried out, but the choice of the velocity implicitly depends on the time of maximum approach. Moreover, such a formulation leads to a straightforward integration both of multiple interactions, like in [Helbing and Molnár, 1995], and of additional components, as we show below for the grouping attraction term. Contrary to [Helbing and Molnár, 1995], however, the LTA function does not specify a reaction force. Rather, the function minima are searched to retrieve the chosen velocity.

2.2 Steering Model

We introduce in this section a generalization of the LTA model [Pellegrini *et al.*, 2009] for simulating human behavior. LTA is an energy based motion model that makes a linear velocity prediction to anticipate collisions, i.e., it assumes that each subject adapts its velocity based on its present state and on the position and velocity of other visible objects. Such a collision avoidance model based on *anticipating* trajectories of other objects is motivated by real human behavior [Pettré *et al.*, 2009].

2.2.1 Model Variables

In the remaining of the chapter, we assume that the scene, at time t consists of a set of N pedestrians, $\mathcal{N}^t$. Each pedestrian i is represented by a state vector $\mathbf{s}_i^t = (\mathbf{p}_i^t, \mathbf{v}_i^t)$, comprising the 2D position $\mathbf{p}_i^t$ and 2D velocity $\mathbf{v}_i^t$. It is assumed that each pedestrian prefers to move at a given comfortable speed u_i to reach a destination $\mathbf{z}_i^t$. The state $\mathbf{s}_i^t$ of a pedestrian is updated by

$$\mathbf{v}_i^t = \alpha \mathbf{v}_i^{t-1} + (1-\alpha)\overset{\star}{\mathbf{v}}_i^t \tag{2.1}$$

$$\mathbf{p}_i^t = \mathbf{p}_i^{t-1} + \Delta \mathbf{v}_i^t \tag{2.2}$$

where Δ is the time step and α is a smoothing coefficient, while $\overset{\star}{\mathbf{v}}_i^t$ is obtained through the LTA energy minimization that is discussed in Sec. 2.2.3. Other objects like benches, trees or walls are represented by a set of static points $\mathcal{O}^t$.

2.2.2 Visibility

Each pedestrian i views only a portion of the scene defined by the gaze and the occlusion by other objects. To this end, we introduce the set of visible objects for pedestrian i at time t, $\mathcal{V}_i^t$. To populate this set, we carry out a Delaunay triangulation over the set of points $\mathcal{N}^t \cup \mathcal{O}^t$. From the triangulation, we obtain a neighboring relation between pairs of objects. In particular, we term $\mathcal{D}_i^t$ the set of objects connected to i by the triangulation and ϕ_{ij} the angle between the vector $\mathbf{p}_j^t - \mathbf{p}_i^t$ and the gaze direction of pedestrian i, which is estimated by the previous velocity $\mathbf{v}_i^{t-1}$. We further limit the visibility by a maximum distance $d_v = 6m$ and a maximum field of view angle $\phi_v = 91°$. The visibility set for pedestrian i is thus defined by

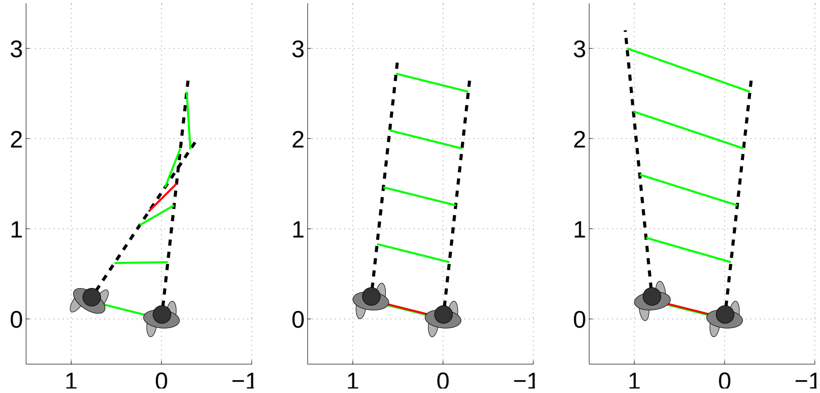
$$\mathcal{V}_i^t = \mathcal{D}_i^t \cap \{j | d(\mathbf{p}_i^t, \mathbf{p}_j^t) < d_v \wedge |\phi_{ij}| < \phi_v\}. \tag{2.3}$$

2.2.3 Avoidance Behavior

Given a simulated pedestrian i at time $t-1$ and another pedestrian or object $j \in \mathcal{V}_i^{t-1}$, we want to know the repulsion energy that i *feels* when choosing velocity $\mathbf{v}_i^t$. Without loss of generality [1], we assume that $\mathbf{p}_j^{t-1} = \mathbf{0}$ and $\mathbf{v}_j^{t-1} = (\kappa, 0)^T$ with $\kappa \geqslant 0$. In other words we use an egocentric reference system centered on j. As already mentioned, the avoidance behavior is based on a linear projection in the future of the

---

[1] A coordinate transformation can be applied to make the assumption valid.



(a) (b) (c)

***Figure 2.1**: Three possible cases of interaction between two people. See text for comments. The red line shows the distance at $\overset{\star}{\tau}$. In case (a) the moment of maximum approach is $\overset{\star}{\tau} = \min(T, -\frac{(\mathbf{p}_i^{t-1}-\mathbf{p}_j^{t-1})\cdot(\mathbf{v}_i^t-\mathbf{v}_j^{t-1})}{||\mathbf{v}_i^t-\mathbf{v}_j^{t-1}||^2})$, while in cases (b) and (c) it holds $\overset{\star}{\tau} = 0^+$.*

current state. High energies are assigned to those velocities that are expected to bring the pedestrian too close to other objects. To this end, the repulsion energy between i and j depends on the distance vector

$$\mathbf{d}_{ij}(\mathbf{v}_i^t, \tau) = \mathbf{p}_i^{t-1} + \tau\mathbf{v}_i^t - (\mathbf{p}_j^{t-1} + \tau\mathbf{v}_j^{t-1}) \; , \tag{2.4}$$

such that $\mathbf{d}_{ij}(\mathbf{v}_i^t, \tau)$ is the vector that represents the expected position of i from j's point of view τ seconds after t. Whether the current state will lead to a collision in a time range $(0, T]$ will depend on the magnitude of this distance vector. In fact, it will depend only on the moment of maximum approach $\overset{\star}{\tau}$, defined as

$$\overset{\star}{\tau} = \underset{\tau\in(0,T]}{\arg\min} \, ||\mathbf{d}_{ij}(\mathbf{v}_i^t, \tau)|| \; . \tag{2.5}$$

A few possible situations are shown in Fig. 2.1. For converging trajectories (Fig. 2.1 (a)) the magnitude of $\mathbf{d}_{ij}(\mathbf{v}_i^t, \tau)$ decreases to a minimum that can be in the range $(0, T]$, in which case

$$\overset{\star}{\tau} = -\frac{(\mathbf{p}_i^{t-1} - \mathbf{p}_j^{t-1}) \cdot (\mathbf{v}_i^t - \mathbf{v}_j^{t-1})}{||\mathbf{v}_i^t - \mathbf{v}_j^{t-1}||^2} \tag{2.6}$$

or greater than T, in which case it is easy to show that $\overset{\star}{\tau} = T$. For diverging trajectories (Fig. 2.1 (c)), the magnitude of $\mathbf{d}_{ij}(\mathbf{v}_i^t, \tau)$ increases with τ. $\overset{\star}{\tau}$ is in this case 0^+. In practice, we use a small constant ($\epsilon = 0.01$) to maintain the dependency on the velocity in Eq. 2.4, i.e., $\overset{\star}{\tau} = \epsilon$. For the special case of parallel trajectories (Fig. 2.1 (b)), $\mathbf{d}_{ij}(\mathbf{v}_i^t, \tau)$ is constant and any value in $(0, T]$ is a good candidate for $\overset{\star}{\tau}$. We use $\overset{\star}{\tau} = \epsilon$. We can therefore express the distance of maximum approach as $\overset{\star}{\mathbf{d}}{}_{ij}^t(\mathbf{v}_i^t) = \mathbf{d}_{ij}(\mathbf{v}_i^t, \overset{\star}{\tau})$. Note that we bounded $\overset{\star}{\tau}$ with a maximum value T. This is done in order to avoid pedestrians to feel repulsion for interactions that happens too far in the future. The benefits of this are shown below.

Intuitively, the energy should be high when $\overset{\star}{\mathbf{d}}{}_{ij}^t(\mathbf{v}_i^t)$ has small magnitude. Furthermore, a different repulsion might be expected if at $\overset{\star}{\tau}$, subject i is in front of j than when i is on j's side. Evidence [Gérin-Lajoie *et al.*, 2005] suggests that the *personal area* has an elliptical shape, rather than circular. We therefore define the interaction term as

$$I_{ij}(\mathbf{v}_i^t) = \exp\left(-\overset{\star}{\mathbf{d}}{}_{ij}^t(\mathbf{v}_i^t)^T \begin{bmatrix} \lambda_{I,1} & 0 \\ 0 & \lambda_{I,2} \end{bmatrix} \overset{\star}{\mathbf{d}}{}_{ij}^t(\mathbf{v}_i^t) \right) \tag{2.7}$$

In the particular case when $\mathbf{v}_j^{t-1} = \mathbf{0}$, as the gaze direction of j is unknown, we cannot make a front-side distinction for the interaction. In this case we set $\lambda_{I,1} = \lambda_{I,2}$, thus obtaining the original LTA formulation of the interaction energy. See Fig. 2.2 for an illustration.

Multiple subjects interaction energies are combined into a single energy as in the original LTA model:

$$I_i(\mathbf{v}_i^t) = \sum_{j \in \mathcal{V}_i^t} w_{ij}^{t-1} I_{ij}(\mathbf{v}_i^t) \ , \tag{2.8}$$

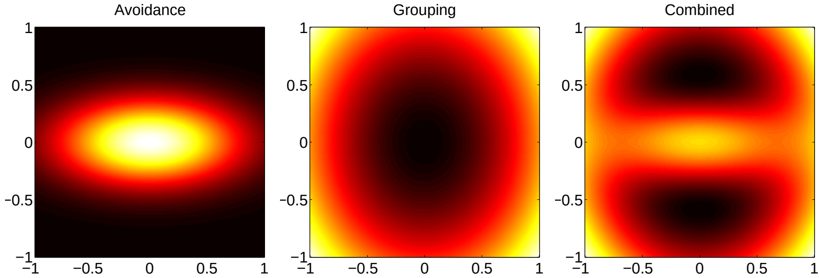


Figure 2.2: *Shape of the energy functions: we show the avoidance term (left), the grouping term (center) and their combination (right). Note that we use j egocentric reference system, assumed to move in the horizontal direction.*

with the weight w_{ij}^{t-1} defined as

$$w_{ij}^{t-1} = \eta_{ij}^{t-1}\nu_{ij}^{t-1} \tag{2.9}$$

$$\eta_{ij}^{t-1} = \exp\left(-\frac{||\mathbf{p}_i^{t-1} - \mathbf{p}_j^{t-1}||^2}{2\sigma_w^2}\right) \tag{2.10}$$

$$\nu_{ij}^{t-1} = \left((1+\cos(\phi_{ij}^{t-1}))/2\right)^{\beta} . \tag{2.11}$$

While η_{ij}^{t-1} down-weights the interaction with pedestrians that are currently far from i, ν_{ij}^{t-1} gives more importance to the interaction with j when the angle ϕ_{ij}^{t-1} under which i sees j is small.

2.2.4 Groups

Interaction among subjects is not only repulsion and avoidance. Here we introduce in LTA the interaction between people belonging to the same group, modeled as attraction.

Note that while groups have not been explicitly modeled yet, in a sense they arise from the avoidance term during the obstacle avoidance maneuver. See Fig. 2.3 for an example. What however is missing is the modeling of the desire of *walking together* in the same group. We define a group as a set of individuals that always share desired speed u and

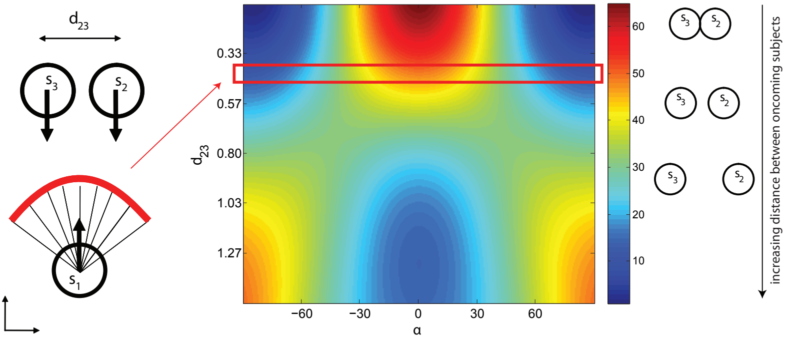


***Figure 2.3**: Energy seen by subject 1 when making a choice of changing its heading (horizontal axis) as it approaches two subjects moving in opposite direction. Each column of the figure describes the energy for a different direction of the velocity vector (keeping the speed fixed), while each row indicates different distance between subjects 2 and 3. One can see that as a consequence of the avoidance term shape, a local minimum in the middle exists only when the gap between the two oncoming subjects is sufficiently large. The wider the gap between 2 and 3 (vertical axis), the easier it is to pass between them (bottom of graph, minimum in middle) instead of steering around the pair (top, minima on the side).*

have *similar* desired destination $\mathbf{z}$. In particular, we define a set $\mathcal{G}_i$ as the set of pedestrians that belong to the same group as pedestrian i.

Let us stay under the assumption used before for the calculation of $\overset{\star}{\mathbf{d}}{}^t_{ij}$. We then model the attraction between a subject i and another subject j as

$$G_{ij}(\mathbf{v}_i^t) = \overset{\star}{\mathbf{d}}{}^t_{ij}(\mathbf{v}_i^t)^T \begin{bmatrix} \lambda_{G,1} & 0 \\ 0 & \lambda_{G,2} \end{bmatrix} \overset{\star}{\mathbf{d}}{}^t_{ij}(\mathbf{v}_i^t) \; . \qquad (2.12)$$

Note that by setting the two parameters $\lambda_{G,1}$ and $\lambda_{G,2}$ to different values, one obtains different behavior patterns. In particular, if $\lambda_{G,1} > \lambda_{G,2}$, the energy decreases faster on the side of the attracting subject, thus favoring a side attraction. When instead $\lambda_{G,1} < \lambda_{G,2}$, a tandem configuration is favored. As in the previous case, when j is not moving and therefore the gaze direction is not known, we set $\lambda_{G,2} = \lambda_{G,1}$.
In Eq. 2.12 the attraction depends on the distance of closest approach $\overset{\star}{\mathbf{d}}{}^t_{ij}$ and is zero when this distance is zero. The attraction toward zero

energy, and therefore zero distance, is balanced by the avoidance term of Eq. 2.7, that is always combined with the group attraction (Fig. 2.2).

The grouping interaction are combined for multiple subjects as

$$G_i(\mathbf{v}_i^t) = \sum_{j \in \mathcal{G}_i \cap \mathcal{V}_i^t} w_{ij}^{t-1} G_{ij}(\mathbf{v}_i^t) \ . \tag{2.13}$$

Fig. 2.4 shows different behaviors obtained with different instantiations

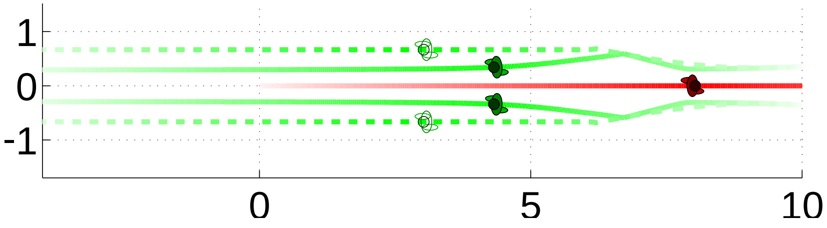


***Figure 2.4**: Two subjects (green) avoid another (red) coming in the middle of them. Without the limit T on the time horizon, agents (dashed green line) are affected by predicted collision that happen too far in the future. The continuous green line shows the simulation for the LTA presented in this chapter.*

of the model. Note that without a bound on the time horizon, the agents become extremely careful and find it difficult to converge to a more comfortable distance when being in the same group.

2.2.5 Destinations

To complete the motion model, LTA uses energy terms to make sure that subjects walk toward the destination $\mathbf{z}_i^t$,

$$D_i(\mathbf{v}_i^t) = \left(1 - \frac{(\mathbf{z}_i^t - \mathbf{p}_i^{t-1})}{||\mathbf{z}_i^t - \mathbf{p}_i^{t-1}||} \cdot \frac{\mathbf{v}_i^t}{||\mathbf{v}_i^t||}\right) \lambda_D \ , \tag{2.14}$$

and with a certain desired speed,

$$S_i(\mathbf{v}_i^t) = (u_i - ||\mathbf{v}_i^t||)^2 \lambda_S \ . \tag{2.15}$$

As the parameters λ_I (2.7) and λ_G (2.12), λ_D and λ_S steer the impact of the corresponding terms.

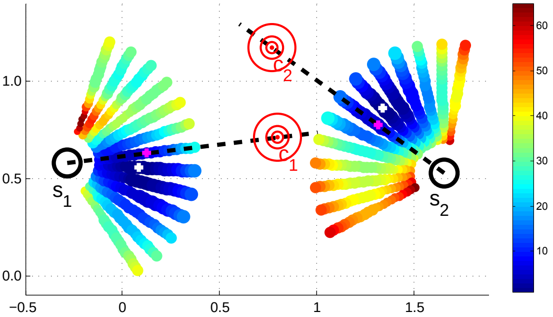


***Figure 2.5**: Two subjects, with their current directions (black) and velocities (magenta). Subject 1 feels the repulsion from subject 2's expected point of closest approach c_2, and vice versa. Colors denote energies for different velocities (drawn in the ground plane by multiplication with a constant time factor), white dots mark the respective minima. Note how subject 2 accelerates and turns right in order to avoid subject 1, while subject 2 slows down and turns to his right.*

2.2.6 Model Summary

The energy terms described in this section are combined as.

$$E(\mathbf{v}_i^t)=\sum_{j\in\mathcal{V}_i^t} w_{ij}^{t-1} I_i(\mathbf{v}_i^t)+\sum_{j\in\mathcal{G}_i\cap\mathcal{V}_i^t} w_{ij}^{t-1} G_{ij}(\mathbf{v}_i^t)+S_i(\mathbf{v}_i^t)+D_i(\mathbf{v}_i^t)\ . \quad (2.16)$$

At each time step and for each subject, the velocity $\overset{\star}{\mathbf{v}}_i^t$ is obtained by minimizing Eq. 2.16. This can be done, for instance, by using gradient descent. However only local optima can be found in this way, as the components are analytics and gradient can be computed easily. In Chapter 5 we investigate further this issue. This velocity is used for the state update in Eq. 2.1. Fig. 2.5 shows an illustration of an energy landscape seen by interacting subjects.

Fig. 2.6 shows the graphical model of the space and time dependencies among the model variables.

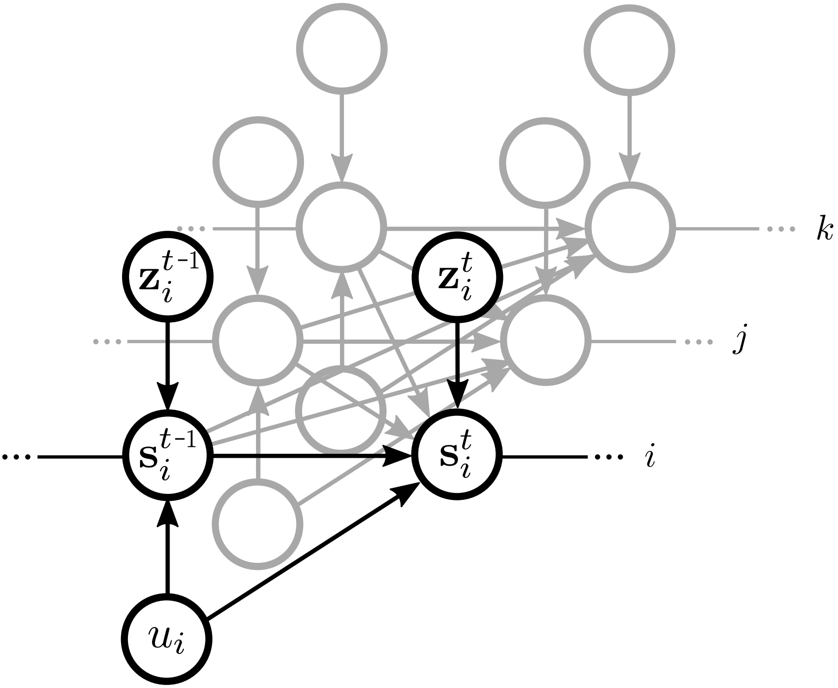


Figure 2.6: *Dependencies among the LTA variables.* i*,* j *and* k *are the indices for three pedestrians.*

2.3 Static Obstacles

So far, we only took dynamic obstacles in the form of pedestrians into account. In most common scenes, however, people try to avoid static obstacles, as well. Following other authors [Johansson *et al.*, 2007] we model such obstacles as subjects with zero velocity. Several choices are possible. A first one is to represent the obstacle's position, at every time step, as the point closest to the pedestrian. While being a coarse approximation, this works well for small and sparse obstacles. As an alternative, we could regularly sample points along the obstacle boundary, and represent the obstacles with these. This approach is general and works well with bigger, complex obstacles. However it comes at the cost of efficiency, as each sample is seen as an interaction target in the LTA energy function.

2.4 Conclusions

In this chapter we presented a model for steering clear of static and dynamic obstacles, as well as for interacting with other people in the scene. The model is described by an energy function that is minimized to obtain the velocity of choice for the subject. As the collision is handled in advance, the model is capable of showing human like avoidance behavior.

In the rest of the dissertation the model is used in different applications. In particular it is used for crowd simulation (Chapter 3), for tracking (Chapter 4) and for trajectory prediction (Chapter 5). The model is slightly adapted or extended in each of these applications and the details are given within the corresponding chapter.

3

Scene-dependent Crowd Simulation

3.1 Introduction

In this chapter we propose a model to simulate crowds within a specific context. Indeed, depending on the environment there are different motion patterns. Whereas people may leisurely walk in a park, they probably are quite hasty in a business district. Different environments, which could also be indoor, also contain different sources, attractors, and sinks where people resp. appear, often go to, or disappear. All these factors must be accounted for when animating a virtual crowd. Manual specification (e.g. by means of scripting) is common, but also very tedious and time consuming. An alternative approach is to extract real trajectories from an actual crowd and use these. This is not always doable, is limited to the amount of data available from that crowd and may still require directing it.
To overcome these limitations, we propose to learn motion behaviors directly from the scene. We do this both at the goal setting and steering levels. In order to maximally benefit from the real trajectories that one might have at one's disposal, we propose the use of mixed crowds, consisting of a blend of real and simulated trajectories.[1]

We extend the model presented in Chapter 2 by adding a probabilistic goal selection level that enables virtual agents to purposively move

---

[1] We call real agents those crowd members the trajectory of whom is derived directly from captured data, and virtual agents those that are generated by simulation.

about. Behaviors like “wait at the tram stop”, “stroll-around” or “go and grab a coffee” are not explicitly scripted but learned from real examples. This said, user interaction is easy to integrate in the process, and an arbitrary behavior can be chosen.

Our approach only requires a few minutes (5 to 10) of video recording with real people moving in the scene. Their trajectories and group memberships are extracted by means of the semi-automatic tracker based on the work presented in Chapter 7. This video data is processed both to analyze lower level properties such as typical personal area [Gérin-Lajoie *et al.*, 2005] and to extract the regions in the scene that are relevant for crowd motion. At simulation time, the mix of real vs. virtual people is a free parameter, allowing the user to adapt crowd density.

3.2 Related Work

We discussed already in Chapter 2 works on steering models. In this section we review works dealing with crowd simulation.
An example of crowd simulation is the work of [Treuille *et al.*, 2006]. Inspired by [Hughes, 2003], the authors propose a real-time large scale crowd simulation designed to handle large groups of people moving with a common goal, without using an agent based perspective. A different kind of simulation is presented in [Lerner *et al.*, 2007]. This is an example-based model, that uses a database build with real world trajectories. During the simulation, the database is searched for the closest match to the simulated scene state and the subject trajectories are updated copying stored ones. A more detailed review can be found in [Pelechano *et al.*, 2008; Thalmann and Musse, 2007].

Group behavior. Rather than based on the single pedestrian, [Musse and Thalmann, 2001] present a crowd simulation architecture where groups are instead the central entities. The simulation allows the specification of interest point and external events, either through a scripting interface or through external commands. Always related to group, but data-driven, is the approach of [Lee *et al.*, 2007]. Here, state-velocity pairs are extracted from real data and at simulation time a combination of them is employed in order to simulate group behaviors. Another example of group behavior is presented in [Loscos *et al.*, 2003]. In this case the group behavior is implemented through a leader-followers paradigm and the leader is actually not displayed in the simulation for lack of realism in her positioning.

Mixed Reality. An interesting addition to crowd simulation, is the possibility of mixing real and virtual people. [Zhang *et al.*, 2011] propose online integration of few virtual characters in a sparse real scene. Only the simulated people are aware of the real ones and the focus of the paper is mostly on the rendering of the mixed video sequence. In the perspective of an immersive virtual experience, [Olivier *et al.*, 2010] studies whether real humans are able to perceive and anticipate virtual motions.

Decision behavior. Another important aspect of an autonomous virtual agent is the goal selection. In [Tu and Terzopoulos, 1994] the intentions of different types of agents are represented by different flow-charts. In [Musse and Thalmann, 2001], when in the *autonomous mode*, the crowd responds to events following the rules specified by the animator. [Shao and Terzopoulos, 2005] proposes a hierarchical architecture comprising behavioral and cognitive aspects that is capable of simulating pedestrians in large-scale urban environments. In [Barros *et al.*, 2004], the intentions of the agents are represented through a Finite State Machine FSM, and the transition from one state to another depends on the knowledge and the status of the agent, that in turn is affected by scene events. A FSM is the model of choice also in [Braun *et al.*, 2005]. This is a physically based model that extends [Helbing and Molnár, 1995] with a set of attributes like *mobility* or *altruism* and includes an elaborate representation of the environment. Some of the FSM transitions depend on a random number generator, thus allowing for non-deterministic simulations.
In [Penn and Turner, 2002] the authors show that simulation with rational looking patterns of motions are possible even without explicitly specifying destination points. Moreover, they show that by simply choosing a random next step in the visibility area of each agent, a significant correlation with recorded pedestrian flow at the gates of a store area is achievable. This correlation is actually bigger than the one resulting from simulating agents that look for *junctions* across the environment. This suggests that at least exploratory motion behavior is strongly influenced and driven by the static spatial configuration of the available

space. A similar technique, albeit in a much larger scale simulation, is shown in [Loscos *et al.*, 2003], where junctions in this case are automatically detected pedestrian crossings in a city map.
In the works we have discussed, the region structure and/or the transition rules are manually specified. In this chapter we reduce the concept of intention, or goal, to that of a destination in space and we let the transitions be learned automatically from the scene.

Closely related to the work presented in this chapter, is the study presented in Chapter 6 of [Thalmann and Musse, 2007]. The authors here first describe which scene information could be useful to reproduce a real scene in a simulation. In particular they identify the scene structure, the basic behaviors and the events/reactions as the main aspects to reproduce. However, the information extraction at this point is manual, and requires a customization for each scene. In the second part of the same chapter, they propose to use real tracked trajectories to extract scene specific velocity fields. The velocity fields are then clustered and virtual agents at simulation time are driven by these fields. The simulation represents people entering and exiting a T-shaped pathway. Our approach is different in several ways. First, we do not use fully automated tracker, as in our scenes trackers cannot be fully relied upon. Second, rather than extract velocity fields, we compute region of interests and transitions between them. This allows us to simulate in a stochastic manner complex scenes, where repetitive behaviors are possible. Last, we study the mixing of real and virtual agents in the reproduced scene.

3.3 Scene Transitions

In Chapter 2, we have assumed that the destination $\mathbf{z}_i^t$ and the desired speed u_i are known for each subject i. In order to be able to mimic the motion patterns in a specific scene, we propose to learn a probabilistic model for these quantities. To this end, we extract the trajectories of real persons and the groups $\mathcal{G}$ form a video with an interactive tracker based on the work presented in Chapter 7 and label the static objects $\mathcal{O}^t$ manually. We use top-view cameras (Fig. 3.1) in order to capture the entire scene and to facilitate the tracker job. The tracker requires only to know the ground plane. We also define a set of scene events, $\mathcal{E}$ and the subset $\mathcal{A}^t \subseteq \mathcal{E}$ of active events at time t.

Based on the trajectories, we model the desired speed u_i by a normal distribution, where mean μ and variance Σ are estimated from the observed velocities. We set an additional threshold of $0.5ms^{-1}$ to avoid slow walkers. In our model, all the subjects in the same group share the same desired speed. To learn the sequence of destinations for each group of subjects, we need to know the *intentional* destination of the recorded pedestrians. To this end, we extract interesting points from the scene by segmenting the trajectories.

3.3.1 Segmenting the Trajectories

Given a trajectory $\mathcal{Q}_i = [\mathbf{p}_i^0 \dots \mathbf{p}_i^T]$, we are interested in a sequence of points $\mathbf{p}_i^0 \dots \mathbf{p}_i^{t_c} \dots \mathbf{p}_i^T$ that split the trajectory in a sequence of sub-tracks. Each of these sub-tracks specifies a unit of motion that a subject should be able to undertake without any complex path planning operation. With the exception of turns done for avoiding obstacles, these sub-tracks should be almost straight lines. Since thresholding the velocity or the curvature of the trajectories turned out not to be robust, the problem is solved by a shortest path search in a graph, as in [Mann *et al.*, 2002]. The graph is obtained by connecting each $\mathbf{p}_i^t \in \mathcal{Q}_i$ with all subsequent points. The cost of the transition form $\mathbf{p}_i^{t_a}$ to $\mathbf{p}_i^{t_b}$ is defined by

$$\gamma(t_a, t_b) = \kappa + \sum_{t=t_a}^{t_b} ||\mathbf{p}_i^t - (\mathbf{p}_i^{t_a} + \frac{t - t_a}{t_b - t_a}(\mathbf{p}_i^{t_b} - \mathbf{p}_i^{t_a}))||^2 \,, \tag{3.1}$$

where κ is a fixed cost associated to each split to regularize the number of splits. The summation in Eq. 3.1 is the cost of approximating the portion of the trajectory from t_a to t_b with a straight line. Note that this cost takes into account also the time t. This is done in order to split the trajectory at points where the speed changes, e.g., when the pedestrian stops. The impact of the regularizer κ is shown in Fig. 3.1. We set $\kappa = 10$ in this chapter. Few other examples are shown in Fig. 3.2.

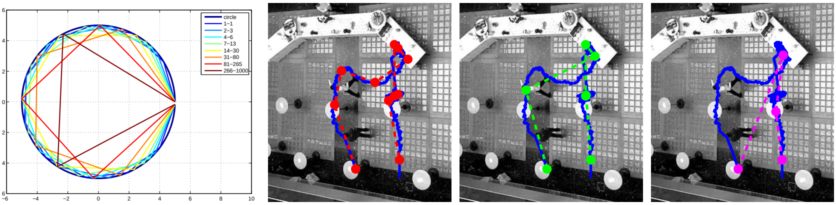


Figure 3.1*:* *Approximations of curve with straight lines for different κ values.* ***Left****: The result of approximating a circle while changing the κ in Eq. 3.1. Each regular polygon corresponds to a range of κ values, shown in the legend. The trajectory starts and ends at point $(5, 0)$. The number of splits decreases with increasing values of κ.* ***Center to Right****: The approximation of a real trajectory (in blue) for $\kappa = 1$ (red), $\kappa = 10$ (green) and $\kappa = 100$ (magenta). Note how a too small value of κ introduces too many corners while the large value misses some.*

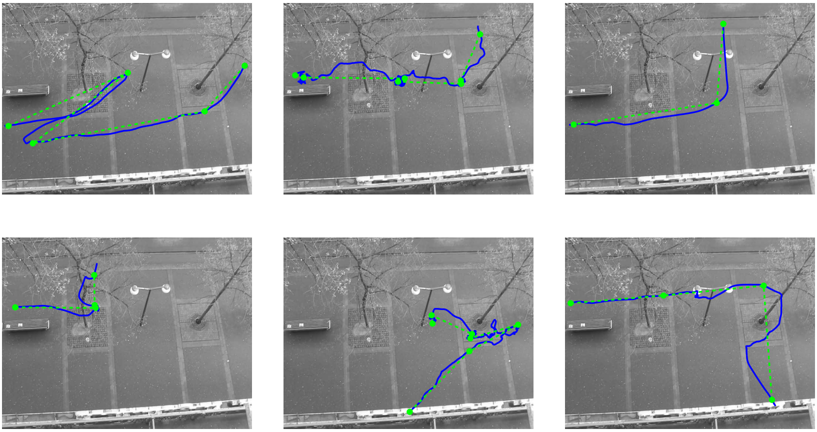


Figure 3.2*: A few examples of trajectory segmentation ($\kappa = 10$). Note how small oscillations, probably due to avoidance behaviors, do not affect the segmentation.*

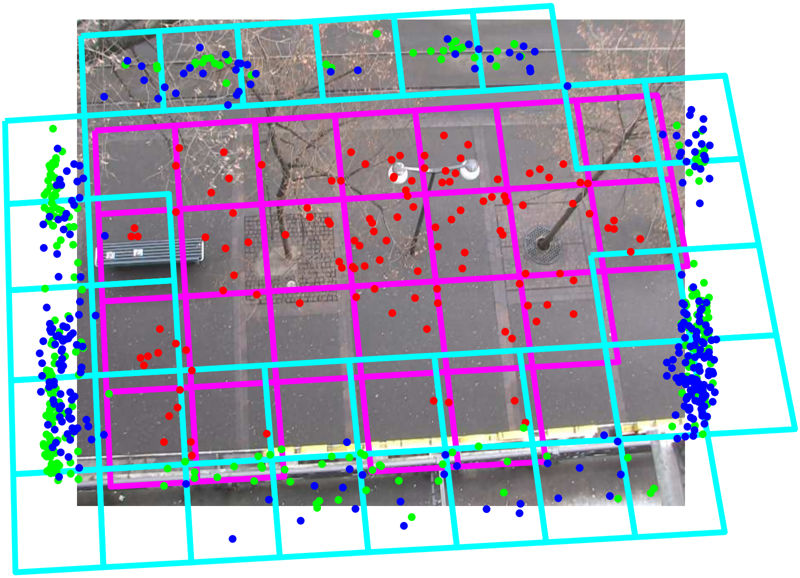


***Figure 3.3**: The grids used for the destination flow. We show the* 3 *kind of corners: entrance (green), exit (blue) and turn (red). The magenta grid is the one for turning points, while the two entrance and exit grid are overlapping and shown in cyan. Grid cells with no corners are not shown.*

3.3.2 Destination Flow

Once we have segmented the trajectories, we can estimate the transition probabilities. We first assign each corner to a region r in the scene, where we distinguish between entrance, exit and transition corners and regions depending on whether they are at the beginning, in the middle or at the end of the trajectories. In our experiments, we use square cells to represent each region as shown in Fig. 3.3. The size of each region cell affects the destination flows. Intuitively, a bigger cell size has more corners in it. However, a bigger cell size over-smooths meaningful transition patterns. We found a good compromise at $2m$ edge size. We also define two special regions, *create* and *destroy* to handle agents initialization and termination, respectively. The same solution can be used for different kind of region shapes, not necessarily square cells, as we show in 3.3.3. The user could also interactively select, modify, move or discard some of the regions.

In this chapter, we assume that a group of subjects always share the same intentions, and therefore the same destination region r. The splitting described in Sec. 3.3.1 can be generalized to a group of trajectories by simply averaging the transition costs in Eq. 3.1 of the subjects within the same group. In practice, due to the problems produced by groups that are at the edge of two cells, we count the transitions at the individual level. A region transition $r_a \rightarrow r_b$ is therefore defined by a pair of consecutive corners in a subject trajectory. The transition probability $p(r_a|r_b)$ is estimated by the normalized count of the transitions $r_a \rightarrow r_b$ that occur, with each subjects contribution divided by the size of the group he belongs to, to account for overcounting.
The probabilities, however, might change over time based on the events that are active. For instance, people that walk to a tram stop might wait until the tram arrives to decide for a new destination. We therefore model the transition probabilities for each active event set $\mathcal{A}^t$, by accumulating in each $p^{\mathcal{A}^t}(r_a|r_b)$ only the transitions that are observed when the corresponding event set is active. We call the set of regions and transitions the destination flow of a region.

3.3.3 A Clustering Variant

In the previous section we introduced the destination flow using square cells. However the method does not rely on the cells being square cells in a grid layout. An alternative solution would be to cluster the corners of the trajectories. Clustering might indeed highlight relevant regions of the image and suffers less of the border effects discussed above. As it often happens with clustering, a first drawback is that the number of clusters is not known in advance. While clustering methods like k-mean can be adapted to cope with such situations by introducing a score that accounts for the number of clusters, we decided to use Dirichlet Process Mixture Model [Rasmussen, 2000], using the a publicly available library [Eisenstein, 2007]. This stochastic method has the advantage that it does not require knowing in advance the number of clusters. We use the assumption that the populations within a cluster are normally

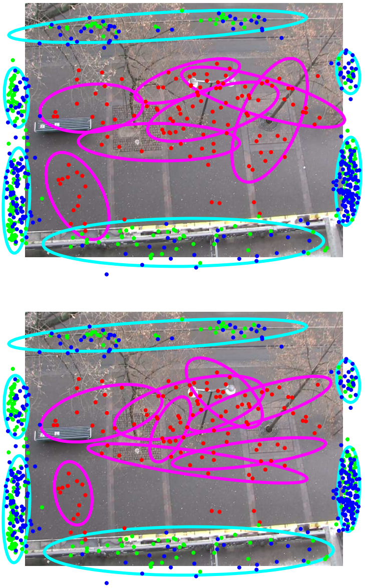


***Figure 3.4**: Four samples from the DPMM method. For each cluster we plot the elliptical region within 2 standard deviations. The color code is the same used in 3.3.*

distributed. We show the results of this clustering method in Fig. 3.4, where we ran the method once for the turning points (red) and once for the entrance and exit points (green and blue, respectively). Note that while the clustering method produces robust results for the entrance and exit points, this is not true for the turning points. This happens because these points do not seem to be normally distributed. While the transitions could be defined on random clusters of the corner points, we preferred to use the square cells in the grid layout because they offer a simple, stable, reproducible solution.

3.3.4 Simulation

The simulation of a group of people g starts with a transition from the *create* region to an entrance region, according to the transition probabil-

ity $p^{\mathcal{A}^t}(r_g|create)$. The composition of the group and the time at which a new group is created, can reflect the distributions of these quantities in the scene, or be arbitrary. Every time a new transition r_g is sampled for a group, for each subject i in the group a new destination point $\mathbf{z}_i^t$ is sampled uniformly from the destination region r_g. The motion from the starting region to the end region is demanded to the steering model of Sec. 2.2. Note that no navigation skill is used for the virtual agents, as the destination flow is made mostly of straight paths.
A group reaches the destination when each subject i in the group individually reaches her destination point $\mathbf{z}_i$, e.g. when the distance $d(\mathbf{p}_i^t, \mathbf{z}_i^t)$ is small. We define an indicator binary variable n_g^t, that is 1 when the group reached destination, otherwise it is 0. A group selects a new region r_i (possibly the same) at time t only when $n_g^{t-1} = 1$. More formally, the region transitions that determine the motion pattern of simulated pedestrians is modeled as

$$p(r_g^t|r_g^{t-1}, \mathcal{A}^t, n_g^{t-1}) = \begin{cases} \delta_{r_i^t, r_i^{t-1}} & n_g^{t-1} = 0 \\ p^{\mathcal{A}^t}(r_i^t|r_i^{t-1}) & \text{otherwise} \end{cases} \tag{3.2}$$

where $\delta_{i,j}$ is the Kronecker function that is 1 if $i = j$, and 0 otherwise. The dependencies of the variables described in this section are shown in the graphical model in Fig. 3.5. Note how the groups share the same quantities.

3.4 Mixed Reality

While learning the scene motion behavior helps capturing some of the scene semantic without the limitation of sticking to a collection of trajectories, it might still not be sufficient to reproduce the variety and the uniqueness of real trajectories. Adding real agents to a simulated scene, or the other way around, can be beneficial to enrich the simulation. The main problem is that the real agents are not aware of the simulated ones. Simply adding them together, results in general in unlikely configurations, especially when the density of the simulated scene increases. For example, a real agent might collide with slower simulated agents when the former is not in the field of view of the latter. We propose two possible solutions to address this problem.

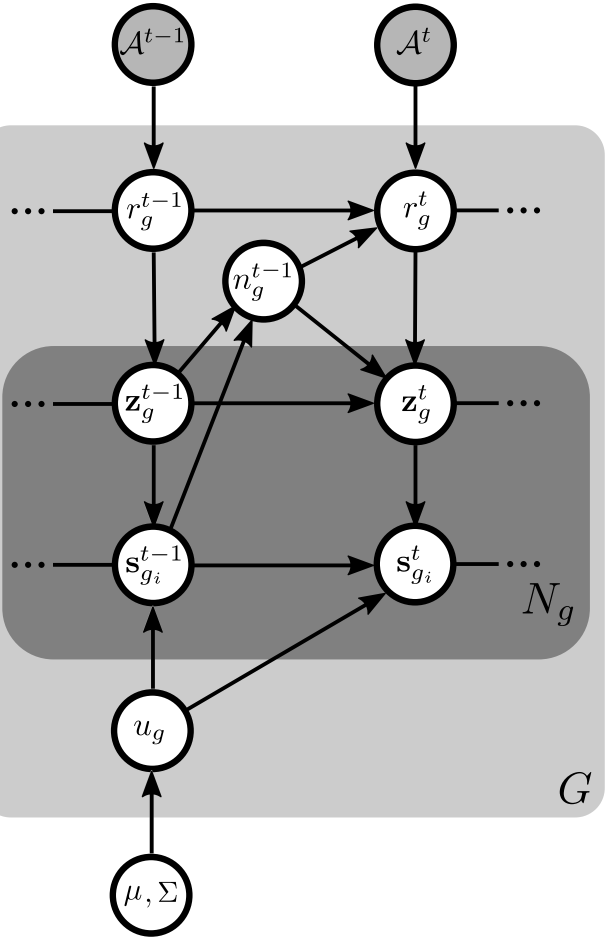


Figure 3.5: *The graphical model shows the dependency of the variables of the destination selection layer. Here we use plates to simplify the model.* G *is the number of groups and* N_g *is the number of subjects in the* g*-th group. Shaded nodes are observed variables.*

Sampling: The scene model presented in this chapter is composed by a probabilistic destination selection layer Sec. 3.3 and a deterministic steering one Chapter 2. Sampling from the model is relatively straightforward, as the model dependencies contain no directed cycle; see Fig. 2.6 and Fig. 3.5. As the samples are extracted from the same distribution, they are all valid candidates for the scene simulation. Our first solution is therefore to sample from the model until a certain criterion is met. In this chapter, we use as a criterion the number of collisions between real and virtual agents. A collision is counted every time the distance between two subjects is less than $0.4m$.

Path-following: Another solution is to allow real agents for small deviations from their real trajectory. We implement this solution by using a path following strategy for each subject. In particular, each real subject becomes a simulated one with the position of the following time step as destination. The desired speed is the speed necessary to complete the real step within one time step. In this way, the real subject still follows the original trajectory, but the limited freedom granted by the path following strategy favors a reduction in the number of collisions. Fig. 3.6 shows an example of the benefits coming from this strategy. While a reduction in the number of collision seems to be expected, one needs to check also that the deviations are not too big (see Sec. 3.5). Finally, this strategy can be combined with the previous one, by repeatedly sampling in order to minimize the number of collisions.

3.5 Experiments

In this section, we evaluate the motion model. We first compare LTA with other steering models. Next, we analyze the different mixing strategies proposed in Sec. 3.4. Finally, we provide the results of a user study that we carried out to investigate the user perception of the scene reproduction and mixed simulation.

3.5.1 The Circle Experiment

[Ondřej *et al.*, 2010] propose an interesting synthetic experiment to show some properties of their motion model. The experiment consists in hav-

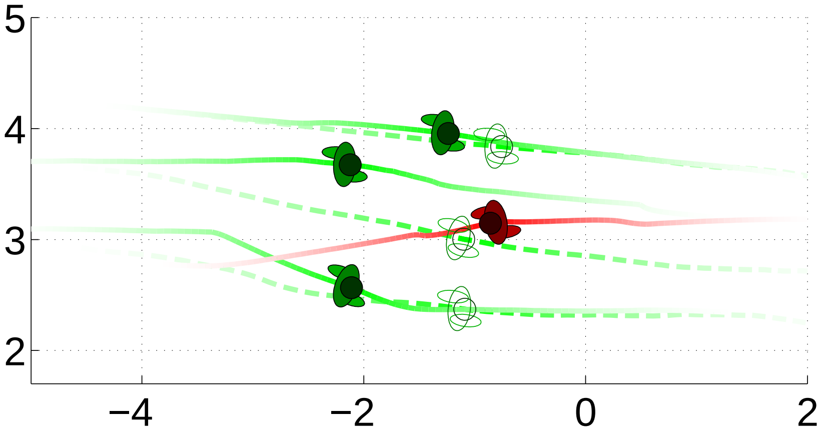


***Figure 3.6**: An illustration of the path-following strategy behavior. The simulation show the real agents (green) in the same group using the path following strategy while avoiding a simulated agent (red). The dashed lines and the transparent subjects show the original trajectory and position, respectively, of the real agents.*

ing a number of subjects starting in a circle configuration (see Fig. 3.7). Each simulated subject must reach the diametrically opposed point in the circle. One of the interesting properties of this experiment is that it highlights the difference between reactive and anticipating models. Anticipating models are able to avoid getting stuck in the center of the circle. Many agents avoid passing through the center and this reduces the density at the middle, thus creating no blocking situation.
Fig. 3.7 shows the result of our model (top row) when using a big T and compares it with the results of [Ondřej *et al.*, 2010] (bottom row). For a visual comparison to the Helbing's model and the RVO library, we refer to [Ondřej *et al.*, 2010].

The reason why both models succeed in avoiding getting stuck in a high density area is due to the fact that these models anticipate the collision and avoid it when it is still possible. To reproduce the problem encountered by more reactive steering models, we lower the time horizon T and show the result in Fig. 3.7 (middle row). Reducing the time horizon to such a small value corresponds to taking away most of the

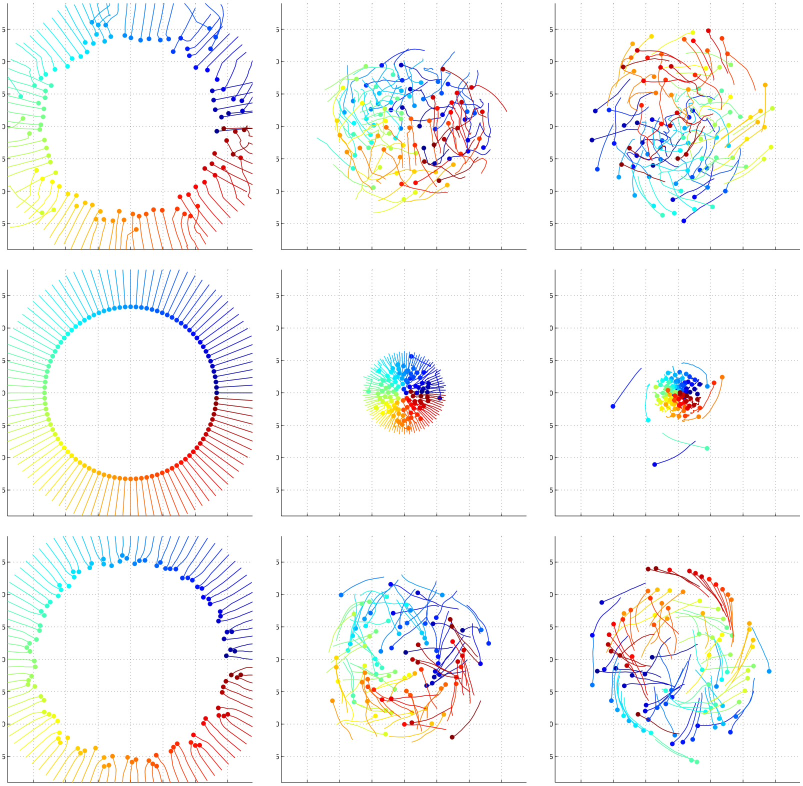
	LTA	VB	SF	RVO
max. travel time	57s	53s	90s	63s
% of slow walkers	0.87%	0.97%	30.4%	13.0%

Table 3.1: *Maximum travel time and percentage of slow walkers for the circle experiments. We show the results for our steering model (LTA) and compare it with the results reported in [Ondřej* et al.*, 2010] for their vision based model (VB), the Social Force model (SF) and the RVO library.*

prediction ability from the model. At each timestep, indeed, each simulated subject *feels comfortable*, as in the short time horizon no collision is predicted to happen. When the subjects, however, reach the center and start feeling each other's repulsion, it is too late to find an alternative route. Visual inspection of the first experiments shows a different behavior of our model when compared to [Ondřej *et al.*, 2010]. In the latter, subjects behave in a more ordered way. Group of subjects with similar motion patterns are clearly visible. In our model, instead, these patterns are less clear and shorter, subjects change direction of motion and sometimes stop. A reason why this happens might be that in our results a bigger portion of agents pass through the center. Specifically these are those agents that manage to get there first. In what circumstances one model is more realistic than the other is a question that would deserve to be investigated by comparison with a real realization of the experiment.
Tab. 3.1 shows the maximum traveling time and the percentage of slow walkers in the circle experiment. We report also the results published in [Ondřej *et al.*, 2010] for comparison. The quantitative results confirm what the qualitative analysis suggested, i.e. anticipating collision results in smoother and more effective motion.

3.5.2 User Study

To validate the quality of the reproduced scene, we set up a user study. We used 3 different video sequences (see Fig. 3.8):
Students : This ~ 3.5 minutes outdoor sequence has been provided by a third party [Lerner *et al.*, 2007]. The scene represents people walking



$t = 10s$ $t = 20s$ $t = 30s$

***Figure 3.7**: Circle experiment. Results for LTA with* $T = 15s$ *(top row), LTA with* $T = 0.1s$ *(middle row) and the model [Ondřej et al., 2010] (bottom row) at 3 different timesteps (in seconds). See text for comments.*

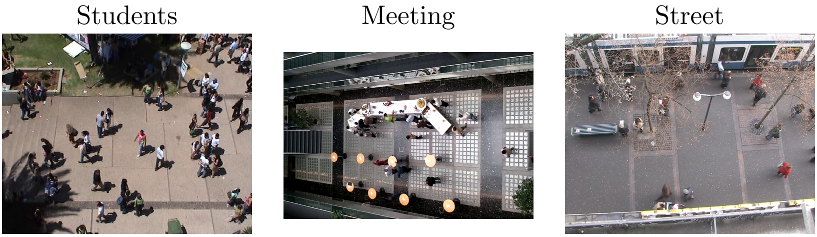


***Figure 3.8**: A frame from the image sequences used for the experiments.*

freely as there are almost no obstacles. Ground truth trajectory were available. Grouping relations have been manually annotated.

Meeting : This 10 minutes sequence has been recorded in an indoor hall during the coffee break of a project meeting. The main motion pattern is the one of people coming from the meeting room to the coffee table and then to the small tables. Ground truth was extracted with the help of the tracker.

Street : This sequence contains people walking in a busy street. Beside people walking along the street, there is also the motion pattern of people waiting and using the tram. Furthermore this sequence shows the event "tram", that is active when the tram is at the stop with open doors. We annotated the frames at which this happens. The ground truth was extracted with the help of the tracker.

Five mixtures of real and virtual agents were generated for these sequences: 0% (purely simulated), 25%, 50%, 75% and 100% (real sequence without the path-following strategy of Sec. 3.4). Each simulation lasts 45 seconds. For each simulation, the entrance time and the composition of the group was decided by the real sequence. For instance, if a group of 2 people enters at frame 500 in the real sequence, in each mixed simulation a group of 2 people is initialized in the *create* state at frame 500 and from there it would enter the scene in the region r sampled from $p(r|create)$. This was done in order to keep the different mixtures comparable. We render[2] the sequence in 3D with [Zhao *et al.*, 2009]. A simplified reconstruction of the environment has been used and the static obstacles have been modeled using the second alternative described in Sec. 2.3, *i.e.* we

---

[2] We recently conducted a similar study in 2D with similar results. Details can be found in [Pellegrini *et al.*, 2012].

regularly sample points along the obstacles boundaries and use them all in the simulation.

The interaction parameters were set to match those of the simulated scene. In Fig. 3.9, we report the 2D histograms that show the frequency of the displacement of two subjects, one of which sits in the center of the histogram $\mathbf{p} = \mathbf{0}$ with velocity positive only along the horizontal axis. Left-right symmetry (with respect to the subject in the middle of the histogram) is enforced. These histograms were used to fit the interaction parameters $\lambda_{I,1}, \lambda_{I,2}, \lambda_{G,1}, \lambda_{G,2}$, while we manually set for all the sequences $\lambda_S = 0.7$ and $\lambda_D = 0.3$. To fit the interaction parameters, we use a Gibbs measure interpretation of the energy terms, and we minimize the sum of squared residual between the histogram and

$$(1/Z)\exp\left(-\omega(I(\mathbf{d}) + G(\mathbf{d}))\right) \tag{3.3}$$

where I and G are from Eq. 2.7 and Eq. 2.12, respectively, and Z is a normalizing constant to ensure that Eq. 3.3 sums to 1. ω is a parameter that is used for the conversion from energy to probability. The minimization of energy in Eq. 2.16 used during the simulation is invariant with respect to a multiplication by a factor. The ω parameter is therefore only used for the fitting. Note that we use the energy terms I and G here as function of distance directly, rather than velocity. Fig. 2.2 shows the result of the fitted energy for the Street sequence.

The user study was made available to volunteers on the web. There were two different settings. In the first, the users were given a sample of the original sequence and in the next page they were shown all the 5 mixed simulations. This setting was repeated for the Students and the Street sequences. In the second setting, after the sample video of the original sequence, we show the user 3 videos of 45 seconds: the 100% real, the simulated one (0% real) and a completely simulated with a random transition matrix[3]. In this second setting, we used the Meeting sequence. The users could watch each video several times. The question was always: “The videos below refer to the scenario shown in the video of the previous page. How realistic does it look to you?”. The users had to answer with a score from 1 to 10 for each video. 28 people gave an answer for the Students sequence, 24 for the Street and 26 for the

---

[3] We still kept the feasibility of the transitions, for example, we allowed the transition to the *destroy* region only from the exit regions.

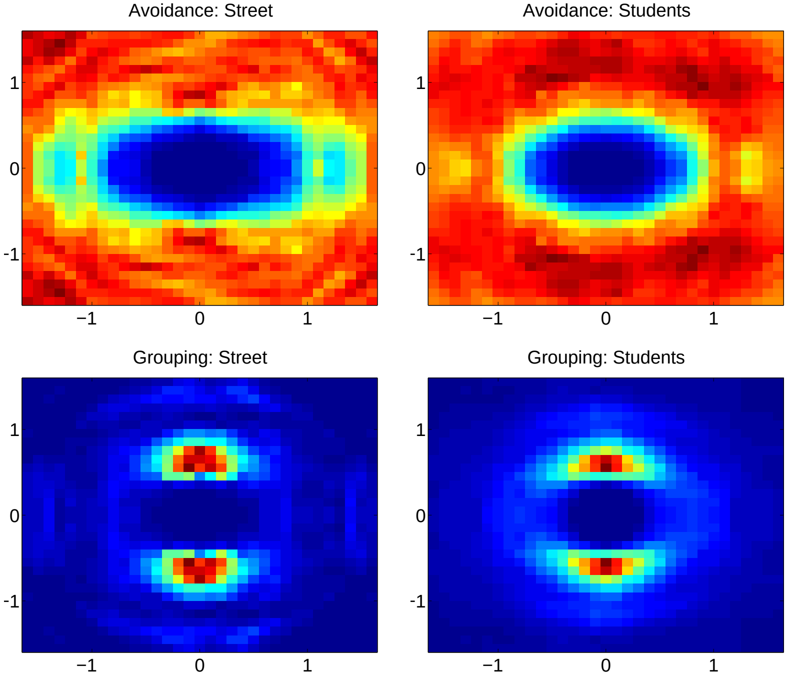


Figure 3.9: *The occupancy histogram, in egocentric reference system, for the sequences Street and Students. See text for comments.*

Meeting one. The results are shown in Fig. 3.10. For the first setting, the p-value is greater than 0.05 for both sequences (0.601 for Students and 0.053 for Street). For the second setting the analysis returns a p-value less than 0.05. This is true also when comparing the sequence simulated with the random transition matrix and the one simulated with the learned transition (p-value equal 0.02). Examples of the sequences used in the user study are shown in Fig. 3.12, Fig. 3.13. An example of the 2D rendering is shown in Fig. 3.11.

3.5.3 Mixing Real and Simulated People

As discussed in Sec. 3.4, an interesting possibility is that of using real and simulated agents in the same simulation. There we proposed 2 possible

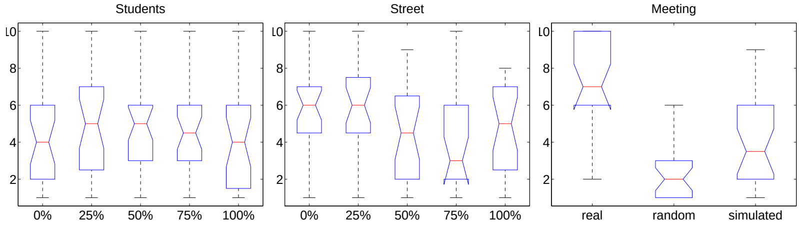


***Figure 3.10**: The results of the user study for the first setting (left, center) and the second one. The central red bar is the median and the box extends from the 25-th to the 75th percentile. The dashed lines go to the most extreme data points not considered as outlier.*

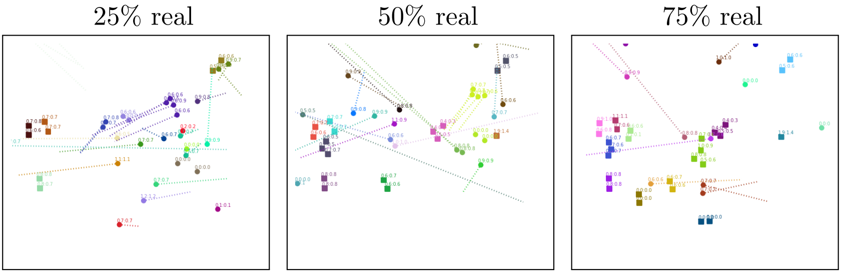


***Figure 3.11**: Three different blends of real (squares) and virtual (circles) agents for the simulated scene. The dotted lines shows the destination of the simulated agents. The numbers associated with each agent show the current speed and the desired speed for each agent.*

strategy (and their combination) to achieve such goal. Fig. 3.14, left, shows the comparison of the strategies. In detail, we extract 100 sample simulations of the Students sequence, with a 50% mixture, once with (red histogram) and once without (blue histogram) the path-following strategy, and for each sample we collect the number of collisions between real agents and virtual ones. The effect of the path-following strategy in reducing the collisions is evident. This, however, comes at the cost of slight deviations from the real agent's original trajectory. This is shown in Fig. 3.14 right. The mean of the histogram is $0.22m$, while the median is $0.12m$. This suggests that the path-following strategy is to be

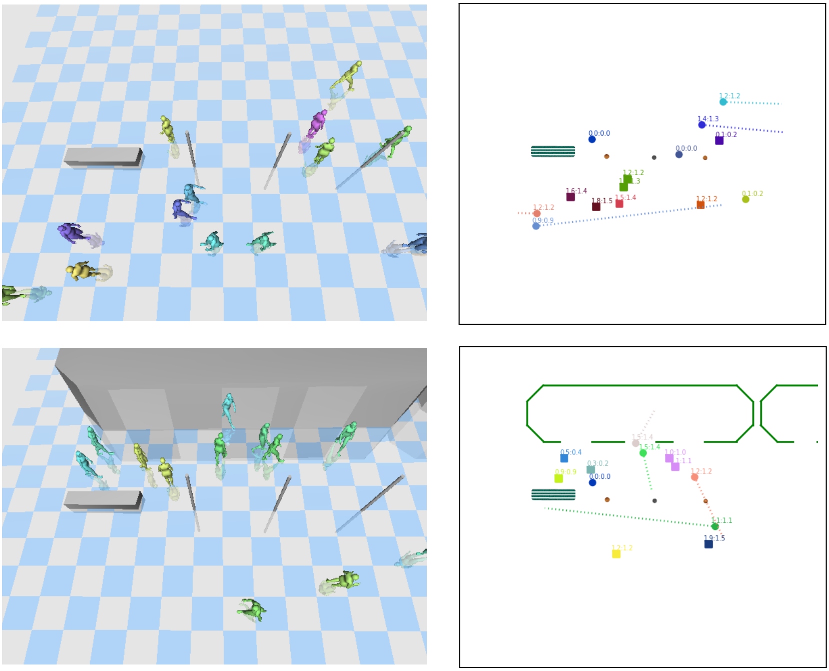


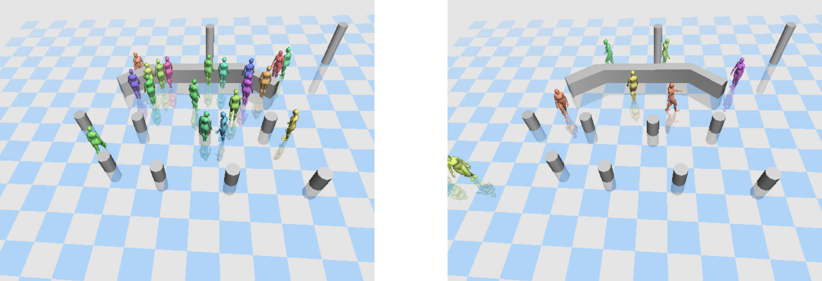
***Figure 3.12**: Two frames of the Street sequence. The left column shows the 3D rendered simulation, while the right column shows the 2D one. Note that the 3D rendering causes a small delay for the agents. Also the tram is treated as a static obstacle here. See Fig. 3.11 for an explanation of the 2D map symbols.*

preferred when no particular restriction is imposed on the fidelity of the real agent trajectories.

3.5.4 Timings and Limitations

In reproducing a specific scene, the most time consuming phase is certainly the track extraction from the real sequence. For the Meeting sequence, it took about 4 hours to extract tracks and group memberships, although most of the time the tracker requires no interaction. The processing of the trajectories and the computation of the transition matrix

Real agents Random transitions



Simulated agents Modified scene

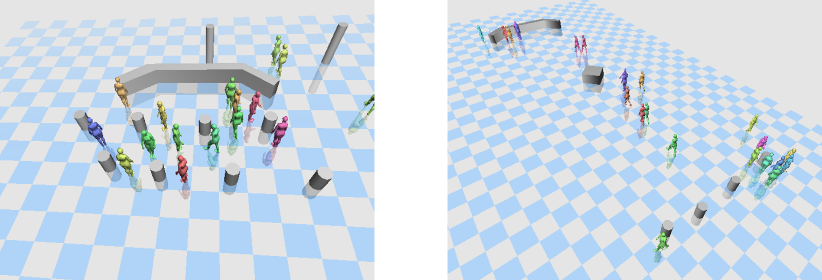


***Figure 3.13**: Examples from the Meeting sequence. **Top-left**: the position of the agents are obtained from the real trajectories. Note that the agents are divided between the big and the small tables. **Top-right**: the simulation is obtained by randomly changing the transition matrix. Note that the agents ignore the tables. The low density is due to the fact that agents leave the scene too early. **Bottom-left**: when the transition matrix extracted from the scene is used, agents repeat the motion pattern of the real scene. In this case more agents are at the small table with respect to the real sequence. **Bottom-right**: the user can interact and change the scene layout and the transition matrix. In this example, the transition matrix has been modified so that the agents are attracted by only one of the four small tables.*

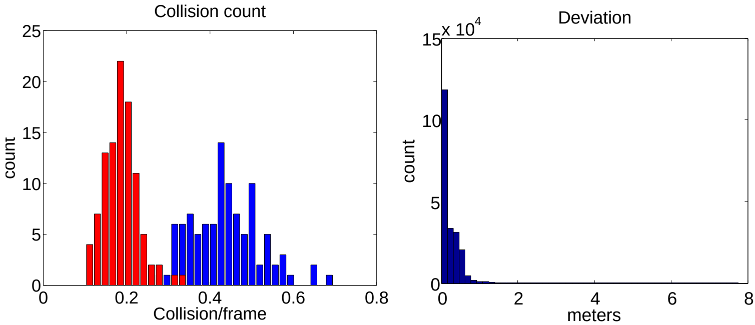


Figure 3.14: *Left: The number of collision per frame when using the path-following strategy (red) and without (blue). Right: The distribution of the deviation of the real agents from the original trajectory when using the path-following strategy.*

require little time, in the order of seconds, like the parameter estimation described in the previous section. The simulation of the Students sequence, 5400 frames, with an average of 40 agents per frame requires about $45s$ on an Intel i7 CPU @ 2.67GHz.

One limitation of our system is the first order assumption of the destination selection model, which is not always adequate. For instance, the area in front of the tram stop is used both by people entering the tram, waiting for the tram, and people getting off the tram. Since the first order model does not distinguish between these people, it averages the probabilities. Therefore, agents in the simulation that get off the tram might wait some time before they leave to another destination in the scene. Although this can be realistic as well, the probability with which it really happens is likely different from the one we captured in the transition matrix. A higher order model would alleviate this problem. However, the higher the order, the greater the need of real data, as the meaningful transitions become sparser in the set of possible ones.

Similar to other works, we rely on a linear velocity prediction model due to efficiency. Although the model shows a reasonable prediction accuracy of real trajectories, it needs to be investigated in the future whether non-linear models would improve the simulations. Since our simulation currently provides trajectories, the agents have been rendered after the simulation and do not follow exactly the trajectories. Although this introduces some artifacts that could be compensated by a more advanced renderer, the results demonstrate the capabilities of the proposed model.

3.6 Conclusions

In this chapter we have presented a scene dependent crowd simulation. Our goal was to produce a model that could adapt to the semantics of a particular scene, and reproduce it with small effort. Once the virtual agents behave similarly to real people, it is easier to integrate them in the same environment. We validated this possibility by conducting a user study. Users do not seem to be able to tell apart different mixtures of virtual and real agents. This is instead not true when the scene transitions do not reflect the semantics of the scene, as it is shown by the user replies for the Meeting sequence.

Even if the goal selection layer in this work is reduced to a destination selection, a series of interesting behaviors emerge. For example, people gather around tables thus forming new groups, even if we have no notion of group merging in our model. The action of getting in a tram when it comes, is also the pure result of a learned ”go-to” behavior. Although probably less visible, the customization of the steering model to the particular scene has been effective in reproducing its features.

4

Tracking with LTA

4.1 Introduction

Object tracking has seen considerable progress in recent years, with current systems able to handle long and challenging sequences automatically with high precision. The progress is mostly due to improved object models—either generic appearance models or detectors for specific kinds of objects—or better optimization strategies. One aspect that was hardly explored so far however is the dynamic model, another key component of every tracking approach. Typically, a standard first-order model is used, which does not account for the real complexity of human behavior.

In particular, physical exclusion in space is often modeled only indirectly, by allowing at most one detection to be assigned to a trajectory, while at the same time making sure that detections are sufficiently far from each other. In practice this amounts to non-maximum suppression in 2D image space. In situations where full occlusions are common (*e.g.* in street scenes seen by a street-level observer), such an image-based approach fails to adequately differentiate collisions from occlusions.

We believe that one main problem in this context is the dynamic model, typically a first- or second-order approximation applied *independently* to each subject, *e.g.* using an Extended Kalman Filter (EKF). The fact that people proactively anticipate future states of their environment during path planning, rather than only react to others once a collision is imminent, has largely been ignored in the literature. This goes to the extent that standard motion models do not even take into account the elementary fact that people have a destination, and hence steer back to

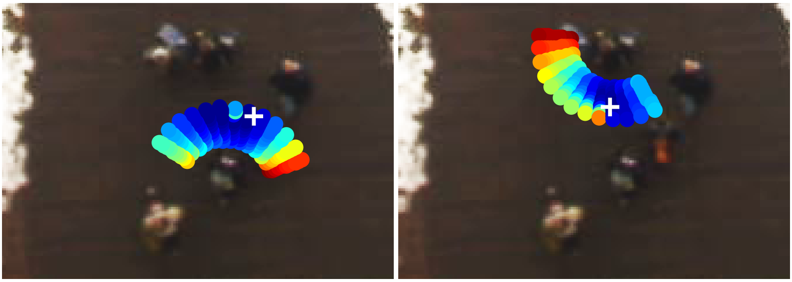


Figure 4.1: *While walking among other people, several factors influence short-term path planning. Smoothness of motion, intended destination, and interactions with others limit one's choice of direction and speed. In the example (same scene, two pedestrians' perspectives), blue indicates good choices for velocity, red signals "no-go"s. The white cross shows the actually chosen velocity.*

their desired direction after deviating around an obstacle.
Due to the complexity of human motion patterns, longer prediction horizons become unreliable; *very* short ones do not require sophisticated models, since displacements are so small that linear extrapolation is sufficient. Hence, the effect of LTA is best seen in busy scenarios with frequent short-term occlusions, or when framerate is low and the data association procedure is less reliable.

The steering model (Chapter 2) operates in physical world coordinates and can be applied to any tracker which operates in a metric frame. We show how the model parameters can be learned from birds-eye view data (Sec. 4.4), and apply it both in a simple patch-based tracker operating on oblique views, and in a detection-based tracker operating on footage from a moving camera (Sec. 4.5).

4.2 Related Work

Tracking is one of the core problems in the computer vision community (see [Yilmaz *et al.*, 2006] for a survey on object tracking). Early theoretical works on the topic can be tracked back to the work of [Sittler, 1964], where most of the multi-target concepts are introduced. In this section we review the literature related to the arguments treated in the next three chapters, putting more emphasis on the applications that use interactions within their framework.

Data Association. When multiple targets need to be tracked, the problem of associating new observations to tracks (possibly to no track, *i.e.* a false alarm) arises. This is a combinatorial problem [Morefield, 1977] that has been tackled in different ways in the literature [Cox, 1993]. [Morefield, 1977] proposes to first construct feasible tracks and then choose the best non-conflicting ones by formulating the problem as an integer programming one. [Reid, 1979] introduces the multiple hypothesis tracking (MHT) algorithm, later improved in [Cox and Hingorani, 1996], where the data association is carried out over multiple time steps. A set of hypotheses is maintained, where each hypothesis represents a possible assignments of measurements to targets. However in this approach the number of hypothesis quickly increases exponentially, and pruning strategies must be used to achieve feasibility. With emphasis on performance instead, the Joint Probabilistic Data Association (JPDA) [Fortmann *et al.*, 1983], proposes to use a weighted combination of all measurements for all the targets. More recently, in their "space-time event-cone tracking", [Leibe *et al.*, 2008] explicitly model physical exclusion between subjects in world coordinates.
Data association is usually carried out at the level of the single target. In [Gennari and Hager, 2004] the authors instead describe an extension of the JPDA based on group of individuals. Group merging and splitting are also handled by the method. Linear dynamics are still assumed, and false measurements and missed detections are not taken into account. With the same motivation, [Lau *et al.*, 2010] extend the MHT algorithm by hypothesizing both over data association and interaction events, like group splitting or merging. Both works carry out clustering of the observations at each time step.

In the approaches just described, the physical exclusion is restricted to the selection of the best trajectory hypotheses only—the important step of creating these hypotheses is done independently and does not cater for interactions, as we instead do in the work presented in the next chapters. In this respect, also the work of [Khan *et al.*, 2005] accounts for interactions within the motion model. The work proposes a joint particle filter, where the avoidance behavior is modeled with a Markov Random Field that penalizes overlapping targets configurations, *i.e.* a reactive repulsion based on distance only. The approach that we describe in Chapter 7 is similar to the one just described. A different model is however employed that, *e.g.*, includes grouping interactions and a different, distributed, inference strategy is used.

Motion model. When designing the motion model of a tracker, the classical approach is to assume linear dynamics and use a Kalman Filter [Welch and Bishop, 1995]. Extensions to this method have been proposed to handle some degree of nonlinearity, like the Extended Kalman Filter or the Unscented Kalman Filter. All these methods have problems when the system significantly violate the non-linearity assumption. An approach devised to handle such non-linearity is the to use particle filters [Isard and Blake, 1998] and its variants [Cappe *et al.*, 2007]. This method has been successfully used for tracking (*e.g.* [Breitenstein *et al.*, 2011]) and we describe a related approach in Chapter 7.

Beyond the choice of how to propagate the uncertainty in the system, there is the problem of deciding where subjects are headed to. The knowledge that subjects are headed toward goal points has been used to influence tracking in several works [Ali and Shah, 2008; Huang *et al.*, 2008; Kaucic *et al.*, 2005; Pellegrini *et al.*, 2009; Antonini *et al.*, 2006]. More interestingly, the specific motion patterns of a scene have been used to influence the target motion. The work of [Ali and Shah, 2008] in crowded scenarios is an interesting example of how scene constraints can help a tracking application. Inspired by cellular automata for evacuation dynamics (see Chapter 2.1), they use a set of Floor Fields as a prior for pedestrian motion. While this is directed at scenarios with a single dominant motion pattern, the work of [Rodriguez *et al.*, 2009] proposes an elegant solution to exploit the multiple patterns of motion in the scene. In both these works the inferred motion patterns are tied to a specific scene. More recently instead, [Rodriguez *et al.*, 2011] propose to use a video database of crowd behaviors to learn several possible motion priors.

Some other works have instead focused on building motion model independently of the specific scene. The work of [Antonini *et al.*, 2006], similarly to what we present in this chapter, improves the tracking performance by using a more informative motion prior. The modeling of the prior is however different from ours, as the authors use a discretization of the possible choices for the target, limited to a time horizon of about 1 second. Similar results to what we reported in [Pellegrini *et al.*, 2009], have later been confirmed by [Luber *et al.*, 2010], that show that substituting the constant velocity model with the social force model in the MHT framework results in better tracking performance.

Observation model. Fostered by recent progress in object detection [Dalal and Triggs, 2005; Felzenszwalb *et al.*, 2008; Gall and Lempitsky, 2009; Dollár *et al.*, 2011], there is an impressive body of work in people tracking-by-detection [Breitenstein *et al.*, 2011; Ess *et al.*, 2008; Li *et al.*, 2009; Okuma *et al.*, 2004; Wu and Nevatia, 2007; Zhang *et al.*, 2008]. All propose different ways of handling the data association problem (see below), but do not take advantage of any social factors beyond spatial exclusion principles.

In some circumstances, *e.g.* when there are not many targets in the scene or when the appearance is discriminative enough, trackers often rely mostly on the appearance model. One of the most promising approach to build such appearance model is that of online selecting the best features to track. In this context, [Grabner *et al.*, 2010] propose to use online boosting to do feature selection. [Kuo *et al.*, 2010] use AdaBoost to online learn discriminative appearance models, where the appearance model are then used to determine the association among the extracted tracklets. Pure appearance based trackers however suffer from the drift problem over long sequences. [Santner *et al.*, 2010] alleviate this problem by combining template matching, online random forests and optical flow in a cascade. Another way of combining trackers has been proposed by [Kwon and Lee, 2011]. In this work the uncertainty of the tracker is accounted with a sampling method that samples not only the states of the targets, but also the appearance model and the motion model. Using the appearance to discriminate between targets is not always a viable option. In some situations low resolution or similar target appearance [Khan *et al.*, 2005] prevent from relying on appearance only. To reduce this problem, [Stalder *et al.*, 2010] propose to use scene information, such as ground plane or background model, to further improve the detector output.

Some authors have investigated the possibility of exploiting low level information for tracking. An interesting approach to couple low-level image segmentation and high-level occlusion reasoning and tracking has been proposed in [Wang *et al.*, 2009]. The authors use a graphical model framework to jointly process the image support and enforce ordering constraints. [Andriluka *et al.*, 2008] instead carry out tracking together with a simplified articulated pose detection that enables their approach to exploit gait information for walking pedestrians.

Trajectory Prediction. Predicting future locations of people is interesting both for computer vision and robotics. In robotics, predicting people motion is used in path-planning applications. In this context, [Ziebart *et al.*, 2009] learn people interaction with environment features and use this information during path-planning to avoid the robot hindering people natural behavior. With a similar motivation, [Trautman and Krause, 2010] proposes an elegant solution to account for the fact that when robot plans its path through the crowd, also the people in the crowd cooperatively avoid the robot. They use Interactive Gaussian Processes to model the predicted path, and employ importance sampling to carry out the inference. [Scovanner and Tappen, 2009] propose a steering model similar to LTA. The main difference is in the way the avoidance component is built. In their case the avoidance accounts for the next position of the steering subject together with a sequence of positions of the pedestrian to be avoided, obtained through linear velocity prediction. LTA instead uses only a single point for interaction, namely the point of maximum approach, possibly within a time horizon. It must be noted that their model is designed to allow parameter learning, in their case carried out with Mode Variational Learning. [Vasquez *et al.*, 2008] predict trajectories with a Hidden Markov Model, but do not account for interactions among agents. Furthermore their system is scene specific. [Yamaguchi *et al.*, 2011] build a prediction model that explicitly exploits grouping and destination estimates. They show how the destination prediction accuracy does not improve significantly when more frames from the past are available, while an improvement is observed in the group prediction accuracy.

Tracking with Interactions. Recently, some works have shown the benefits of exploiting grouping interactions for tracking [French, 2006; Pellegrini *et al.*, 2010; Pellegrini and Gool, 2012; Choi and Savarese, 2010; Gennari and Hager, 2004; Lau *et al.*, 2010]. In particular the work of Choi *et al*. [Choi and Savarese, 2010], published contemporary with ours [Pellegrini *et al.*, 2010], is closely related to our approach. Among the differences that separate the two works, we propose a model that propagates the grouping interactions and uses a different inference strategy. Albeit with a different motivation, the work of [Grabner *et al.*, 2010] exploits the correlation among features to predict the position of possibly invisible targets. A Generalized Hough Transform is used to let these correlated features vote for the target of interest. Similarly, [French, 2006] exploits the correlation among targets by having one target using the motion parameters of another target with probability proportional to the correlation between the two.

Recently several authors tackled the multi-target tracking using a graph formulation of the problem, by representing single or sequence of detections as graph nodes and modeling interactions through, possibly weighted, edges. In this context [Brendel *et al.*, 2011] build a graph with tracklets as nodes and edges connecting two tracklets that have at least one detection in common. The set of nodes that (approximately) solve the maximum weighted independent set problem on this graph is then used to carry out multi-target tracking. Interactions such as grouping or avoidance are not explicitly modeled, but tracklets that rapidly change the velocity correlation over time are penalized by connecting them with a weighted edge. [Yang *et al.*, 2011] also cast the multi-target tracking problem in a graphical model framework. Differently from the approach we propose in the next chapters, their nodes represent pairs of tracklets built on the output of a detector. The learned pairwise terms among the nodes encode both motion and occlusion dependencies. Finally, the problem is solved using simulated annealing, starting from an initial solution obtained by applying the Hungarian algorithm on a unary-only instantiation of the graph. Single tracklets are modeled instead by nodes in [Song *et al.*, 2010], while the edges associate pairs of tracklets. In their paper, the prediction in motion and feature space of one tracklet is used to compute a similarity measure with other tracklets. This similarity is then used as a weight for the graph edges and the optimal solution is computed using the Hungarian algorithm. Incorrect associations are compensated for by using a graph evolution strategy that adapts the weights of the graph based on long-term consistency of the connected tracklet features. Other authors [Zhang *et al.*, 2008; Leal-Taixé *et al.*, 2011; Pirsiavash *et al.*, 2011] use a minimum cost-flow algorithm. [Leal-Taixé *et al.*, 2011] take into account both avoidance, similarly to the Social Force model, and some form of grouping behavior. These approaches, differently from what we propose in this work, solve for the Maximum A Posteriori (MAP) estimate of their equivalent probabilistic formulation of the problem.

Understanding interactions. Tracking is often used as a preprocessing step to carry out scene analysis. The knowledge of people tracks is an effective input for understanding people interaction and analyzing group composition. In Chapter 6 and Chapter 7 we show how we can extract and use similar information *while* tracking, and not *after*.

Few works have focused on the task of interaction classification. [Oliver *et al.*, 2000] uses Coupled Hidden Markov Model to model five simple pair interaction patterns like "approach, meet and go separately". The authors use a Kalman filter and background subtraction to extract trajectories that are used to build the feature vector employed in the interaction classification. While in this work the possible interaction behaviors are manually specified, in [Galata *et al.*, 2002] a set of pair interaction patterns are learned using a variant of Vector Quantization. These learned prototypes are then employed as states in a Variable Length Markov Model. This work is applied in the context of traffic surveillance, but the formulation is general and applies to other types of targets. [Bose *et al.*, 2007] track foreground blobs in the scene and then use proximity and coherent motion to classify the tracked blobs into fragments of objects, objects and group of objects.
Other works have instead directly carried out group detection, *i.e.* extraction of the groups in the scene. [Ge *et al.*, 2009] uses a bottom up clustering algorithm to discover the groups in the crowd while in [Ryoo and Aggarwal, 2011] the group detection is carried out jointly with group activities. In [Cristani *et al.*, 2011] the groups are detected using the head orientation in a voting scheme. A probabilistic group relationship is instead used in [Chang *et al.*, 2011] to recognize group behaviors.

In these works tracker is applied before the interaction classifications and the features used for the task are simple quantities, like relative distance, velocities and accelerations. We also show in Chapter 6 that when trajectories are available the task of classifying whether two subjects belong to the same group or not is generally feasible. Furthermore, we investigate to what extent the prior knowledge about interactions can be built *into* the tracker (Chapter 7).

4.3 Model Setup

In this chapter we use the steering model introduced in Chapter 2 for a tracking application. In particular, we use a simplified version of the model, as no grouping term is used ($\lambda_{G,1} = \lambda_{G,2} = 0$) and the interaction term behaves symmetrically ($\lambda_{I,1} = \lambda_{I,2}$). Also, for the visibility discussed in Sec. 2.2.2 we did not use a Delaunay triangulation, rather all the subjects within a certain distance and a certain angle are visible to the steering pedestrian. More details are given below. For the static obstacles, we use the first alternative described in Sec. 2.3, *i.e.* the single closest point to the steering pedestrian represents the static obstacle at each timestep. Finally the time horizon is not truncated to a finite value ($T = \infty$). The LTA minimization, as in Chapter 2, is be carried out using gradient descent. The model parameters are learned from real data, as it is explained in the next section.

4.4 Training

The model, as defined in Chapter 2 and further specified in the previous section, has six free parameters, which need to be learned from training sequences: the interaction parameter λ_I, the direction weight λ_D (Eq. 2.14) and the speed weight λ_S (Eq. 2.15), the radius of interest σ_w (Eq. 2.10), the "peakiness" β of the subject's field of view (Eq. 2.11), and the update rate α (Eq. 2.1). We fix Δ (Eq. 2.2) to 0.4 seconds.

To train our model, we have recorded two data sets from birds-eye view and annotated them manually. This gave a total of 650 tracks over 25 minutes.[1] A sample image including annotation can be seen in Fig. 4.2. In both scenes, goal points were labeled and the desired direction for each subject was set towards the closest goal. For each pedestrian, the

---

[1] Data and videos available at www.vision.ee.ethz.ch/~stefpell/lta

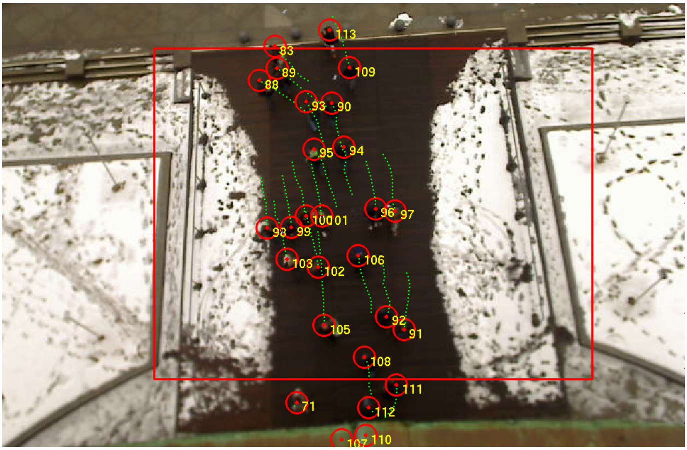


***Figure 4.2**: Sample frame from one of the training sequences.*

λ_I	σ_W	λ_D	λ_S	β	α
3.84	2.088	2.33	2.073	1.462	0.730

***Table 4.1**: Model parameters obtained from training sequences.*

desired speed was set to the mode of his speed histogram. The field-of-view was restricted to ± 90 degrees (i.e., $\nu = 0$ for $|\phi| > \frac{\pi}{2}$ in Eq. 2.11). People standing or strolling aimlessly were ignored.

To find an optimal set of parameters we have experimented with two optimization strategies, namely gradient descent starting from multiple random initializations, and a variant of genetic algorithms (GA). We found that among the returned local optima of the parameters vector, several performed equally well. For the following experiments, we always use the local optimum with the lowest error (which resulted from the GA optimization).

In one iteration round, each subject is simulated in turn, holding the others fixed at the ground truth. The simulation is started every 1.2 seconds along the subject's path, and continues for 4.8 seconds, similar to [Johansson *et al.*, 2007]. The sum of squared errors (distances to ground truth) over all simulations in the round is minimized.

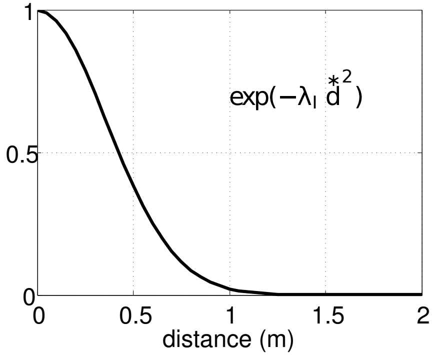


***Figure 4.3**: The interaction energy as function of the distance of maximum approach (see Eq. 2.7). Note that here we set $\lambda_{I,1} = \lambda_{I,2} = \lambda_I$*

We obtained the parameters given in Tab. 4.1. At first glance, $\lambda_I = 3.84$ looks reasonable (see Fig. 4.3), as it suggest that at an expected distance of about $1.3m$ almost no repulsion is felt; $\sigma_w = 2.1$ means that people further away than ≈ 6 meters do not influence path planning; β suggests a relevant peak of attention in the center of the field of view. Note that the restricted field-of-view and the value of σ_w imply that pedestrians are actually only aware of a limited portion of the scene.

4.5 Results

To experimentally evaluate the trained steering model, we test it in three different settings. First, we measure its mere quality as a predictor, which is *e.g.* of interest for path planning in robotics. Then, we apply it inside two tracking methods, a simple patch-based tracker, as well as a state-of-the-art multi-person tracking system.

4.5.1 Prediction

To test the prediction performance of our model, we use annotated data provided by the authors of [Lerner *et al.*, 2007]. The video shows part of a shopping street from an oblique view. We evaluate on a subsequence of about 3 minutes @ 2.5 FPS containing 86 trajectories annotated with splines. With the same simulation setting used during training (see Sec. 4.4) this yields $\approx$ 300 simulations. A homography from image to ground plane was estimated from four manually clicked points on the footpath to transfer image to world coordinates. As destinations we chose two points far outside the left and right image borders, which holds for most subjects.

We compare our model with a simple baseline ("LIN"), that merely extrapolates using the previous velocity, and with a re-implementation of the *social force* model ("SF") with elliptical potentials [Johansson *et al.*, 2007]. Parameters for the latter are learned using the procedure discussed in Sec. 4.4. For our LTA model, we explore two possible parameter sets: the first one was trained without interaction term, adding only the drive towards a destination ("DEST"), whereas the other one ("LTA") also caters for interaction among subjects.

As error measure, the average Euclidean distance between predictions and ground truth is measured in each simulation step. The experiments show an improvement of 6 % in prediction error for the LTA model compared to SF and DEST, and of 24% compared to the LIN model. A closer look at the distribution of the errors sheds more light on the differences between models. For this purpose, we define a trajectory as *correctly predicted* when for each timestep of its simulation, the distance from prediction to ground truth lies within a threshold H. The curve in Fig. 4.4 shows the result of this analysis, plotting the percentage of the correctly predicted trajectories over varying H. At a threshold of 1 meter, $\approx$50% of the trajectories are already correctly predicted using linear extrapolation (LIN). Adding goal-direction (DES) increases the correctly predicted trajectories to $\approx$63%. The SF model performs only slightly better than the DES model. Another $\approx$7% boost is achieved using LTA, reaching a total of $\approx$70%.

There are two issues to note here. Firstly, the scene is only moderately crowded, and a large part of the trajectories are almost straight. For these, all models give satisfactory results, which washes out the average difference. Secondly, the error distribution of LTA has a light but long tail with a small number of very large errors. These happen when the model in its present deterministic form avoids other pedestrians by

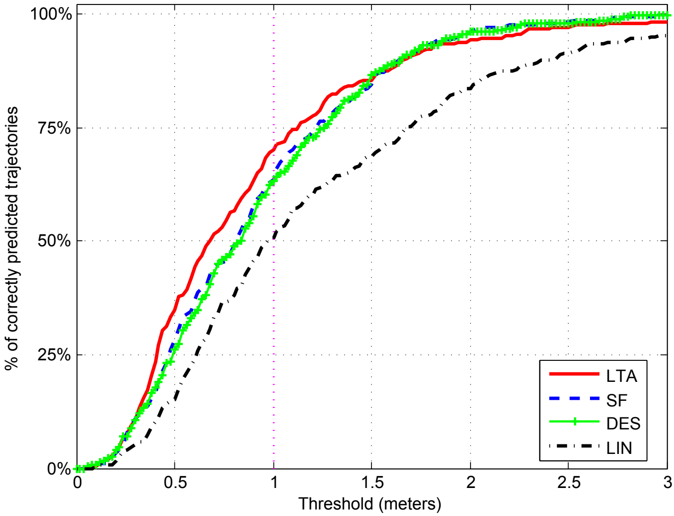


Figure 4.4: *performance of the LTA model (solid red) against a trained model that uses destinations but no interactions (crossed green), the social force model [Johansson et al., 2007] (dashed blue) and simple linear extrapolation (dash-dot black).*

walking around the wrong side, see Fig. 4.5. Although from a tracking perspective, bumping into an obstacle is a no less severe failure than passing it on the wrong side, the latter adds twice as large errors and thereby distorts the comparison. A stochastic variant of the model could help here, as we show in Chapter 5.

4.5.2 Patch-based Tracking

To highlight the effect of the dynamic model and compare it to the LIN model, we have implemented a simple patch-based tracker, using the normalized cross-correlation (NCC) as similarity measure. In the first frame a rectangular patch is manually initialized at each person's location $\mathbf{p}_i^0$ as appearance model, and the speed of all targets is initialized

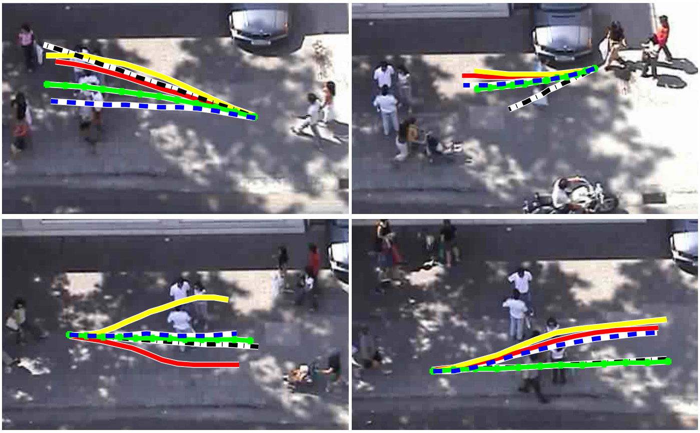


***Figure 4.5**: Example extrapolations: the model smoothly avoids the standing crowd* (top left, *yellow=groundtruth*)*, sometimes however suggests meaningful, but wrong paths* (bottom left)*. Using only goal-directed prediction is effective in some cases* (top right) *but in general better prediction is obtained by taking into account interaction among pedestrians* (bottom right)*.*

to $\|\mathbf{v}_i\| = 0$. At each new time step t, the target location $\mathbf{p}_i^t$ is predicted with the dynamic model, and a Gaussian centered at the prediction gives the location prior $P_{pred}(\mathbf{p}) = \frac{1}{Z}\exp\left(-(\frac{\|\mathbf{p}-\mathbf{p}_i^t\|}{2\sigma_{pred}})^2\right)$. In the surroundings of the predicted location, the squared exponential $P_{data}(\mathbf{p}) = \frac{1}{Y}\exp\big(-(NCC(\mathbf{p},\mathbf{p}_i^0)-1)^2\big)$ is employed as data likelihood, and the maximum of the posterior $P_{pred} \cdot P_{data}$ gives the new target location.

This simple tracker was applied to short, interesting sub-sequences of the footpath sequence (non-overlapping with the ones used above). For the dynamic model, we plug in either the LIN (constant velocity) model or our LTA model, leaving the other parameters unchanged. For the LTA model, the desired direction (standing, left-to-right, or right-to-left) is set for each person according to their last displacement, and the desired speed is set to a constant value for all people.

Tracking was performed at 2.5 FPS, leaving 0.4 seconds between consecutive frames. In this scenario with low framerate, multiple interactions,

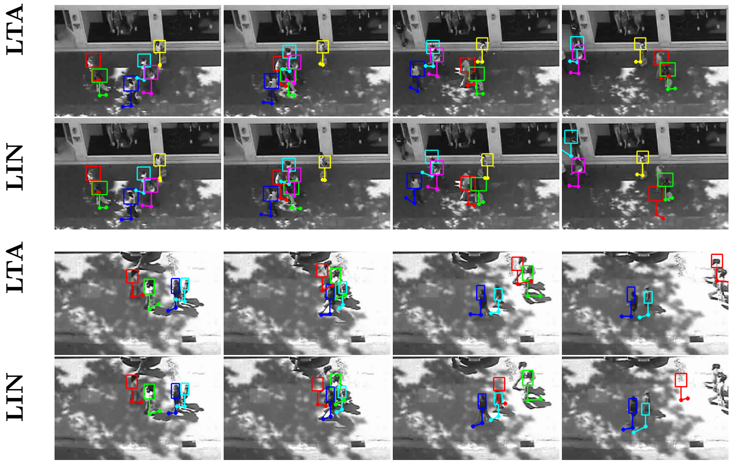


Figure 4.6: *LTA model vs. constant velocity (LIN) model. Selected frames from two tests with the patch-based tracker.* Top: *When using the LTA model, the pedestrian marked in red is constrained by people walking nearby. The LIN model overshoots when he maneuvers around an oncoming person and loses track.* Bottom: *the LIN model for the person marked in red makes a significantly wrong prediction and loses track, whereas the LTA model tries to avoid oncoming people and predicts correctly. Note also how in both examples the persons marked in cyan drift away at the end, because they are not steering towards a target direction.*

and low data quality, a strong dynamic prior is important to enable tracking at all. As can be seen in the examples of Fig. 4.6, the simple constant-velocity model loses track of several targets, when they pass others and have to adjust their speed and direction. The examples also show how the trajectories fail to swing back without a target direction. On the other hand, LTA successfully tracks all people in the two examples.

4.5.3 Tracking with a Moving Observer

To further demonstrate the versatility of the approach, we apply the LTA model (as learned from birds-eye view) to tracking from a moving observer. We use the tracking-by-detection framework [Ess *et al.*, 2008], and plug in both the LIN and the LTA models for modeling pedestrian dynamics. Both versions are then evaluated on two (publicly available) sequences from that work.

In a nutshell, the approach projects the output of a pedestrian detector—in our case the HOG framework [Dalal and Triggs, 2005]—to 3D world coordinates with the help of visual odometry and a ground plane assumption. The tracking system then generates a set of trajectory hypotheses based on the object detections and a dynamic model, and prunes that set to a minimal consistent explanation with model selection. This pruning relies on the assumption that all actual trajectories are present in the set of hypotheses, thus requiring correct tracking even when no data is available to immediately correct the motion model, mainly during to occlusions. Here the LTA model comes into play.

To make the method amenable to our problem formulation, we adapt as follows: first, instead of creating all trajectory hypotheses independently, we introduce a trajectory extension step that updates all currently active object hypotheses in parallel, making them fight for available detections similar to the greedy approaches used by [Wu and Nevatia, 2007; Okuma *et al.*, 2004]. This then allows the second, crucial change: in the extension step, we apply the LTA model for each hypothesis in turn, making them anticipate the other subjects' movements in order to avoid them. Especially during occlusion, this ensures that blind trajectory extrapolation takes into account other subjects, and increases the chance that a subject's trajectory leaves the occlusion at the right position, so that tracking can continue correctly. To also detect static obstacles, we additionally project the depth map from stereo images onto a polar occupancy map.

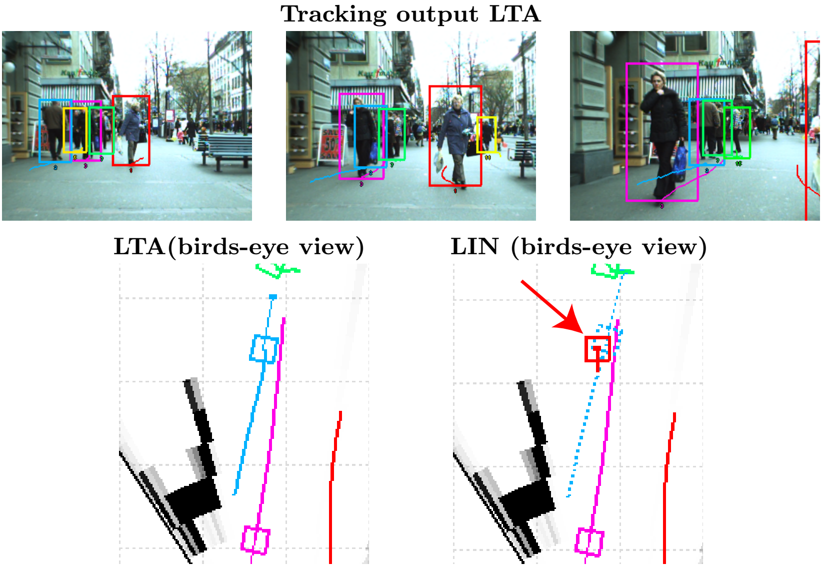
LTA requires a desired orientation and velocity. Assuming very little scene knowledge, we set the desired orientation parallel to the road, pointing in the respective pedestrian's previous direction. The desired velocity is set to the last measured speed of the hypothesis.

As the tracker builds on a quite reliable set of pedestrian detections, we expect an advantage of the LTA model mainly in case of occlusion. The improvement is therefore bounded by the frequency of occlusion events. Then, LTA's extrapolation which is constrained by other agents should outperform a standard linear model, thus preventing possible data association problems when the occlusion is over.

To quantitatively relate the two approaches with each other, we compare tracking output with annotated ground-truth using the CLEAR evaluation metrics [Bernardin and Stiefelhagen, 2008], which measure ID switches and the percentage of false negative / false positive bounding boxes. In Tab. 4.2, we compare the two dynamic models by varying the threshold on the Mahalanobis distance d used in the data association. The reasoning behind this procedure is the intuition that a larger search area could possibly compensate for the disadvantages of a less accurate prediction. When using LTA, the number of ID switches is constantly lower, while the number of misses and false positives stays about the same. While consistent, the automatic evaluation tends to over-estimate the number of ID-switches with increasing number of occlusion events. For $d = 3$, we thus manually re-counted the ID switches for the two sequences. In the first sequence, using LTA yields 31 as opposed to 36 ID switches with LIN. In the second sequence, these figures are 18 (LTA) and 26 (LIN). Here, many people leave the field of view and enter again, which is always flagged as a new ID by the tracker. Leaving out these "unrecoverable" cases, the last comparison gets down to 10 (LTA) vs. 18 (LIN), a 44% improvement.

	ID switches				misses				false positives			
	1.5	2.0	2.5	3.0	1.5	2.0	2.5	3.0	1.5	2.0	2.5	3.0
	Seq#1											
LIN	55	55	51	48	0.29	0.28	0.28	0.28	0.19	0.19	0.19	0.19
LTA	48	42	45	41	0.28	0.28	0.28	0.28	0.19	0.19	0.19	0.19
	Seq#2											
LIN	35	33	31	31	0.21	0.21	0.21	0.21	0.08	0.08	0.09	0.09
LTA	31	30	26	25	0.21	0.21	0.20	0.20	0.08	0.09	0.08	0.09

***Table 4.2**: Comparison of the dynamic models for differing data association thresholds based on the CLEAR evaluation metrics.*

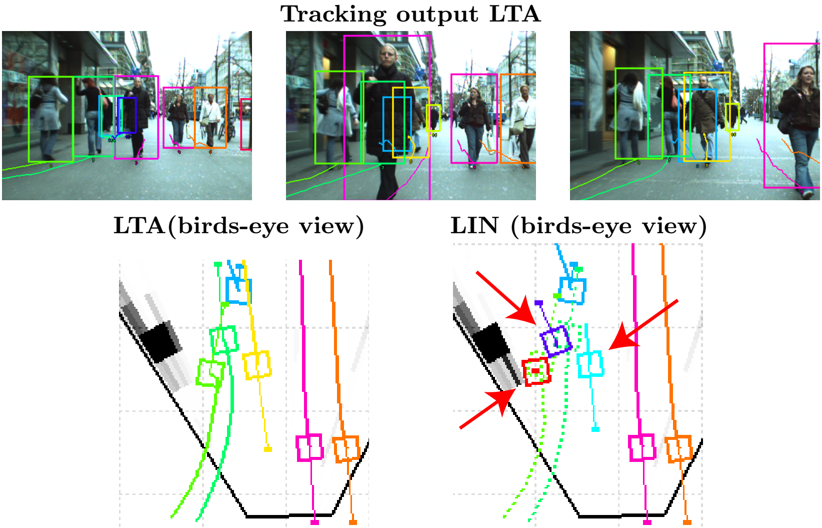

Tracking output LTA

LTA(birds-eye view)

LIN (birds-eye view)

Figure 4.7: *Examples where LTA improves the performance of multi-body tracking.* ***First row***: *short sequences with occlusion events, tracking results with LTA.* ***Second row***: *birds-eye view for the middle frame, using LTA(left) and using linear model (right). Black areas are static obstacles, red arrows mark ID switches, dotted lines show the pre-switch trajectories still being extrapolated—these disappear after ≈5 frames as they fail to find supporting detections. Note that the man on the left is successfully recovered from occlusion.*

A few interesting situations from the two sequences are shown in Fig. 4.7, Fig. 4.8 and Fig. 4.9. The first row shows the sequence including the occlusion event as tracked by LTA, then in the second row two plots in birds-eye view contrast the results for LTA with those for LIN. Note the ID switches (red arrows), and the missing track in the third example(Fig. 4.9). This last example is especially interesting, because the person in the very front is only detected as a static obstacle. Nevertheless it influences the man in the striped sweater, who successfully steers around it, whereas LIN looses track.



Tracking output LTA

LTA(birds-eye view)

LIN (birds-eye view)

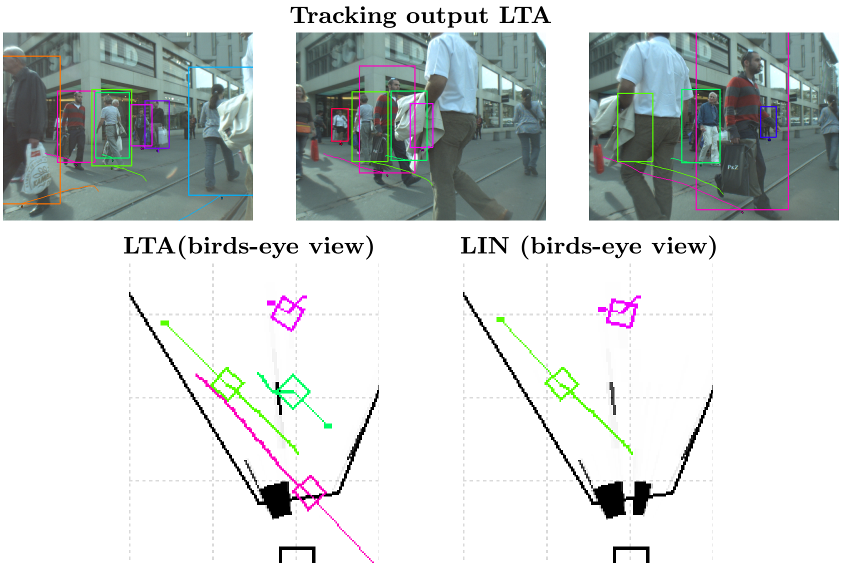
Figure 4.8: *Constrained by the oncoming person, both ladies and the oncoming man are picked up again.*

4.6 Conclusion

In this chapter we have presented the integration of the steering model of Chapter 2 in a tracking application. The LTA model is not dependent on any specific tracker or scene, it merely needs the subjects to reside in a space that allows one to calculate metric distances.

The LTA model takes into account both simple scene information in the form of destinations or desired directions, and interactions between different targets. As it operates in world coordinates, the model can be trained offline on training sequences, and then applied elsewhere. We have also shown experimentally that the model yields better predictions, and consistently improves tracking performance compared to dynamic models which disregard social interaction. The improvement comes at negligible computational cost (less than 10 ms for a frame with 15 subjects).

We draw attention to an additional lesson learned from the study: a person's destination is valuable information and should always be used.


Tracking output LTA

LTA(birds-eye view)

LIN (birds-eye view)

Figure 4.9: *While the man in the front is not detected, he is integrated into the obstacle map, thus constraining the man in the red-black sweater.*

While this finding is by no means new, *e.g.* [Kaucic *et al.*, 2005; Huang *et al.*, 2008], we emphasize that it is true even when the destinations are incomplete or inaccurate. We have shown that even roughly guessed target directions help to make more meaningful predictions. This is particularly interesting for the case of mobile cameras, where the destination cannot be learned from continuous observation.

In the present state, we do not model groups of people walking together. This would be possible, as the steering model supports this feature. However it still remains the problem of how to assign people in the scene to the same group. In Chapter 6 and Chapter 7 we propose a solution to this problem. A further interesting direction is the stochastic application of the proposed energy functional. We explore this possibility in the next chapter.

5

Stochastic LTA for Prediction

5.1 Introduction

The behavioral patterns that regulate pedestrian interactions have been studied in the field of social psychology for a long time [Hall, 1966; Freedman, 1975; McPhail and Wohlstein, 1982; Goffman, 1971]. We know from several studies that these patterns are subject to variation that depend on culture, gender and other factors. Nevertheless, to a certain extent, it is possible to exploit this knowledge by the means of models for pedestrian motion prediction whose goal is not only that of describing, but also and moreover, of synthesizing and predicting. Accurately modeling a pedestrian's future path in a deterministic way is extremely difficult: on the one hand, the observed information is incomplete, either because it is invisible to the camera (but visible to the pedestrian in the scene), or because it is part of a pedestrian's individual preferences (*e.g.* some people like to walk in the shade, others do not). On the other hand, model complexity is limited by computational power and inclusion of further elements should be handled with care. Instead of modeling more and more such elements, *e.g.* individual preferences or scene specific behaviors, an alternative approach is that of making the model more robust. A probabilistic motion model is robust to fluctuations in the behavioral patterns of the modeled pedestrians.

In the following, we show how a stochastic motion model can handle situations as those depicted in Fig. 5.1, where the two possible evading trajectories seem equally likely. We use the stochastic formulation of the LTA model presented in Chapter 2, referred to as sLTA. sLTA uses the

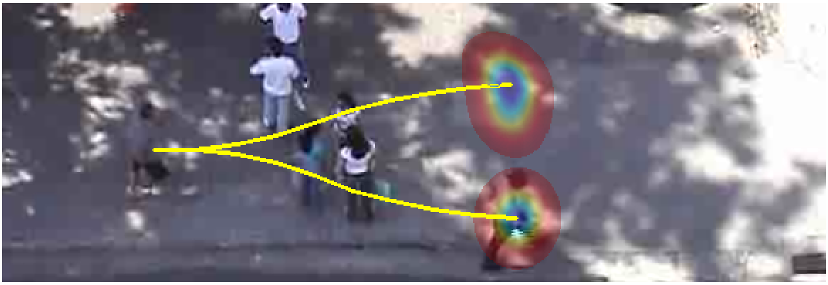


Figure 5.1: *When moving through a scene, a person takes a variety of factors into account, such as steering clear of other people. In many cases, the prediction of the motion cannot be well described by a deterministic algorithm: in the above example the pedestrian on the left hand side could either evade the group by going on its left or right side, as indicated by the yellow lines. We therefore propose a stochastic extension of LTA that can deal with the uncertain future motion of a pedestrian.*

same energy potential formulation of the original LTA model, but within a Gibbs measure to turn the potential into a probability.

One specific question that is then addressed is the usability of the motion model for tracking. In Chapter 4, it has already been shown that a motion prior has better predictive power than linear extrapolation and that a tracker can benefit from its use in situations where the observation is unreliable (*e.g.*, during occlusions). Here, we investigate this issue further by conducting a set of systematic experiments using an appearance-based tracker.

5.2 Stochastic LTA

To account for the uncertain future motion of a pedestrian, we extend LTA in a multi-hypothesis fashion. We term the new model sLTA. In this chapter we use the simplified version of the steering model described in Sec. 4.3. The joint state of all the subjects at time t is given by $\mathbf{S}^t = [\mathbf{p}_1^t, \mathbf{v}_1^t, \ldots \mathbf{p}_N^t, \mathbf{v}_N^t]$. We call such a joint state a *world model*. To make explicit the dependence of the energy from the past world model, we refer to it as $E(\mathbf{v}_i^t; \mathbf{S}^{t-1})$ [1]. Therefore we still assume that a pedestrian, at each time step t, makes a decision for his next velocity based on its past observations of the environment (see Fig. 2.6). As opposed to standard LTA however, we allow multiple *hypotheses* $[\mathbf{p}_{ih}^t, \mathbf{v}_{ih}^t]$ with $h \in \{1 \dots H^t\}$ to represent the state of subject i at time t. We assume that the number of hypotheses is the same for all the subjects, only for ease of explanation. By using a mixture of Gaussians to model the subject state, we can associate a *weight* and an *uncertainty* to each hypothesis. We model therefore the distribution over the state for each subject as

$$p(\mathbf{p}_i^t, \mathbf{v}_i^t) = \sum_{h=1}^{H^t} w_h \mathcal{N}(\mathbf{p}_i^t, \mathbf{v}_i^t; \boldsymbol{\mu}_{ih}^t, \boldsymbol{\Sigma}_{ih}^t) \,. \tag{5.1}$$

At time t we factor the distribution $p(\mathbf{S}^t)$ over the world models as

$$p(\mathbf{S}^t) = \prod_{i=1}^{N} p(\mathbf{p}_i^t, \mathbf{v}_i^t) \,. \tag{5.2}$$

Therefore also $p(\mathbf{S}^t)$ is a mixture of Gaussians

$$p(\mathbf{S}^t) = \sum_{m=1}^{M^t} w_m \mathcal{N}(\mathbf{S}^t; \boldsymbol{\mu}_{\mathbf{S}_m}^t, \boldsymbol{\Sigma}_{\mathbf{S}_m}^t) \,. \tag{5.3}$$

with $M^t = (H^t)^N$. In order to account for multiple velocity *choices* for each subject, we move towards a stochastic formulation of Eq. (2.16).

---

[1] In the original model, the desired velocity is linearly filtered for smoothness (see Eq. 2.16). Here, we use an equivalent energy potential that includes already the same smoothing, by introducing a simple coordinate transformation:

$$E(\mathbf{v}^t; \mathbf{S}^{t-1}) = E_{LTA}(\frac{\mathbf{v}^t - \alpha * \mathbf{v}^{t-1}}{1-\alpha}; S^{t-1})$$

where E_{LTA} is the formulation of the energy given in [Pellegrini *et al.*, 2009]. Note that this is an entirely equivalent formulation, but has the advantage of being more compact.

Based on the energy potential formulation, we define the posterior probability of a pedestrian's velocity $p(\mathbf{v}_i^t|\mathbf{S}^{t-1})$ as a Gibbs measure, for each pedestrian as

$$p(\mathbf{v}_i^t|\mathbf{S}^{t-1}) = e^{-\omega E(\mathbf{v}_i^t;\mathbf{S}^{t-1})}/Z \ , \tag{5.4}$$

where Z is a normalization constant and ω is a free parameter that is discussed later.

Rather than working directly with Eq. 5.4, we fit to Eq. 5.4 a mixture of Gaussians

$$p(\mathbf{v}_i^t|\mathbf{S}^{t-1}) \approx \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{v}_i^t; \tilde{\mathbf{v}}_{ik}^t, \tilde{\boldsymbol{\Psi}}_{ik}^t) \ , \tag{5.5}$$

where w_k, $\tilde{\boldsymbol{v}}_{ik}^t$ and $\tilde{\boldsymbol{\Psi}}_{ik}^t$ are estimated from 5.4, as discussed next. In Eq. 5.5 each mixture component represents a choice of subject i for the next velocity [2]. This mixture could be fit with standard methods such as Expectation Maximization or iterative function fitting techniques. However, to keep the system applicable to real-time scenarios, we opt to use the following heuristic to estimate the mixture parameters:

1. Discretize the distribution of Eq. (5.4). The number of components K of the mixture is decided by counting the local maxima in the discretized distribution.

2. Run a BFGS [Nocedal and Wright, 2000] maximization for each mode to refine the mode estimate. These mode estimates are assumed to be the locations of the means $\tilde{\mathbf{v}}_{ik}^t$ of the mixture components, with $k \in \{1 \dots K\}$.

3. Compute the gradient in the central point of each cell of the discretized distribution of Eq. (5.4). Assign the cell to the mode with the smallest angle between the gradient vector and the vector originating from the cell center and ending in the mode. Be this mode k. Estimate the covariances $\tilde{\boldsymbol{\Psi}}_{ik}^t$ by fitting a Gaussian distribution to the central points of the cells assigned to the mode.

---

[2] We assume that K and N are time independent and the same for all the subjects. This need not to be the case. However, we drop the dependencies for the sake of readability. The generalization is straightforward.

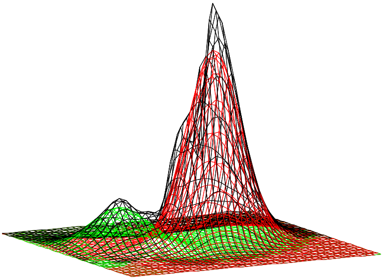


***Figure 5.2**: The energy potential is brought into an analytical form by fitting a mixture of Gaussians using a fast approximate method (see text).*

4. The weight w_k of each mode is computed for each component independently, by setting the k^{th} component's mode of the mixture equal to the energy at that point
$$w_k = \frac{exp(-\omega E(\tilde{\mathbf{v}}^t_{ik}))/Z}{\mathcal{N}(\tilde{\mathbf{v}}^t_{ik}; \tilde{\mathbf{v}}^t_{ik}, \tilde{\Psi}^t_{ik})} \,. \tag{5.6}$$
These weights are finally normalized so that their sum is one (therefore, the equality in Eq. 5.6 does not necessarily hold anymore (see also Fig. 5.2).

This is obviously a rough estimate of the parameters, that becomes worse the less the Gaussians are separated. Nevertheless, it turned out to be sufficient for our purposes (see Fig. 5.2 for an example fit). In Sec. 5.4, we explain why the algorithm is robust in this respect.

We are ultimately interested in the probability $p(\mathbf{p}^t_i, \mathbf{v}^t_i)$ of the state for a subject i at time t. From Eq. 5.1 we know that it is a mixture of Gaussians, but we need to find an explicit formula for the mixture parameters. We can write the state distribution as

$$\begin{aligned} p(\mathbf{p}^t_i, \mathbf{v}^t_i) &= \int p(\mathbf{p}^t_i, \mathbf{v}^t_i, \mathbf{S}^{t-1}) d\mathbf{S}^{t-1} \qquad (5.7) \\ &= \int p(\mathbf{p}^t_i | \mathbf{v}^t_i, \mathbf{p}^{t-1}_i) p(\mathbf{v}^t_i | \mathbf{S}^{t-1}) p(\mathbf{S}^{t-1}) d\mathbf{S}^{t-1} \,. \qquad (5.8) \end{aligned}$$

Note that we already have an approximation for $p(\mathbf{v}_i^t|\mathbf{S}^{t-1})$ from Eq. 5.5. For $p(\mathbf{p}_i^t|\mathbf{v}_i^t,\mathbf{p}_i^{t-1})$, by modeling the update of the position as the linear process

$$\mathbf{p}_i^t = \mathbf{p}_i^{t-1} + \Delta\mathbf{v}_i^t + \gamma \text{ with } \gamma \sim \mathcal{N}(\mathbf{0},\Gamma)\,, \tag{5.9}$$

we can write

$$p(\mathbf{p}_i^t|\mathbf{p}_i^{t-1},\mathbf{v}_i^t) = \mathcal{N}(\mathbf{p}_i^t;\mathbf{p}_i^{t-1} + \Delta\mathbf{v}_i^t,\Gamma)\,. \tag{5.10}$$

Finally, we can use Eq. 5.5, Eq. 5.10 and Eq. 5.3, to write

$$p(\mathbf{p}_i^t,\mathbf{v}_i^t) \quad \approx \quad \sum_{m=1}^{M^t}\sum_{k=1}^{K} w_{mk}\mathcal{N}(\mathbf{p}_i^t,\mathbf{v}_i^t;\boldsymbol{\mu}_{imk}^t,\boldsymbol{\Sigma}_{imk}^t)\,. \tag{5.11}$$

where

$$\boldsymbol{\mu}_{imk}^t = \begin{bmatrix} \mu_{\mathbf{p}_{im}}^{t-1} + \boldsymbol{\Delta}\tilde{\mathbf{v}}_{imk}^t \\ \tilde{\mathbf{v}}_{imk}^t \end{bmatrix}, \tag{5.12}$$

$$\boldsymbol{\Sigma}_{imk}^t = \begin{bmatrix} \Gamma + \Delta^2\tilde{\Psi}_{imk}^t + \boldsymbol{\Sigma}_{imk}^{t-1} & \Delta\tilde{\Psi}_{imk}^t \\ (\Delta\tilde{\Psi}_{imk}^t)^T & \tilde{\Psi}_{imk}^t \end{bmatrix}. \tag{5.13}$$

(The complete derivation of this approximation is given in App. A). As Eq. 5.1 shows, the distribution over the subject position has the form of a mixture of Gaussians with $M^{t-1}K = H^t$components. As a consequence of Eq. 5.11 and Eq. 5.2 the distribution over the world models has now $(M^tK)^N$ mixture components. This is the number of world models in the next iteration. This clearly leads to a combinatorial explosion of the number of state mixture components, or world models. Fig. 5.3 shows an example of this process. To prevent this from happening, we limit the maximum number of world models to a value $\hat{M}$. If the splitting process at a certain time step generates more than $\hat{M}$ world models, the most likely $\hat{M}$ are used, while the others are discarded. Further, we limit the combinatorial explosion in Eq. (5.11) by pruning the mixture components when $w_{mk} < \epsilon = 0.1$. Since the value of w_{mk} decreases with time because of the splitting, at a certain point the splitting ceases.

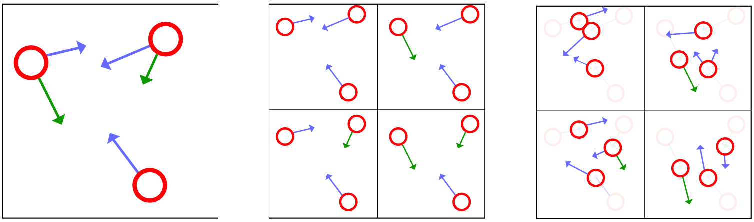


***Figure 5.3**: A toy illustration of the state evolution. A red circle represents a subject position and the arrows show the subject velocities. For each subject, a circle together with one of the corresponding arrows represents a single hypothesis. Uncertainties and hypotheses weights are not shown. **Left**: the state at time $t-1$. **Center**: the* world models *at time $t-1$. Each world model is a 3-tuple in the Cartesian product of the subject state hypotheses. **Right**: the state at time t after the propagation from the previous time step. The position is updated with Eq. 5.10 while the new velocities are computed with Eq. 5.5. Note the growth in the number of subject hypotheses.*

Note that in the special case when $\hat{M} = 1$, the model is deterministic and almost the same as the original LTA . The main difference is that in the original LTA, the next velocity $\hat{\boldsymbol{v}}$ was computed with a gradient descent over the energy potential E_{LTA}, while now the heuristic just described is used.

Note also that this general approach of handling multiple possible world states is conceptually similar to multi-hypothesis tracking [Reid, 1979], in which each world corresponds to a possible data association between trajectories and observations.

5.2.1 Why not a Particle Filter Framework?

Eq. (5.4) could be easily used in a particle filter framework as a propagation function (see Fig. 5.4). It is reasonable to expect that the results, for a sufficient number of particles, are more accurate than those obtained with an approximation by a mixture of Gaussians. However, there are at least two reasons why to refrain from taking this approach.

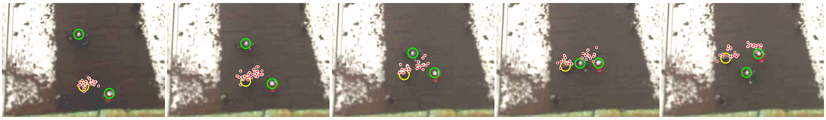


***Figure 5.4**: Particle filter experiment: when simulating a person (yellow circle) given the other people (green circles) using a particle filter embodiment of the model, multiple modes (red particles) form naturally. While both options of steering clear of the oncoming persons are found, such a solution is computationally prohibitive (see text).*

The first reason is related to computational requirements. Since we want to represent the interactions between subjects, the state space cannot be easily factored into independent particle filters. The state should rather be represented jointly by the positions and velocities of all the subjects. With the ensuing rapidly growing state dimension, the number of particles increases exponentially. For each particle, the basic LTA procedure should be evaluated for each subject, which is computationally prohibitive. In contrast, in our formulation the LTA procedure is only invoked for each mode of the mixture.

Even if a particle filter were computationally feasible, we believe that the commonly used resampling stage [Arulampalam *et al.*, 2002] introduces a higher logic that we assume a pedestrian not to have in the LTA model: if a mode of the sampled distribution happens to die out at some point, *e.g.* due to higher likelihood of the other modes in the resampling stage, the history of the particles belonging to the cloud until that point is meaningless. Once an alternative has been created, it cannot cease to exist simply because, a posteriori, other alternatives are more suited. This would imply that pedestrians predict their complete possible future trajectories in advance, even with information that is unavailable to them at present, and then choose the feasible ones. This assumption is not part of the LTA model and also does not seem to be realistic.

5.2.2 Training

The model parameters are set to the values learned with the procedure described in Sec. 4.4. The stochastic variant presented in this chapter

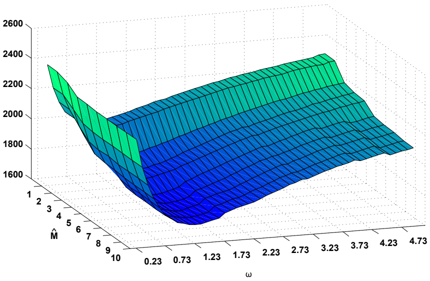


***Figure 5.5**: Negative log-likelihood (vertical axis) while changing the number of world models $\hat{M}$ and free parameter ω. Increasing $\hat{M}$ always improves the result.*

introduces other two parameters, *i.e.* $\hat{M}$ and ω. We inspect the effect of these parameters in the experiments section.

5.3 Experiments

As noted before, a pedestrian motion model has a multitude of uses. For instance, in robotics, its predictions can be used for path-planning purposes. Alternatively, the model output can help improving data association in a tracking context when appearance is unreliable. In the following, we first evaluate the sLTA model, comparing its prediction capabilities for different parameter settings. We then show the application of the model in a tracking experiment, highlighting the importance of a good motion model in data association. For these experiments, we use annotated data provided by the authors of [Lerner *et al.*, 2007]. The video shows part of a shopping street from an oblique view. A homography from image to ground plane was estimated using four manually clicked points on the footpath to transfer image to world coordinates. Standing and erratically moving people were marked; for these, a simple extrapolation is used. As destinations we chose two points far outside the left and right image borders, which holds for most subjects. Static obstacles (*i.e.*, the building and the parked car) were also annotated.

5.3.1 Prediction

To test the prediction capabilities of our model, we evaluate on a subsequence of about 3 minutes @ 2.5 *fps*, containing 86 trajectories annotated with splines. We simulate all the subjects in parallel. Note that this is different from the prediction experiment in Sec. 4.5.1, where each subject was simulated in turn, while using the ground truth positions and velocities of all the others. Starting one simulation every 1.2 seconds with a prediction horizon of 4.8 seconds yields ≈ 200 simulations. To highlight the importance of using multiple modes, as well as the effect of the parameter ω, we run multiple simulations over all subjects, varying both the maximum number of world models $\hat{M}$, as well as the free parameter ω. For each simulation, we report the negative log-likelihood $\log p(GT|\hat{M}, \omega)$ of $\hat{M}$ and ω based on the ground truth trajectories GT, Fig. 5.5. As can be seen, increasing the number of world models, and therefore of modes, always improves the prediction result, irrespective of the chosen ω: this indicates that even with multiple modes, the model is conservative enough as not to allow completely improbable predictions. The parameter ω relates to how certain each hypothesis is. When ω is zero, the probability becomes uniform, while for bigger values of ω, the uncertainty around each mode decreases. Fig. 5.5 shows a small yet interesting positive correlation between the value of ω and $\hat{M}$. This can be interpreted saying that when increasing the number of world models, less uncertainty per mode is *allowed*.

Some example images when using 10 modes are shown in Fig. 5.6. Red lines indicate the ground truth, yellow lines indicate the predicted path of a person, blue circles correspond to the standard deviation of the fitted Gaussians at the respective end positions. Green lines indicate the linear extrapolations of people that are standing or moving erratically, white boxes the set of used obstacle points. Please note that the model operates in ground-plane coordinates, hence all drawings correspond to people's feet in the image. For each image, we show the final image after 4 s of extrapolation. As can be seen, the model manages to find the correct extrapolation for almost all persons in one of its modes, while keeping the number of modes at a minimum. Multiple possibilities can be especially seen when people are walking towards other groups of people.

In the deterministic setting ($\hat{M} = 1$), extrapolations in easy situations remain the same (Fig. 5.7, left; these images correspond to the left column of Fig. 5.6). In more difficult situations, only the stronger mode remains, which can either be correct (middle) or wrong (bottom). Thus, from a prediction point of view, it is indeed beneficial to use multiple

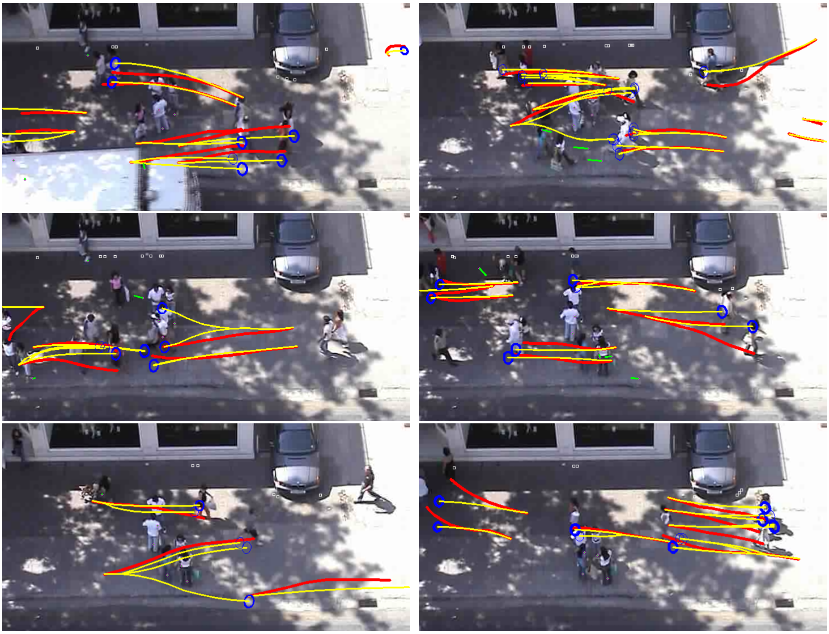


***Figure 5.6**: Example extrapolations. The possible paths for a given person are shown in yellow, with blue circles indicating the σ-confidence of the fitted Gaussians. Note that the model operates in ground-plane coordinates, the lines and circles thus correspond to people's feet. Also note that all the subjects are simulated in parallel.*

modes in a stochastic fashion. Finally, Fig. 5.7, right column, shows some typical failures of the model. These are not all failures in the hard sense, as the stochastic options often also includes the correct solution: in the top-right image, the model splits too much because it is unsure what to do with two persons walking with each other in a group, but slightly changing positions to each other. It splits, but keeps the correct hypothesis. In the middle image, another person is wrongly extrapolated (green line in middle of image), causing a split, but the correct hypothesis is also kept. In the bottom-right image, the lower extrapolation is wrong, with the correct solution (going above the standing group) not identified: this is a special case of the first case, where two people walking in a group feel repulsion rather than staying together.

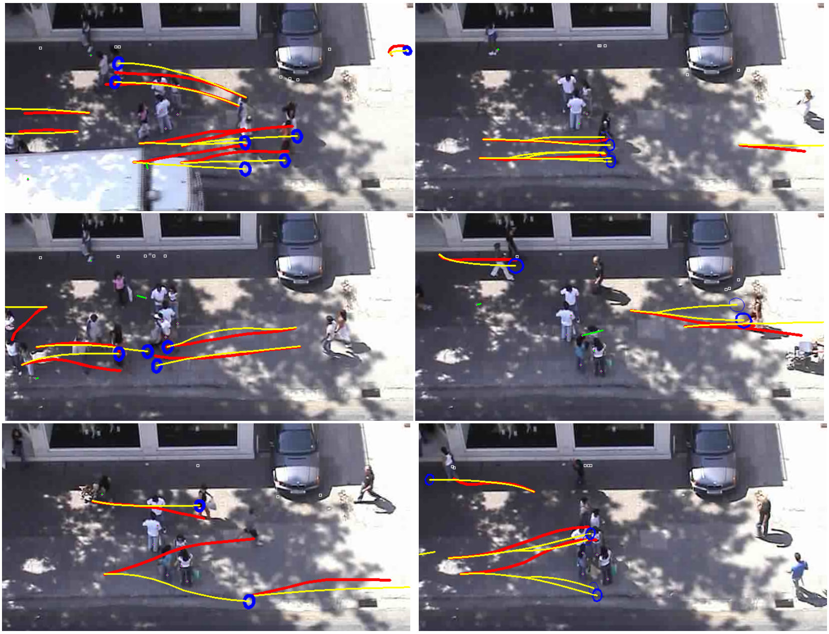
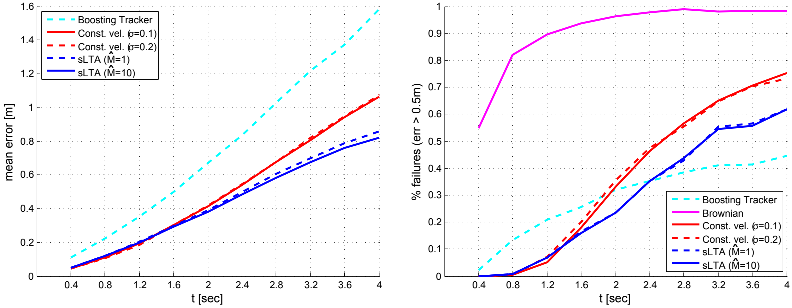


Figure 5.7: ***Left column*** *: Extrapolations when just using one mode, corresponding to a deterministic model. (see text).* ***Right column****: Typical failures of the sLTA: (top, middle) unnecessary splittings can occur due to other wrong extrapolations, but are handled in the multi-hypothesis framework. (bottom) without the knowledge of people walking in groups, wrong extrapolations can occur. (see text for details)*

5.3.2 Tracking

To explore the effect of a stochastic motion prior on tracking performance, we present the following experiment: for each person, and for increasing time horizons, we perform an NCC-based template matching between a subject in a reference frame and its possible location in a later frame. The chosen motion model defines the search radius for the matching; the solution is found as the peak NCC-response, weighted by the motion models' uncertainty. The error in distance between this solution and the ground truth is accumulated for all persons and by starting



(a) (b)

***Figure 5.8**: (a) Mean error (in meters) of tracking using different motion models, for increasing frame gaps. (b) Number of tracking failures (error* $>$ 0.5 *m) using different motion models, for increasing frame gaps.*

the tracking every 1.2 seconds. As the model is trained in steps of 0.4 seconds (10 frames), we also keep this spacing for the experiment.

This experiment should highlight the advantage of a good motion model: a correct search region should prevent the tracker from drifting by guiding the data association. Instead of including the model into a complicated tracker, where many side-effects can influence the result, we therefore keep the experiment as simple as possible to see the real merit of a motion model.

We specifically compare a simple Brownian motion model with a constant velocity one, as well as different instantiations of sLTA. For the experiment, we use templates of 30 × 30 pixels on people's head positions. As an additional baseline, we use an adaptive tracker based on online boosting [Grabner and Bischof, 2006] that uses *all* intermediate frames (as opposed to steps of 10 frames). In the given sequences, purely appearance-based matching is especially tricky due to low contrast, cast shadows, and interlacing and compression artifacts. The motion model uncertainty is chosen as follows: for the Brownian model, the uncertainty is assumed uniform in the search region (which is bounded by a statistic on the maximum walking speed); for the constant-velocity one, we use a single Gaussian centered around the prediction (we plot results for two choices of the uncertainty); for sLTA, the mixture of Gaussians as introduced above is used. In this systematic experiment, velocities are inferred from the past frame's ground-truth. While this does not reflect the actual tracking application, it still allows for a fair comparison between the different models, and their (ideal) influence on appearance-based tracking.

Fig. 5.8 (a) plots the mean error in meters for all approaches. We furthermore report the number of actual tracking errors (deviation from ground truth > 0.5 m) in Fig. 5.8 (b). For increasing frame gaps, an uninformed motion model makes tracking virtually impossible ("Brownian", mean error not plotted in (a) due to large error). For small time horizons, the result of a constant velocity model ("Const. vel.") is virtually the same as with any more advanced model, as small motions can be sufficiently approximated by a linear extrapolation. However, for increasing time horizons, the positive effect of sLTA becomes more pronounced. This is also in line with what we reported in Chapter 4, where the effects of the strong motion model were mostly visible in cases of missing data, *e.g.* due to occlusion.

As an additional baseline, we show the result of purely appearance-based tracker, which uses *all* available intermediate frames while learning the model of the appearance ("Boosting Tracker"). Using all available data from the image produces fewer hard failures, still, the high mean error indicates that when the tracker starts drifting, it's totally lost. We therefore believe a strong motion model to be important for tracking.

Accounting for a pedestrian's future motion in a probabilistic manner, *i.e.*, using $\hat{M} = 10$ instead of $\hat{M} = 1$, does not seem to have a considerable effect on tracking performance: both the mean error and the fraction of tracking errors seems to only improve slightly when allowing multiple modes. The important thing to note here is that in the presented sequence, there is only a limited number of "splittings" in general, and only in a fraction of these, the deterministic model chooses the wrong mode. While the effect thus seems limited, this still means that in such cases, the tracker would fail and lose an object for multiple seconds, searching in the wrong location. Employing a stochastic model therefore definitely helps in extreme situations, which can also be expected more frequently in more crowded scenarios.

5.4 Discussion and Conclusions

This chapter presented a stochastic simulation-based motion model for pedestrians. The probabilistic formulation is based on using the LTA energy function in a Gibbs measure. Then, by using a multi-hypothesis approach with uncertainty propagation, a set of possible future world states is obtained. To achieve a good compromise between accuracy and tractability, we fit a Gaussian mixture model to the Gibbs measure. Although the fitting is rather approximate, we found it to work well in our experiments. This is due to the fact that the actual choices of pedestrians seem to be limited to one or two, for each timestep. Therefore the potential function has only one or two modes. Furthermore, the modes corresponding to alternatives of a choice for a pedestrian, tend to separate apart with time. This allows us to *wait* for the modes to be well separated before fitting the mixture (when their distance is below an empirical threshold, we group them and consider them as a single mode).

In our prediction experiments, we showed that the log-likelihood of the prediction increases considerably as we go from a deterministic instantiation to a stochastic one, showing the benefits of such a non-deterministic solution.

For tracking, a clear advantage over simpler motion models was demonstrated, the effect of a stochastic model is however not as pronounced as expected. While more complicated scenes would probably show an advantage of using a probabilistic formulation, this difference is only present at higher frame gaps, which could be *e.g.* due to occlusion. Generally, it thus seems that the prediction would be more suited to tasks in, *e.g.*, robot navigation, where safety is a crucial issue.

6

Towards Joint Grouping and Tracking

6.1 Introduction

Tracking algorithms are an indispensable prerequisite for many higher-level computer vision tasks, ranging from surveillance to animation to automotive applications. Advances in observation models, such as object detectors or classification-based appearance models, have enabled tracking in previously infeasible scenarios. Still, tracking remains a challenging problem, especially in crowded environments. Tracking high numbers of pedestrians in such cases is even hard for humans. Usually, a manual annotator has to rely on higher-level reasoning, such as temporal information (that can go into the future) or social factors. Recent advances in the literature suggest that the latter can improve tracking performance. Typically employed social factors include a pedestrian's *destination*, *desired speed*, and *repulsion* from other individuals (see Chapter 4). Another factor is grouping behavior, which so far however has been largely ignored. For one, this is due to the fact that the grouping information (do two persons belong to the same group?) is not easily available. Still, groups constitute an important part of a pedestrian's motion. As we show in this chapter, people behave differently when walking in groups as opposed to alone: when alone, they tend to keep a certain distance from others, passing by closely only if necessary, but mostly at different speeds. When in groups, they try to stay close enough with other members, walking at the same speed.

In this chapter, we investigate the group classification problem and its relation to tracking. First we look at the problem when the trajectories of people are given (Sec. 6.2. Then we exploit the interaction between different people for data association (Sec. 6.3), having in mind a fully functional tracking application (see Chapter 7). In particular, for this second problem, we model group relations and study their effect on trajectory prediction. The grouping between pedestrians is treated as a latent variable, which is estimated jointly together with the trajectory information. Our model of choice is a Conditional Random Field (CRF), with nodes in the lower level corresponding to pedestrians, connected by third-order links that represent possible groupings. Recent advances in discrete optimization provide powerful tools for carrying out (approximate) inference in such models (Sec. 6.5).

6.2 Group Classification

Although they seem to represent a significant aspect of social walking, groups have largely been ignored. Probably one reason is that the knowledge of whether two people belong to the same group or not is not directly available in the image, but requires further processing. Here, we want to show some preliminary studies on the group classification task in order to suggest the amount of effort that is required for such a task and the results that one might expect. We use only a proximity clue to perform the classification. In detail, we look at the distances among pair of subjects across time, and try to answer the question whether the two subjects belong to the same group or not. Let us call d_{ij}^t the Euclidean distance among two subjects i and j at time t. If, starting from t, we concatenate d_{ij}^t in a vector over time, let us say over a time window of l time steps, we get $\mathbf{d}_{ij}^{t,l} = [d_{ij}^t \ \ldots \ d_{i,j}^{t+l-1}]$. If the time window is longer than the subject trajectories, the vector $\mathbf{d}_{ij}$ is opportunely trimmed to the shortest of the two trajectories.

Instead of using directly the whole $\mathbf{d}_{ij}^{t,l}$, we use simple features extracted from it. The feature vector used for the group classification task is

$$[\ mean(\mathbf{d}_{ij}^{t,l}),\ max(\mathbf{d}_{ij}^{t,l}),\ min(\mathbf{d}_{ij}^{t,l}),\ std\_dev(\mathbf{d}_{ij}^{t,l}),\ length(\mathbf{d}_{ij}^{t,l}))\ ]\ . \tag{6.1}$$

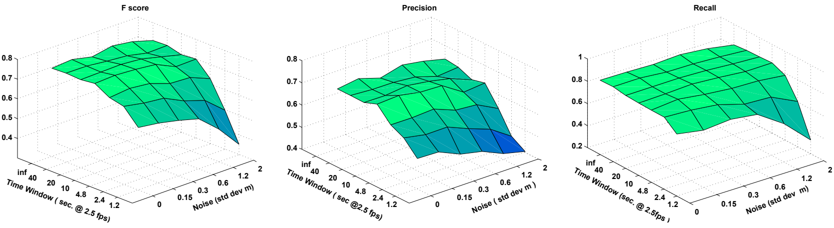


***Figure 6.1**: Group classification results. From left to right we report Fscore, precision and recall for different time windows and noise values. Noise values refer to standard deviation of the Gaussian noise (in meters) and the time window is measured in seconds (at 2.5 frames per second)*

We used an off-the-shelf SVM library [Chang and Lin, 2001] to perform the classification. For the evaluation, we performed a 20-fold crossvalidation on the ETH dataset [Pellegrini *et al.*, 2009], and we averaged the results for F score, precision and recall. We repeated the same experiment for different values of the time window l and for changing Gaussian noise added to the $\mathbf{d}$ vector. The results are reported in Fig. 6.1. As one can see, the proximity alone, based on simple statistics over time, is already a powerful clue for the classification task. As one might expect, the results degrade rather quickly with a shorter time window l, but already after ~ 5 seconds the precision of the classification reaches $\sim 70\%$.

These preliminary results suggest that grouping might be included in the model with a reasonable effort. Other clues could be used together with proximity. For example one could try to estimate whether two subjects are talking or at least looking at each other. Further investigation is clearly required to better understand costs and potentialities of including this aspect in a pedestrian motion model.

6.3 Group CRF

In the previous section, we have seen promising results for the group classification task when trajectories are given. Unfortunately this is rarely the case. In the remaining of the chapter, we investigate the

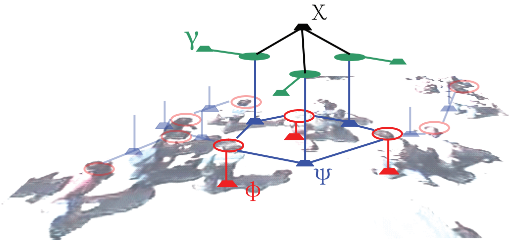


Figure 6.2: *Assumed higher-order model for joint trajectory and group finding (see text).*

interplay between tracking and group classification. To improve data association in crowded scenarios, we want to jointly estimate pedestrian trajectories and their group relations. Fig. 6.2 shows the factor graph for the third-order CRF model we assume for this problem.

Given a starting frame, each tracking target i ($i = 1 \dots N$) is modeled as a variable node (red empty circle, Fig. 6.2), where each possible state corresponds to the choice of one local trajectory hypothesis $\mathbf{h}_i^m \in \mathcal{H}_i = \{\mathbf{h}_i^m\}^{m=1 \dots M_i}$, with $\mathcal{H}_i$ the set of hypotheses for one person. As a trajectory hypothesis $\mathbf{h}_i^m$, we consider a single subject's possible future within a short time window. A joint assignment of hypotheses to all the subjects is defined as $\mathbf{H}^q = [\mathbf{h}_1^{q(1)} \dots \mathbf{h}_N^{q(N)}]$, where q is an assignment function that assigns each target i to one hypothesis in $\mathcal{H}_i$.

To set up the links between individuals, Delaunay triangulation is performed on the subjects positions in the input frame. Links longer than 3 meters are canceled. This results in a set $\mathcal{D}$ of pairs $\{i, j\}$ that are mutually connected. For each pair, the group variable g_{ij} (green filled circle, Fig. 6.2) indicates the group relation among the subjects i and j,

$$g_{ij} = \begin{cases} 1 & \text{if subject } i \text{ and } j \text{ belong to the same group} \\ 0 & \text{otherwise} \end{cases} . \qquad (6.2)$$

Two subjects i and j and the group variable g_{ij} are linked by a factor ψ (blue factor in Fig. 6.2). This link variable is essential to take advantage of grouping relations in our model. The joint assignment of grouping variables g_{ij} is defined as $\mathbf{L} = [g_{ij} \dots]$ with $\{i, j\} \in \mathcal{D}$.
In our definition, grouping is an equivalence relation, i.e. it fulfills reflexivity, symmetry, and transitivity. While reflexivity and symmetry

are enforced by the graph construction, the transitivity constraint is encoded in a factor χ (black factor in Fig. 6.2). We define the set $\mathcal{T}$ of triples $\{i, j, k\}$, such that $\{i, j\} \in \mathcal{D}$, $\{i, k\} \in \mathcal{D}$ and $\{k, j\} \in \mathcal{D}$. The transitivity constraint is then applied to each $\{i, j, k\} \in \mathcal{T}$ as

$$(g_{ij} \wedge g_{ik}) \rightarrow g_{kj} \ . \tag{6.3}$$

The log-probability of a set of trajectories $\mathbf{H}^q$ and a set of grouping relations $\mathbf{L}$, given an image sequence $\mathbf{I}$ and parameters Θ, is given by

$$\begin{aligned} &\log P(\mathbf{H}^q, \mathbf{L} | \mathbf{I}, \Theta) = \\ &\quad \sum_{i=1}^{N} \left(\phi^{mot}(\mathbf{h}_i^{q(i)} | \Theta_{\phi^{mot}}) + \phi^{app}(\mathbf{h}_i^{q(i)} | \mathbf{I}, \Theta_{\phi}^{app})\right) + \sum_{\{i,j\} \in \mathcal{D}} \gamma(g_{ij} | \Theta_{\gamma}) + \\ &\quad \sum_{\{i,j\} \in \mathcal{D}} \left(\psi^{pos}(\mathbf{h}_i^{q(i)}, \mathbf{h}_j^{q(j)}, g_{ij} | \Theta_{\psi}^{pos}) + \psi^{ang}(\mathbf{h}_i^{q(i)}, \mathbf{h}_j^{q(j)}, g_{ij} | \Theta_{\psi}^{ang})\right) + \\ &\quad \sum_{\{i,j,k\} \in \mathcal{T}} \chi(g_{ij}, g_{jk}, g_{ki} | \Theta_{\chi}) - \log Z(I, \Theta), \end{aligned} \tag{6.4}$$

where ϕ^{app} and ϕ^{mot} model, respectively, the appearance and motion of a trajectory, γ models the prior over a relation being of type group or not, ψ^{pos}, ψ^{ang} model the grouping relation and $Z(I, \Theta)$ is the usual partition function making sure that the probability density function sums to one.

6.4 Learning the parameters

Learning the parameters of the model in Eq. 6.4 could be done by maximizing the conditional likelihood of the parameters given the data. However, this is hard because of the partition function Z. Instead, inspired by piecewise training [Sutton and McCallum, 2005], we learn simple statistics from the data and define the terms in the Eq. 6.4 as a combination of these statistics. In particular we overparametrize the trajectory $\mathbf{h}$ as a sequence $[\mathbf{p}^0, s^0, \alpha^0 \ldots \mathbf{p}^{T-1}, s^{T-1}, \alpha^{T-1}]$ of, respectively, position, speed and orientation and extract simple statistics over these terms, rather than over the whole trajectory. In doing so, we use a non-parametric approach, by building histograms to estimate densities. The parameters Θ can be interpreted as the entries of these histograms. To reduce the notational clutter we drop in the following the dependence on Θ.

In the analysis of the data, we make, when appropriate, a distinction between people walking and people standing. Besides believing that these two classes can have different statistics indeed, we are motivated for making this distinction by a technical limitation: the orientation estimate is hard and unreliable for standing people, while it can be approximated by the direction of motion for moving people. We therefore choose an empirical threshold of $0.15m/s$ to distinguish between the two modes.

In the following, we show the relevant statistics that we used in our model.

6.4.1 Dataset

The data used to extract the statistics has been kindly provided by Lerner *et al*. [Lerner *et al.*, 2007]. The employed sequence shows a busy square from a stationary camera, oblique view, with a total of 450 subjects in 5400 frames. Most of the subjects walk from one of the borders of the scene to another and stay within the scene for about 15 seconds, while some stand longer in the scene talking to other subjects or waiting. An example frame is shown in Fig. 3.8, left. The sequence is particularly challenging due to low image resolution, interlacing and compression artifacts, cast shadows, as well as the large number of people. We manually annotated the head position of each subject and estimated a homography matrix to retrieve metric properties. In a second step, we annotated groups in the sequence, by relying on several cues, such as people talking to each other or holding hands, for example. For our purposes, we split the sequence in a training (3400 frames) and testing section (2000 frames).

6.4.2 Independent Motion and Appearance

Pedestrians change the walking direction smoothly. Furthermore, the walking speed is not arbitrary. This information is commonly exploited

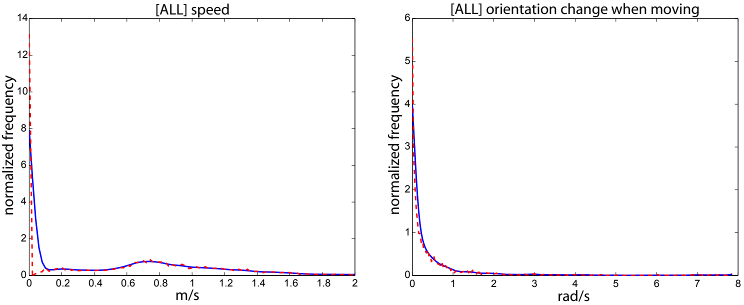


Figure 6.3: *Statistics over a person's movement:* ***Left***: *the distribution $P(s^t)$ over speeds shows two peaks for people standing and walking.* ***Right***: *the figure shows $P_{s^t \geq 0.15}(\alpha^t|\alpha^{t-1})$. For walking people, there is a preference to keep the current heading. Red indicates the original data points, blue the histogram estimate.*

in motion prior for pedestrians in a constant velocity model. To model these factors we define the motion term of Eq. 6.4 as

$$\phi^{mot}(\mathbf{h}) = \sum_{t=0}^{T-1} \log[P_{s^t<0.15}(\alpha^t|\alpha^{t-1}) + P_{s^t\geq 0.15}(\alpha^t|\alpha^{t-1})] + \sum_{t=0}^{T-1} \log P(s^t) \; . \tag{6.5}$$

$P_{s^t<0.15}(\alpha^t|\alpha^{t-1})$ is assumed uniform while $P(s^t)$ and $P_{s^t\geq 0.15}(\alpha^t|\alpha^{t-1})$ are estimated by building a normalized histogram (smoothed with a Gaussian kernel) of the angles and speeds extracted from the training set and are shown in Fig. 6.3. As one can expect, from the speed statistics it is easy to distinguish two modes, corresponding to standing and walking people. Fig. 6.3 shows also that the the choice of $0.15 m/s$ for telling apart walking and standing pedestrian is a reasonable one.

For the appearance term, we directly use the output of the tracker (see Sec 6.6).

$$\phi^{app}(\mathbf{h}|\mathbf{I}) = \log f^{app}(\mathbf{h}|\mathbf{I}) \; . \tag{6.6}$$

6.4.3 Grouping Relations

Given two pedestrians, one of the obvious features that makes it possible to guess whether they belong to the same group or not, is proximity. So, when two pedestrians belong to the same group, their distance is kept to a certain value. If they are walking, the estimate of the orientation can give us further information on how they are positioned with respect to each other. For two pedestrians belonging to the same group, we therefore define

$$\psi^{pos}(\mathbf{h}_i, \mathbf{h}_j, g_{ij} = 1) = \sum_{t=0}^{T-1} \log[P_{s_i^t \geq 0.15 \wedge s_j^t \geq 0.15}(\mathbf{p}_i^t | \mathbf{p}_j^t \alpha_j^t, g_{ij}) + P_{s_i^t < 0.15 \vee s_j^t < 0.15}(d_{ij}^t | g_{ij})] \ , \tag{6.7}$$

where d_{ij}^t is the Euclidean distance between the positions $\mathbf{p}_i^t$ and $\mathbf{p}_j^t$. As we did before, we estimate $P_{s_i^t \geq 0.15 \wedge s_j^t \geq 0.15}(\mathbf{p}_i^t | \mathbf{p}_j^t \alpha_j^t, g_{ij} = 1)$ and $P_{s_i^t < 0.15 \vee s_j^t < 0.15}(d_{ij}^t | g_{ij} = 1)$ using histograms and we shown them in Fig. 6.4. For pedestrians that do not belong to the same group, we found it unnecessary to distinguish between walking or standing. The main feature, when dealing with the position of two individual pedestrians, seems to be the *repulsion* effect: individuals try not to come close to each other unless necessary. In this case, we define the motion term as

$$\psi^{pos}(\mathbf{h}_i, \mathbf{h}_j, g_{ij} = 0) \ = \ \sum_{t=0}^{T-1} \log P(d_{ij}^t | g_{ij} = 0) \ , \tag{6.8}$$

where $P(d_{ij}^t | g_{ij} = 0)$ is again estimated using histograms and shown in Fig. 6.4. Another important feature of people when walking in the same group, is that they have the same orientation. We therefore define

$$\psi^{ang}(\mathbf{h}_i, \mathbf{h}_j, g_{ij} = 1) = \sum_{t=0}^{T-1} \log P_{(s_i^t \geq 0.15 \wedge s_j^t \geq 0.15)}(\alpha_i^t, \alpha_j^t | g_{ij} = 1) \ . \tag{6.9}$$

As before, this term is estimated with a smoothed histogram approach. The density is shown in Fig. 6.4 and, as expected, shows that subjects that walk together keep the same orientation. We did not observe an interesting orientation pattern among pedestrians that are not in the same group, therefore we assume uniform $P_{(s_i^t \geq 0.15 \wedge s_j^t \geq 0.15)}(\alpha_i^t, \alpha_j^t | g_{ij} = 0)$.

Finally, $\gamma(g_{ij})$ could be set by looking at the fraction of grouping relations over the total number of relations. Although the correct value for the fraction would be $\sim 23\%$ for our dataset, we vary this value to measure the robustness of our model in the experiments below (see Sec. 6.6).

6.4.4 Transitivity Constraints

The hard constraint in Eq. 6.3 is modeled by penalizing impossible configurations with an opportunely large constant cost.

6.5 Inference

We are looking for the most probable joint assignment of the trajectories $\mathbf{H}^q$ together with the grouping relations $\mathbf{L}$ in Eq. 6.4. Exact inference is intractable, as the graph contains cycles and the potentials are not restricted to a particular kind (e.g., submodular). For the inference, we use Dual Decomposition (DD) [Komodakis *et al.*, 2007], building on the code made available by [Torresani *et al.*, 2008]. DD optimizes the Lagrangian dual of the LP-relaxation of the original problem, by decomposing the problem into a set of subproblems, each of which can be solved efficiently. By optimizing the dual, it gives a lower bound that can be used to check whether the method converged to a global optimum (i.e. when the solution given by the primal has the same energy as the solution of the dual problem).

In our case, we decompose the original graph first into a constraint layer containing only transitivity constraints factors and a data layer containing all the other factors. Then these sub-graphs are further decomposed into spanning trees. We optimize each tree separately using max-product algorithm, using publicly available code [Mooij and al., 2010]. The primal solution, and therefore the upper bound to the optimal solution, is found by using a heuristic similar to that described in [Komodakis *et al.*, 2007].

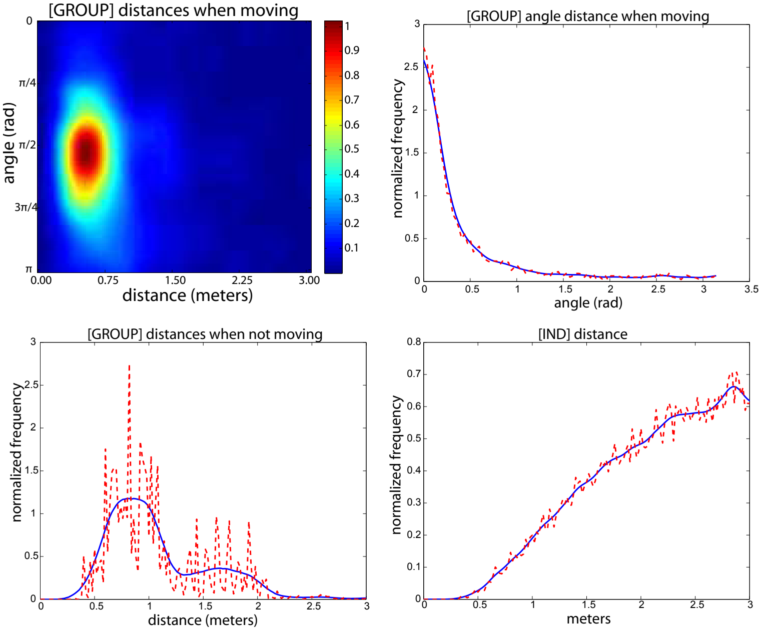


Figure 6.4: *Statistics over interacting people.* ***Top-left***: $P_{s_i^t \geq 0.15 \wedge s_j^t \geq 0.15}(\mathbf{p}_i^t | \mathbf{p}_j^t \alpha_j^t, g_{ij} = 1)$ *in polar coordinates, such that radius is the distance* d_{ij}^t *and the angle is the angle under which* j*, with absolute orientation* α_j^t *sees* i*. When moving in groups, people keep a low distance from each other, trying to walk side by side.* ***Top-right***: *the figure shows* $P_{(s_i^t \geq 0.15 \wedge s_j^t \geq 0.15)}(\alpha_i^t, \alpha_j^t | g_{ij} = 1)$*. As expected, people that walk together are headed in the same direction.* ***Bottom-Left***: *the figure shows* $P_{s_i^t < 0.15 \vee s_j^t < 0.15}(d_{ij}^t | g_{ij} = 1)$*. The distribution is less peaked than the distribution shown in the top-left figure, probably reflecting the fact that when people are standing in groups, they allow for more flexible configurations.* ***Bottom-Right***: *the figure shows* $P(d_{ij}^t | g_{ij} = 0)$*. Like for groups, the repulsion effect between individuals is evident from the low value around 0.*

6.6 Experiments

The proposed model requires a set of hypotheses to choose from. In this section, we therefore first describe how to build up the model given an input frame, before presenting experiments on real-world data.

6.6.1 Model Construction

Hypothesis Generation. Given a starting frame t_0, a separate set of hypotheses $\mathcal{H}_i$ is generated for each currently tracked person i. Each hypothesis $\mathbf{h}_i$ describes a possible motion of the subject between $t = t_0 \dots t_{T-1}$. To this end, we start a single-person tracker for each person i at t_0, at each time step following the cost function recursively according to a best-first paradigm. Following at each time step t the M best options therefore yield a maximum of M^T hypotheses per person. As a cost function, we employ several cues: as a motion and appearance model, we use a constant velocity assumption, respectively an HSV-color histogram $\mathbf{a}_i^t$ on the subject's head. The product of the Bhattacharrya coefficients $b(\cdot, \cdot)$ along the trajectory is then used to define $f^{app}(\mathbf{h}_i|\mathbf{I}) = \prod_{t=1}^{T-1} b(\mathbf{a}_i^t|\mathbf{a}_i^{t-1})$.

As a third cue, we consider a discrete set of detections in the current hypothesis' vicinity. The detections are obtained from a voting-based detector [Gall and Lempitsky, 2009], trained on both head and upper bodies from a total of 1145 positive and 1208 negative examples. Even though specifically trained on the same setup's data, the detector only reaches an equal error rate of 0.65 (head) respectively 0.76 (torso). The reason for this low performance is a higher number of false positives on strong cast shadows, as well as some false negatives when people are standing very closely together. To account for frequent false negatives, up to 50% of missing detections are allowed inside a trajectory, where the missing parts are interpolated using the constant velocity model.

To handle the case of persons leaving the scene, we introduce a set of virtual detections at the border of the image. Once a tracker selects such a detection, it is terminated, and the corresponding trajectory corresponds to a linear extrapolation starting from that time step.

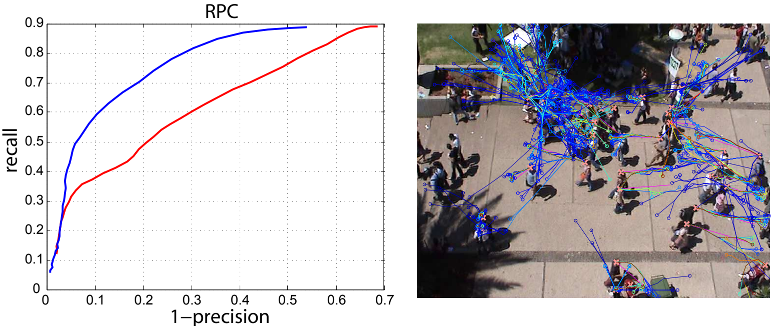


Figure 6.5: **Left:** *RPC curves for head detector (red) and torso detector (blue).* **Right:** *Sample hypotheses for one frame, with blue corresponding to low confidence and red to high. Especially in crowded areas, many possible hypotheses can be generated.*

For computational reasons, in the presented experiments, we use a time step of 0.2 seconds, and set $T = 10$ (thus always considering time windows of 2 seconds) and $M = 4$, yielding an average of 147 hypotheses per subject. We run the experiment each 2 s for all pedestrians, starting from 40 different frames. This results in 1236 subject tracks (see Fig. 6.5).

6.6.2 Ground-truth

Before using an actual detector to drive hypothesis generation, we perform a baseline experiment, where we use the ground-truth annotations as detections (note that we are operating on the test sequence, i.e., the training of the model did not use this data at all). To measure the effect of the proposed model, we compare the output of the inference stage with locally selecting the best trajectories (i.e., the hypothesis with the maximum unary term). We report the number of correctly selected trajectories as the ones that coincide with the ground-truth completely. In Fig. 6.6 (left), we run this experiment for different values of the grouping prior γ. As can be seen, the model does not blindly trust the prior (unless set to the extreme positions), but moves towards the true fraction of groupings (0.23) disregarding the starting position. The performance of the model with respect to trajectory selection is hardly affected by

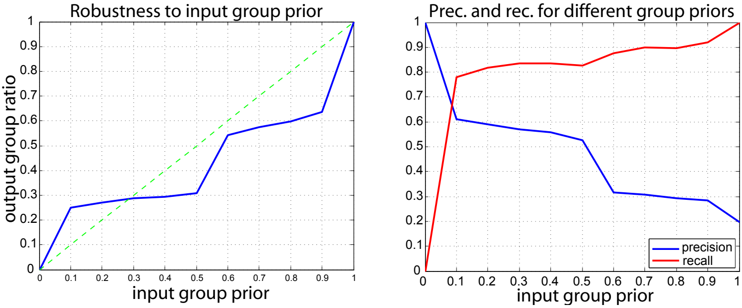


***Figure 6.6**: **Left:** Estimate of groups relations while varying the grouping prior γ. **Right:** precision and recall curves for group relations for different γ values.*

the grouping: the unary makes 34 mistakes, whereas the full model, depending on the chosen grouping prior, performs considerably better with 12 ± 2 mistakes. Only when $\gamma = 0$, our model makes 22 mistakes. In Fig. 6.6 (right), we furthermore plot recall and precision of finding groups, again varying over the prior γ. The numbers stay quite constant for a large range of γ, underlining the stability of the model. Choosing extreme values naturally also leads to inferior results, either in favor of groups or not. In the upcoming experiments, we use an uninformed prior, $\gamma = 0.5$.

6.6.3 Detector

When starting from a ground-truth point and generating hypotheses using detections, the generation step has to deal with a considerable number of false positives (generating excess wrong trajectories) and false negatives (in the worst case, missing an entire trajectory). Due to these inaccuracies, we change the notion of correct trajectory to an error $<$ 0.5 m from the ground-truth at the last trajectory position. The subject errors and the group statistics are reported in Table 6.1. Note that this experiment is considerably harder, so the number of errors in absolute terms increases. Still, our method improves $\approx$ 10% w.r.t. using only the unary terms, i.e. without grouping. The group statistics show a precision of 46% (about twice above the chance level of 23%) and 82%

	Wrong Trajectories	Groups			
		TP	TN	FP	FN
Local	401	-	-	-	-
Group CRF	363	389	1526	449	84

***Table 6.1**: Performance of model when using raw detections as input. The proposed model not only improves the correctly chosen trajectories, but also recovers groups with high recall and good precision.*

recall. Note that this performance is similar to the one obtained in the previous experiment, using ground-truth detections. Considering also the results of Sec. 6.2, we can conclude that group classification can be carried out effectively also when trajectories are not very accurate.

Some example images, comparing the two methods, are shown in Fig. 6.7. For each sample, we report both the trajectories found by either choosing the local optimum or the group CRF, as well as the recovered grouping by our model. In the top row, the grouping information gives a twofold improvement, encouraging the two persons to move together to the left side, as opposed to choosing intersecting trajectories (yellow arrows). One single wrong link between the two correctly inferred groups spurs the creation of additional wrong links through transitivity. In the second row, grouping correctly enforces the two people in the middle to walk together to the left as opposed to the local solution, which erroneously goes to the right (yellow arrows). In the third row, the joint reasoning keeps the group CRF from choosing the wrong path leading through all the pedestrians (yellow arrows), thus highlighting the spatial exclusion constraint. Finally, in the last row, grouping encourages smoother trajectories that stay well separated, with the group on the left correctly estimated.

6.7 Conclusions

In this chapter we investigated the influence of pedestrian interactions on data association in crowded scenes, having in mind a tracking application. Statistics learned on natural video data show that people walking in groups behave differently from people walking alone. Commonly hard-

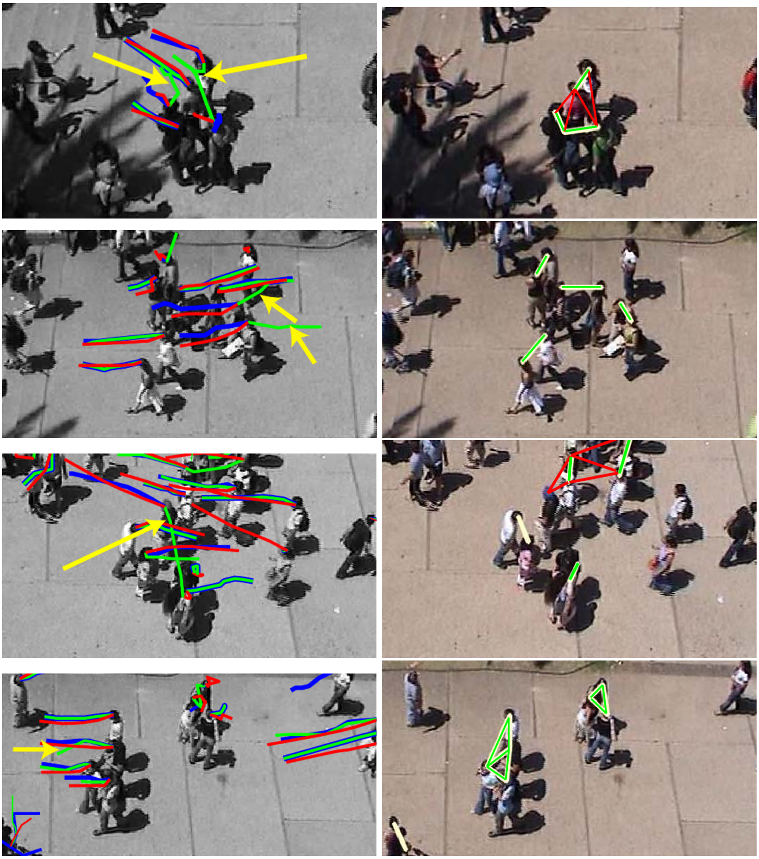


***Figure 6.7**: Example situations (close-ups).* **Left:** *trajectories, with ground truth (red) and solutions found by the unary term alone (green) and the group CRF (blue).* **Right:** *grouping, with ground truth (white), true/false positives (green/red) (see text).*

coded effects such as repulsion/avoidance were also clearly visible in the data. These statistics were used to train a graphical model encoding the interactions between pedestrians in a principled manner. The model was optimized for the MAP estimate with a state of the art approximate inference engine, giving a joint estimate about correct trajectories and group memberships in the data.

The preliminary results reported in this chapter, show that interactions should be taken into account when reasoning about people trajectories. We not only showed that joint optimization is beneficial in terms of tracking error, but we were able to recover, with a good recall and a sufficient precision, group statistics. Such statistics are relatively easy to extract once the trajectories are available, as we reported in Sec. 6.2. In this chapter we wanted to show that it is possible to tackle the joint problem of estimating the trajectory and grouping in a uniform framework.

The running time depends on the number of people in the scene. Our current implementation of the system, far from being optimized, takes few minutes ($\approx$ 10) to output trajectories of length 2 seconds and grouping relations.

The focus of this chapter was rather the effect of interactions as opposed to a complete tracking application. We therefore only showed results on short time windows initialized from ground-truth locations, not forming entire trajectories automatically. A full tracker needs however to output the whole trajectory over a longer time. Furthermore, automatic initialization and termination of targets is a necessary feature of a complete tracking system. In the next chapter, inspired by the results presented here, we describe a tracking application that includes all the mentioned requirements.

7

Tracking with Interactions

7.1 Introduction

The performance of a tracker is greatly affected by the target density in the scene. We can call a scene *sparse*, when the low target density causes very few occlusions and for a short period of time. Targets in a sparse scene move freely toward their destination and little intra-target interactions can be observed. On the other end of the spectrum, we have a *crowded* scene, where no subject is fully visible and each target motion is constrained by the motion of other targets. Our work focuses on the gray area between these two extremes. Our goal is to tackle those scenarios where there is more than a single dominant scene motion pattern and the targets interact with one another while moving towards their destinations. We call this scene a *busy* scene.

A multi-target tracker is a complex system. Among the many modules that usually collaborate in a successful tracker, we have :

- an observation model, to search the images for likely target positions based on what we know about the target appearance,
- a motion model, to exploit the temporal correlation among successive target positions,
- a way to associate the available observations to the multiple targets.

Furthermore, a fully autonomous tracker should be able to initialize and terminate tracks for targets entering and leaving the scene, respectively. Given the need to employ such heterogeneous components, a tracker often resorts to ad-hoc solutions and scene specific customizations to improve the performance.

In contrast, we propose here a single model that contains most of the tracker components just introduced. We aim to define simple functions, each of which models a particular aspect of the tracker. One such function, for instance, models the appearance similarity of a target with a certain portion of the image. Another function models the temporal correlation of the tracks. More interestingly, we define functions to model the interactions among targets, based on position and velocity configurations. The challenge that we face is how to combine these many functions. To do this in a principled way, we use the graphical model formalism as our *language*. One of the contributions of this work is the construction of an easily configurable, extensible and modular tracking model.

At design time, the model has been built mostly disregarding inference feasibility, as the focus has been kept on making it accurate and rich with features. However, a functional tracker needs to provide results in a reasonable time. In a second contribution, starting from a general inference technique, we propose a customization that exploits the application structure to achieve better performance.

This chapter builds on the work presented in Chapter 6. However it strongly differentiates from it for several aspects:

- The model comprises of continuous state variables for the targets, repeated at each time step, rather than only discrete ones representing candidate trajectories provided by an external tracker module. Tracking is therefore done inside the proposed framework.

- In this chapter, we propose a fully autonomous tracker, capable of initialization and destruction of targets, in contrast with a ground truth based initialization and a lack of termination mechanism.

- As a consequence of the presence of continuous variables, we need to employ a different inference method.

- We still perform, albeit differently, inference within a time window, but we propose a way to concatenate the results in a fully functional tracker and we show human intervention free results on minutes of video.

This chapter is structured as follows. We show how to model a tracker in a probabilistic framework in Sec. 7.2. While describing the modeling phase, we pay little attention at the computational constraints and focus only on modeling accurately the main aspects of a tracker. In Sec. 7.3 however, when describing the inference phase, we have to relax some model constraint in order to achieve feasibility. We do this in a principled way, trying to keep as much as possible of the original model. Finally, we show the experimental results in Sec. 7.4 and we conclude in Sec. 7.5.

7.2 Model

We use graphical models, and in particular factor graphs [Bishop, 2006], to write as many tracker components as possible, including the observation model, the motion model, the interaction model and the termination procedure. We use a log-linear Conditional Random Field [Lafferty *et al.*, 2001]

$$p(\mathbf{a}|\mathbf{D};\theta) = \frac{1}{Z(\theta)} e^{-\sum\limits_{k} f^k(\mathbf{a}_k)\cdot\theta_k}, \tag{7.1}$$

where $\mathbf{a}$ is the vector that concatenates all the model variables, $\mathbf{D} = [D^0 \dots D^T]$ is the vector of the evidence, θ is the vector of model parameters, $Z(\theta)$ is the partition function and f^k are the feature functions on a subset $\mathbf{a}_k$ of variables, with the corresponding subset of parameters θ_k. Casting the problem in a well known and studied framework makes it easy to exploit the techniques and state of the art methods that have been developed for that framework. Furthermore, although sometimes underestimated, re-usability, extensibility and modularity are positively affected by this choice.

The effort in writing the tracker as a graphical model consists of defining the variables of interest $\mathbf{a}$ (the nodes of the graph) and specifying

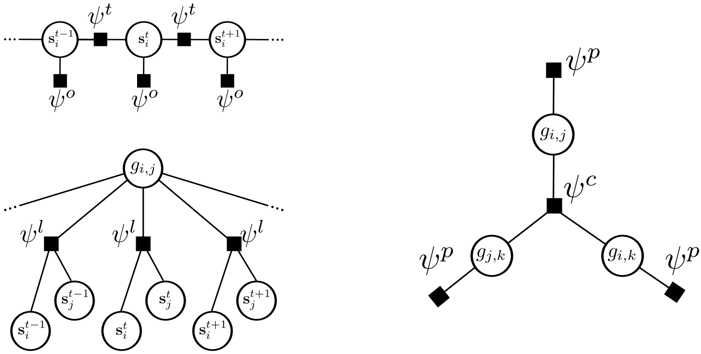


Figure 7.1: *The model sub-structures.* ***Upper left***: *the chain portion of the graph that represents a target i. The ψ^o factors implement the observation model, while the ψ^t ones implement the motion model.* ***Lower left***: *the portion of the graph the deals with modeling the interactions between two targets i and j. Note that all the interactions factors ψ^l are connected with the unique time independent variable g_{ij}.* ***Right***: *the portion of the graph that handles the consistency among the group assignments with the transitivity factor ψ^c together with the prior ψ^p on the group assignment.*

feature functions f_k (the links of the graph) to model the properties of individual, and group of, variables.

7.2.1 Model Variables

The core variable of interest is the position at timestep t for a target $i = 1 \dots N$. A target is represented by the 2D projection on the plane of a reference point $\mathbf{p}^t : [p_x^t \; p_y^t]$. We do not use a scale parameter to model the appearance, but this extension is possible and straightforward. In order to better exploit temporal consistency, it is very useful to also include in the state an estimate of the target velocity $\mathbf{v}^t : [v_x^t \; v_y^t]$.

We want to account for false positive initialization and in general of tracker mistakes in a coherent way. Once again, we would like to keep this reasoning within the graphical model framework. In order to do so, we define a binary variable u^t for each target. This variable takes the value

1 when the estimate, for any reason, is believed to be *unreliable*. All the subject specific variables can be concatenated in the mixed continuous-discrete state vector

$$\mathbf{s}^t : [\mathbf{p}^t \ \mathbf{v}^t \ u^t] \ . \tag{7.2}$$

Note that the positions and velocities, albeit continuous, are assumed to be bounded by a limited region of the world and by a maximum reasonable speed, respectively. $\mathbf{s}_i^t$ is represented as a node in the graphical model, as can be seen in Fig. 7.1.

The other variable that we are interested in estimating is the *grouping* variable. The group membership among two targets i and j can be represented with a binary variable g_{ij}, that takes value 1 if the two subjects are in the same group and 0 otherwise. As in Chapter 2, we say that two people belong to the same group if they walk or stand together.

We can finally concatenate all the state and group variables to write $\mathbf{a} = [\mathbf{s}_i^0 \dots \mathbf{s}_i^T \dots \mathbf{g}_{ij}]$, with $i, j = 1 \dots N$.

7.2.2 Motion Model

For each target, a pair of temporally consecutive state variables are strongly correlated. This correlation is usually exploited in a tracker by means of a motion model. The most common choice of a motion model in this regard is represented by a constant velocity model with some added noise, often normally distributed. We have shown that using a more complex motion model can lead to better performance in certain circumstances (see Chapter 4 and Chapter 5). In this chapter we separate the motion model from the interaction model. Therefore, we use a constant velocity model in the form of pairwise energy functions $f^m(\mathbf{s}^t, \mathbf{s}^{t+1}) \to \mathbb{R}$, that assign low energy to pairs of arguments that deviate little from a constant velocity assumption (see Tab. 7.1 for more details). We also need to deal with the case of unreliable estimate. If at time t a target i is believed unreliable ($u_i^t = 1$), we forbid a possible recovery by assigning an infinite cost to the transition $u_i^t = 1 \to u_i^{t+1} = 0$. This is done in order to avoid modeling the target motion when a track is lost, which is particularly complex, since we make no assumptions about the circumstances that caused the tracking loss. We prefer to re-initialize the track for the target as a means to recover from failure.

The graphical model factor that represents the motion model is defined as

$$\psi^m(s_i^t, s_i^{t+1}) = e^{-\mathbf{f}^m \cdot \theta^m} \tag{7.3}$$

with

$$\mathbf{f}^m = \begin{bmatrix} f_p^m(s_i^t, s_i^{t+1}) \\ f_v^m(s_i^t, s_i^{t+1}) \end{bmatrix} \qquad \theta^m = \begin{bmatrix} \theta_p^m \\ \theta_v^m \end{bmatrix}$$

where f_p^m and f_p^m are defined in Tab. 7.1. The factor ψ^m is shown in Fig. 7.1.

7.2.3 Observation Model

A crucial part of every tracker is the observation model. This component scores state hypotheses by evaluating feature correspondences between the target model and the data. As the observation model operates independently for each subject and each timestep, we represent it with unary energy functions of the form $f^o(\mathbf{s}^t) \to \mathbb{R}$. In particular, we use a target-class specific detector trained offline [Gall and Lempitsky, 2009] and then wrapped in a trained logistic function. The detector function, f_d^o, assigns low energy when the detector output is high. Also, we model the appearance by encapsulating an on-line classifier [Saffari *et al.*, 2010] in another feature function, f_a^o. Finally, with the function f_u^o, we model within the observation model the cost of being in the unreliable state (see Tab. 7.1).

The graphical model factor that represents the observation model is defined as

$$\psi^o(\mathbf{s}_i^t|\mathbf{D}) = e^{-\mathbf{f}^o \cdot \theta^o} \tag{7.4}$$

with

$$\mathbf{f}^o = \begin{bmatrix} f_a^o(\mathbf{s}_i^t) \\ f_d^o(\mathbf{s}_i^t) \\ f_u^o(\mathbf{s}_i^t) \end{bmatrix} \qquad \theta^o = \begin{bmatrix} \theta_a^o \\ \theta_d^o \\ \theta_u^o \end{bmatrix}$$

where the f^o functions are defined in Tab. 7.1.

7.2.4 Interaction Model

The model described so far is a simple chain graph (Fig. 7.1, upper left), where each target is modeled independently from all the others. As we are interested in multi-target tracking, it is reasonable to consider what kind of interactions we might exploit. One first kind of interaction is the physical one that forbids two targets to occupy the same position at the same time. But there is more to the simple exclusion, as we already mentioned in Sec. 7.1. In particular, it is reasonable to expect that the interaction is different when the two subjects belong to the same group than when they do not know each other [McPhail and Wohlstein, 1982]. We define a set of energy functions $f^l(\mathbf{s}_i^t, \mathbf{s}_j^t, g_{ij}) \to \mathbb{R}$ that assign a low value to more likely configurations of state and grouping variables. As mentioned we are interested in modeling the physical avoidance or repulsion (f_r^l), but also the attraction when two subjects belong to the same group (f_a^l) and the fact that two people in the same group have similar velocity (f_s^l). See again Tab. 7.1 for the specific choices of the feature functions.

Note that the interaction functions all return a fixed cost when one of the interacting arguments is in the unreliable state. The reason for this choice is that the interaction, while propagating useful information from one variable to the neighboring ones, can lead to wrong estimates when erroneous information is propagated during inference. Say, for instance, that there is a target that is estimated to occupy a certain position $\mathbf{p}_i^t$. According to the interaction model that we are describing, we want to forbid any other target to get too close to the same region of space in the same time. Now, if the estimate $\mathbf{p}_i^t$ is unreliable and there is actually no real target in that location, this limitation is a mistake. We want therefore to *turn off* this interaction whenever we infer that an estimate is unreliable. Assigning a fixed cost κ in the interaction functions whenever one of the two arguments is in the unreliable state, allows to achieve the desired result.

The graphical model factor (see Fig. 7.1) that represents the interaction term is defined as

$$\psi^l(\mathbf{s}_i^t, \mathbf{s}_j^t, g_{ij}) = e^{-\mathbf{f}^l \cdot \theta^l} \tag{7.5}$$

with

$$\mathbf{f}^l = \begin{bmatrix} f_r^l(\mathbf{s}_i^t, \mathbf{s}_j^t, g_{ij}) \\ f_a^l(\mathbf{s}_i^t, \mathbf{s}_j^t, g_{ij}) \\ f_s^l(\mathbf{s}_i^t, \mathbf{s}_j^t, g_{ij}) \\ f_b^l(\mathbf{s}_i^t, \mathbf{s}_j^t, g_{ij}) \end{bmatrix} \qquad \theta^l = \begin{bmatrix} \theta_r^l \\ \theta_a^l \\ \theta_s^l \\ \theta_b^l \end{bmatrix}$$

We choose not to integrate out the grouping variable g_{ij} in order to be able to explicitly model some relational properties on it, similarly to other works [Black and Rangarajan, 1994; Stein *et al.*, 2007]. In our definition, grouping is an equivalence relation, i.e. it fulfills reflexivity, symmetry, and transitivity. While reflexivity and symmetry are enforced by the graph construction, the transitivity is not. We therefore define a function $f^c(g_{ij}, g_{jk}, g_{ki}) \to \{0, 1\}$ that assigns zero energy to triplet of arguments that respect the transitivity property. In other words, when i and j are assigned to the same group (so that g_{ij} takes the value 1) and j and k belong to the same group (g_{jk} takes values 1), then k and i must also belong to the same group (therefore g_{ik} should be 1).
The graphical model factor that encodes this constraint is defined as

$$\psi^c(g_{ij}, g_{jk}, g_{ki}) = e^{-f^c\theta^c} \tag{7.6}$$

Finally, we also define a function $f^p(g_{ij}) \to \{0, 1\}$ that acts as a prior on the grouping variable. The graphical model factor for the group prior is defined as

$$\psi^p(g_{ij}) = e^{-f^p\theta^p} \tag{7.7}$$

7.2.5 Connectivity

So far, we deliberately decided not to specify which pairs of subjects nodes are connected to each other (and to the corresponding grouping variable). We could indeed use link ψ^l to connect all the pair of subjects that co-exist in the same time step (as a consequence, we could connect all the possible triplets of nodes g_{ij}). Doing so would mean that the interactions occur regardless of the distance and the reciprocal visibility of a pair of targets. Furthermore, the increase in connectivity, would increase the inference computational requirements. We instead assume that a target i interacts only with the neighboring targets j. Let us define the set $\mathcal{D}^t(\mathbf{P}^t)$ as the set of pairs of targets i, j that are connected after a Delaunay triangulation on the target positions $\mathbf{P}^t := [\mathbf{p}_1^t \dots \mathbf{p}_N^t]$. We can now define the set $\mathcal{N}^t$ of neighboring targets at time t as

$$\{i, j\} \in \mathcal{N}^t \quad \text{if} \quad \{i, j\} \in \mathcal{D}^t \;\wedge\; d(\mathbf{p}_i^t, \mathbf{p}_j^t) \leq \lambda_{con} \tag{7.8}$$

where $d(\cdot, \cdot)$ is the Euclidean distance function and λ_{con} is a threshold parameter. We also choose to apply the transitivity constraint locally.

In particular we add a transitivity function for a triplet of subjects reciprocally connected by the Delaunay triangulation. While this solution still enforces the property locally, it does not model global transitivity, as Fig. 7.2 shows. We could prevent this violation by increasing the number of ψ^c factors to connect more and more grouping nodes. However, we believe that the local transitivity is a sufficient constraint and further complexity would offer little benefit.

7.3 Inference

In Sec. 7.2 we focused on building a model that could properly represent the main component of a multi-target tracker. In this section we deal with how to estimate the model variables. We did not concern ourselves with the computational feasibility issues so far. It turns out that the model that we described is inherently complex. One first issue is the connectivity of the graph. In the general case, the graph contains loops. Also, no assumption was made on any particular structure for the feature functions. Another, more important obstacle for exact inference lays in the fact that some of the variables of interest are, at least partially, continuous, albeit bounded. This is the case for the position $\mathbf{p}$ and the velocity $\mathbf{v}$ variables. Discretizing this variables jointly would require choosing a proper resolution parameter and, above all, would probably have unfeasible memory requirements. Finally, there is a problem due to the size of the problem itself. Multiple targets can be present in the scene for a long time. For each target we have a node at each timestep. We tackle all these problems in this section.

7.3.1 Belief Propagation with Continuous Variables

A common strategy to adopt when dealing with loopy graph, is Loopy Belief Propagation (LBP) [Murphy *et al.*, 1999]. Although in the general case it gives no optimality guarantees, this method and its variants have led to good results in the literature. In our case, as already mentioned, we have the additional problem that some the variables are continuous. Let us consider, as an example, the messages m exchanged between a state node of a target i and one of the neighboring interactions link ψ^l

Observation functions.

$f_a^o(\mathbf{s}_i^t) = \begin{cases} app(\mathbf{p}_i^t) & \text{if } u_i^t = 0 \\ 0 & \text{else} \end{cases}$	An appearance based online classifier $app(\cdot)$. Lower energy are assigned to locations that are likely to contain the object.
$f_d^o(\mathbf{s}_i^t) = \begin{cases} det(\mathbf{p}_i^t) & \text{if } u_i^t = 0 \\ 0 & \text{else} \end{cases}$	The detector function $det(\cdot)$ assign low energy to input which correspond to positions in the image with high detector score.
$f_u^o(\mathbf{s}_i^t) = u_i^t$	The cost of being in the *unreliable* state.

Motion functions.

$f_p^m(\mathbf{s}_i^t, \mathbf{s}_i^{t+1}) = \begin{cases} \Vert\mathbf{p}_i^{t+1} - (\mathbf{p}_i^t + \Delta\mathbf{v}_i^t)\Vert^2 & \text{if } u_i^t = u_i^{t+1} = 0 \\ \kappa & \text{if } u_i^{t+1} = 1 \\ \infty & \text{else} \end{cases}$	These motion functions assign a cost to the deviation from the constant velocity model $\begin{bmatrix} \mathbf{p}_i^{t+1} \\ \mathbf{v}_i^{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_i^t + \Delta\mathbf{v}_i^t \\ \mathbf{v}_i^t \end{bmatrix},$ while an infinite cost is assigned for the transition from $u_i^t = 0$ to $u_i^{t+1} = 1$). The transition to the unreliable state has a fixed cost.
$f_v^m(\mathbf{s}_i^t, \mathbf{s}_i^{t+1}) = \begin{cases} \Vert\mathbf{v}_i^{t+1} - \mathbf{v}_i^t\Vert^2 & \text{if } u_i^t = u_i^{t+1} = 0 \\ \kappa & \text{if } u_i^{t+1} = 1 \\ \infty & \text{else} \end{cases}$	

Group functions.

$f^c(g_{ij}, g_{jk}, g_{ki}) = \begin{cases} 1 & \text{if } g_{ij} + g_{jk} + g_{ki} = 2 \\ 0 & \text{else} \end{cases}$	It returns a cost for those configurations of grouping that violate the transitivity property.
$f^p(g_{ij}) = g_{ij}$	A fixed prior on the grouping variable g_{ij}.

Interaction functions.

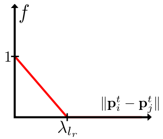
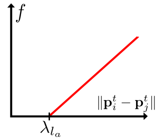
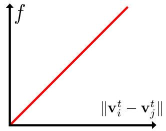
$f^l_r(\mathbf{s}^t_i, \mathbf{s}^t_j, g_{ij}) = \begin{cases} \max(0, 1 - \dfrac{\Vert\mathbf{p}^t_i - \mathbf{p}^t_j\Vert}{\lambda_{l_r}}) & \text{if } \; u^t_i = u^t_j = 0 \\ \kappa & \text{else} \end{cases}$	It models a repulsion effect when the targets distance is less than λ_{l_r}. It does not depend on whether the two targets are not in the same group or not (g_{ij} is not used). 
$f^l_a(\mathbf{s}^t_i, \mathbf{s}^t_j, g_{ij}) = \begin{cases} \max(0, \dfrac{\Vert\mathbf{p}^t_i - \mathbf{p}^t_j\Vert}{\lambda_{l_a}} - 1)g_{ij} & \text{if } \; u^t_i = u^t_j = 0 \\ \kappa & \text{else} \end{cases}$	It models the attraction between two targets of the same group. It is zero if the targets distance is less than λ_{l_a}. If targets i and j are not in the same group, the function has no effect. 
$f^l_s(\mathbf{s}^t_i, \mathbf{s}^t_j, g_{ij}) = \begin{cases} \Vert\mathbf{v}^t_i - \mathbf{v}^t_j\Vert g_{ij} & \text{if } \; u^t_i = u^t_j = 0 \\ \kappa & \text{else} \end{cases}$	A pair of targets in the same group have similar velocity. A linear cost is assigned to the norm of the velocity difference. 
$f^l_b(\mathbf{s}^t_i, \mathbf{s}^t_j, g_{ij}) = \begin{cases} g_{ij} & \text{if } \; u^t_i = u^t_j = 0 \\ \kappa & \text{else} \end{cases}$	This function acts as a bias term. It returns one it the two targets are not in the unreliable state and if they are not in the same group.

***Table 7.1**: Detailed description of the energy functions used to model the various building blocks of a tracker.*

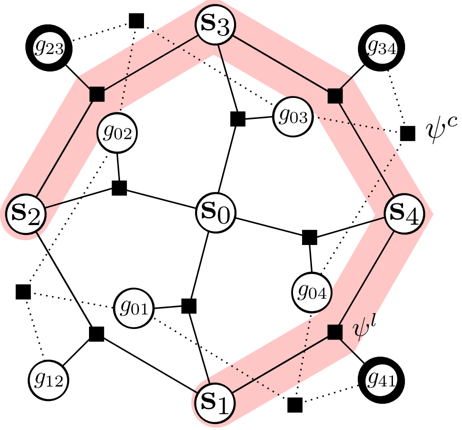


Figure 7.2: *The grouping nodes with bold borders are assigned a value 1, meaning they assign the two neighboring target nodes,* e.g. $\mathbf{s}_1$ *and* $\mathbf{s}_4$, *to the same group. The others are assigned a value of zero. Dotted lines connect grouping nodes to transitivity factors* ψ^c. *Note that* $g_{1,2}$ *is assigned a zero value, implying that the subjects* 1 *and* 2 *do not belong to the same group. However, we could as well infer that subject* 1 *and* 4 *belong to the same group* ($g_{1,4} = 1$)*, that subjects* 4 *and* 3 *belong to the same group* ($g_{3,4} = 1$) *and that subjects* 3 *and* 2 *belong to the same group* ($g_{2,3} = 1$)*. Therefore, assuming a transitivity property for group membership, we would conclude that also subjects* 1 *and* 2 *belong to the same group. However this is not true in our model. Global transitivity is not enforced, only local transitivity (by means of the* ψ^c *factors) is. This example was built with the purpose of showing this aspect of the model, but such situations are in fact extremely rare.*

at a certain time. The message passing adopted in belief propagation strategies could be generalized to the presence of continuous variables as follows

$$m_{i \rightarrow \psi^l}(\mathbf{s}_i) = \prod_{\psi \in ne(i)/\psi^l} m_{\psi \rightarrow i}(\mathbf{s}_i) \tag{7.9}$$

$$m_{\psi^l \rightarrow i}(\mathbf{s}_i) = \sum_{g_{ij} \in \{0,1\}} \int \psi^l(\mathbf{s}_i, \mathbf{s}_j, g_{ij}) m_{j \rightarrow \psi^l}(\mathbf{s}_j) m_{g_{ij} \rightarrow \psi^l}(g_{ij}) d\mathbf{s}_j \ , \tag{7.10}$$

where $ne(i)$ is the set of neighboring factors of the state node of target i. The integral in Eq. 7.10 is intractable in the general case. One solution is to resort to sampling in alternation with belief propagation [Ihler and McAllester, 2009; Sudderth *et al.*, 2003]. We use the Particle Belief Propagation (PBP) [Ihler and McAllester, 2009] for our purposes. The idea behind this method is to approximately solve the integral in Eq. 7.10 with importance sampling. Given an importance function $q(\mathbf{s}_i)$ we can derive the sampled approximation to Eq. 7.10

$$m_{\psi^l \rightarrow i}(\mathbf{s}_i) \approx \sum_{g_{ij} \in \{0,1\}} \sum_{\mathbf{s}_j \in \mathcal{S}_j} \psi^l(\mathbf{s}_i, \mathbf{s}_j, g_{ij}) \frac{m_{j \rightarrow \psi^l}(\mathbf{s}_j)}{q(\mathbf{s}_j)} m_{g_{ij} \rightarrow \psi^l}(g_{ij}) \ , \tag{7.11}$$

where $\mathcal{S}_j$ is the set of samples on the state of target j drawn from $q(\mathbf{s}_j)$. Note that the elements of the summation are now divided by the importance weight $q(\mathbf{s}_j)$. Also, note that the message can be evaluated on *any* value in the continuous range of the variable $\mathbf{s}_i$.

We have therefore a way to compute the messages m, but we did not specify the importance distribution q. We discuss this in the next subsection.

7.3.2 Sampling Strategies

In [Ihler and McAllester, 2009], the authors show that the best choice of importance distribution $q(\mathbf{s}_i)$ is the marginal distribution $b(\mathbf{s}_i)$ for the variable $\mathbf{s}_i$. This distribution is not available however, and the best approximation is only available at the end of the inference process. An increasingly more accurate approximation $\tilde{b}(\mathbf{s}_i)$ to this marginal distribution is however available at each iteration of the belief propagation, simply obtained by multiplying the incoming messages at each node

$$\tilde{b}(\mathbf{s}_i) = \prod_{\psi \in ne(i)} m_{\psi \to i}(\mathbf{s}_i) \; . \tag{7.12}$$

As shown in [Ihler and McAllester, 2009] this is a effective choice. Note that this product is a continuous function.
Sampling from the product of incoming messages, in the continuous case, is not a trivial task. A possible solution is to use a Markov Chain Monte Carlo (MCMC) sampling method, such as Metropolis-Hastings.
While perfectly valid, this strategy requires using MCMC to sample from each target node at each iteration of the algorithm. This is significantly demanding in terms of computation. We are not forced in principle to use the product of messages as the sampling distribution. The MCMC sampling works in the general framework of PBP, but it does not exploit all the structure of the specific problem, in our case, the tracker application. It was already mentioned that there is a strong correlation among two consecutive state nodes for each target. We exploit this temporal correlation to *propagate* the samples from one to the next target nodes. We therefore devise an importance sampling scheme, similar to Briers *et al.* [Arnaud *et al.*, 2005], but within the PBP framework.

The nodes that we need to sample are the target nodes. The grouping node indeed represents a binary variable that needs no sampling. The idea here is to sample from the target nodes sequentially. Let us assume a directed graphical model like the one in Fig. 7.3 for each target. This model does not need to be identical to the one in Fig. 7.1, top-left. Let us define the distribution from which we would like to sample as

$$q^*(\mathbf{s}^t | D^{0\ldots t}) \tag{7.13}$$

where $D^{0\ldots t}$ is the vector of the available evidence up to the current timestep t. Assuming the model of Fig. 7.3, we can rewrite the distribution as follows

$$q^*(\mathbf{s}^t | D^{0\ldots t}) = \int q^*(\mathbf{s}^t, \mathbf{s}^{t-1} | D^{0\ldots t}) d\mathbf{s}^{t-1} \tag{7.14}$$

$$= \frac{q^*(D^t | \mathbf{s}^t)}{q^*(D^t)} \int q^*(\mathbf{s}^t | \mathbf{s}^{t-1}) q^*(\mathbf{s}^{t-1} | D^{0\ldots t-1}) d\mathbf{s}^{t-1} \tag{7.15}$$

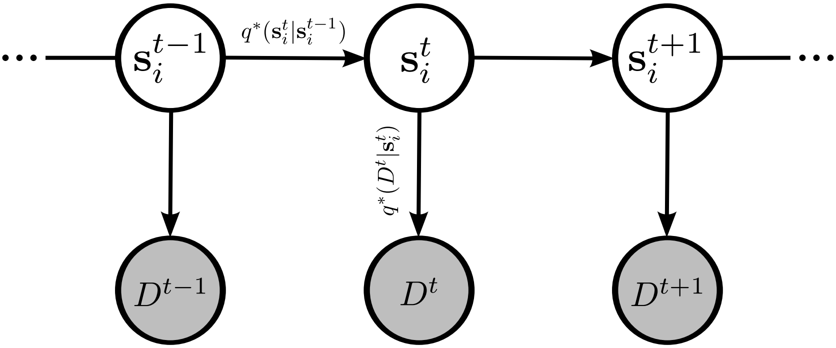


Figure 7.3: *The directed graph model for sequential sampling. Rather than sampling from the individual belief estimates as in the PBP framework, we propose to use sequential sampling to exploit the sequential structure of the problem. The chain in the figure is used for this purpose. Here* $q^*(\mathbf{s}_i^t|\mathbf{s}_i^{t-1})$ *is the time propagation density and is defined in Eq. 7.17, while* $q^*(D^t|\mathbf{s}^t)$ *is the observation density and is defined in Eq. 7.18.*

If, for the moment, we assume that samples $\mathcal{S}_j^{t-1}$ are available from the distribution $q^*(\mathbf{s}^{t-1}|D^{0\ldots t-1})$, then we can write

$$q^*(\mathbf{s}^t|D^{0\ldots t}) \approx \frac{q^*(D^t|\mathbf{s}^t)}{q^*(D^t)} \sum_{\mathbf{s}^{t-1}\in\mathcal{S}_j} q^*(\mathbf{s}^t|\mathbf{s}^{t-1}) \tag{7.16}$$

Eq. 7.16 has the form of a mixture model. Note that the denominator $q^*(D^t)$ is not dependent on $\mathbf{s}^t$ and therefore it can be treated as a constant. Sampling from Eq. 7.16 therefore reduces to a weighted sampling of the propagation density $q^*(\mathbf{s}^t|\mathbf{s}^{t-1})$. We see how to this below. Now, to finish showing that we can sequentially sample from this distribution, we need to show that we can sample from $q^*(\mathbf{s}^0|D^0)$. We can assume that the first node of the chain for each target, is a discrete node, therefore we can always sample from it. In Sec. 7.3.4 we see that the target initialization provides us with a state from which we can easily sample. Finally, we need to define the propagation density $q^*(\mathbf{s}^t|\mathbf{s}^{t-1})$ and the observation density $q^*(D^t|\mathbf{s}^t)$. We keep in mind that our goal is to have the importance function as similar as possible to the marginal for $\mathbf{s}^t$ [Ihler and McAllester, 2009]. For the propagation function, we use a function similar to the motion function described in Tab. 7.1. These energy functions implement a motion model from which we cannot directly sample, because of the presence of the unreliable state. We use instead the following propagation function

$$q^*(\mathbf{s}_i^t|\mathbf{s}_i^{t-1}) = \alpha\delta(\mathbf{s}_i^t - \hat{\mathbf{s}}^{t-1}) + (1-\alpha)\begin{bmatrix} \mathcal{N}\left(\begin{bmatrix} \mathbf{p}_i^t \\ \mathbf{v}_i^t \end{bmatrix}; \mu_i^t, \mathbf{\Sigma}\right) \\ \delta(u_i^t - u_i^{t-1}) \end{bmatrix} \tag{7.17}$$

with

$$\hat{\mathbf{s}}^{t-1} = \begin{bmatrix} \mathbf{p}_i^{t-1} \\ \mathbf{v}_i^{t-1} \\ 1 \end{bmatrix} \qquad \mu_i^t = \begin{bmatrix} \mathbf{p}_i^{t-1} + \Delta\mathbf{v}_i^{t-1} \\ \mathbf{v}_i^{t-1} \end{bmatrix}$$

$$\mathbf{\Sigma} = \frac{1}{2}\begin{bmatrix} (\theta_p^m)^{-2} & 0 & 0 & 0 \\ 0 & (\theta_p^m)^{-2} & 0 & 0 \\ 0 & 0 & (\theta_v^m)^{-2} & 0 \\ 0 & 0 & 0 & (\theta_v^m)^{-2} \end{bmatrix},$$

where $\delta(\cdot)$ has value zero everywhere except when its argument is $\mathbf{0}$, α is a parameter that regulates the random *switch* from the reliable to the unreliable state and the variance parameters $\mathbf{\Sigma}$ are the same parameters that are used to multiply the motion functions in the motion factor ψ^m. Notice that in the propagation function of Eq. 7.17, the u_i can only change to 1 and never to 0, in accordance with the motion model energy functions. We can sample from Eq. 7.17 by randomly choosing one of the two terms: we choose the first term with probability α and the second term with probability $(1-\alpha)$. If the first term is chosen, there is only one choice for the sample of the state $\mathbf{s}^t$, *i.e.* $\hat{\mathbf{s}}^{t-1}$. If the second term is chosen, position and velocity are sampled from a multivariate normal density function, while the u_i^t keeps the old value u_i^t.
The observation function $q^*(D^t|\mathbf{s}^t)$ acts as a weight for the samples from Eq. 7.16. As we already described for the MH sampling strategy, at each iteration of the belief propagation algorithm, we have a set of messages being sent to all the nodes. It seems therefore reasonable to use the messages coming from the neighboring factors of node $\mathbf{s}_i^t$, with the exclusion

of the time factors ψ^m (that are already accounted for in the propagation function), to implement the observation function

$$q^*(D^t|\mathbf{s}^t) = \prod_{\psi \in ne(i)/\psi^m} m_{\psi \to i}(\mathbf{s}_i) \; . \tag{7.18}$$

Here, slightly abusing the notation, we included the messages in the available evidence. In this way the weights convey the information coming not only from the images, but also from the interacting targets.

7.3.3 Splitting the Inference

So far we have seen how to adapt LBP to a graph with continuous variables. There are two further considerations. The first is that performing inference for all the graph at the same time might be prohibitive, especially in terms of memory requirements. The tracker might be used for long sequences, depending on the task, and the amount of data (samples and images) necessary for the inference might be too big to fit in memory. The other consideration is that, sometimes, tracking applications require an estimate as soon as new data becomes available. Therefore waiting for the end of the sequence to perform inference would not be appropriate in these situations.

The solution we employ is to perform the inference in temporal windows. The idea is to split the graph into partially overlapping subgraphs, and perform inference on each of these. Fig. 7.4 shows an illustrative example. For the sake of clarity we show only the portion of the graph relative to a single target (i) for the first 3 time steps. The graph is split in such a way that the all the target chains end at a certain time (time 1 in the figure) and restart with the same node in the next subgraph. Note that the grouping nodes, like g_{ij}, are repeated in all the subgraphs. The splitting shown in Fig. 7.4 detaches the first graph slice from the whole graph. The procedure can be repeated (on the rightmost subgraph in figure) to obtain the other time slices.

The inference is carried out for each graph slice in chronological order. To exploit the information coming from previous slices of the graph, the messages to overlapping nodes (g_{ij} and $\mathbf{s}_i^1$ in the figure) coming from factors ψ that are not repeated in the successive slice are multiplied and collected in special connection factors $\tilde{\psi}$. This is not the same as performing the inference with the whole graph at the same time. The main difference is that the information can propagate only from one graph slice to the following ones, but not backwards. Also, the overlapping target nodes are sampled only in the first graph slice in which they appear. At the end of the inference for that slice, the samples and the sampling weights are stored. When the overlapping nodes are used again in the following graph slice, they are not resampled, but the stored values are re-used.

The sequential sampling scheme introduced in Sec. 7.3.2 is also adapted to cope with the graph splitting. Since the samples and the sampling weights of the overlapping target nodes ($\mathbf{s}_i^1$ in figure) are stored for the PBP on the next graph slice, they can also readily be used for sequential sampling.

We employed this approximation also to be able to provide an estimate at the end of each graph slice. Sometimes, for example for visualization or for evaluation purposes, a hard decision for each variable is necessary. Rather than a max-marginal estimate for each node, we use dynamic programming for each target chain within the graph slice to obtain smoother results. The dynamic programming uses Eq. 7.18 to compute the unary costs for each chain node, thus accounting for observation model and interactions. The transition costs from one node to the next are evaluated using the ψ^m factor of the model. In particular, if the estimate returns the unreliable state for a target, that target is not propagated to the next graph slice.

7.3.4 External Modules

In this subsection we discuss the tracker components that we did not manage to include in the graphical model framework.

One external component is initialization, that is, the component that decides whether one or more targets have to be initialized for tracking. We have already seen that the inference is not carried out in one single step for all the graph, but rather it is performed in a temporal window. When a new inference window is initialized, new targets might have entered the scene. We use an external module that, given the detector output and the target estimates until the current time step, infers whether there are new targets to be initialized. This is done by looking at the local max-

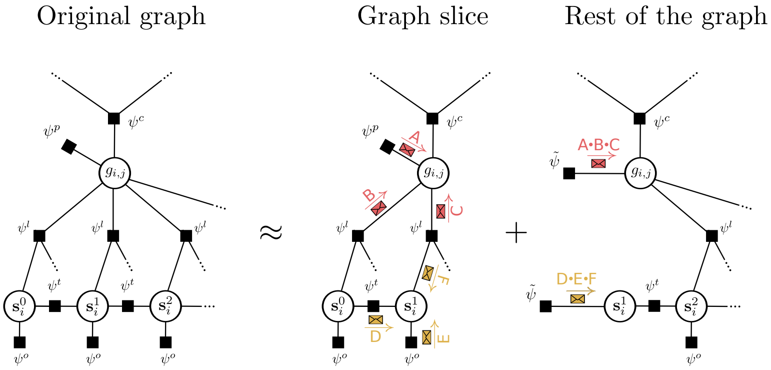


***Figure 7.4**: Graph slicing. The undirected graphical model on the left shows a section of the graph built for an entire sequence (See Sec. 7.2). Both because of the limited computational resources and to allow the method to return results as new data becomes available, we extract and compute the result for one slice at a time (center). This slice contains all the subject nodes for a certain time window and all the grouping nodes. In this figure, the slice contains only two consecutive time-steps, but this need not be the case in general. As the rest of the graph (right) is not used for the current computation, it does not even need to be known. This is the reason why we can apply the method sequentially as new data becomes available. Once the inference stage is complete, we repeat the same slicing operation on the remaining graph. The slices are not treated completely independently however. The result of the inference on previous slices is propagated to the following ones, as illustrated by the A,B,C (for the grouping variables) and D,E,F (for the subject variables) messages that are sent from one slice to the next. Note, however, that the information does not travel backwards.*

ima of the detector image, and while visiting them in order of decreasing detector score, accepting only those that do not overlap with more than a certain threshold to those that have already been initialized. This process is performed only at the beginning of each inference window. Since from the single detector image we have no estimate of the velocity of the newly initialized target, we initialize the first state samples for each target with the position given by the detector and the velocity sampled uniformly in all directions and within a reasonable speed. The first node can be considered therefore a discrete node, with all the discrete choices equally likely and therefore easy to sample from, as required in Sec. 7.3.2. Finally, to be robust with respect to poor initialization, the target for the first inference window is not linked by interaction factors ψ^l to the other targets. In other words, for the duration of the first inference window, it is tracked independently.

The other component that we could not cast into the framework is the one responsible for updating the appearance model. Once a target is initialized, the appearance model is built by using as a positive example the initialized position, and as negative examples positions taken at random in the vicinity of the positive example. The appearance model needs to be updated however, because of the well known appearance change that targets undergo. These updates are carried out using the same strategy employed for building the appearance model, but as positive examples we use samples with high detector score, similarly to [Breitenstein *et al.*, 2011].

7.3.5 Implementation Remarks

Finally, in this section we account for some technical non-trivial caveats that are necessary to properly implement the tracker.

With regard to propagation to the unreliable state (Eq. 7.17), when a certain number of samples are propagated to the unreliable state, the effective number of samples that represent the position and velocity part of the state decreases. Therefore the accuracy of the representation decreases and it becomes more likely for the samples to be propagated to the unreliable state. This process can favor the premature termination of a target. In order to avoid this, while resampling in Eq. 7.17, we try to keep the number of samples that are in the reliable state equal to their initial number, by increasing the number of output samples as new unreliable samples are drawn. However, we do not sample more than twice the initial number of samples.

The sampling procedure described in the previous section, both in the MCMC form and in the sequential sampling one, are parallelizable. In the former case, each node can be sampled independently, while in the latter this is true for each target chain. In the same way, the caching of the energy function values, as described above, can be carried out in parallel. With the widespread growth of parallel computing, and seen the resource requirement of the tracker presented in this work, this becomes necessary, rather than an option.

7.4 Experiments

In the following we show the experiments we carried out in order to evaluate the tracker. For the evaluation, we need data that can challenge all the major aspects of a multi-target tracker. In particular, the tracker should work on a scene with several naive targets. Instructing targets to walk in various configurations might fail in reproducing those interaction patterns that we want to model. Furthermore, since we need many targets to be visible in the scene at the same time, we chose sequences with cameras set high enough to capture a big portion of the scene.

7.4.1 Datasets and Setup

Students. This is an outdoor sequence provided by a third party [Lerner *et al.*, 2007] that we manually annotated to collect ground truth. The sequence is particularly challenging due to the high number of subjects in the scene, the multiple patterns of motion, the compression artifacts and the strong shadows. Although the background is static, we do not use this information and rely on a person detector [Gall and Lempitsky, 2009]. Given the position of the camera, the detector was trained on the same scene from a small set of frames (50). We used another small subset of the frames (300) to choose the best parameters θ, κ and λ. To choose the parameters we used a simple gradient descent with fixed step

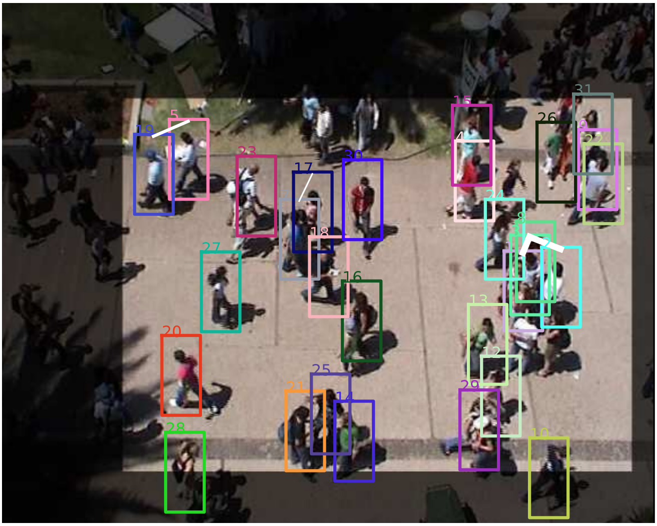


***Figure 7.5**: A result frame from the Students sequence. The white line connecting the bounding boxes shows the group relationship estimate. The thicker the line, the stronger is the estimate that the two connected subjects belong to the same group.*

size, one coordinate at the time, optimizing for accuracy. After removing all the frames used for training, we were left with 4400 frames at 25fps (i.e., about 3 minutes of video) that we used for testing. Fig. 7.5 shows a screenshot of the tracker results. The darker area in the image has not been used for tracking. This was done both to avoid border effects and to avoid dark shadows and stairs (for the top and left part of the images).

BIWI-Walking. These sequences have been made available by [Pellegrini *et al.*, 2009]. They have been captured with almost top-view cameras and show people walking in a busy street (Fig. 7.6) and at the entrance of a public building (Fig. 7.7). We kept the parameters used for the Students sequence. The detector has been re-trained for the busy street sequence on a portion of the data while for the building entrance, because of low resolution and compression artifacts, a simple background subtraction has been employed.

Mycoplasma. We also used a video sequence from [Uenoyama and Miyata, 2005], showing moving bacteria. As a detector we used a logistic function (trained on a single frame) on the intensity on the image. The bacteria intensity is substantially different from the background and provides a simple but effective detector. Like before, we cut out the borders of the image to overcome border effects. No ground truth was available for this sequence. For the parameter settings, we used the same parameters from the Students sequence, with the only exception of the velocity parameter θ_v^m that was reduced to account for more irregular target motion.

For the following experiments, we set the size of the inference window to 5, as a compromise between the computational requirements, the need of producing results as soon as new data becomes available and the desire of exploiting the temporal correlation. We use a fixed aspect ratio and scale within each sequence for all the targets, as these quantities do not vary much in the images. Finally, we use 100 samples per target and we set the interaction threshold λ_{con} from Eq. 7.8 to $2m$.

7.4.2 Grouping Influence

To highlight the importance of the grouping component of the tracking, we show the output of the system when tracking groups with two different setups. In the first we manually set $g_{ij} = 1$ for all the pair of subjects i and j within the same group. In the second, we instead manually set all the groups relations to 0, that is $g_{ij} = 0$ for all i and j in the scene. A first example is shown in Fig. 7.6. The sequence is extracted from the BIWI-Walking dataset. Note that subject 1 is tracked properly in both setups, while 2 is not. The observation model is indeed weaker because of the partial occlusion of the tree branches. In the setup with the group correctly initialized the failure is avoided because the velocity of subject 2 is affected by the velocity of subject 1, that, as stated, is well tracked and *pushes* 2 forward.

Another example is shown in Fig. 7.7. This sequence is extracted from the same dataset as the previous one and shows the entrance of a public building. The two tracker settings are the ones used in the previous experiment, with the difference that now several groups are extracted. The figure shows the comparison of the two tracker setups. While in this

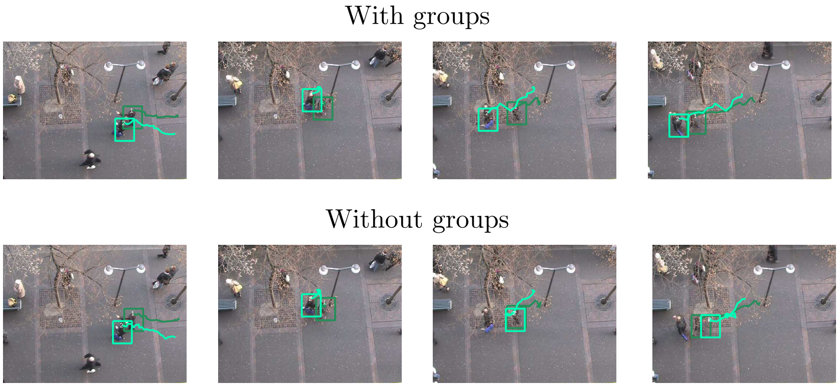


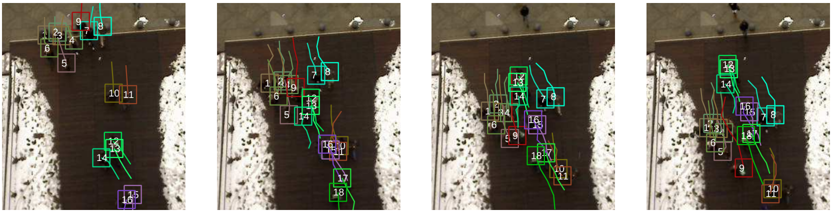
Figure 7.6: *Two subjects belonging to the same group. Two setups are compared: the one with the group correctly initialized (top row) and the one in which the two subjects do not belong to the same group (see text for more details). Note that subject 1 is properly tracked in both setups. The grouping component of the tracking uses the velocity of subject 1 to estimate the velocity of subject 2 when the observation model is weak due to the partial occlusion of the tree branches.*

sequence, because of the poor quality of the background subtraction, the trajectory precision is much lower, the difference in accuracy still favors the setup with the groups correctly initialized. Note how the group prior favors the cohesion of the trajectories.

7.4.3 Group Formation and Splitting

Although our modeling of groups is only static and no merging or splitting are explicitly defined, we observe similar dynamics emerging from the inference strategy that we use. In particular, by doing inference on a time window basis, we actually update the estimate of the group relation as new evidence becomes available. Fig. 7.8 shows the progressive estimate of the grouping relation among few subjects. Note that eventually the local transitivity property holds for all the triplets of subjects in the figure. The group relationship is a result by itself that can be used, for example, for scene understanding [Chang *et al.*, 2011].
Group splitting is also not directly modeled in our framework. However

With groups



Without groups

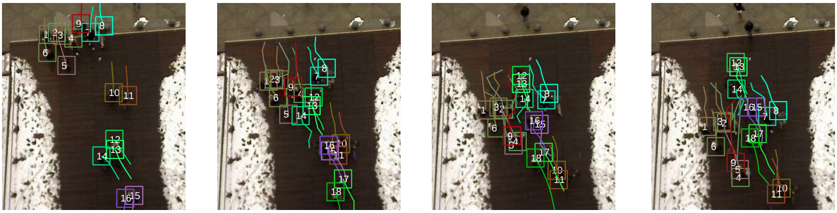


***Figure 7.7**: Several groups interacting while walking in opposite directions. Subjects within the same group are assigned similar colors and ID number. In particular, we have the following groups: {1, 2, 3, 4, 5}, {7, 8}, {9}, {10, 11}, {12, 13, 14}, {15, 16} and {17, 18}. Two tracker settings are compared (see text and Fig. 7.6 for more details). Note that subject 9, in the second image, is incorrectly estimated going through the left-most group in the setup without groups.*

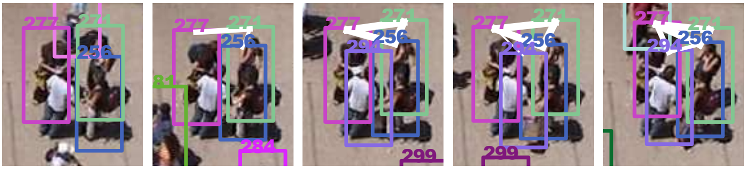


***Figure 7.8**: Group formation. The thickness of the white line is proportional to the estimated probability of the two subjects being in the same group. The frames being shown are, from left to right, 4000, 4100, 4300, 4320 and 4350.*

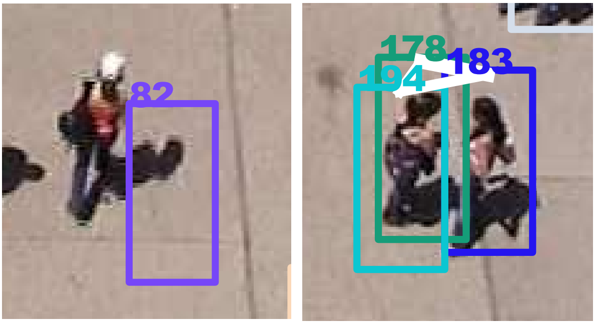


Figure 7.9: *Tracker errors. Some of the tracker errors are caused by a wrong initialization. In the* ***left*** *figure, the shadow is detected instead of the person. Furthermore, the shadow moves indeed like a human, and this makes it harder for the inference to infer that there is a failure. The* ***right*** *figure shows another mistake. This time is the bag that is being recognized as a person and is also being assigned the same group as the other two subjects.*

group sometimes split and the model should not hallucinate members walking together. If the observation model is reliable enough, this does not happen. Fig. 7.10 shows a case in the Students sequence. This is also an example of recovery from wrong initialization. One major reason for false positives in the tracker is given by wrong initialization, as the detector fires often on shadows and backpacks (see Fig. 7.9). The tracker is anyhow capable of alleviating this problem by not trusting completely the initializer. The inference is indeed able to assign the unreliable state to those tracks that violate the interaction model and/or do not have much support from the observation model.

7.4.4 Avoidance

While the appearance model provides already a means for disambiguation among targets, when the targets look similar and the resolution is relatively low, the appearance model is not enough to prevent the stronger observation explaining multiple tracks. In Fig. 7.11 we compare

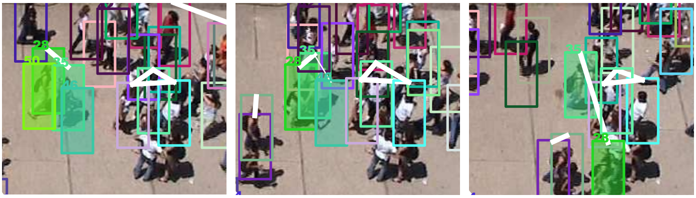


Figure 7.10: *Tracker failure detection. Sometimes, the initialization or the tracker itself lead to a wrong estimate. The leftmost figure shows such a case, where for the two subjects with ID 28 and 35, there is another pair of wrong estimates, namely 50 and 46. The tracks of interest are represented with a shaded bounding box. The tracker finds out that the estimates are wrong in the following frames. This happens because the wrong tracks violate the avoidance behaviors, since they overlap with other tracks and lack of a strong observation support. We show frames, from left to right 1175, 1200 and 1250.*

the result of the tracker when using the interaction terms as previously described and when not using them.

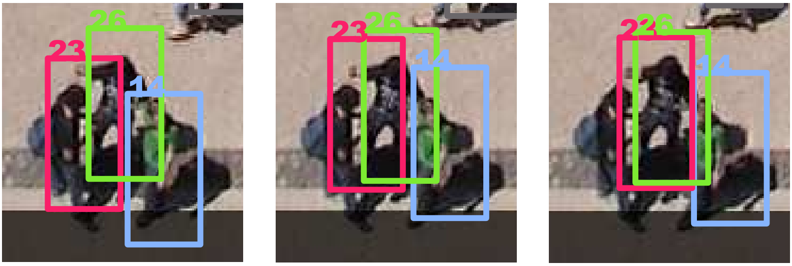
In Fig. 7.12 we show few frames from the result of the bacteria tracking experiment. In the right part of the figure we follow a single bacterium, from the initialization to when it leaves of the scene. Each frame reports an interaction with other bacteria. Even if they come very close to each other, the id of the targets are preserved. Note that there is no group formation in the bacteria experiment. This happens because these kind of bacteria rarely move together.

7.4.5 Quantitative Results

The output of the tracker strongly relies on the quality of the detector. In Fig. 7.13 we plot the precision and accuracy for the detector alone for the Students sequence. We also evaluate the tracker as a detector by using the best estimate (see Sec. 7.3.3) of tracked targets at each frame. The results show that our system is capable of keeping track of the targets also when these are not clearly visible to the detector.

We finally evaluate the performance of the tracker on the Students

Without interactions



With interactions

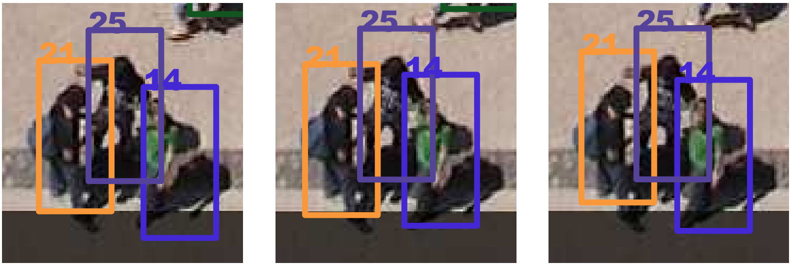


Figure 7.11: ***Top***: *results when using no interaction terms. When the appearance difference is not strong enough, as it happens for the two similarly looking subjects, the tracks overlap as the one with the stronger observation support (ID 26, in this figure) attracts the other (23).* ***Bottom***: *this problem is solved using an interaction term, in this case specifically the avoidance term.*

sequence, using the CLEARMOT metric [Bernardin and Stiefelhagen, 2008]. We compare our method with a baseline instantiation of the tracker that does not use the interaction model The tracker relies therefore on the detector and on the online classifier to disambiguate multiple tracks. We also use an instantiation of the tracker that assigns all the subjects to separate groups, so that group interactions are not used. The result are shown in Tab. 7.2. As discussed already in the previous subsection, interactions, especially the avoidance behavior, are indeed necessary to avoid multiple tracks being explained by the same observa-

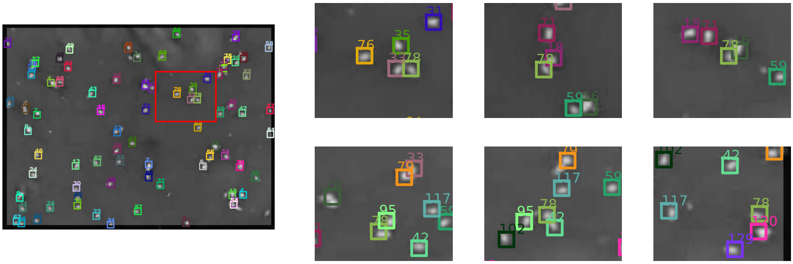


__Figure 7.12__: The figure on the __Left__ shows a screenshot of the tracker applied to the Mycoplasma dataset. The red bounding box identifies the bacterium with ID 78 at the moment of initialization. The bacterium is followed in the __right__ side in the smaller images. Note that the bacterium approaches other almost identical bacteria.

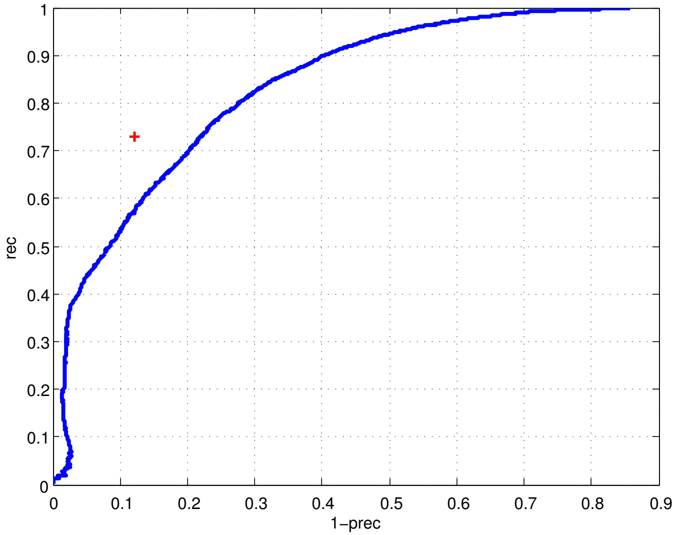


__Figure 7.13__: Recall/(1 - Precision) curve for the detector. The red cross shows the result of the tracker output when treated as a detector.

Method	MOTA	MOTP	FN	FP	MM
Full	**67.3%**	75%	21.7%	9.4%	1.6%
No Group	65.3%	74%	26.2%	6.9%	1.7%
No Int.	52.5%	70%	20.2%	25.2%	2.1%

***Table 7.2**: CLEARMOT [Bernardin and Stiefelhagen, 2008] evaluation, showing accuracy (MOTA), precision (MOTP), false negative rate (FN), false positive rate (FP) and the mismatches rate (MM). We compare the results of the full tracker presented in this chapter ("Full"), with an instantiation that does not use grouping ("No Group") and with one that does not make use of interactions at all ("No Int.").*

tion. Grouping offers a marginal improvement, due mostly to a reduction in false negatives, while it produces more false positives.

In our previous work [Pellegrini *et al.*, 2010], we present a data association method that we apply on short (2 seconds) sub-sequences of the Students dataset. Although the tracker we presented is capable of automatic initialization and can cope with longer sequence, we use the same initialization from ground truth and the same experiment length in order to compare the two performances. Using the same test set (2000 frames, from frame 1000 to frame 3000) we initialize the tracker at each of the 40 sub-sequences using the ground truth annotation. As the sequences are short there is no need of using the unreliable state in the state variables for the subject. Using our approach we achieve $\sim$ 79% correctly predicted trajectories[1] when interactions are not used and $\sim$ 90% with interactions. This offers a considerable improvement over the $\sim$ 70% result reported in [Pellegrini *et al.*, 2010].
As for the grouping results, we plot the precision and recall in Fig. 7.14. We achieve an Equal Error Rate of $\sim$ 62%, compared to the $\sim$ 46% precision and $\sim$ 82% recall reported in [Pellegrini *et al.*, 2010]. For the same sequence, the authors of [Leal-Taixé *et al.*, 2011] report 80.5% precision

---

[1] As in [Pellegrini *et al.*, 2010], a trajectory is correctly predicted when the tracked position at the end of the 2 seconds is within a threshold of 0.5 meters from the ground truth.

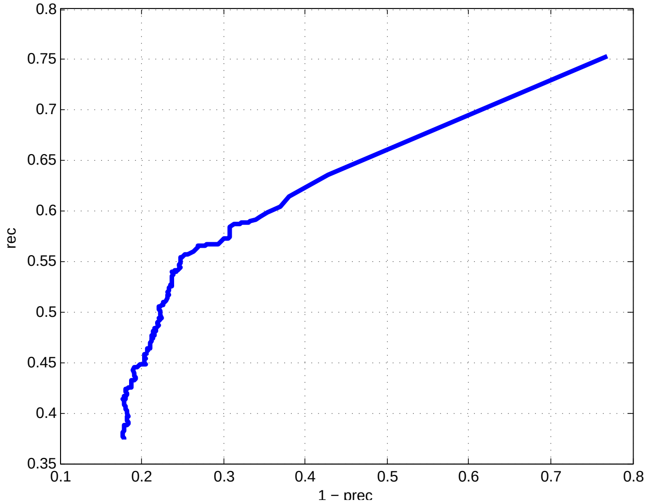


Figure 7.14: *Recall/(1 - Precision) curve for the group classification. To produce this result, the tracker has been run on 40 consecutive subsequences of the Student sequence, each lasting 2 seconds. The targets were initialized using the ground truth.*

and 77% recall. When comparing the results we should note that in our case, grouping variables are only present where a link of the Delaunay triangulation is present. Therefore, we cannot achieve 100% recall no matter what threshold we use. Furthermore, in our case grouping and tracking are carried out jointly.

7.5 Conclusions

In this chapter we presented a principled model for a tracker that unifies an observation model, motion model and interaction model within the same framework. We cope with initialization errors and with tracker failures in the same way by means of an additional state variable. The target tracks and the group memberships are estimated jointly, so that each of them can propagate information to the other during the inference. The interactions themselves, namely the grouping relationship, are also propagated by means of an additional layer of connections that models local transitivity.

The proposed system can be seen as a generalization of Particle Filters, in that it uses a sequential sampling scheme. However it is different from a Particle Filter solution in many aspects, including the use of Particle Belief Propagation and the interaction factors.

One of our goals was to propose a model that is modular, suitable to extensions and modifications. It should be easy, as a result, to add other functions to represent more properties. In our tracker we used a relatively small number of functions, and training the parameters has been carried out with a simple search on a validation set. While adding many other functions is possible, with the increasing number of parameters, a different, more efficient learning strategy must be planned. Since a log-linear model has been used, learning strategies like Contrastive Divergence [Hinton, 2000] might be exploited.

Another advantage of modeling the whole tracker as a graphical model, is that it can readily deal with externally provided information, such as user annotation. This allows to easily extend the tracker to an interactive one. The user annotation forces the node variables to a particular state and the rest of the inference process stays the same. We developed such an interactive interface and we are currently using it to assist the tracker.

The tracker presented in this chapter is not capable of real-time performance. The computational time depends on the number of samples used, but also on the number and density of targets in the scene. As an example, for the Students sequence, with about 30 people per frame in the scene and 100 samples per target, the full tracker requires about 3 seconds per frame. Code optimization and more parallel computation would alleviate this limitation.

8

Conclusions

The aim of the research in this thesis has been to investigate the benefits of using social interactions for a series of applications, from tracking to simulation. We first proposed a novel steering model that accounts for the anticipation with which people carry out obstacle avoidance and interact with other people in the same group. We further extended this model to include a goal selection layer and demonstrated its capability to simulate specific real scenes with minimal effort. This allowed us to build a mixed reality simulation, with both simulated and real agents. The motion prior knowledge has been then used for tracking, in a combined multi-hypotheses system, and for trajectory prediction. Finally, we investigated the interplay between tracking and interactions (grouping in particular). We now conclude the thesis discussing the presented contributions and giving an outlook of possible research directions.

8.1 Discussions

When designing LTA (see Chapter 2), we accounted for the way that people anticipate static or dynamic obstacles. In doing so, we used a linear velocity predictor and focus on the moment of maximum approach, as estimated through this predictor. In this dissertation, we showed that we can reproduce walking paths similar to those of humans (see Chapter 3, Chapter 4 and Chapter 5). Also, properties like elliptical personal area, that naturally arise from the specification of the LTA avoidance component, have been confirmed by other studies [Gérin-Lajoie *et al.*, 2005]. Furthermore, we showed that we can include this knowledge in a tracker motion model (Chapter 4 and Chapter 5) to improve its performance. Predicting the target future path using LTA is expected to improve the tracker performance when the observation is unreliable for a longer time period, for example when occlusions occur.

LTA has been designed also to allow for easy extension. Including or not group membership in the model can be done without affecting the model architecture. Other features can be included to build a more accurate model. Personal, gender or cultural factors are known [Patterson *et al.*, 2007] to affect the individual's walking behavior. It is unlikely, however, that increasing the model complexity *horizontally* would yield significantly better results, while would certainly increase the computational cost. Rather then exploring this direction, we preferred extending the model in a stochastic way (Chapter 5), accounting for multiple alternatives for each pedestrian. We deem this approach more robust and still computationally feasible and we showed promising results in predicting trajectories within a time horizon of about five seconds.

Another direction in which we extended the model is that of adding the goal selection layer (Chapter 3). This *vertical* extension proved very useful in the task of crowd simulation, as it allowed reproducing the motion patterns for a specific scene with little effort. As a consequence, we were capable of smoothly mixing real and virtual agents in our simulation. This was possible because both kinds of agents behave in a meaningful way, once interpreted in the context of the particular scene.

An important aspect of social walkers, is that they walk in groups of people. Although this is a very common experience [Moussaïd *et al.*, 2010], it is rarely exploited in motion models. This might also be the case because the knowledge of whether two people are in the same group or not is not immediately available in the image, but requires further effort, be it processing or manual annotation. In this thesis we showed some preliminary results on the task of group classification (Chapter 6). It appears that once the trajectories are available, albeit not completely accurate, the goal of extracting group membership is a feasible one. Unfortunately trajectories are not available most of the time. It is reasonable to expect, on the other hand, that if we knew already group memberships for the targets, the tracking performance could be improved. This reasoning is based on the observation that people in the same group walk together, therefore the location of one target offers a prior on the location of the other targets in the same group. Therefore, rather than approaching the two problems separately, we decided to cope with a joint inference problem for tracking and grouping. We showed that the group classification performance was similar to the one obtained when the trajectories are given. Furthermore, the tracking seems to benefit as well from the group membership information.

The tracking system that we showed in Chapter 4 is a combined system that uses many clues, such as ground plane or depth information, to achieve a good performance. In many situations, however, it is not possible to rely on a stereo camera or the camera might be far from people, thus offering low resolution images of each target. We investigated whether in this situations it is still meaningful to use targets interactions. Rather than a combined system, we decided to build the interactions within the tracker (Chapter 7). One of our goals indeed, was to build an extensible and modular tracker, where interactions could be treated as any ordinary component, *e.g.* the observation model. The system that resulted was a complex but accurate model of the space and time behaviors of the targets. To cope with the complexity we devised an inference strategy capable of exploiting the structure of this specific problem. We were finally able to successfully track targets over different and challenging sequences, showing the benefits of the chosen approach.

8.2 Outlook

We list now some research directions for future work.

Density and LTA. The LTA model has been tested in different scenes and applications. As discussed, one of the main features of LTA is the fact that the predicted trajectory anticipates the avoidance with other obstacles. This is reasonable when the density of the scene is not too high. In situations like those that arise in panic scenarios, probably the behavior of pedestrians is better modeled by reactive models, like the Social Force model [Helbing and Molnár, 1995]. In general LTA does not explicitly account for density variations. The selection of T in Eq. 2.5 implicitly allows to focus on a different time horizon. When T is particularly small the model behaves reactively (see Chapter 3). In this thesis we used a fixed value of T for all the subjects. To achieve a more flexible behavior, however, different values of T can be used, based on the density or on the group relationship between individuals. In general, a more explicit inclusion of density in the model might be beneficial.

Cognitive LTA. In Chapter 3, we have shown how simple behaviors emerge from goal driven simulations. The agents are not given any specific instructions about performing a particular action, yet they seem behaving as if they are waiting the tram or gathering around a table. This is possible because most of these behaviors are characterized by the place in which they occur. More complex behaviors could be included in the simulation by extending the goal selection layer, for example including proper actions. One of the goals in the work we presented has been that of showing the benefits of blending computer vision and computer graphics techniques. Adding an action recognition within the framework would then represent a continuation in the same direction.

Visual clues. In this thesis the grouping, both in synthesis and analysis, has been modeled only through proximity features. People that belong to the same group, both when walking or standing, often look at each other, gesticulate and interact through body language. In those tracking experiments where we employed grouping, the low resolution prevented us from relying on detailed body pose information. If instead higher quality image sequences are available, an interesting extension would be to use visual clues to aid the task of group classification.

Pose and Scene. As discussed before, the tracker presented in Chapter 7 has been built with the goal of realizing a modular and extensible system. Beyond the improvement in the tracking itself, we believe that an interesting research direction would be that of including more top down information, like scene related knowledge and bottom up information, *e.g.* by using articulated motion [Andriluka *et al.*, 2008]. We did some preliminary work in this direction [Pellegrini *et al.*, 2008] but further investigation is clearly needed.

A

Derivation of the marginal probabilities

In this appendix we are going to show a derivation for $p(\mathbf{p}_i^t, \mathbf{v}_i^t)$, and $p(\mathbf{p}_i^t)$, from Chapter 5, *Stochastic LTA for Prediction*,.

We start by factorizing the probability of the state $\mathbf{S}^t$

$$p(\mathbf{S}^t) = \prod_{i=1}^{N} p(\mathbf{p}_i^t, \mathbf{v}_i^t) \ . \tag{A.1}$$

Now, we can assume that $p(\mathbf{p}_i^0, \mathbf{v}_i^0)$ is initially given as a mixture of Gaussians (possibly with a single component). To show, by induction, that the the marginal $p(\mathbf{p}_i^t, \mathbf{v}_i^t)$ will have the form of a mixture of gaussians, we assume that at time $t-1$ the distribution $p(\mathbf{p}_i^{t-1}, \mathbf{v}_i^{t-1})$ is already a mixture of gaussians and prove that this is sufficient for $p(\mathbf{p}_i^t, \mathbf{v}_i^t)$ to have the same form.

For the moment, the fact that the each factor of Eq. A.1 is a mixture of gaussians, say with H^t component, means that $P(\mathbf{S}^t)$ will be itself a mixture of gaussians with $M^t = (H^t)^N$ components

$$p(\mathbf{S}^t) = \prod_{i=1}^{N} \sum_{k=1}^{H^t} w_{ih} \mathcal{N}(\mathbf{p}_i^t, \mathbf{v}_i^t; \mu_{ih}^t, \mathbf{\Sigma}_{ih}^t) \tag{A.2}$$

$$= \sum_{m=1}^{M^t} w_m \mathcal{N}(\mathbf{S}^t; \mu_{\mathbf{S}_m}^t, \mathbf{\Sigma}_{\mathbf{S}_m}^t) \ , \tag{A.3}$$

where we use a mapping function $\phi : \{1 \dots H\}^N \to \{1 \dots M\}$ from each N-tuple in the Cartesian product of the h indices to a single m and we define $w_m = w_{1h_1} w_{2h_2} \dots w_{3h_3}$. μ^t_{ih} and $\mathbf{\Sigma}^t_{ih}$ are the mean and the covariance matrix for the h^{th} component of the mixture for subject i at time t, and $\mu^t_{\mathbf{S}_m}$ and $\mathbf{\Sigma}^t_{\mathbf{S}_m}$ are the mean and the covariance matrix for the joint state $\mathbf{S}^t$. Note that $\boldsymbol{\mu}^t_{\mathbf{S}_m}$ is obtained by simply concatenating the $\mu^t_{ih_i}$ for each subject, and the covariance matrix $\mathbf{\Sigma}^t_{\mathbf{S}_m}$ is a block diagonal matrix with blocks $\mathbf{\Sigma}^t_{ih_i}$. Also, note that the ϕ mapping describes the possible world models for the set of subjects, as in each component of the mixture in Eq. A.3 there is a single component h_i selected from the mixture $p(\mathbf{p}^t_i, \mathbf{v}^t_i)$.

Now let us see how to derive a single marginal $p(\mathbf{p}^t_i, \mathbf{v}^t_i)$:

$$p(\mathbf{p}^t_i, \mathbf{v}^t_i) = \int p(\mathbf{p}^t_i, \mathbf{v}^t_i | \mathbf{S}^{t-1}) p(\mathbf{S}^{t-1}) d\mathbf{S}^{t-1} \tag{A.4}$$

to meet real-time requirements, here we simplify the mixture of Gaussians in Eq. A.3 by substituting each normal distribution with a of Dirac function:

$$p(\mathbf{S}^{t-1}) \approx \sum_{m=1}^{M^{t-1}} w_m \delta(\mathbf{S}^{t-1}_m - \mu^{t-1}_{\mathbf{S}_m}) \ , \tag{A.5}$$

so that we can rewrite the integral in Eq.A.4 as

$$p(\mathbf{p}^t_i, \mathbf{v}^t_i) = \sum_{m=1}^{M^{t-1}} w_m p(\mathbf{p}^t_i, \mathbf{v}^t_i | \mu^{t-1}_{\mathbf{S}_m}) \ . \tag{A.6}$$

By this approximation, we retain the mean and weight of the mixture components, but we discard the covariances. We will compensate for this in the empirical covariance that we will introduce later on. Continuing with the derivation we have

$$p(\mathbf{p}_i^t, \mathbf{v}_i^t) = \sum_{m=1}^{M^{t-1}} w_m p(\mathbf{p}_i^t | \mathbf{v}_i^t, \mu_{\mathbf{S}_m}^{t-1}) p(\mathbf{v}_i^t | \mu_{\mathbf{S}_m}^{t-1}) \tag{A.7}$$

$$= \sum_{m=1}^{M^{t-1}} w_m \mathcal{N}(\mathbf{p}_i^t; \mu_{\mathbf{p}_{im}}^{t-1} + \boldsymbol{\Delta}\boldsymbol{v}_i^t, \Gamma) \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{v}_i^t; \tilde{\mathbf{v}}_{imk}^t, \tilde{\Psi}_{imk}^t) \tag{A.8}$$

$$= \sum_{m=1}^{M^{t-1}} \sum_{k=1}^{K} w_m w_k \mathcal{N}(\mathbf{p}_i^t; \mu_{\mathbf{p}_{im}}^{t-1} + \boldsymbol{\Delta}\boldsymbol{v}_i^t, \Gamma) \mathcal{N}(\mathbf{v}_i^t; \tilde{\mathbf{v}}_{imk}^t, \tilde{\Psi}_{imk}^t) \tag{A.9}$$

$$= \sum_{m=1}^{M^{t-1}} \sum_{k=1}^{K} w_{mk} \mathcal{N}(\mathbf{p}_i^t, \mathbf{v}_i^t; \mu_{imk}^t, \mathbf{\Sigma}_{imk}^t), \tag{A.10}$$

where

$$\mu_{imk}^t = \begin{bmatrix} \mu_{\mathbf{p}_{im}}^{t-1} + \boldsymbol{\Delta}\tilde{\mathbf{v}}_{imk}^t \\ \tilde{\mathbf{v}}_{imk}^t \end{bmatrix}, \tag{A.11}$$

$$\mathbf{\Sigma}_{imk}^t = \begin{bmatrix} \Gamma + \Delta^2 \tilde{\Psi}_{imk}^t & \Delta \tilde{\Psi}_{imk}^t \\ (\Delta \tilde{\Psi}_{imk}^t)^T & \tilde{\Psi}_{imk}^t \end{bmatrix} \tag{A.12}$$

and where we define $w_{mk} = w_m w_k$. So we show that $p(\mathbf{p}_i^t, \mathbf{v}_i^t)$ has a mixture of Gaussian form with $H^t = M^{t-1}K$ components. As we noted above, because of the approximation in Eq. A.5, we are discarding the uncertainty information included in the covariance matrix $\mathbf{\Sigma}_{\mathbf{S}}^{t-1}$. We can partly compensate for this by appropriately modifying the covariance $\mathbf{\Sigma}_{mk}^t$. In particular we modify Eq. A.12 by adding the position covariance at the previous time step as

$$\mathbf{\Sigma}_{imk}^t = \begin{bmatrix} \Gamma + \Delta^2 \tilde{\Psi}_{imk}^t + \mathbf{\Sigma}_{imk}^{t-1} & \Delta \tilde{\Psi}_{imk}^t \\ (\Delta \tilde{\Psi}_{imk}^t)^T & \tilde{\Psi}_{imk}^t \end{bmatrix}. \tag{A.13}$$

Note that the velocity components of the covariance are in fact discarded, but we partially account for this in the empirical setting of $\mathbf{\Gamma}$.

Finally, let us derive $p(\mathbf{p}_i^t)$ as

$$p(\mathbf{p}_i^t) = \int p(\mathbf{p}_i^t, \mathbf{v}_i^t) d\mathbf{v}_i^t \tag{A.14}$$

$$= \sum_{m=1}^{M} \sum_{k=1}^{K} w_{mk} \mathcal{N}(\mathbf{p}_i^t; \mu_{\mathbf{p}_{im}}^{t-1} + \boldsymbol{\Delta} \tilde{\mathbf{v}}_{imk}^t, \Gamma + \Delta^2 \tilde{\Psi}_{imk}^t + \boldsymbol{\Sigma}_{imk}^{t-1}), \tag{A.15}$$

where we made use of the marginalization property of the Gaussian distribution.

Bibliography

[Ali and Shah, 2008] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *European Conference on Computer Vision (ECCV)*, pages 1–14, 2008. 4.2

[Andriluka *et al.*, 2008] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 4.2, 8.2

[Antonini *et al.*, 2006] G. Antonini, S. V. Martinez, M. Bierlaire, and J. Thiran. Behavioral priors for detection and tracking of pedestrians in video sequences. *International Journal of Computer Vision (IJCV)*, 69:159–180, 2006. 2.1, 4.2

[Arnaud *et al.*, 2005] M. B. Arnaud, A. Doucet, and S. S. Singh. Sequential auxiliary particle belief propagation. In *International Conference on Information Fusion*, pages 705–711, 2005. 7.3.2

[Arulampalam *et al.*, 2002] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002. 5.2.1

[Barros *et al.*, 2004] L. M. Barros, A. T. da Silva, and S. R. Musse. Petrosim: An architecture to manage virtual crowds in panic situations. In *Computer Animations and Social Agents (CASA)*, pages 111–120, 2004. 3.2

[Beaulieu, 2004] C. Beaulieu. Intercultural study of personal space: A case study. *Journal of Applied Social Psychology*, 34(4):794–805, April 2004. 2.1

[Bernardin and Stiefelhagen, 2008] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Jounal of Image and Video Processing*, 2008:1:1–1:10, January 2008. 4.5.3, 7.4.5, 7.2

[Bishop, 2006] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1 edition, 2006. 7.2

[Black and Rangarajan, 1994] M. J. Black and A. Rangarajan. The outlier process: Unifying line processes and robust statistics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15–22, 1994. 7.2.4

[Bose *et al.*, 2007] B. Bose, X. Wang, and E. Grimson. Multi-class object tracking algorithm that handles fragmentation and grouping. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1 –8, 2007. 4.2

[Braun *et al.*, 2005] A. Braun, B. Bodman, and S. R. Musse. Simulating virtual crowds in emergency situations. In *ACM Symposium on Virtual Reality Software and Technology (VRST)*, pages 244–252, 2005. 3.2

[Breitenstein *et al.*, 2011] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33:1820–1833, 2011. 4.2, 7.3.4

[Brendel *et al.*, 2011] W. Brendel, M. R. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1273–1280, 2011. 4.2

[Brogan and Johnson, 2003] D. Brogan and N. Johnson. Realistic human walking paths. In *Computer Animations and Social Agents (CASA)*, pages 94 – 101, 2003. 2.1

[Cappe *et al.*, 2007] O. Cappe, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007. 4.2

[Chang and Lin, 2001] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 6.2

[Chang *et al.*, 2011] M.-C. Chang, N. Krahnstoever, and W. Ge. Probabilistic group-level motion analysis and scenario recognition. In *International Conference on Computer Vision (ICCV)*, pages 747–754, 2011. 4.2, 7.4.3

[Choi and Savarese, 2010] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *European Conference on Computer Vision (ECCV)*, pages 553–567, 2010. 4.2

[Cox and Hingorani, 1996] I. J. Cox and S. L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(2):138–150, 1996. 4.2

[Cox, 1993] I. J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision (IJCV)*, 10:53–66, 1993. 4.2

[Cristani *et al.*, 2011] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. Del Bue, G. Menegez, and V. Murino. Social interaction discovery by statistical analysis of F-formations. In *British Machine Vision Conference (BMVC)*, pages 1–12, 2011. 4.2

[Dalal and Triggs, 2005] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. 4.2, 4.5.3

[Dollár *et al.*, 2011] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 99(PrePrints), 2011. 4.2

[Eisenstein, 2007] J. Eisenstein. Dirichlet process mixture models in matlab. http://www.cc.gatech.edu/ jeisenst/software.html, 2007. 3.3.3

[Ess *et al.*, 2008] A. Ess, B. Leibe, K. Schindler, and L. V. Gool. A mobile vision system for robust multi-person tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 4.2, 4.5.3

[Felzenszwalb *et al.*, 2008] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model.

In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 4.2

[Fiorini and Shiller, 1998] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research (IJRR)*, 17(7):760–772, 1998. 2.1

[Fortmann *et al.*, 1983] T. E. Fortmann, Y. Bar Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, July 1983. 4.2

[Freedman, 1975] J. L. Freedman. *Crowding and behavior*. Viking Press, 1975. 2.1, 5.1

[French, 2006] A. French. *Visual Tracking: From An Individual To Groups Of Animals*. PhD thesis, University of Nottingham, 2006. 4.2

[Galata *et al.*, 2002] A. Galata, A. Cohn, D. Magee, and D. Hogg. Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models. In *European Conference on Artificial Intelligence (ECAI)*, pages 741–745, 2002. 4.2

[Gall and Lempitsky, 2009] J. Gall and V. Lempitsky. Class-specic hough forests for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1022–1029, 2009. 4.2, 6.6.1, 7.2.3, 7.4.1

[Ge *et al.*, 2009] W. Ge, R. Collins, and B. Ruback. Automatically detecting the small group structure of a crowd. In *Workshop on Applications of Computer Vision (WACV)*, pages 1–8, dec. 2009. 4.2

[Gennari and Hager, 2004] G. Gennari and G. D. Hager. Probabilistic data association methods in visual tracking of groups. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 876–881, 2004. 4.2

[Gérin-Lajoie *et al.*, 2005] M. Gérin-Lajoie, C. L. Richards, and B. J. McFadyen. The negotiation of stationary and moving obstructions during walking: anticipatory locomotor adaptations and preservation of personal space. *Motor control*, 9(3):242–69, July 2005. 2.1, 2.2.3, 3.1, 8.1

[Goffman, 1971] E. Goffman. *Relations in public: microstudies of the public order*. New York: Basic Books, 1971. 2.1, 5.1

[Grabner and Bischof, 2006] H. Grabner and H. Bischof. On-line boosting and vision. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 260–267, 2006. 5.3.2

[Grabner *et al.*, 2010] H. Grabner, J. Matas, L. V. Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1285–1292, 2010. 4.2

[Guy *et al.*, 2010] S. J. Guy, M. C. Lin, and D. Manocha. Modeling collision avoidance behavior for virtual humans. In *Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 575–582, 2010. 2.1

[Hall, 1966] E. T. Hall. *The Hidden Dimension*. Garden City, 1966. 2.1, 5.1

[Hediger, 1955] H. Hediger. *The Psychology and Behavior of Animals in Zoos and Circuses*. Dover Publications, 1955. 2.1

[Helbing and Molnár, 1995] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review*, 51(5):4282–4286, 1995. 2.1, 3.2, 8.2

[Henderson, 1971] L. F. Henderson. The statistics of crowd fluids. *Nature*, 229(5284):381–3, February 1971. 2.1

[Hinton, 2000] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:2002, 2000. 7.5

[Hoogendoorn, 2003] S. P. Hoogendoorn. Pedestrian travel behavior modeling. In *International Conference On Travel Behavior Research (IATBR)*, pages 507–535, 2003. 2.1

[Huang *et al.*, 2008] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *European Conference on Computer Vision (ECCV)*, 2008. 4.2, 4.6

[Hughes, 2003] R. L. Hughes. The flow of human crowds. *Annual Review of Fluid Mechanics*, 35(1):169–182, January 2003. 3.2

[Ihler and McAllester, 2009] A. Ihler and D. McAllester. Particle belief propagation. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 256–263, 2009. 7.3.1, 7.3.2, 7.3.2, 7.3.2

[Isard and Blake, 1998] M. Isard and A. Blake. CONDENSATION–conditional density propagation for visual tracking. *International Journal of Computer Vision (IJCV)*, 29:5–28, 1998. 4.2

[Johansson *et al.*, 2007] A. Johansson, D. Helbing, and P. K. Shukla. Specification of a microscopic pedestrian model by evolutionary adjustment to video tracking data. *Advances in Complex Systems*, 10(2):271–288, 2007. 2.1, 2.3, 4.4, 4.5.1, 4.4

[Kapadia *et al.*, 2009] M. Kapadia, S. Singh, W. Hewlett, and P. Faloutsos. Egocentric affordance fields in pedestrian steering. In *Symposium on Interactive 3D graphics and games (I3D)*, pages 215–223, 2009. 2.1

[Kaucic *et al.*, 2005] R. Kaucic, A. G. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 990–997, 2005. 4.2, 4.6

[Khan *et al.*, 2005] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(11):1805–1819, 2005. 4.2

[Komodakis *et al.*, 2007] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *International Conference on Computer Vision (ICCV)*, 2007. 6.5

[Kuo *et al.*, 2010] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 685 –692, june 2010. 4.2

[Kwon and Lee, 2011] J. Kwon and M. K. Lee. Tracking by sampling trackers. In *International Conference on Computer Vision (ICCV)*, pages 1195–1202, 2011. 4.2

[Lafferty *et al.*, 2001] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, pages 282–289, 2001. 7.2

[Lau *et al.*, 2010] B. Lau, K. Arras, and W. Burgard. Multi-model hypothesis group tracking and group size estimation. *International Journal of Social Robotics*, 2:19–30, 2010. 4.2

[Leal-Taixé *et al.*, 2011] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: modeling social and grouping behavior on a linear programming multiple people tracker. In *Workshop on Modeling, Simulation and Visual Analysis of Large Crowds*, pages 120–127, 2011. 4.2, 7.4.5

[Lee *et al.*, 2007] K. H. Lee, M. G. Choi, Q. Hong, and J. Lee. Group behavior from video: a data-driven approach to crowd simulation. In *Symposium on Computer Animation (SCA)*, pages 109–118, 2007. 3.2

[Leibe *et al.*, 2008] B. Leibe, K. Schindler, N. Cornelis, and L. 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, 2008. 4.2

[Lerner *et al.*, 2007] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum (Eurographics)*, pages 655–664, 2007. 3.2, 3.5.2, 4.5.1, 5.3, 6.4.1, 7.4.1

[Li *et al.*, 2009] Y. Li, C. Huang, and R. Nevatia. Learning to associate: HybridBoosted multi-target tracker for crowded scene. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2953–2960, 2009. 4.2

[Loscos *et al.*, 2003] C. Loscos, D. Marchal, and A. Meyer. Intuitive crowd behaviour in dense urban environments using local laws. In *Theory and Practice of Computer Graphics (TPCG)*, pages 122–129, 2003. 3.2

[Luber *et al.*, 2010] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras. People tracking with human motion predictions from social forces. In *International Conference on Robotics and Automation (ICRA)*, pages 464–469. IEEE, 2010. 4.2

[Mann *et al.*, 2002] R. Mann, A. D. Jepson, and T. El-Maraghi. Trajectory segmentation using dynamic programming. In *International Conference on Pattern Recognition (ICPR)*, pages 331–334, 2002. 3.3.1

[McPhail and Wohlstein, 1982] C. McPhail and R. T. Wohlstein. Using film to analyze pedestrian behavior. *Sociological Methods Research*, 10(3):347–375, 1982. 2.1, 5.1, 7.2.4

[Mehran *et al.*, 2009] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using Social Force model. In *Conference on*

Computer Vision and Pattern Recognition (CVPR), pages 935–942, 2009. 2.1

[Mooij and al., 2010] J. M. Mooij and al. libDAI 0.2.5: A free/open source C++ library for Discrete Approximate Inference. http://www.libdai.org/, 2010. 6.5

[Morefield, 1977] C. L. Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, pages 302–312, 1977. 4.2

[Moussaïd *et al.*, 2010] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one*, 5(4):e10047, January 2010. 2.1, 8.1

[Murphy *et al.*, 1999] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence (UAI)*, pages 467–475, 1999. 7.3.1

[Musse and Thalmann, 2001] S. R. Musse and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7:152–164, 2001. 3.2

[Nocedal and Wright, 2000] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2000. 2

[Okuma *et al.*, 2004] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision (ECCV)*, pages 28–39, 2004. 4.2, 4.5.3

[Oliver *et al.*, 2000] N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):831–843, 2000. 4.2

[Olivier *et al.*, 2010] A.-H. Olivier, J. Ondřej, J. Pettré, R. Kulpa, and A. Crétual. Interaction between real and virtual humans during walking: perceptual evluation of a simple device. In *Symposium on Applied Perception in Graphics and Visualization (APGV)*, pages 117–124, 2010. 3.2

[Ondřej *et al.*, 2010] J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian. A synthetic-vision based steering approach for crowd simulation. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 123:1–123:9, 2010. 2.1, 3.5.1, 3.1, 3.5.1, 3.7

[Paris *et al.*, 2007] S. Paris, J. Pettré, and S. Donikian. Pedestrian reactive navigation for crowd simulation: a predictive approach. In *Compututer Graphics Forum (Eurographics)*, pages 665–674, 2007. 2.1

[Patterson *et al.*, 2007] M. Patterson, Y. Iizuka, M. Tubbs, J. Ansel, M. Tsutsumi, and J. Anson. Passing encounters east and west: Comparing japanese and american pedestrian interactions. *Journal of Nonverbal Behavior*, 31:155–166, 2007. 8.1

[Pelechano *et al.*, 2008] N. Pelechano, J. A. Norman, and Badler. *Virtual Crowds: Methods, Simulation, and Control (Synthesis Lectures on Computer Graphics and Animation)*. Morgan and Claypool Publishers, October 2008. 2.1, 3.2

[Pellegrini and Gool, 2012] S. Pellegrini and L. V. Gool. Tracking with a mixed continuous-discrete conditional random field. *Computer Vision and Image Understanding (CVIU)*, 2012. In press. 4.2

[Pellegrini *et al.*, 2008] S. Pellegrini, K. Schindler, , and D. Nardi. A generalization of the icp algorithm for articulated bodies. In *British Machine Vision Conference (BMVC)*, 2008. 8.2

[Pellegrini *et al.*, 2009] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *International Conference on Computer Vision (ICCV)*, pages 261–268, 2009. 2.1, 2.2, 4.2, 1, 6.2, 7.4.1

[Pellegrini *et al.*, 2010] S. Pellegrini, A. Ess, and L. V. Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European Conference on Computer Vision (ECCV)*, pages 452–465, 2010. 4.2, 7.4.5, 7.4.5, 1

[Pellegrini *et al.*, 2012] S. Pellegrini, J. Gall, L. Sigal, and L. V. Gool. Destination flow for crowd simulation. In *Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams (ARTEMIS)*, pages 162–171, 2012. 2

[Penn and Turner, 2002] A. Penn and A. Turner. Space syntax based agent simulation. In *Pedestrian and Evacuation Dynamics (PED)*, pages 99–114, 2002. 3.2

[Pettré *et al.*, 2009] J. Pettré, J. Ondřej, A.-H. Olivier, A. Cretual, and S. Donikian. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Symposium on Computer Animation (SCA)*, pages 189–198, 2009. 2.1, 2.2

[Pirsiavash *et al.*, 2011] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1201–1208, 2011. 4.2

[Proust, 1919] M. Proust. *À l'ombre des jeunes filles en fleurs*. Grasset and Gallimard, Translated from the French by C. K. Scott Moncrieff, 1919. 1

[Rasmussen, 2000] C. E. Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 554–560, 2000. 3.3.3

[Reid, 1979] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979. 4.2, 5.2

[Reynolds, 1987] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 25–34, 1987. 2.1

[Reynolds, 1999] C. Reynolds. Steering behaviors for autonomous characters. In *Game Developers Conference*, 1999. 1, 2, 2.1

[Rodriguez *et al.*, 2009] M. Rodriguez, S. Ali, and T. Kanade. Tracking in unstructured crowded scenes. In *International Conference on Computer Vision (ICCV)*, pages 1389–1396, 2009. 4.2

[Rodriguez *et al.*, 2011] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert. Data-driven crowd analysis in videos. In *International Conference on Computer Vision (ICCV)*, pages 1235–1242, 2011. 4.2

[Ryoo and Aggarwal, 2011] M. Ryoo and J. Aggarwal. Stochastic representation and recognition of high-level group activities. *International Journal of Computer Vision (IJCV)*, 93:183–200, 2011. 4.2

[Saffari *et al.*, 2010] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof. Online multi-class LP-boost. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3570–3577, 2010. 7.2.3

[Santner *et al.*, 2010] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 723–730, 2010. 4.2

[Schadschneider, 2001] A. Schadschneider. Cellular automaton approach to pedestrian dynamics-theory. *arXiv preprint cond-mat/0112117*, 2001. 2.1

[Scovanner and Tappen, 2009] P. Scovanner and M. Tappen. Learning pedestrian dynamics from the real world. In *International Conference on Computer Vision (ICCV)*, pages 381–388, 2009. 2.1, 4.2

[Shao and Terzopoulos, 2005] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *Symposium on Computer Animation (SCA)*, pages 19–28, 2005. 3.2

[Sittler, 1964] R. W. Sittler. An optimal data association problem in surveillance theory. *IEEE Transactions on Military Electronics*, 8(2):125–139, 1964. 4.2

[Song *et al.*, 2010] B. Song, T.-Y. Jeng, E. Staudt, and A. K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *European Conference on Computer Vision (ECCV)*, pages 605–619, 2010. 4.2

[Stalder *et al.*, 2010] S. Stalder, H. Grabner, , and L. V. Gool. Cascaded confidence filtering for improved tracking-by-detection. In *European Conference on Computer Vision (ECCV)*, pages 369–382, 2010. 4.2

[Stein *et al.*, 2007] A. N. Stein, D. Hoiem, and M. Hebert. Learning to find object boundaries using motion cues. In *International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 7.2.4

[Sudderth *et al.*, 2003] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 605–612, 2003. 7.3.1

[Sutton and McCallum, 2005] C. Sutton and A. McCallum. Piecewise training of undirected models. In *Uncertainty in Artificial Intelligence (UAI)*, pages 568–575, 2005. 6.4

[Thalmann and Musse, 2007] D. Thalmann and S. R. Musse. *Crowd Simulation*. Springer, 2007. 2.1, 3.2

[Torresani *et al.*, 2008] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *European Conference on Computer Vision (ECCV)*, pages 596–609, 2008. 6.5

[Trautman and Krause, 2010] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Conference on Intelligent Robots and Systems (IROS)*, pages 797–803, 2010. 2.1, 4.2

[Treuille *et al.*, 2006] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. *ACM Transactions on Graphics*, 25(3):1160–1168, July 2006. 3.2

[Tu and Terzopoulos, 1994] X. Tu and D. Terzopoulos. Artificial fishes: physics, locomotion, perception, behavior. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 43–50, 1994. 3.2

[Uenoyama and Miyata, 2005] A. Uenoyama and M. Miyata. Gliding ghosts of mycoplasma mobile. *Proceedings of the National Academy of Sciences of the United States of America*, 102(36):12754–8, September 2005. 7.4.1

[van den Berg *et al.*, 2008] J. P. van den Berg, M. C. Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *International Conference on Robotics and Automation (ICRA)*, pages 1928–1935, 2008. 2.1

[van den Berg *et al.*, 2009] J. van den Berg, S. J. Guy, M. C. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *International Symposium On Robotics Research (ISRR)*, pages 3–19, 2009. 2.1

[Vasquez *et al.*, 2008] D. Vasquez, T. Fraichard, O. Aycard, and C. Laugier. Intentional motion on-line learning and prediction. *Machine Vision Applications*, 19:411–425, September 2008. 4.2

[Wang *et al.*, 2009] C. Wang, M. de La Gorce, and N. Paragios. Segmentation, ordering and multi-object tracking using graphical models. In *International Conference on Computer Vision (ICCV)*, pages 747–754, 2009. 4.2

[Welch and Bishop, 1995] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995. 4.2

[Wu and Nevatia, 2007] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet part detectors. *International Journal of Computer Vision (IJCV)*, 75(2):247–266, Nov. 2007. 4.2, 4.5.3

[Yamaguchi *et al.*, 2011] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1345–1352, 2011. 4.2

[Yang *et al.*, 2011] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a crf model. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1233–1240, 2011. 4.2

[Yilmaz *et al.*, 2006] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Computing Surveys*, 38, December 2006. 4.2

[Zhang *et al.*, 2008] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 4.2

[Zhang *et al.*, 2011] Y. Zhang, J. Pettré, J. Ondřej, X. Qin, Q. Peng, and S. Donikian. Online inserting virtual characters into dynamic videoscenes. *Computer Animation and Virtual Worlds*, 22(6):499–510, 2011. 3.2

[Zhao *et al.*, 2009] L. Zhao, A. Normoyle, S. Khanna, and A. Safonova. Automatic construction of a minimum size motion graph. In *Symposium on Computer Animation (SCA)*, pages 27–35, 2009. 3.5.2

[Ziebart *et al.*, 2009] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Conference on Intelligent Robots and Systems (IROS)*, pages 3931–3936, 2009. 4.2

Curriculum Vitae

Name	Stefano Pellegrini
Date of Birth	17.12.1981
Place of Birth	Soriano nel Cimino, Italy
Citizenship	Italy

Education

2008 – 2012	**Doctor of Science ETH** *ETH Zurich, Computer Vision Laboratory*
2003 – 2005	**M.S. Degree in computer Engineering** *Sapienza University Rome, Dept. of Computer and System Sciences*
2000 – 2003	**B.S. Degree in computer Engineering** *Sapienza University Rome, Dept. of Computer and System Sciences*

Occupations

2011 – 2011	**Intern** *Google, Zurich*
2008 – 2011	**Research Assistant** *ETH Zurich, Computer Vision Laboratory*
2006 – 2006	**Visiting Fellow** *NIH, Bethesda MD, United States*