DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 787

Tobias Klinger

Probabilistic multi-person localisation and tracking

München 2016

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5199-7

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover
ISSN 0174-1454, Nr. 329, Hannover 2016

DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 787

Probabilistic multi-person localisation and tracking

Von der Fakultät für Bauingenieurwesen und Geodäsie
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

von

Dipl.-Ing. Tobias Klinger

München 2016

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission bei der C. H. Beck'schen Verlagsbuchhandlung München

ISSN 0065-5325 ISBN 978-3-7696-5199-7

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover
ISSN 0174-1454, Nr. 329, Hannover 2016

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 – 89 – 23 031 1113 ● Telefax +49 – 89 – 23 031 - 1283 / - 1100

e-mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:

Vorsitzender:	Prof. Dr.-Ing. Ingo Neumann
Referent:	Prof. Dr.-Ing. Christian Heipke
Korreferenten:	Prof. Dr.-Ing. Steffen Schön
	Prof. Dr.-Ing. Stefan Hinz
	Prof. Dr. techn. Franz Rottensteiner

Tag der mündlichen Prüfung: 02.12.2016

ISSN 0065-5325 ISBN 978-3-7696-5199-7

Erklärung

Ich erkläre, dass ich die vorliegende Dissertation selbständig verfasst habe, die benutzten Hilfsmittel vollständig angegeben sind und die Dissertation nicht als Diplomarbeit, Masterarbeit oder andere Prüfungsarbeit verwendet wurde. Weiterhin erkläre ich, dass ich keine anderen Promotionsgesuche eingereicht habe.

Hannover, 04. Oktober 2016

Statement

I state that this dissertation has been written entirely by myself. No further sources besides the ones noted in the bibliography have been used and this dissertation has not been submitted as Diploma thesis, Master thesis or any other written examination. Furthermore I state that I have not applied for any other conferral of a doctorate.

Hanover, October 4th, 2016

Abstract

This dissertation investigates the problem of localising multiple persons in image sequences, while, at the same time, establishing temporal correspondences between single-frame locations. The aim of this work is the improvement of the reliability and precision of the generated trajectories, which is addressed by the formulation and investigation of a joint probabilistic model for the recursive filtering of the estimated positions. The trajectories are estimated in a common 3D object coordinate system, which was previously almost exclusively done in 2D.

Four principle scientific contributions are made in this work. Firstly, the location of persons in single images, which is widely measured by independent single frame person detectors, is no longer treated as observable, but is improved together with the 3D state parameters of each pedestrian. In this way, the update step of the recursive filter is performed using an improved image position of the persons, which, in turn, receives feedback from the optimised solution of the state variables and, thus, generates more accurate samples for the representation of the target appearance. Secondly, the multitude of pedestrians in the approached scenarios is accounted for by the introduction of a new generative model of the pedestrian dynamics. This new model takes account of the motion of every pedestrian in a probabilistic framework and automatically detects interactions among the persons, which assist at the prediction of future states and thereby improve the accuracy of the generated trajectories. Thirdly, a new strategy for the modelling of the appearance used for the localisation and recognition of persons in successive frames is developed on the basis of a classification strategy that is fed with samples that automatically derive from tracking. Finally, the assignment of image-based observations from a generic person detector to existing trajectories is formulated as an optimisation problem solved at each time step, using a new model for the similarity measures based on the new model of appearance and the improved predictive model.

The new method is applied to image sequences from complex real-world scenarios in the context of different applications. Experimental results show that the method improves the state-of-the-art in multi-person localisation and tracking in the context of visual surveillance, while the results lag behind those of the related work in the context of autonomous driving. The proposed method and the insights achieved by the experiments motivate future research in the direction of probabilistic modelling for multi-person localisation and tracking.

Keywords: Detection, classification, localisation, tracking, pedestrians, context, 3D, Dynamic Bayesian Networks, Gaussian Process Regression, Linear Programming

Zusammenfassung

Die vorliegende Dissertation erforscht das Problem der Positionsbestimmung von Fußgängern in Bildsequenzen, unter Berücksichtigung zeitlicher Korrespondenzen zwischen einzelnen Positionen. Das Ziel dieser Arbeit ist die Verbesserung der Zuverlässigkeit und der geometrischen Genauigkeit der erzeugten Trajektorien gegenüber dem Stand der Forschung. Dieses Ziel wird durch die Entwicklung eines neuartigen probabilistischen graphischen Modells zur rekursiven Zustandsbestimmung der Fußgänger angegangen. Während bisherige Arbeiten fast ausschließlich die Trajektorien im Bildraum bestimmen, setzt das hier vorgestellte Verfahren auf die Auswertung im 3D Objektraum.

Vier wesentliche Beiträge werden in dieser Arbeit zur Erreichung des Forschungsziels geleistet. Erstens wird die Bildposition von Fußgängern, die gemeinhin als Beobachtung modelliert wird, zusammen mit den Zustandsparametern in einem gemeinsamen Modell bestimmt. Auf diese Weise erfolgt die Korrektur des rekursiven Filters mit einer verbesserten Bildposition, welche wiederum Rückschlüsse aus dem verbesserten Systemzustand zieht. Daraus lässt sich ein zuverlässiges Modell für das Aussehen der Personen erlernen. Zweitens wird der Vielzahl an Personen in einer Szene dadurch Rechnung getragen, dass ein gemeinsames Bewegungsmodell für alle Personen entwickelt wird. Dieses Modell erkennt mögliche Interaktionen zwischen Personen zur Laufzeit automatisch und berücksichtigt diese bei der Prädiktion von neuen Systemzuständen. Drittens wird eine neue Strategie zum inkrementellen Erlernen personenspezifischen Aussehens auf Grundlage eines Klassifikators vorgestellt, dessen Trainingsbeispiele automatisch aus den Trackingergebnissen abgeleitet werden. Zuletzt wird das Zuordnungsproblem zwischen Detektionen und Trajektorien durch ein neues Modell für die Berechnung von Zuordnungskosten, basierend auf dem verfeinerten Erscheinungs- und Bewegungsmodell, unterstützt.

Die Methode wird anhand realer Bildsequenzen, die für unterschiedliche Anwendungszwecke charakteristisch sind, evaluiert. Experimentelle Ergebnisse zeigen, dass der Stand der Forschung in Hinblick auf die Lokalisierung und Verfolgung von Personen im Kontext der Videoüberwachung durch die neue Methode übertroffen wird. Tests mit Datensätzen aus dem Bereich des autonomen Fahrens zeigen hingegen, dass die erzielten Ergebnisse denen aus dem Stand der Forschung unterlegen sind. Die Erkenntnisse, die aus den experimentellen Ergebnissen gezogen werden, motivieren die Weiterentwicklung dieses Verfahrens in Hinblick auf die Anwendbarkeit im Kontext des autonomen Fahrens.

Schlagworte: Detektion, Klassifikation, Lokalisierung, Verfolgung, Fußgänger, Kontext, 3D, Dynamische Bayes-Netze, Gauß-Prozesse, Lineare Programmierung

Notation and symbols

General notation

a, b, α, β, x, y	Scalars
$\mathbf{a}$, $\mathbf{b}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\mathbf{x}$, $\mathbf{y}$	Vectors
A, B, X, Y, Σ	Matrices
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{W}$	Sets
$\mathcal{N}(\cdot)$, $\mathcal{GP}(\cdot)$	Probability distributions

Notational conventions

$p(x)$	Marginal probability of x
$p(x, y)$	Joint probability of x and y
$p(x\|y)$	Conditional probability of x given y
$\mathcal{N}(\cdot)$	Normal distribution
$\mathcal{GP}(\cdot)$	Gaussian Process
μ_x	Mean of x
$\boldsymbol{\mu}_{\mathbf{x}}$	Mean vector of $\mathbf{x}$
σ_x	Standard-deviation of x
$\Sigma_{\mathbf{xx}}$	Covariance matrix of $\mathbf{x}$
$m(\cdot)$	Mean function
$k(\cdot)$	Covariance function
$E(\cdot)$	Expected value
$\hat{x}$	Estimate of x
x^+	Predicted value of x
θ_x	Threshold for the value of x
$\eta_{\mathcal{A}}$	Cardinality of a set $\mathcal{A}$
ρ_p	Coefficient of a parameter p
$[\mathcal{A}]$	Set of indices of a set $\mathcal{A}$
$m_{v_1 \to v_2}$	Message sent from variable v_1 to v_2

Symbols

α_{ij}	Angular displacement between two motion trajectories i and j
a_i^k	Association event of person i with detection k
$c_{i,t}$	Confidence of the classifier about the presence of person i at time t
C_t	Camera orientation at time t

Symbols

$d_{i,t}$	Confidence of the detector about the presence of person i at time t
$\mathbf{d}_k$	Detection with unique index k
ϵ_{nms}	Parameter of the non-maximum suppression
H_i	Height of person i in object space
h_k	Hypothesis about a new tracking candidate based on a detection k
$\mathbf{i}$	System innovation
IP	Interesting places, representing prior knowledge about the scene
i	Index of the entities in a set of persons
j	Index of the entities in a set of persons other than i
K	Kalman gain matrix
$K_{ij,t}$	Covariance of two persons i and j at time t
l	Characteristic length scale
M	Measurement matrix
n_i	Uncertainty about the position of person i w.r.t. $o_{i,t}$ and IP
$o_{i,t}$	Occlusion of person i at time t
π_t	Ground plane
Ψ	Transition matrix
$\mathbf{r}_i$	Rectangle surrounding person i
σ_f^2	Signal variance
σ_n^2	Noise variance
$\mathcal{T}_i$	Trajectory of person i
θ_α	Angular threshold
$\mathbf{v}_i$	Horizontal velocity of person i
w_i^k	Weight of the association event a_i^k
$w(\cdot)$	Angular function
$\mathbf{w}_{i,t}$	State vector of person i at time t
$\mathbf{x}_{i,t}^F$	Image coordinates of the reference point of person i at time t
$\mathbf{x}_{i,t}^H$	Image coordinates of the top-most point of person i at time t
X_i, Y_i, Z_i	World coordinates of the reference point of person i
Y_π	Distance of ground plane from the camera

Table of contents

1 Introduction

1.1 Motivation

In the twenty-first century, where cameras are omnipresent in many areas around the globe, and decades of research in the fields of photogrammetry and computer vision have brought progress in the geometric reconstruction and semantic interpretation of a filmed scene, the analysis of image sequences is one of the most versatile technologies for many applications such as human-machine interaction and autonomous driving. If one considers cameras as the resemblance of one of the human's most important sensory organs, the potential applications of this technology are nearly limitless.

Today's consumer cameras and smartphones are equipped with software for face detection and game consoles aim at the tracking of human bodies to enable the interaction with the filmed persons. The detection and tracking of persons in such situations over time allows for an understanding of their actions. Bringing semantics into the filmed scene also allows for the reduction of the human effort in the inspection of surveillance footage. For autonomous vehicles and robots, interacting with persons in a common space, the detection and tracking of persons from imaging sensors must be addressed with the highest possible reliability and accuracy. Currently, the analysis of image sequences already allows for the localisation and tracking of the scene content on an object level, but the required quality of the generated trajectories is yet far from being achieved. Consequently, the improvement of the reliability and accuracy of motion trajectories, obtained from tracking persons in image sequences, is the envisaged goal of this research.

In the applications previously mentioned, the trajectories often need to be available in real time. To this end, many available systems apply *tracking-by-detection*, i.e., *object detection* in single frames to find an approximate position of persons in a single image, *data association* for linking the detections to trajectories and *recursive filtering* to find a synthesis between image-based measurements, given by the object detection results, and a motion model. This approach is subjected to several challenges in the addressed scenarios. The tracking of individual persons is a challenge in itself, as the person of interest performs articulated movements, so that its appearance in the image changes, which needs to be considered by the detection strategy. The surrounding conditions of illumination and the visible background often change gradually, which must be accounted for by the association strategy. If persons appear in crowds, a disambiguation in the assignment of single measurements to individual targets must be performed. Furthermore, mutual occlusions are inherent in crowded scenes, so that measurements in the image can often not be accomplished. Many approaches available in the tracking literature focus on solving the tasks of a correct assignment of measurements to tracked objects, aiming at a high completeness and correctness of the trajectories. By contrast, only few papers focus on the

geometric accuracy of the generated trajectories. In the context of autonomous car navigation, the aim of the tracking algorithms must be to generate trajectories as complete and accurate as possible, to decide, for instance, whether a pedestrian actually enters a vehicle path or not (Gavrila and Munder, 2007).

Probabilistic models are widely used to guide the estimation of the dynamic state of a person, due to their ability to produce estimates of the desired model variables, accompanied by measures of uncertainty that enable the assessment of these variables. Models such as the recursive Bayes filter perform the estimation in a recursive way, predicting the system state based on previous measurements, and correcting the prediction with new measurements. Due to the sequential processing, errors committed in either of these steps can often not be corrected later and thus pose a severe reliability risk for applications using these models. If the detections are imprecise, the trajectories may be updated towards wrong positions, affecting the accuracy of the estimated state and increasing the risk to take wrong decisions.

As measured by current results on relevant tracking benchmarks, e.g. (Geiger et al., 2012) and (Leal-Taixé et al., 2015), the automatic detection and tracking of persons with the intended quality is far from being solved. This work applies tracking-by-detection by recursive Bayesian estimation and tries to improve the quality of the trajectories of multiple persons by improving the state-of-the-art in temporal modelling of the moving patterns and in the localisation of each individual in the single frames. The new predictive model performs state estimation under consideration of *motion context*, as reflected by the state information about all tracked persons in a common scene. The geometric accuracy of the trajectories is improved by modelling the image position of each object as a hidden variable of the probabilistic model, enabling redundant information about the position to contribute to the correction step of the estimation framework.

1.2 Research objectives and contributions

Against the background of the posed challenges for a tracking system and the desired properties with respect to reliability and accuracy of the generated trajectories, the research objective of this dissertation can be stated as follows: *This work aims at the improvement of the geometric accuracy and reliability of generated trajectories by integration of all available image-based observations and person-specific models of motion and appearance, under consideration of their uncertainties in a recursive estimation framework.*

To achieve this research goal, the following scientific contributions are made within this dissertation:

- A new probabilistic model for the joint estimation of the state vectors of multiple persons in object space and their locations in the image is proposed. Given three different image-based observations, this model allows the automatic detections to be geometrically corrected before they are incorporated into the recursive filter.
- A new model for the representation of the appearance of individual persons is proposed on the basis of online adaptable Random Forest classifiers. The new strategy can model a changing

environment with varying appearance and changing numbers of tracked persons.

- A new temporal model for the predictive function of the recursive filter is developed, which takes into account information about the motion of all pedestrians in the scene. The proposed method determines interactions between multiple persons at runtime and performs the prediction based on this information.
- A new model for the assessment of similarities between single-frame detections and tracked targets is developed that integrates the improved temporal model as well as the instance-specific classification. This model is used to compute similarity measures to solve the data association problem.

1.3 Outline of the dissertation

The remainder of the dissertation is structured as follows. After this introduction, the theoretical foundations of the central building blocks of the proposed method are given in Chapter 2. This includes the fundamentals of probabilistic modelling with the focus on Bayesian Networks, recursive Bayesian estimation, and Gaussian Process Regression. In Chapter 3, the literature related to the topic of this dissertation is reviewed in three sections, with the focus on general tracking approaches, pedestrian detection and localisation in single images and temporal modelling. The new probabilistic approach for multi-person localisation and tracking is introduced in Chapter 4. This chapter is further divided into six sections, related to the structure of the new probabilistic model for the recursive filtering, the observation model and the temporal model of the recursive estimation framework, the data association, the inference procedure used to determine the hidden parameters of the system, and the discussion of the new method. Experimental results are given in Chapter 5. Firstly, the datasets and evaluation metrics are introduced and the free parameters are investigated, then the proposed method is evaluated with respect to the importance of the individual model components, which is followed by the evaluation of the reliability and geometric accuracy of the generated trajectories. After that, the experiments show a comparison of the achieved results with results from related work. In Chapter 6, the results are critically discussed. Finally, Chapter 7 concludes the dissertation and gives an outlook on future work.

2 Basics

This chapter presents the theoretical basis of the methods used to achieve the research goal of this dissertation. First and foremost some fundamentals of probabilistic modelling will be explained in Chapter 2.1, which constitute the basic principles for much of the methodology. Chapter 2.2 describes the generalisation of the probabilistic models to the modelling of temporal phenomena. As part of this chapter, the Kalman Filter is introduced as a special case of the general recursive estimation framework. Finally, Chapter 2.3 presents the principles of Gaussian Process Regression, which is used in this work for the prediction of pedestrian velocities.

2.1 Probabilistic modelling

Probabilistic modelling plays a central role in the fields of geodesy, computer vision and machine learning. In the field of geodesy, probabilistic modelling has been practised at the latest since the publication of the "survey of heavenly bodies" by C.F. Gauss (Gauss, 1809), which accounts for the inherent uncertainties about the correctness of measurements by assigning them probability distributions that are modelled from sets of redundant measurements. This way of dealing with probabilities is related to today's frequentist view on probabilities (Bishop, 2006). In the more general case, however, frequent observations of an event are often not possible to make, but probability distributions for the event may still be given as defined for instance by expert knowledge. Such problems often occur in computer vision or remote sensing applications. Assuming one desires the susceptibility of a geographic area to land slides and one is given information about land cover, surface elevation and meteorological measurements, an expert has a belief in the occurrence of the land slide event, but his belief is uncertain (e.g., due to possible errors in the input information). The Bayesian concept of probabilities integrates arbitrary probability distributions that are related to prior information about an event (e.g., the temporal frequency of previous land slides), the likelihood of the event given observed data (e.g., a forecast for heavy rain) and evidence (i.e., the probability that the forecast is correct). Both, the frequentist and the Bayesian concept of probability, model known or unknown quantities as *random variables*, whose actual outcome depends on chance and cannot be explained causally. Probabilistic modelling provides a framework to express the *belief* and the *uncertainties* about an event and to draw conclusions from the observed data. Subsequently, the basic principles of Bayesian probability theory are presented, followed by the framework of Bayesian networks to handle more complex problems and by an explanation of a common inference strategy.

2.1.1 Bayesian probabilities

Probabilistic modelling essentially enables to draw conclusions about unobserved variables from other variables that are observed. Most probabilistic calculations are based on two fundamental rules of probability, the sum rule of probability, Equation 2.1, and the product rule of probability, Equation 2.2,

$$p(x) = \sum_y p(x, y), \tag{2.1}$$

$$p(x, y) = p(y|x)\, p(x)\,. \tag{2.2}$$

The sum rule states that the marginal probability $p(x)$ of a random variable x can be expressed as the sum over the joint probability $p(x, y)$ for all possible values of another variable y. The product rule states that the joint probability $p(x, y)$ is identical to the product of the conditional probability $p(y|x)$ of y given x and the marginal probability of x. For discrete variables, such probability functions are referred to as probability mass functions, whereas, for continuous variables, they are referred to as probability density functions (pdfs). By application of the product rule and considering the symmetry property $p(x, y) = p(y, x)$, the theorem of Bayes,

$$p(x|y) = \frac{p(y|x)\, p(x)}{p(y)}, \tag{2.3}$$

can be derived. In this context, the term $p(y|x)$ is called *likelihood* and is the probability to observe the variable y if the value of x is known. The probability $p(x|y)$ is referred to as the *posterior* probability of x given y. $p(x)$ is also referred to as the *prior* probability of x and $p(y)$ as the *evidence*, which is often expressed as $p(y) = \sum_x p(y|x)\, p(x)$ in case of discrete variables and in accordance with the sum rule and the product rule.

2.1.2 Bayesian networks

For more complicated probabilistic models, *probabilistic graphical models* (Bishop, 2006) become advantageous for essentially two reasons. Firstly, probabilistic graphical models represent the factorisation of the joint probability of all random variables. Because the factors usually have fewer random variables as arguments than the joint probability, fewer parameters need to be determined, so that the amount of required training data is also reduced. Secondly, for certain graph types, inference can be performed efficiently using standard algorithms.

Generally, a probabilistic graphical model consists of nodes and edges, where the nodes represent random variables or fixed variables and the edges represent relations between the variables. The edges of the graph can be directed, pointing from a parent node to a child node, indicating that the value of the random variable associated to the child node depends on the value of the random variable of the parent node, or undirected. A directed graphical model without cycles is referred to as *Bayesian network*. Bayesian networks can be applied to problems where only causal relationships between variables exist. Random variables can be further categorised as observed or unknown. In accordance with the convention of (Bishop, 2006), observed random variables are drawn as circles shaded in grey,

unknown random variables as blank circles, and fixed variables are drawn as black solid circles in this work. In Figure 2.1 an illustration of Bayesian networks involving two variables x and y is given.

Figure 2.1: Simple Bayesian network: Graphical representation of two random variables. (a) the cause is unknown and the effect is observed; (b) the cause is observed and the effect is unknown; (c) both are unknown; (d) the cause is fixed and the effect is unknown.

For a Bayesian network, the joint probability distribution of the set of K random variables $x_1, x_2, ..., x_k, ..., x_K$ in the network can be *factorised* into a product of conditional probabilities $p(x_k|pa_k)$, where pa_k is the set of random variables associated to all parent nodes of x_k, so that

$$p(x_1, x_2, ..., x_k, ..., x_K) = \prod_k p(x_k|pa_k), \tag{2.4}$$

with $p(x_k|pa_k) = p(x_k)$ if x_k has no parents.

Example. Given the example from the introduction to this section, the problem of predicting the land slide susceptibility can be represented by a Bayesian network as shown in Figure 2.2. Consider the variables *precipitation* P, *slope* S, *land slide susceptibility* L, *land cover class* C, and *image data* I. The graphical model indicates that the image data, as well as the slope and precipitation, are observed, and that the land cover class and the land slide susceptibility are unknown. The directed edges of the graph model the causal relationship between the random variables: the slope of the terrain, the amount of rain, as well as the land cover influence the susceptibility for land slides. The graphical model indicates that the land cover affects the data observed in the image. Further dependencies between the variables are neglected for simplicity. The joint probability of the involved random variables can be factorised in accordance with Equation 2.4 and with the network structure in Figure 2.2:

$$p(P, S, L, C, I) = p(L|P, S, C)\, p(S)\, p(P)\, p(I|C)\, p(C)\,. \tag{2.5}$$

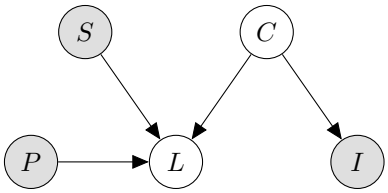


Figure 2.2: Bayesian network representation of the problem of land slide prediction. The variables *precipitation* P, *slope* S and image data I are observed and the *land slide susceptibility* L and *land cover class* C are unknown.

2.1.3 Marginalisation and inference

Previously, probabilistic graphical models were introduced as a tool to express the causal relations between variables. The general aim of probabilistic modelling, however, is to draw conclusions about

unknown variables based on observations. Two tasks in this context are to find the marginal probability of single random variables (*marginalisation*), cf. Equation 2.1, and to find the values of all random variables that maximise the joint probability of the unknowns and the observations (*inference*). If the network can be arranged as a tree, i.e., the graph does not contain loops, these problems can be solved exactly. Both marginalisation and inference can be conducted by so-called *message passing* algorithms. *Belief propagation* (Pearl, 1988) is a message passing algorithm that is guaranteed to find the optimal solution in tree-structured graphs, and can also be used for approximate inference on general graphs. Belief propagation algorithms exploit the way in which a joint probability distribution can be factorised, and perform inference by successive evaluation of local functions associated to the factors, i.e., probability functions, of the joint distribution. A graphical representation of the structure of the factorisation of a joint distribution is provided by *factor graphs* (Kschischang et al., 2001). By definition, a factor graph is a bipartite undirected graphical model that comprises one node for every variable and an additional *factor node* (generally represented by a black square) for every factor in the representation of the joint probability according to Equation 2.4. Every factor node $f_s(\mathbf{x}_s)$ represents a function of the subset $\mathbf{x}_s$ of all variables $\mathbf{x}$, whose members are connected to that node. The joint probability of a set of variables can thus be factorised according to

$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s). \tag{2.6}$$

Both, marginalisation and inference can be conducted by passing messages through the graph twice: firstly, from the leaf nodes to the root node, and then from the root node to the leave nodes. When all incoming messages of a node can be evaluated, the marginal distribution of the associated variable can be computed. When the marginal probabilities of multiple variables are required, the same computations, i.e. the local message passing along the edges of the factor graph, lead to the solution of all marginals, which makes these algorithms efficient also for more complex problems.

A factor graph representation of the example of land slide prediction is given in Figure 2.3. In accordance with the factor graph representation, the joint distribution of the variables in that example can be factorised as

$$p(P, S, L, C, I) = f_1(L, S, P, C)\, f_2(S)\, f_3(P)\, f_4(C, I)\, f_5(C). \tag{2.7}$$

In Figure 2.3 the blue arrows represent messages sent from the leave nodes towards the root node, and the green arrows represent messages sent from the root node towards the leave nodes. The variable that is represented by the root node can be chosen arbitrarily. Formally, the marginal probabilities of the individual variables are computed using the *sum-product* algorithm and the best agreement of all variables with the observations can be computed using the *max-sum* algorithm. A schematic description of these algorithms is given in the following paragraphs.

Sum-product algorithm. Generalising the sum-rule of probability (Equation 2.1) to the modelling of multiple discrete variables, the marginal distribution of a desired variable can be found by summing over all variables that are connected to the required variable. The sum-product algorithm (Kschischang

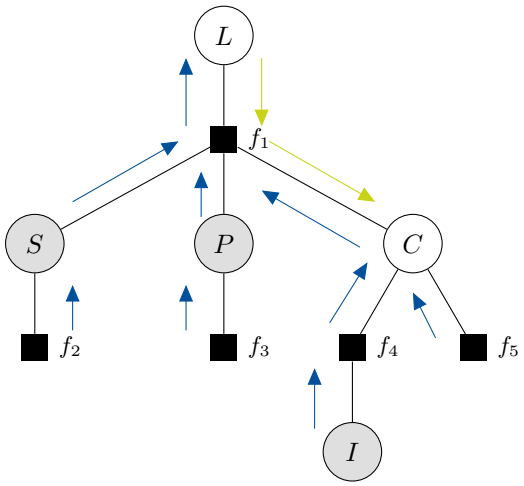


Figure 2.3: Factor graph representation of the Bayesian Network in Figure 2.2. The variable nodes are drawn as circles and the factor nodes as squares. Blue arrows represent messages in the forward recursion and the green arrows represent messages in the backward recursion.

et al., 2001) computes marginal distributions by means of message passing. It takes account of the way in which a joint distribution can be factorised by replacing the summands of the marginalisation step with products of local functions, according to Equation 2.6. Marginalisation takes place recursively by firstly sending messages from the leave nodes towards the root node, and then in the reverse direction, so that every variable is updated based on information about all other variables. Different computation rules for the messages sent from a variable node x to a factor node f_s, denoted by $m_{x\to f_s}(x)$, and for the messages sent from a factor node to a variable node, denoted by $m_{f_s\to x}(x)$, are defined. In the first case, the message sent from a variable node x to a factor node f_s is the product of all messages sent to that variable node from all connected factor nodes, except the factor node the message is sent to:

$$m_{x_m\to f_s}(x_m) = \prod_{f_l\in ne(x_m)\setminus f_s} m_{f_l\to x_m}(x_m), \tag{2.8}$$

where $ne(x_m)$ is the set of neighbouring nodes of x_m. The messages sent from a factor node to a variable node are evaluated as the product of all messages arriving at the factor node, multiplied by the function associated to that factor node, and a marginalisation over all variables associated to the incoming messages (Kschischang et al., 2001):

$$m_{f_s\to x}(x) = \sum_{x_1\ldots x_M} f_s(x, x_1, \ldots, x_M) \prod_{x_m\in ne(f_s)\setminus x} m_{x_m\to f_s}(x_m). \tag{2.9}$$

The outgoing messages from leaf nodes are set to $\mathbf{1}$ in case the leaf node is a variable node and equal the function related to the factor node in case the leaf node is a factor node. After the messages have been passed through every edge in the factor graph the marginal probability of each variable x can be computed as the product of all messages arriving at the variable node:

$$p(x) = \prod_{f_s\in ne(x)} m_{f_s\to x}. \tag{2.10}$$

The message passing steps and the computation of the marginal probabilities (referred to as *belief update* in accordance with Pearl, 1988) of the toy example are given in Table 2.1.

Forward recursion

1: $m_{I \to f_4}(I) = 1$
2: $m_{f_2 \to S}(S) = f_2(S)$
3: $m_{f_3 \to P}(P) = f_3(P)$
4: $m_{f_4 \to C}(C) = m_{I \to f_4}(I)$
5: $m_{f_5 \to C}(C) = f_5(C)$
6: $m_{S \to f_1}(S) = m_{f_2 \to S}(S)$
7: $m_{P \to f_1}(P) = m_{f_3 \to P}(P)$
8: $m_{C \to f_1}(C) = m_{f_4 \to C}(C) \cdot m_{f_5 \to C}(C)$
9: $m_{f_1 \to L}(L) = \sum_C \left(f_1(L, S, P, C) \prod_{i \in \{S,P,C\}} m_{i \to f_1}(i) \right)$

Backward recursion

1: $m_{L \to f_1}(L) = 1$
2: $m_{f_1 \to C}(C) = \sum_L \left(f_1(L, S, P, C) \prod_{i \in \{L,S,P\}} m_{i \to f_1}(i) \right)$

Belief update

$p(L) = m_{f_1 \to L}(L)$
$p(C) = m_{f_1 \to C}(C) \cdot m_{f_4 \to C}(C) \cdot m_{f_5 \to C}(C)$

Table 2.1: Message passing and belief update steps for the sum-product algorithm in the toy example.

Max-sum algorithm. The set of variable values that maximise the joint probability of all variables can be computed using the max-sum algorithm (Bishop, 2006). The procedure is similar to the message update rules 2.8 and 2.9 of the sum-product algorithm, with the difference that the sum in Equation 2.9 is replaced by the *max*-operator. In practical applications, the product of the involved probability functions is typically replaced by the sum of the logarithms of these functions to prevent the system from numerical underflow.

2.2 Recursive Bayesian estimation

This section describes the probabilistic reasoning about random variables that evolve over time. The concept for the modelling of such temporally changing phenomena inherently integrates into the context of Bayesian estimation. The values of the variables of a dynamic system that are related over time are referred to as the system *state*. In the context of this work, the state describes the position and velocity parameters of moving objects, i.e. pedestrians. Inference tasks in that context can be categorised into *prediction*, *filtering* and *smoothing*. Prediction aims at the estimation of future system states given measurements up to the current time step. The task of filtering is to compute the posterior state of a system at a current time given all measurements up to that time step. Smoothing is the task of computing the posterior state at past time steps given all measurements up to the

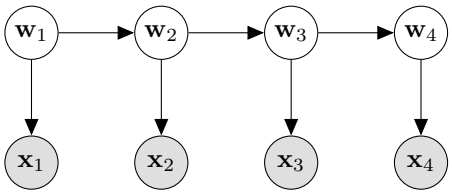


Figure 2.4: Bayesian network representation of the recursive Bayes filter for four time steps $t = 1...4$. $\mathbf{x}_t$ are the measurements and $\mathbf{w}_t$ are the hidden system states.

current date. This work only focusses on prediction and filtering, because these methods are suitable for online applications. As opposed to the models previously described, the prediction and filtering is designed to compute optimal state parameters at every time step, rather than to find the parameters that maximise the common posterior of all time steps. In the context of Bayesian probability theory, prediction and filtering of a system state can be performed in a recursive way, and is, thus, referred to as recursive Bayesian estimation. Formally, recursive Bayesian estimation is a framework for the estimation of state parameters at discrete moments in time from a sequence of noisy measurements. The concept of recursive Bayesian estimation and, based on that concept, the Kalman Filter model and Dynamic Bayesian Networks are described in this section based on the textbook of Prince (2012).

Figure 2.4 shows a directed graphical model representing a dynamic system, where the unknown parameters $\mathbf{w}_{1...4}$ give rise to the measurements $\mathbf{x}_{1...4}$ and are connected over time, with all variables having parent nodes only defined at the same or at the preceding time step. The joint probability of the variables can be written as

$$p(\mathbf{w}_{1...4}, \mathbf{x}_{1...4}) = p(\mathbf{w}_1) \prod_{t=2}^{4} p(\mathbf{w}_t|\mathbf{w}_{t-1}) \prod_{t=1}^{4} p(\mathbf{x}_t|\mathbf{w}_t). \tag{2.11}$$

In this model, inference about the posterior marginals can be performed using the *forward-backward algorithm*, and the most likely sequence of variables can be found using the *Viterbi algorithm* (Rabiner, 1989). In online applications, one is primarily interested in the current system state rather than in a repetitive estimation of the posterior marginals of all variables, including those from past time steps. To this end, the so called *Markov assumption* is used to express that the current state only depends on the preceding state parameter and current observations, i.e. $p(\mathbf{w}_t|\mathbf{w}_{1...t-1}) = p(\mathbf{w}_t|\mathbf{w}_{t-1})$. At every time step t, the posterior distribution of state parameters $\mathbf{w}_t$, given all the data $\mathbf{x}_{1...t}$ observed up to this moment, is desired. At time $t = 1$ the posterior only depends on what is observed at that time step and on a prior distribution (here, continuous variables are assumed):

$$p(\mathbf{w}_1|\mathbf{x}_1) = \frac{p(\mathbf{x}_1|\mathbf{w}_1)\, p(\mathbf{w}_1)}{\int_{\mathbf{w}_1} p(\mathbf{x}_1|\mathbf{w}_1)\, p(\mathbf{w}_1)\, d\mathbf{w}_1}. \tag{2.12}$$

As the process evolves, new measurements arrive, so that, at an arbitrary time step $t > 1$, the posterior must be evaluated based on all information from the first until the current time step:

$$p(\mathbf{w}_t|\mathbf{x}_{1...t}) = \frac{p(\mathbf{x}_t|\mathbf{w}_t)\, p(\mathbf{w}_t|\mathbf{x}_{1...t-1})}{\int_{\mathbf{w}_t} p(\mathbf{x}_t|\mathbf{w}_t)\, p(\mathbf{w}_t|\mathbf{x}_{1...t-1})\, d\mathbf{w}_t}. \tag{2.13}$$

According to the law of total probability the marginal probability can also be expressed by an integral over conditional probabilities, so that the prior term in Equation 2.13 can be conditioned on the state of the previous time step, $\mathbf{w}_{t-1}$:

$$p\left(\mathbf{w}_t|\mathbf{x}_{1\ldots t-1}\right) = \int_{\mathbf{w}_{t-1}} p\left(\mathbf{w}_t|\mathbf{w}_{t-1}\right) p\left(\mathbf{w}_{t-1}|\mathbf{x}_{1\ldots t-1}\right) d\mathbf{w}_{t-1}. \tag{2.14}$$

The first term in the integrand in Equation 2.14 is referred to as the *temporal model* and reflects the belief about the system state given the previous state, and the second term reflects the uncertainty about the previous state. The Markov property enables the computation of the posterior probability of the system state by only regarding the previous state. This keeps the computation effort constant and, thus, allows for the application of this method in real-time systems. The estimation of the posterior state by means of Equation 2.13 essentially depends on two steps: Firstly, the computation of the prior probability of the state using Equation 2.14 and, secondly, the computation of the conditional probability of the data given the state, which is defined via a *measurement model*. The first step computes the belief about the current state given the state of the previous time step, which is referred to as *prediction*. The second step relates the state parameters to the measurements and is hence referred to as the *measurement incorporation* or *update* step.

2.2.1 The Kalman filter model

The Kalman filter model (Kalman, 1960) is a realisation of the recursive Bayesian estimation framework, in which the temporal model and the measurement model are linear systems that are modelled to be affected by Gaussian noise. Consequently, the belief about the system state at time t can be expressed by a multivariate Gaussian distribution, $p\left(\mathbf{w}_t\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{w},t}, \Sigma_{\mathbf{ww},t}\right)$, where $\boldsymbol{\mu}_{\mathbf{w},t}$ is the mean vector and $\Sigma_{\mathbf{ww},t}$ is the covariance matrix of state vector $\mathbf{w}$ at time t. The conditional probability of the state given the observed data is expressed as posterior distribution using the theorem of Bayes. Given the posterior state of the previous time step, $p\left(\mathbf{w}_{t-1}|\mathbf{x}_{1\ldots t-1}\right) = \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{w},t-1}, \Sigma_{\mathbf{ww},t-1}\right)$, the prior probability of the state parameters at the current time step t can be computed using Equation 2.14, once the temporal model is defined. Given a measurement at the current time step, the posterior probability of the current state can be computed via Equation 2.13, once the measurement model is defined. The temporal model and the measurement model are defined exemplary in the following paragraphs.

Temporal model. The temporal model translates the state from the previous time step $t-1$ to the current time step t, based on the assumption that the state $\mathbf{w}_t$ evolves from the state $\mathbf{w}_{t-1}$ of the previous time step, via a linear model

$$\mathbf{w}_t = \Psi \mathbf{w}_{t-1} + \epsilon_t, \tag{2.15}$$

where the matrix Ψ is referred to as the transition matrix and ϵ_t is referred to as the process noise which is assumed to follow a zero-mean normal distribution with covariance matrix Σ_p. The covariance matrix of the process noise accounts for deviations from the expected dynamic behaviour of the system

expressed via the transition matrix. Given an exemplary state vector $\mathbf{w}_t = [x_t, y_t, v_{x,t}, v_{y,t}]^\top$ that comprises the position $[x_t, y_t]^\top$ and the velocities $[v_{x,t}, v_{y,t}]^\top$ of a dynamic system in 2D, and assuming that the velocity is constant, the predicted state variables $\mathbf{w}_t^+$ are computed according to:

$$\mathbf{w}_t^+ = \begin{bmatrix} x_t \\ y_t \\ v_{x,t} \\ v_{y,t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ v_{x,t-1} \\ v_{y,t-1} \end{bmatrix}. \tag{2.16}$$

In Equation 2.16, the predicted position is the sum of the previous position and the distance covered within the time span Δt between time $t-1$ and t with constant velocity $[v_{x,t-1}, v_{y,t-1}]^\top$. The uncertainty about the state vector $\mathbf{w}_t^+$ is modelled by the covariance of the process noise, which results from deviations from the constant velocity assumption due to unforeseen accelerations $\mathbf{a}_t = [a_{x,t}, a_{y,t}]^\top$ with expectation $E(\mathbf{a}_t) = 0$ and covariance matrix Σ_a. The covariance matrix of the process noise thus takes the form $\Sigma_p = G\Sigma_a G^\top$, where the effect of the accelerations is related to the covariance of the process noise via the matrix G, which is defined as

$$G = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{2.17}$$

By application of the state transition model, the conditional pdf of the current state, given the previous state, can be written as a normal distribution, whose mean is a linear function of the expected value of the previous state with covariance Σ_p:

$$p\left(\mathbf{w}_t | \mathbf{w}_{t-1}\right) = \mathcal{N}\left(\Psi \boldsymbol{\mu}_{\mathbf{w},t-1}, \Sigma_p\right). \tag{2.18}$$

According to Equation 2.14, the belief about the system state, given the information from the past, can be expressed via the integral over the product of the prior distribution $p(\mathbf{w}_{t-1}|\mathbf{x}_{1\ldots t-1})$ and the state transition model previously defined, which yields the probability $p(\mathbf{w}_t|\mathbf{w}_{t-1})$. The pdf of the predicted state is thus found by application of Equation 2.14:

$$p\left(\mathbf{w}_t | \mathbf{x}_{1\ldots t-1}\right) = \int_{\mathbf{w}_{t-1}} \mathcal{N}\left(\Psi \boldsymbol{\mu}_{\mathbf{w},t-1}, \Sigma_p\right) \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{w},t-1}, \Sigma_{\mathbf{ww},t-1}\right) d\mathbf{w}_{t-1}. \tag{2.19}$$

By applying the calculation rules for marginal and conditional Gaussians (cf. (Bishop, 2006), Chapters 2.3.2 and 2.3.3) to Equation 2.19, the pdf of the predicted state can be reformulated as:

$$\begin{aligned} p\left(\mathbf{w}_t | \mathbf{x}_{1\ldots t-1}\right) &= \mathcal{N}\left(\Psi \boldsymbol{\mu}_{\mathbf{w},t-1}, \Psi \Sigma_{\mathbf{ww},t-1} \Psi^\top + \Sigma_p\right) \\ &= \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{w}}^+, \Sigma_{\mathbf{ww}}^+\right), \end{aligned} \tag{2.20}$$

where $\boldsymbol{\mu}_{\mathbf{w}}^+$ denotes the mean of the predicted state at time t and $\Sigma_{\mathbf{ww}}^+$ denotes its covariance matrix. In

Equation 2.20, the probability of the state given all and only the information from the past is expressed by a normal distribution, whose mean is a linear function of the mean of the preceding state, and the covariance of the predicted state is the sum of the covariance matrix of the previous state propagated in time using the state transition model and the covariance matrix of the process noise.

Measurement model. Given a measurement $\mathbf{x}_t = [x_t, y_t]^\top$ of the current position of the dynamic system, the measurement model translates the predicted state to the posterior state by application of Equation 2.13. The measurement is assumed to be drawn from the noisy measurement process

$$\mathbf{x}_t = \mathrm{M}\mathbf{w}_t + \epsilon_m, \tag{2.21}$$

where the matrix M is referred to as the measurement matrix that translates the state to the measurements with additive zero-mean Gaussian noise $\epsilon_m \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{xx},t})$. For the exemplary state vector defined in the previous paragraphs and given that the measurements are defined in the same coordinate system as the state variables, the measurement matrix takes the form $\mathrm{M} = [\mathbf{I}\ \mathbf{0}]$, where $\mathbf{I}$ is the identity matrix of size 2, and $\mathbf{0}$ is a matrix of the same size with all elements set to zero, because the velocity components of the state vector are not observed. In probabilistic form, the term $p(\mathbf{x}_t|\mathbf{w}_t)$ in Equation 2.13 can be expressed as

$$p(\mathbf{x}_t|\mathbf{w}_t) = \mathcal{N}\left(\mathrm{M}\boldsymbol{\mu}_{\mathbf{w},t}, \Sigma_{\mathbf{xx},t}\right). \tag{2.22}$$

The posterior probability of the state can be found by integrating Equations 2.20 and 2.22 into Equation 2.13:

$$p(\mathbf{w}_t|\mathbf{x}_{1\ldots t}) = \frac{\mathcal{N}\left(\mathrm{M}\boldsymbol{\mu}_{\mathbf{w},t}, \Sigma_{\mathbf{xx},t}\right)\mathcal{N}\left(\boldsymbol{\mu}^+_{\mathbf{w}}, \Sigma^+_{\mathbf{ww}}\right)}{\int_{\mathbf{w}_t}\mathcal{N}\left(\mathrm{M}\boldsymbol{\mu}_{\mathbf{w},t}, \Sigma_{\mathbf{xx},t}\right)\mathcal{N}\left(\boldsymbol{\mu}^+_{\mathbf{w}}, \Sigma^+_{\mathbf{ww}}\right)d\mathbf{w}_t}. \tag{2.23}$$

As the product of two Gaussian distributions is proportional to another Gaussian distribution, Equation 2.23 can be rewritten as

$$p(\mathbf{w}_t|\mathbf{x}_{1\ldots t}) \propto \mathcal{N}\left(\left(\mathrm{M}^\top\Sigma^{-1}_{\mathbf{xx},t}\mathrm{M} + {\Sigma^+_{\mathbf{ww}}}^{-1}\right)^{-1}\left(\mathrm{M}^\top\Sigma^{-1}_{\mathbf{xx},t}\boldsymbol{\mu}_{\mathbf{w},t} + {\Sigma^+_{\mathbf{ww}}}^{-1}\boldsymbol{\mu}_+\right), \left(\mathrm{M}^\top\Sigma^{-1}_{\mathbf{xx},t}\mathrm{M} + {\Sigma^+_{\mathbf{ww}}}^{-1}\right)^{-1}\right). \tag{2.24}$$

By introducing the Kalman Gain matrix,

$$K = \Sigma^+_{\mathbf{ww}}\mathrm{M}^\top\left(\Sigma_{\mathbf{xx},t} + \mathrm{M}\Sigma^+_{\mathbf{ww}}\mathrm{M}^\top\right)^{-1}, \tag{2.25}$$

Equation 2.24 can be simplified to

$$\begin{aligned} p(\mathbf{w}_t|\mathbf{x}_{1\ldots t}) &= \mathcal{N}\left(\boldsymbol{\mu}^+_{\mathbf{w}} + \mathrm{K}\left(\mathbf{x}_t - \mathrm{M}\boldsymbol{\mu}^+_{\mathbf{w}}\right), \Sigma^+_{\mathbf{ww}} - \mathrm{KM}\Sigma^+_{\mathbf{ww}}\right) \\ &= \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{w},t}, \Sigma_{\mathbf{ww},t}\right). \end{aligned} \tag{2.26}$$

For a derivation of Equation 2.25 see, e.g., (Prince, 2012). Using the Kalman gain matrix, a trade-off between the measured state and the predicted state is found, which is influenced by the covariance matrices of the prediction and the measurement. The term in brackets in the expression for the mean

in Equation 2.26 is called *innovation* $\mathbf{i} = \mathbf{x}_t - \mathrm{M}\boldsymbol{\mu}_{\mathbf{w}}^{+}$ with covariance

$$\Sigma_{\mathbf{ii}} = \Sigma_{\mathbf{xx}} + \mathrm{M}\Sigma_{\mathbf{ww}}^{+}\mathrm{M}^{\top}. \tag{2.27}$$

Summary. The recursive prediction and update steps in a Kalman filtering model can be summarised as follows:

Prediction:

$$\boldsymbol{\mu}_{\mathbf{w}}^{+} = \Psi\mu_{\mathbf{w},t-1} \tag{2.28}$$

$$\Sigma_{\mathbf{ww}}^{+} = \Psi\Sigma_{\mathbf{ww},t-1}\Psi^{\top} + \Sigma_p \tag{2.29}$$

Update:

$$\boldsymbol{\mu}_{\mathbf{w},t} = \boldsymbol{\mu}_{\mathbf{w}}^{+} + \mathrm{K}\left(\mathbf{x}_t - \mathrm{M}\boldsymbol{\mu}_{\mathbf{w}}^{+}\right) \tag{2.30}$$

$$\Sigma_{\mathbf{ww},t} = \Sigma_{\mathbf{ww}}^{+} - \mathrm{KM}\Sigma_{\mathbf{ww}}^{+} \tag{2.31}$$

Intuitively, the prediction propagates the previous state according to the temporal model, and the variance of the predicted state depends on the variance of the previous state and an additional covariance matrix that models deviations from the assumed motion model. The updated state then equals the predicted state plus a trade-off between the measurements and the predicted state weighted by the covariances of the respective entities. Note that Kalman filtering is equivalent to message passing in the forward direction, whereas the posterior marginals can be found by application of the so-called Kalman smoother equations (Kschischang et al., 2001).

The Kalman filter model assumes linear models for the measurement generation and for the prediction. If one or both of these models involve non-linear, yet differentiable, functions, Ψ and/or M are replaced by local linear approximations of these functions for the prediction and update of the covariance matrices. Such a model is referred to as *Extended Kalman filter* (EKF, Gelb, 1974). Another variant of the Kalman filter model is the *Unscented Kalman filter* (Julier and Uhlmann, 1997), which approximates isolines of the Gaussian distributions by particles. These particles are propagated by the possibly non-linear measurement and/or update equations and the transformed particles resemble the new distributions. The aforementioned models all assume Gaussian distributions for the noise models. In contrast to these models, *particle filtering* models (Deutscher et al., 2000) circumvent the assumption of a uni-modal state by representing the distribution of the state by a set of independent particles. In this way, arbitrary multi-modal distributions can be handled.

2.2.2 Dynamic Bayesian Networks

A key limitation of Kalman Filter models (KFM) is that these models only handle (vectors of) single random variables to represent a dynamic system state. A *Dynamic Bayesian Network* (DBN) can be understood as an extension of a Bayesian network to the state-space domain (Murphy, 2002; Russell et al., 2003). According to the properties of Bayesian networks, a DBN allows for the representation

of a system state in a factorised form. Because DBNs are nothing more than Bayesian Networks set into a temporal context, inference algorithms such as Belief propagation can be used readily for reasoning about hidden variables in such models (Russell et al., 2003). In contrast to Kalman Filter models, DBNs alleviate the need for linear models for the conditional probabilities with Gaussian noise and allow for arbitrary conditional probability functions, and the variables may be either discrete, continuous or both. This allows for the modelling of more complex connections of random variables while maintaining the computational efficiency of inference algorithms available for Bayesian networks.

This way of modelling can also be applied to the example of land slide prediction given in the earlier sections of this chapter. In this example, the land cover class was modelled as hidden variable. The land cover class can be expected to depend on its value at the previous time step, so that the probability distribution for the land cover class C_t at time $t = 2$ can be written as conditional probability of C_2 given C_1, $p(C_2|C_1)$. Such a DBN is depicted in Figure 2.5, where the index refers the variables to the time step. Again, inference in this Dynamic Bayesian Network can be performed using Belief Propagation, e.g., in order to find the most likely sequence of land cover class labels and land slide susceptibilities, or to reason about the current system state given the estimates from the past and current measurements.

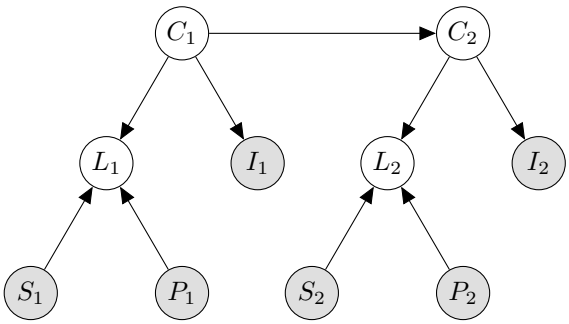


Figure 2.5: Dynamic Bayesian network representation of the problem of land slide prediction (cf. Figure 2.2) for two successive time steps, indicated by the indices of the variables. The directed edge between C_1 and C_2 represents the conditional probability of the land cover class C_2 at time $t = 2$, given the land cover class C_1 of the preceding time step.

2.3 Gaussian Process Regression

This section addresses the class of stochastic processes, which provide a mathematical description of (time) ordered events, whose outcome at every step (in time) depends on chance. A Gaussian Process (GP) is a realisation of a stochastic process, in which each point in input space is associated with a target variable that is normally distributed, and in which every finite subset of random variables has a multivariate normal distribution (Rasmussen, 2006). By analogy with a Gaussian distribution over scalar (or vectorial) random variables, which are defined by a mean (vector) and a (co)variance(matrix), a Gaussian Process can be thought of as a distribution over functions, uniquely defined by a mean function $m(x)$ and a covariance function $k(x, x')$ of two points x and x' in a common input space.

The probability distribution over a function $f(x)$ can, thus, be denoted by

$$f(x) \sim \mathcal{GP}\left(m(x), k\left(x, x'\right)\right). \tag{2.32}$$

Gaussian Processes can be used for the estimation of continuous target variables from a given set of observed target variables (i.e., for regression), and for classification, given that the target variables are discrete (Rasmussen, 2006). Generally, regression aims at the determination of function values of a target variable at arbitrary positions in an input space by modelling deterministic functions that best approximate a set of observed variables, e.g., in terms of least squares. Gaussian Process Regression refines this approach by modelling stochastic relations between the variables. In this respect, Gaussian Process Regression is related to *kriging*, as referred to in the field of geo-statistics (Krige, 1951), and *collocation*, as referred to in the field of geodesy (Moritz, 1973). Here, the terminology of GP is used for consistency with the related work in the fields of computer vision and machine learning (Rasmussen, 2006; Urtasun et al., 2006). Formally, it is assumed that the function values of a target variable y, evaluated for an input variable x, are drawn from a noisy process,

$$y = f(x) + n = m(x) + s + n, \tag{2.33}$$

with Gaussian white noise n with variance σ_n^2. The regression function $f(x) = m(x) + s$ is composed of a deterministic part $m(x)$, also referred to as the *trend*, and a stochastic part $s \sim \mathcal{N}(0, K)$, which is referred to as the *signal* and which follows a zero-mean normal distribution with covariance K. It is further assumed that the signals at close positions are correlated, and that these correlations are described by the covariance function. Gaussian Process Regression enables the joint estimation of the parameters of the trend and the function values at new input points. The trend is defined by the mean function, which can be formulated, e.g., by polynomial regression or by non-parametric approaches for regression using kernel functions (Jaakkola and Haussler, 1999). The covariance function quantises the expected correlations between signals as a function of distance in input space. One of the most prominent realisations of the covariance function is the squared exponential or Gaussian function (Rasmussen, 2006):

$$k(x_i, x_j) = \sigma_f^2 \cdot exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right) + \sigma_n^2 \cdot \delta(i = j). \tag{2.34}$$

Equation 2.34 is a Gaussian function in which the covariance of two input points x_i and x_j depends on their distance with decreasing covariance at growing distances. In Equation 2.34, σ_n^2 is the noise variance accounted for in the diagonal elements of K, and $\delta(i = j)$ is the Kronecker delta function, which is 1 for $i = j$ and 0 otherwise. Depending on the noise variance of an input variable, the target variable is more or less affected by input points near its position in input space. The characteristic length-scale l basically controls the range of correlations in input space. The signal variance σ_f^2 controls the uncertainty of predictions at input points far from observed data.

An illustration of a Gaussian Process Regression model is given in Figure 2.6. The dashed line resembles the trend (in the figure, it is assumed to be zero) and the green line the signal. Three observed input points are shown in the figure. The black solid lines show the expected 1σ interval over the target variables at each position in input.

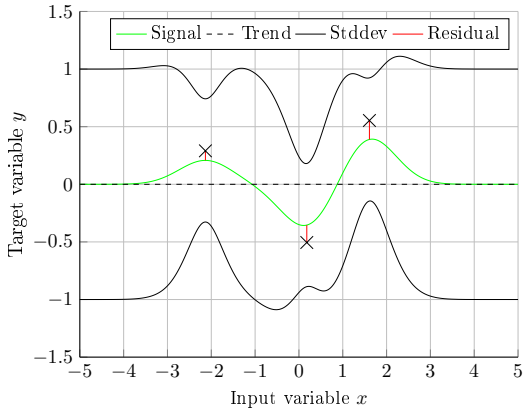


Figure 2.6: Illustration of a Gaussian Process with a zero-mean function and three pairs of observed input and target variables, marked by '×' symbols.

Given a set of observed input and target variables $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$, the aim is to predict the target variable y_* for a new input point x_*. Since any finite subset of the random variables in a GP has a Gaussian joint distribution, the joint distribution of an unknown target variable y_* and the observed data $\mathbf{y} = \{y_1, ..., y_n\}$ is also Gaussian and can be modelled according to Equation 2.35,

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} E(\mathbf{y}) \\ E(y_*) \end{bmatrix}, \begin{bmatrix} K & K_*^\top \\ K_* & K_{**} \end{bmatrix} \right) \tag{2.35}$$

where $E(\mathbf{y})$ and $E(y_*)$ are the expected values of the observed and the unknown target variables that correspond to the trend. The covariance matrix is computed using the covariance function $k(x_i, x_j)$. The matrix K is the covariance matrix of the observed target variables, such that $K(i,j) = k(x_i, x_j)$, K_* is the vector of covariances of the observed and the unknown target variables, such that $K_*(i) = k(x_*, x_i)$, and $K_{**} = k(x_*, x_*)$ is the variance of the unknown target variable.

The prediction of a new target variable for the input point x_* is realised by constructing the conditional distribution of the desired target variable y_* based on Equation 2.35. Since the distribution in Equation 2.35 is Gaussian, the same holds true for the conditional probability of the unknown target variable. In a Gaussian Process Regression model, the distribution over the predicted target variable can be written as Equation 2.36,

$$P(y_*|\mathbf{y}) \sim \mathcal{N}\left(\mathcal{GP}_\mu(x_*, \mathcal{D}), \mathcal{GP}_\Sigma(x_*, \mathcal{D})\right), \tag{2.36}$$

with mean

$$\hat{y}_* = \mathcal{GP}_\mu(x_*, \mathcal{D}) = m(x_*) + K_* K^{-1}(\mathbf{y} - m(\mathbf{x})), \tag{2.37}$$

and covariance

$$\hat{\sigma}^2_{y_*} = \mathcal{GP}_\Sigma(x_*, \mathcal{D}) = K_{**} - K_* K^{-1} K_*^\top. \tag{2.38}$$

The estimated value $\hat{y}_*$ is the sum of the trend and the signal and corresponds to the mean of Equation 2.36 and $\hat{\sigma}^2_{y_*}$ is the variance of the estimated target variable.

3 Related work

This chapter gives an overview over related work on the topic of multi-person detection and tracking from monocular image sequences. Following Smeulders et al. (2014), visual object tracking can be categorised according to the way in which the object position in image space is acquired. *Matching-based* approaches used for instance in (Shi and Tomasi, 1994; Isard and Blake, 1996; Comaniciu et al., 2003) generally continue the trajectory towards positions that best coincide with target representations of previous time steps, regardless of whether or not the target is actually visible in the current frame. *Detection-based* approaches, also referred to as *tracking-by-detection* (Andriluka et al., 2008), typically use classifiers to discriminate the regarded object class(es), and only update a trajectory when an object is actually detected. In the context of multi-person tracking, where mutual occlusions are inherent, the trajectories are easily distracted from the actual target when using matching-based approaches. In this respect, and with the progress of automatic pedestrian detection in the last decade (Dollár et al., 2011), detection-based approaches for tracking have moved into the focus of interest in most of the related work on multi-person tracking. In the context of single object tracking, pedestrian detection provides measurements of the desired object positions in individual frames, which can be readily evaluated, e.g. by filtering or smoothing techniques. Because the detections do not carry information about the identity of the detected objects, the application to multi-object tracking is not straightforward. The possible assignment of detections to different object trajectories gives rise to an ambiguity problem, often referred to as the problem of *data association*.

The remainder of this chapter begins with an overview over general tracking approaches available in the literature (Section 3.1). The subsequent sections provide a description of methods for single-frame detection and localisation (Section 3.2), and temporal modelling strategies for state prediction (Section 3.3). Section 3.4 describes common approaches for data association on a frame-by-frame basis. Finally, in Section 3.5, the strengths and weaknesses of the related work are discussed and the research gaps, which motivate the further work on this topic, are identified.

3.1 Tracking approaches

The general aim of tracking is to establish a target's position over time. This can be either achieved by successive continuation of trajectories at every time step, or by regarding multiple time steps simultaneously. This section provides a review of common approaches for generating the trajectories in the literature and is separated into three parts, associated to state space models and filtering techniques, tracklet-based approaches and offline tracking.

State space models and filtering techniques

In the context of recursive estimation of the trajectories, tracking-by-detection requires the detection of persons in individual frames, the association to trajectories, i.e. data association, and the filtering to find a synthesis between the position measured by the detector and a motion model. Approaches designed for applications that run in real time only obtain information from the current and the previous frames, having to decide for the detection-to-trajectory assignments instantaneously. In recursive estimation frameworks, the dynamic state of a tracked pedestrian is often modelled by its position and derivatives of the position w.r.t. time in a common state space. In this context, trajectories can be seen as the connecting lines between the filtered positions in state space. Available approaches for filtering generally make use of variants of the recursive Bayesian estimation framework, and model the pedestrian position and its velocity as state variables. If the predictive function is linear and if the noise is normally distributed, the system is often referred to as linear dynamical system (Bishop, 2006). For the recursive state estimation of a linear dynamical system from a sequence of measurements affected by Gaussian noise, the Kalman filter model (Kalman, 1960) yields optimal solutions and is applied in a wide range of tracking applications (Zhao and Nevatia, 2004a; Luber et al., 2010; Shu et al., 2012; Yoon et al., 2015). The authors of these papers model the state variables in the image domain, which has the advantage of being independent of the camera orientation as long as the orientation does not change, thus being more generic in terms of the potential application scenarios. The drawback of these methods is the missing scale information, which can be exploited when modelling the pedestrian state in a common 3D object coordinate system. However, the recursive state estimation in Euclidean 3D coordinates, based on measurements obtained in the projected 2D space of the image, requires a non-linear transformation. To this end, methods that model the system state in 3D make use of Extended Kalman filter models (Schindler et al., 2010), linearising the measurement equations using a Taylor expansion, or Unscented Kalman Filter models (Meuter et al., 2008). Particle filters are used in (Okuma et al., 2004; Zhao and Nevatia, 2004b; Breitenstein et al., 2011; Tran and Manzanera, 2015) and (Zhang et al., 2015). Kalman Filter models facilitate the parametric modelling of confidences and, therefore, also the application of hypothesis testing. For instance, the parametric modelling of probability densities in Kalman filter models allows for the detection of outliers in the measurements by looking at the system innovations. Particle filtering techniques avoid the need for uni-modal expressions of the belief about the system state, but, on the other hand, lack the ability of performing statistical tests on the basis of a parametric expression of the uncertainties.

An alternative view on the state space model is to formulate the state vector as target variable in a Gaussian Process Regression model. Ko and Fox (2009) formulate the dynamic system of a robotic platform as a Gaussian Process, and model the predictive function, as well as the measurement functions, of a recursive Bayes filter as a Gaussian Process Regression problem. It is emphasised that the new prediction and observation models do not require an explicit parametric model of the functional relationship, and only require a model for the mean and covariance function.

Tracklet-based approaches

Tracklet-based approaches for tracking decompose the task into smaller problems and first generate trajectory fragments for which the association can be done reliably, and then associate the generated tracklets over time. Multiple Hypothesis Tracking was proposed by Reid (1979) and revisited only recently (Hinz and Schmidt, 2015; Kim et al., 2015) for multi-person tracking. This method builds a tree of potential trajectory candidates originating from each single-frame detection with branches growing through detections of successive frames in a way that no detection is assigned to more than one trajectory at a time. Each path through the tree corresponds to a trajectory candidate and the set of candidates is reduced (or *pruned*) based on a tracking score function prior to the processing of the next frame to keep the computation time tractable. A similar approach is pursued by Ess et al. (2008) and Schindler et al. (2010), generating multiple trajectory hypotheses and optimising the combination of successive hypotheses in a separate step. Yang and Nevatia (2012) and Choi (2015) propose the use of Conditional Random Fields (CRF) to estimate correspondences between tracklets. Similarly to the work on frame-wise association, methods based on the Hungarian algorithm (Perera et al., 2006) and on linear programming (Jiang et al., 2007) are also used for the tracklet-to-track assignment. Such approaches delay the trajectory generation, because a set of images must be available before a decision is made, but operate in near-real time if the time window is only small.

Offline tracking

To find a globally optimal solution for the entire image sequence, available approaches commonly formulate the association task as a maximum a posteriori problem, which can be represented by a min-cost flow network (Zhang et al., 2008a). This method uses representations based on graphical models, where temporally ordered nodes correspond to successive detections (Leal-Taixé et al., 2011), or tracklets (Wang et al., 2015a; Ben Shitrit et al., 2014). Directed edges between these nodes represent correspondences. The optimal path through the graph indicates the final trajectories and can be found using linear programming (Berclaz et al., 2009; Leal-Taixé et al., 2011), dynamic programming (Pirsiavash et al., 2011; Wang and Fowlkes, 2015), or other optimisation methods. Because the methods aforementioned require the first and last frame of an image sequence for the optimisation, the processing is restricted to offline applications.

Recent developments investigate deep learning architectures (LeCun et al., 2015) for solving the assignment task in a model-free fashion. Milan et al. (2016) approach the task of data association by training a recurrent neural network that enables a model-free representation of the optimisation procedure. Despite the great success in deep learning architectures in recent years, the network requires vast amounts of training data, which are typically not available for multi-object tracking yet. Siamese convolutional neural networks (CNN) are used for data association of tracklets (Wang et al., 2016) in an offline fashion, and of individual detections (Leal-Taixé et al., 2016). In (Leal-Taixé et al., 2016), associations predicted by the CNN are further optimised in a linear programming approach, which is applied over all frames and, thus, restricts the method to offline processing as well.

3.2 Observations

This section describes the related work on generating observations of pedestrians in single images, which build the basis for detection-based approaches for tracking. The term detection involves the decision that an object of a specific object class is present, and at least a coarse localisation of the object. Both these aspects are provided by pedestrian detectors, which are commonly based on classifiers that are either trained offline for the purpose of generic object detection, or at runtime, to specialise on a specific target appearance. Both approaches, their limitations and possible remedies are discussed in the following paragraphs.

Detection based on classifiers trained offline

State-of-the-art pedestrian detection is commonly guided by classifiers trained offline on large sets of pedestrian images to generalise well across a wide range of different pedestrian characteristics. Available approaches differ in the way in which the evidence about the presence of a pedestrian in an image is generated, and can be roughly categorised into top-down and bottom-up approaches. Top-down or *sliding-window*-based approaches typically train a holistic model of pedestrians (Viola and Jones, 2001; Dalal and Triggs, 2005; Dollár et al., 2014) and directly classify all feasible regions in the scale-space representation of an image either as person or as background. One of the most prominent representatives of sliding-window-based detectors is the approach by Dalal and Triggs (2005) that classifies features derived from Histograms of Oriented Gradients (HOG) using Support Vector Machines (SVM). Basically, a HOG is a frequency distribution of edge gradient orientations, with the votes being weighted by the magnitudes of the gradients and accumulated within local spatial regions. The person descriptor in (Dalal and Triggs, 2005) consists of HOG features from multiple sub-regions arranged inside a search window, contrast-normalised within overlapping blocks, and concatenated to a feature vector. For detection, the search window is placed at every feasible position in the image at different scales and the feature vectors computed inside these windows are classified into *person* or *background* using an SVM. The location and size of a search window classified as person are the parameters that describe a detection geometrically. In the approaches mentioned above, the decision if a pedestrian is present takes place by classification of image regions that are expected to depict the entire persons. If a person is occluded, the detection generally fails, leading to an error of omission or a *false negative* detection. When the detection is applied permissively, i.e. if no threshold w.r.t. the confidence of the SVM is applied, the detection typically yields very high recall rates at the cost of many *false-positive* detections (errors of commission), due to the classification of objects in the background with shapes similar to that of a person.

In contrast to the sliding-window-based approaches, *part-based models* decompose the model of a pedestrian into a set of patches in the vicinity of interest points or body parts whose relative positions from the object centre are known (Leibe et al., 2008; Felzenszwalb et al., 2010). The detection itself takes place in a bottom-up fashion by first finding the individual patches or parts and then generating votes for the object position based on the relative displacements stored in the model. Maxima in the accumulation of the generated votes are taken as detections. Felzenszwalb et al. (2010) build upon

the idea of using HOG features, but represent an object as a deformable part-model, training different HOG descriptors for every single part. In this way, the approach accounts for the non-rigidity of human bodies or other object classes in a better way and is more robust against partial occlusions, so that the risk of omission errors decreases. Models based on interest points and parts mitigate the need for a holistic visibility of the persons, but the visible features of partly occluded objects often do not provide as much information as needed for the re-identification of an object in a crowded scenario.

Although the available approaches for detection provide a solution to the detection and localisation problem at the same time, the results are often not particularly reliable and geometrically accurate. In the scientific community a detection is generally evaluated based on the relative overlap – i.e., the intersection over union score – of a detection and a reference annotation, both represented by a surrounding rectangle (Everingham et al., 2010). In a comprehensive study, Dollár et al. (2011) point out that the recall rates of 16 different pedestrian detectors decrease rapidly if the intersection-over-union score threshold is increased. In fact, the resulting surrounding rectangles used in such approaches may easily be misaligned due to partial occlusions, non-rigid body motion, changing illumination and other disturbing effects. Established challenges on object detection (Everingham et al., 2010), single object tracking (Čehovin et al., 2015) and multi object tracking (PETS, 2009; Geiger et al., 2012; Leal-Taixé et al., 2015), only require an intersection-over-union score of 50% for a detection to be counted as correct. However, for many realistic applications that geometric accuracy is not sufficient, and the improvement of this accuracy is discussed later in this section.

Detection based on classifiers trained online

Classifiers trained at runtime are capable of specialising on a specific target's appearance and allow for the renewed detection of a previously tracked object in successive frames. Available approaches for learning models of appearance for individual persons from image sequences make use of variants of Random Forests (Breiman, 2001; Saffari et al., 2009), Hough Forests (Gall and Lempitsky, 2013), boosting (Okuma et al., 2004; Breitenstein et al., 2011; Godec et al., 2011) and Convolutional Neural Networks (Ma et al., 2015; Wang et al., 2015b). These approaches have the advantage of being adaptive to appearance changes, which makes these approaches more applicable to complex scenes with a wide range of depth, temporary occlusions, and changing lighting conditions. Breitenstein et al. (2011) train one binary classifier based on boosting for every tracked pedestrian against all other persons and complement the results of a generic object detector with the additional instance-specific information to find an improved location of the desired object in the image. In a multi-object tracking environment, every binary classifier is trained with positive samples from a desired object and with negative samples from all other objects and from the background. Because the classifier is expected to discriminate against a potentially high number of different objects, the number of samples for the desired object class is typically much larger than the number of samples for the negative class. Consequently, if all samples from the negative classes are represented by only one class, the confidences given by the classifier upon evaluation of a sample may be delusive. As opposed to boosting-based methods for tracking, methods based on ensembles of decision trees, i.e. Random Forests (Breiman, 2001), are also used for tracking. Random Forests are trained in a breadth-first manner and aggregate

many rather shallow decision trees, each of which represents the frequency distribution of any number of classes (depending on the depth of the trees). Upon evaluation, the average class-frequency of the leave nodes, where a sample propagates to, is taken as confidence about the class-membership of the sample. Random forests are inherently applicable to multi-class problems and are more robust against label noise than boosting-based approaches, which is due to the principle of learning independent ensembles of decision trees. Saffari et al. (2009) propose an online growing procedure for decision trees based on online bagging and boosting (Oza, 2005). Because samples for training are rare from scratch in online applications, Saffari et al. (2009) propose a strategy for incremental training of a Random Forest. Beginning with binary decision stumps with a depth of one (i.e. one root node and two leave nodes), a node is only allowed to split if enough training data have become available and if the split function (selected from a set of randomly generated candidates) separates the classes sufficiently well. To account for potentially changing appearance features of the tracked objects, entire decision trees may be discarded based on the misclassification rate of samples held back from training (out-of-bag-error). These so-called *Online Random Forests* by (Saffari et al., 2009) are not applied to the multi-class domain, where the number of classes changes over time.

Although multi-object tracking is a multi-class problem by nature, the available work on instance-specific classification only uses single binary classifiers for every tracked object. None of the existing methods uses Random Forest classifiers for the tracking of multiple persons.

Non-maximum suppression and false positive reduction

Sliding-window-based approaches for detection typically deliver many positive classifications near the true position of a person in the image. To yield only one detection per object, non-maximum suppression (NMS) is usually applied, keeping only track of detections with confidences (measured, e.g., by the distance of a detection from the hyper-plane of the SVM in feature space) above a pre-defined threshold and grouping these detections based on their similarity in terms of size and position. This way of performing NMS comes along with two drawbacks: Firstly, rejecting detections by a constant threshold leads to a decision about the validity of a detection that is not guaranteed to be optimal, because not all persons are detected with the same confidence. Secondly, using the classification confidence for NMS, all detections are validated irrespectively of their positions relative to the scene. To this end, Hoiem et al. (2008) refine the NMS strategy of available object detectors by taking 3D information about the expected objects and the scene into account. The authors keep track of the distribution of the non-maximum detections, developing a more sophisticated measure of uncertainties of the final detections. This work aims at the recognition of objects in single frames and, therefore, neglects temporal correlations between successive frames in an image sequences.

To decrease the false positive rate while preserving the true positive detections, the detector may be applied permissively, validating the detections with additional clues like foreground-background-separation (Stauffer and Grimson, 2000; Elgammal et al., 2000; Zhao and Nevatia, 2004a), or shape (Leibe et al., 2005; Ramanan, 2007; Gavrila and Munder, 2007) prior to further processing. By pre-processing the images, hard decisions are made sequentially and errors committed in that early stage of processing cannot be corrected later. To overcome this problem, Hoiem et al. (2008), Wojek et

al. (2009), Enzweiler et al. (2010), Schindler et al. (2010) and Choi et al. (2013) integrate different sources of information related to depth, texture, shape, appearance and motion. Wojek et al. (2009) and Enzweiler et al. (2010) use classifiers to jointly evaluate different sources of information related to the recognition of pedestrians. Schindler et al. (2010) apply Bayesian Networks for the joint inference of unknown parameters that are related to the correctness of a detection, the object position, to the parameters of the camera orientation and the scene, and other variables. By modelling the correctness of a detection as a hidden variable, the decision whether to use a detection result for the update of a trajectory is made on the basis of the joint probability of the hidden variables and several observations. Every detection is validated by estimating the correctness of that detection based on probabilistic reasoning in the graphical model. Joint inference of the parameters is performed by means of Belief Propagation, but the global optimum is not guaranteed to be found due to the loopy nature of the underlying graphical model. Choi et al. (2013) combine different cues about the presence of pedestrians based on full pedestrian and part-based models, and achieve robustness against outliers in individual system components by the redundancy of observations. These observations are combined in a graphical model, but only approximate inference is applied to reason about the final detections.

Refined localisation

A better alignment of a detection result to the real object boundaries is, for instance, achieved when the location of the detection window is used as initial location from which a refined segmentation proceeds. This segmentation can be carried out on the basis of pixels (Godec et al., 2011; Dai and Hoiem, 2012), superpixels (Shu et al., 2013; Milan et al., 2015), interest points (Leibe et al., 2008; Ommer et al., 2009; Gall and Lempitsky, 2013; Choi, 2015), object parts (Andriluka et al., 2008; Felzenszwalb et al., 2010), contour models (Leibe et al., 2005; Gavrila and Munder, 2007) or, in the context of rigid object detection, full 3D models (Zia et al., 2013). Ommer et al. (2009) cluster interest points based on their motion and classify these groups to jointly determine and localise the underlying object category. In (Klinger et al., 2014) and Choi (2015), the localisation accuracy of a pedestrian detector is improved using additional cues from the tracking of interest points (Shi and Tomasi, 1994). While being robust to single outliers in the generated interest point correspondences, these approaches are easily distracted from the desired object when the object is occluded. Segmentation-based refinement of a detection on a pixel-basis generally tries to find a trade-off between an alignment of the detection to edges in the image and an internal energy constraint keeping the generated contour smooth. Due to such internal energies, limbs are often not segmented well, so that additional shape models may be required to keep the segmentation close to plausible silhouettes of persons (Leibe et al., 2005; Gavrila and Munder, 2007; Milan et al., 2015). Such models require a massive amount of training data to consider the possible articulations of the body parts. Furthermore, such models assume a holistic visibility of the persons and, thus, fail in case of occlusions, so that additional models may be required (Ouyang and Wang, 2013; Rujikietgumjorn and Collins, 2013; Possegger et al., 2014). In (Schindler et al., 2010), the positions of pedestrians are modelled as hidden variables in a Bayesian network, which are evaluated at every time step together with observations stemming from automatic object detection and from a pair of stereo cameras, which are not further considered in this work.

3.3 Temporal modelling

The aim of temporal modelling in the context of recursive filtering is to find an expression of the belief about a system state at a current time step given only the information from the past. The temporal model supports the trajectory estimation in different respects: It keeps the generated trajectory smooth, it supports data association (detection-based tracking) and preserves spatio-temporal consistency in the absence of detections or in case of an occlusion, i.e. if no measurements can be obtained. In a recursive Bayesian estimation framework, a temporal model consists of a predictive function, mapping the previous state to the current state, and a stochastic model, representing the uncertainties about the predicted state. Available approaches differ in the way of representing the belief and in the expectation of the dynamic behaviour of the system. In online applications, conditional independence between a current state and states further than one step in the past, given the previous state, is often assumed. Higher order motion models are considered, for instance, in (Pellegrini et al., 2009) and (Schindler et al., 2010), and are more commonly found in offline applications (Collins, 2012; Arora and Globerson, 2013; Milan et al., 2014). Different motion models are defined for different action categories in (Keller et al., 2011). In the addressed application scenarios, only walking persons are expected so that different action categories do not promise any improvement of the prediction.

In the context of multi-person tracking, the assumption that pedestrians move in rather straight lines independent of other pedestrians is often not valid in much frequented scenarios. In such situations, people react to their environment due to social forces (Helbing and Molnár, 1995) and physical constraints.

Motion context

The term *motion context* is used ambiguously in the literature either for the joint consideration of body part motion for the understanding of activities (Zhang et al., 2008b), or for the joint consideration of interacting pedestrian motion (Yoon et al., 2015). Here, it is meant to be understood in the latter meaning of the term. The social force model by Helbing and Molnár (1995) reflects the observation that people exhibit mutual patterns of behaviour like walking in groups towards common destinations in a scene. Physical constraints are often considered in the literature to model the fact that people cannot share the same location in space (Scovanner and Tappen, 2009), (Pellegrini et al., 2009), (Leal-Taixé et al., 2011), (Yamaguchi et al., 2011), (Choi et al., 2013), (Milan et al., 2014).

In (Scovanner and Tappen, 2009) and (Milan et al., 2014), trajectory estimation is formulated as an energy minimisation problem, where the energy is the sum of various terms penalising a deviation from an expected behaviour, such as collision avoidance, moving towards a predefined destination in a rather straight line with constant velocity. By being aware of other people's positions in the scene, motion context is incorporated in a way that collisions can be avoided, but possible correlations of the trajectories that indicate mutual patterns of motion are not further evaluated. Ge et al. (2009), Pellegrini et al. (2010), Yamaguchi et al. (2011) and Zhang and van der Maaten (2013) incorporate group models which enable a smooth motion of pedestrians of the same group. Although contextual information w.r.t. the motion of interacting pedestrians is considered in this way, a binary decision

about the group memberships must be made, which neglects potential correlations between subjects of different groups. Pellegrini et al. (2009), Choi et al. (2013), Leal-Taixé et al. (2014) and Yoon et al. (2015) do not apply an explicit grouping. They predict the position of each subject based on the history of all pedestrians. Pellegrini et al. (2009) directly incorporate interactions as well as expected destinations in the scene into the dynamic model of a recursive filter. The degree of interaction between two pedestrians is evaluated by their current distance and by the angular displacement of their trajectories. Choi et al. (2013) use a Markov Random Field, where the current state estimate is conditioned on the previous one and undirected edges are established between neighbouring subjects, modelling the social forces caused by interactions. However, due to the Markov property, interactions of pedestrians which are not direct neighbours in object space are suppressed. As a consequence, potential correlations between subjects that are further apart are neglected. Yoon et al. (2015) and Yoon et al. (2016) also consider the relative motion between subjects by conditioning the current state estimate on the previous state estimate of the same subject and on those of the nearby subjects. In this way, the motion of different interrelated persons is taken into account, but uncertainty estimates are not considered in the estimation of the relative motion. Furthermore, tracking is applied in the image domain, and the non-linear relative motion between two persons is approximated by a linear model, therefore, the approach must rely on a smooth camera motion. The works previously described apply tracking and motion prediction based on single-frame detections. As a consequence, if people are not detected in a frame, the motion context can be computed based on extrapolations of the trajectories only. Leal-Taixé et al. (2014) circumvent the evaluation of context on an object level and estimate context based on optical flow features. The features are used to train a Random Forest classifier prior to the actual processing, so that the application is restricted to offline applications.

Ellis et al. (2009) applied Gaussian Process (GP)-Bayes Filters to the tracking of pedestrians, where the input data are trajectories of different persons observed in the past. The problem is formulated as a regression task, where velocities are estimated on the basis of the previously observed trajectories. For a predictive model which is representative for a complete scene, a high amount of training data may be required (depending on the complexity of the scene). As the trajectories are required a priori, the application is restricted to offline processing. Kim et al. (2011) apply GP based regression for the prediction of motion trajectories of vehicles. Individual trajectories are assigned to clusters and outliers are detected when the trajectories deviate from a so-called mean flow field. By the explicit association of the trajectories to clusters, possible correlations between trajectories from different clusters are not considered further. Later, the same authors applied GP Regression to detect regions of interest for camera orientation, when acquiring images of a football match, by looking at the means of the regression model, which reflect the expected destinations of the involved subjects (football players, Kim et al., 2012). Here, the motion trajectories are not regarded further and the trajectories of persons are correlated based on the spatial distances between their current positions only. In these works, the input data are the 2D locations of the subjects and the target variables are their velocities. In (Trautman et al., 2015), a robot's path through scenarios crowded by humans is estimated by applying inference in a joint probabilistic model of the robot's and the people's trajectories. The motion model is realised by a mixture of Gaussian Processes, each of which is defined similarly to the way in (Brau et al., 2013), with different covariance functions for every Gaussian Process. Motion context is accounted

for by introducing an interaction potential that considers the Euclidean distance between the robot and the persons, penalising small distances with respect to a Gaussian function.

3.4 Data association

The data association is commonly approached by optimising the assignment of detections to trajectories with respect to an objective function that takes account of the similarity between the detections and the trajectories. In the context of multi-object tracking, generic object detection based on offline classifiers requires the data association step to be solved, whereas detection based on instance-specific classifiers circumvent that step, because every class is associated with an individual target inherently. The data association is commonly guided by measures of similarity in terms of spatial distance between detections and expected target positions and appearance. Appearance is often modelled by features of texture and colour, represented by histograms (Okuma et al., 2004; Andriluka et al., 2008; Schindler et al., 2010), or classifiers (Breitenstein et al., 2011; Wang et al., 2015a; Xiang et al., 2015). In the histogram-based approaches, the similarity is measured by the Bhattacharyya distance between the histograms of a detection and the tracked object. In this way, only the (co)occurrence of features is measured and their geometric alignment is not regarded further. Especially in ambiguous situations, e.g. when tracking players in a football match, it is often not sufficient to represent the appearance only using histograms, as these representations do not take account of the geometric alignment of the features. In most real-world applications, the appearance gradually changes over time, which can be accounted for by adaptive learning of target-specific models of appearance. Wang et al. (2015a) apply distance metric learning to find metrics that assess the similarity of tracklets in the way that the distance metric is only small for tracklets that stem from the same person. Discriminative classifiers based on boosting (Breitenstein et al., 2011) or on Random Forests (Saffari et al., 2009) learn characteristic models of appearance at runtime. Because reference data for the supervision of the training procedure is generally not available, training samples are derived from results of previous time steps. Xiang et al. (2015) train a supervised binary classifier to evaluate the similarity of detections in successive frames using reinforcement learning, by which training is conducted recursively with training sets augmented by wrong assignments of a previous round. Yang and Jia (2016) model the temporal evolution of appearance features by a Hidden Markov Model to anticipate changes in the appearance before classification yields the similarity measures for data association.

Online approaches for tracking need to solve the data association problem in every frame. Available approaches include greedy schemes, as, e.g., used in (Breitenstein et al., 2011; Pirsiavash et al., 2011) and (Shu et al., 2012), which sequentially decide for correspondences, leaving a globally best fitting solution out of consideration. In (Oh et al., 2004; Khan et al., 2005; Benfold and Reid, 2011) and (Brau et al., 2013) sampling-based approaches to the data association problem are given, which are not guaranteed to reach the global optimum. If a weight is assigned to every possible correspondence between detections and trajectories (typically based on cues of position and appearance), a globally optimal solution can be found using the Hungarian algorithm (Kuhn, 1955) or probabilistic approaches such as the joint probabilistic data association (JPDA) strategy proposed by Fortmann et al. (1983).

The Hungarian algorithm solves this combinatorial problem in polynomial time ($\mathcal{O}(n^3)$). This method requires a square matrix with association costs, which can be guaranteed using dummy variables if the number of detections does not equal the number of trajectories. This method is applicable for multi-person tracking (Wu and Nevatia, 2007; Solera et al., 2015), but constraints (e.g. regarding the association in case of an occlusion) cannot be considered without using workarounds. Another solution to the JPDA problem can be found by Linear Programming (Dantzig, 1951), which, despite of its computational complexity, is often applied in the tracking literature, where the number of objects typically only lies in the order of tens, keeping the computation feasible (Storms and Spieksma, 2003; Jiang et al., 2007; Rezatofighi et al., 2015). Linear programming inherently provides a way to incorporate constraints to the optimisation. In recursive estimation frameworks, decisions must be made on a per-frame-level, so that errors made in one step propagate to the final solution and cannot be corrected later. This motivates the use of sophisticated models of motion and appearance, in order to derive optimal similarity measures.

3.5 Discussion

This section summarises the key limitations of the related work with respect to the aspired research goal. For each building block of the general multi-object tracking framework previously discussed, the conclusions and motivations for the advancement of the existing work pursued in the remaining chapters of this dissertation are given.

General approaches to multi-person tracking

The available approaches to multi-person tracking can be categorised into recursive filtering techniques, tracklet-based methods and approaches for offline tracking. Tracklet-based methods lead to a delayed trajectory generation and are, thus, not suitable for time-critical applications such as collision avoidance in driver assistance systems. Approaches for offline tracking require a closed system of images, which is typically not given, e.g. in the context of the continuous generation of visual surveillance footage. This work, thus, sticks to the recursive estimation of motion trajectories in favour of real time capability. In this context, available methods largely model the position of a detected person in the image as observed variable in a recursive Bayesian estimation framework, which leads to inaccurate posterior positions in case of misaligned detections, or perform localisation and tracking in separate steps (Schindler et al., 2010). To improve the geometric accuracy of the generated trajectories, this work models both the state vector of a pedestrian in object space and its position in the image as unknowns and combines the inference of these variables in a Dynamic Bayesian Network.

Observation models

In detection-based approaches for tracking, the image position of a person is typically obtained by classification, where the classifier is trained either offline or at runtime. Offline classification frameworks such as the HOG/SVM (Dalal and Triggs, 2005) enable generic object detection and thus the detection

of persons not seen before. In turn, the recall and precision and the geometric accuracy of such methods is typically not sufficient for many realistic applications (Dollár et al., 2011). Classifiers trained online have the advantage of adapting to a target appearance at runtime, which enables a refined localisation and more reliable data association, but the detection easily is distracted from the target when the training data were derived from misaligned samples. Similar to (Breitenstein et al., 2011), the presented research combines the advantages of both, generic and instance-specific classification, in a single tracking framework. Different from that work, the presented approach trains a Random Forest classifier, which inherently suits for multi-class problems and can be trained incrementally (Saffari et al., 2009). The proposed method uses the update equations of an Extended Kalman filter model instead of particle filter (Breitenstein et al., 2011; Choi et al., 2013), so that the posterior state estimates are described by Gaussian distributions, which allows for self-assessment of the tracking results. To reduce the risk of false positive detections, this work improves the non-maximum suppression strategy by accounting for the geometry of the scene and for prior knowledge about interesting places that is learnt from training data.

Temporal modelling

Most of the cited papers dealing with motion context are either explicit about the grouping of pedestrians, so that possible correlations among members of different groups are ignored, or they limit the range of related objects by the definition of neighbourhoods. Gaussian Process Regression was used successfully for the interpolation of velocities in the related work (Ellis et al., 2009; Kim and Davis, 2011), leveraging the need for parametric models of the regression function. The related work on Gaussian Process Regression in the context of tracking does not include any approach for the tracking of pedestrians in which the interactions, which are relevant for the interpolation of velocities, are computed together with the trajectories at runtime. This work follows the related work that models the velocities as target variables in Gaussian Process Regression models. Different from the existing approaches, the correlations of trajectories are evaluated at runtime, taking into account the spatial distance of the current positions as well as the directions of motion in a common object coordinate system. This work exploits the strengths of Gaussian Process Regression for the temporal modelling of interacting pedestrian states, and integrates the new temporal model into the Dynamic Bayesian Network. Like Pellegrini et al. (2009) and Leal-Taixé et al. (2011), the new method models the state parameters in 3D, which allows for a geometric interpretation of the estimated positions in SI units (metres and seconds) in favour of a joint motion model for all pedestrians. This work develops a new probabilistic approach for modelling interactions without the need for explicit grouping. The new model can be integrated into the probabilistic framework in order to derive optimal predictions of future states and to anticipate tracking errors such as identity switches.

Data association

Methods for the optimal assignment of detections to trajectories have been identified by the related work, but the available approaches for the definition of similarity measures leave scope for improvements, because the modelling of appearance and target dynamics in multi-person scenarios is often

approached by simplistic models such as histograms (Schindler et al., 2010) and constant-velocity motion models (Breitenstein et al., 2011). Using these models, the risk of committing tracking errors, such as identity switches, is high, especially if the constant velocity assumption is not fulfilled and the appearance of some persons is similar. The presented work defines measures of similarity, using models of appearance and motion, that account for the multitude of persons present in a scene, computing measures of similarity based on class-conditional probabilities of an Online Random Forest classifier in combination with geometric cues measured by a Mahalanobis distance between a predicted position and a detection. In this way, the number of tracking errors, such as identity switches, is expected to be reduced compared to methods using simpler models to express the similarity. Following Storms and Spieksma (2003), the optimisation is carried out using linear programming for its ability of finding the globally best fitting solution in every frame.

4 A new probabilistic approach for multi-person localisation and tracking

This chapter introduces a new probabilistic approach for the tracking of multiple persons from monocular image sequences in a recursive fashion and for their exact localisation in the image and object space. Motivated by the research goal of this dissertation and by the available literature, the trajectory generation is carried out by recursive Bayesian estimation in a Dynamic Bayesian Network (DBN), which models the state vectors of the tracked persons and their positions in the images using hidden variables. This is in contrast to the commonly practised tracking-by-detection approach, which first determines the image position by detection and then updates a recursive filter using these positions. By modelling the image positions of the persons using hidden variables, the update step of the recursive filter is carried out with a geometrically improved image position, which is expected to lead to a *more accurate* posterior position in state space. The temporal model of the DBN considers the motion of all pedestrians via a new model of interactions, referred to as *Implicit Motion Context*. It is expected that an improved prediction lowers the risk of false positive detections and of identity switches, which leads to a *more reliable* assignment of detections to trajectories and of training samples for the online classifier.

Several observations contribute to the determination of the state and image position of a person. These observations are derived from prior knowledge about the observed scene, generic pedestrian detection and a new model for instance-specific classification of multiple persons. To be aware of the presence of multiple persons, this work further describes a new model for the similarity measures in a joint probabilistic data association strategy and for the handling of mutual occlusions.

The core of the new method is a recursive Bayesian estimation framework, which predicts and updates a pedestrian state using a temporal model and various observations at every time step. The general work flow is depicted in Figure 4.1. At every epoch, a set of trajectories is given, where each trajectory is assigned to one person being tracked. Starting from a prior estimate of the pedestrian positions and given one image from an image sequence at a time, the current positions are determined using an instance-specific classification near the prior position and also by generic pedestrian detection. Data association is applied based on the prior position and the instance-specific classifier. Furthermore, the system estimates the degree to which each person is occluded by other persons based on the predicted positions of the persons. Prior information about interesting places in the scene is modelled as observation and is either valid for the entire scene, if the camera orientation is static, or is modelled in every frame, if the camera orientation changes over time. These observations are incorporated into the update step of the recursive filter, from which the posterior state is derived. From the posterior state, the corrected image position of the tracked person is found by back-projection of the posterior

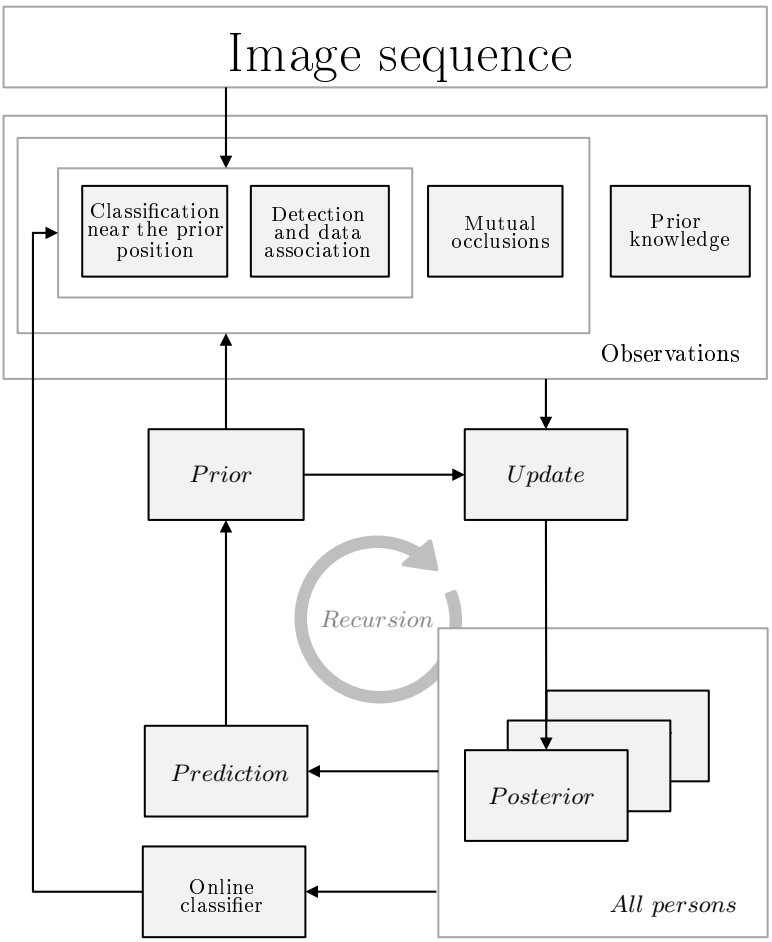


Figure 4.1: Work flow of the proposed method for multi-person localisation and tracking. The core of this method is a recursive estimation framework, which predicts and updates the state of all tracked persons at every point in time. For the update, four different observations are made: The *classification near the prior position* of each person and the *detection and data association*, which are obtained from every image, *mutual occlusions* and *prior knowledge* about the scene. The classification and data association depend on the online classifier, which is updated based on the posterior positions of all persons. The prior position is required for classification, data association and the estimation of mutual occlusions, and is predicted based on the posterior states of all persons.

state to the image. From the corrected image positions of all persons, new training samples are derived in order to update the instance-specific classifier. The prior position of each person in the next image is computed by the predictive model, which takes account of the posterior states of all tracked persons.

The remainder of this chapter is organised as follows. In Section 4.1, the Dynamic Bayesian Network used as recursive estimation framework is described. In Section 4.2, the single observations and their probability densities are described. Section 4.3 explains the proposed temporal model of the recursive estimation framework. This includes the new strategy for modelling motion context. In Section 4.4, the new strategy for data association is presented. The inference strategy for the determination of the hidden state variables and the hidden image position is described in Section 4.5. Finally, Section 4.6 closes this chapter with a discussion of the expected strengths and weaknesses of the proposed method and of the free parameters of the model.

4.1 Problem statement via Dynamic Bayesian Networks

This work aims at the joint estimation of pedestrian states in 3D object space and their exact locations in the images. In terms of Bayesian inference, an optimal reasoning about the unknown variables can only be achieved by modelling these variables in a joint probabilistic model. After describing the functional relationships between the desired variables, this section describes the Dynamic Bayesian Network for the joint inference of the pedestrians' states and their positions in the images.

To leverage the effect of perspective distortions of the trajectories and to model interactions between persons, it is essential to obtain viewpoint-independent results for the positions of pedestrians. To this end, tracking is carried out in a common 3D object coordinate system, see Figure 4.2. The coordinate system is centred at the projection centre of the camera (at time t_0 in case of a moving platform) with the X and Z axes pointing in horizontal directions and Y in the vertical downward direction (right-handed system). To enable the conversion of 2D image coordinates to 3D world coordinates from monocular image sequences, the terrain is assumed to be horizontal and the position of the pedestrians is projected onto the ground plane π_t, which lies at a distance Y_π below the camera. For each person i, a six-dimensional state vector $\mathbf{w}_{i,t} = [X_{i,t}, Y_{i,t}, Z_{i,t}, H_{i,t}, v_{X,i,t}, v_{Z,i,t}]^\top$ that consists of the 3D position $\mathbf{X}_{i,t}=[X_{i,t}, Y_{i,t}, Z_{i,t}]^\top$, the body height $H_{i,t}$ and the velocity components $v_{X,i,t}$ and $v_{Z,i,t}$ in the directions X and Z parallel to the ground plane, is modelled at each time step t. For ease of readability the indices t and i are not specified wherever they are obvious. The image position of each person is described by its position of the feet $\mathbf{x}^F=[x^F, y^F]^\top$, referred to as the reference point of the person, and the position of its head $\mathbf{x}^H=[x^H, y^H]^\top$. These two points jointly define a rectangle $\mathbf{r}_{i,t} = [x^F, y^F, x^H, y^H]^\top$ surrounding the person in the image, where the height of the rectangle is the vertical distance between the reference point and the position of the head, and the width of the rectangle is half of its height.

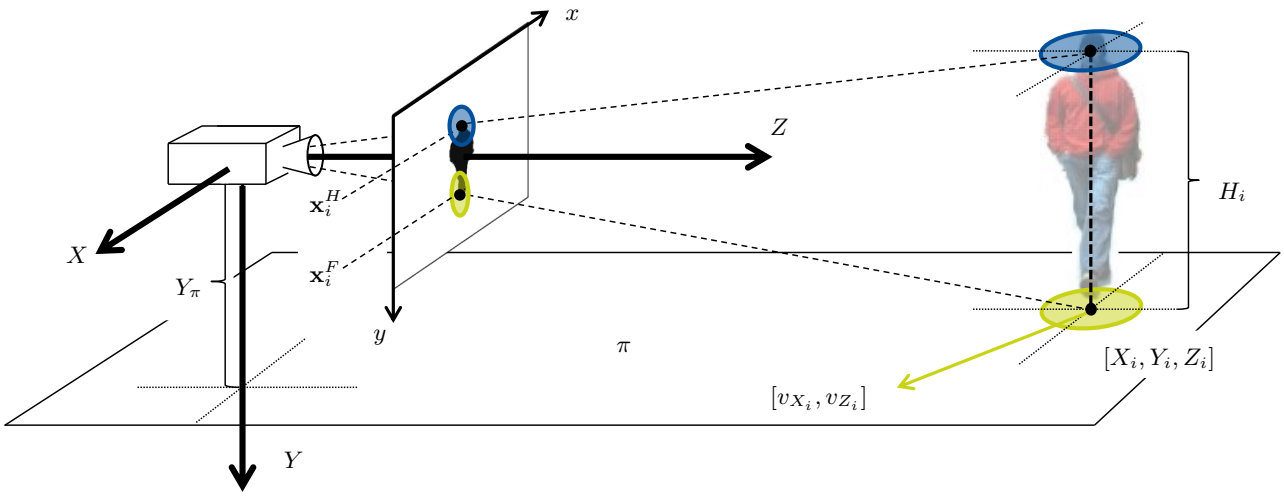


Figure 4.2: Illustration of the functional relationships between image-positions $\mathbf{x}^F=[x^F, y^F]^\top$ and $\mathbf{x}^H=[x^H, y^H]^\top$ and state parameters. Furthermore, the ground plane π and the object coordinate system are depicted (for the first frame, in cases where the camera orientation is dynamic). The ellipses illustrate the confidences about the variables. The surrounding rectangle $\mathbf{r}$ is not shown for clarity.

The functional relationship between image and world coordinates is given by the collinearity equations:

$$x^F = x_o - c \cdot \frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \tag{4.1}$$

$$y^F = y_o - c \cdot \frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \tag{4.2}$$

$$x^H = x_o - c \cdot \frac{r_{11}(X - X_0) + r_{12}((Y - H) - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}((Y - H) - Y_0) + r_{33}(Z - Z_0)} \tag{4.3}$$

$$y^H = y_o - c \cdot \frac{r_{21}(X - X_0) + r_{22}((Y - H) - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}((Y - H) - Y_0) + r_{33}(Z - Z_0)} \tag{4.4}$$

where x_0 and y_0 are the coordinates of the principal point and c is the focal length of the camera, r_{ij} are the elements of the rotation matrix R_t between image and reference frame, and $[X_0, Y_0, Z_0]^\top$ is the perspective centre of the camera. Equations 4.1-4.4 define the measurement model used in the recursive filter of this method. An additional fictitious observation

$$Y_\pi = Y \tag{4.5}$$

takes account of the assumption that pedestrians stand on the ground plane, i.e., that the Y component of the pedestrian's location corresponds to the distance of the ground plane from the camera, and that the terrain is horizontal. The condition that persons stand on the ground plane basically enables the conversion from 2D image coordinates to 3D object coordinates.

To derive optimal decisions about the state variables and the image positions of pedestrians, the DBN models the state vector $\mathbf{w}_{i,t}$, the rectangle $\mathbf{r}_{i,t} = [\mathbf{x}^F_{i,t}, \mathbf{x}^H_{i,t}]^\top$, as well as the reference point $\mathbf{x}^F_{i,t}$ of each person in the image as unknown variables, see Figure 4.3. An additional hidden variable $n_{i,t}$ accounts for uncertainties in the correctness of the image position $\mathbf{r}_{i,t}$ due to mutual occlusions $o_{i,t}$ and prior knowledge about interesting places in the scene IP, which are modelled as observables. The position of the reference point of a person is observed via generic person detection and instance specific classification, and the position of the head is only observed via generic person detection. The detector confidence $d_{i,t}$, the classifier confidence $c_{i,t}$, the image position of the head $\mathbf{x}^H_{i,t}$, observed via the person detector, and the distance of the ground plane from the camera Y_π are modelled as observables. The camera orientation and calibration C_t are introduced as constants. One such model is defined for every pedestrian being tracked. Interactions with other persons are accounted for by the temporal model, defined in Section 4.3.

The aim is to find the parameters of the unknown variables that maximise the joint probability of all variables at every time step, given the observed and fixed values, so that

$$p\left(\mathbf{w}_{j=1\ldots n,t-1}, \mathbf{w}_{i,t}, \mathbf{r}_{i,t}, n_{i,t}, \mathbf{x}^F_{i,t}, \mathbf{x}^H_{i,t}, d_{i,t}, c_{i,t}, C_t, Y_\pi, o_{i,t}, IP_t\right) \rightarrow max \tag{4.6}$$

The parameters of the camera orientation C_t and the prior knowledge about the scene IP_t are defined in image space and are valid for an entire image sequence, if the camera is static, or for every image, if the camera is dynamic, as denoted by the subscript t indicating the time step. All other variables

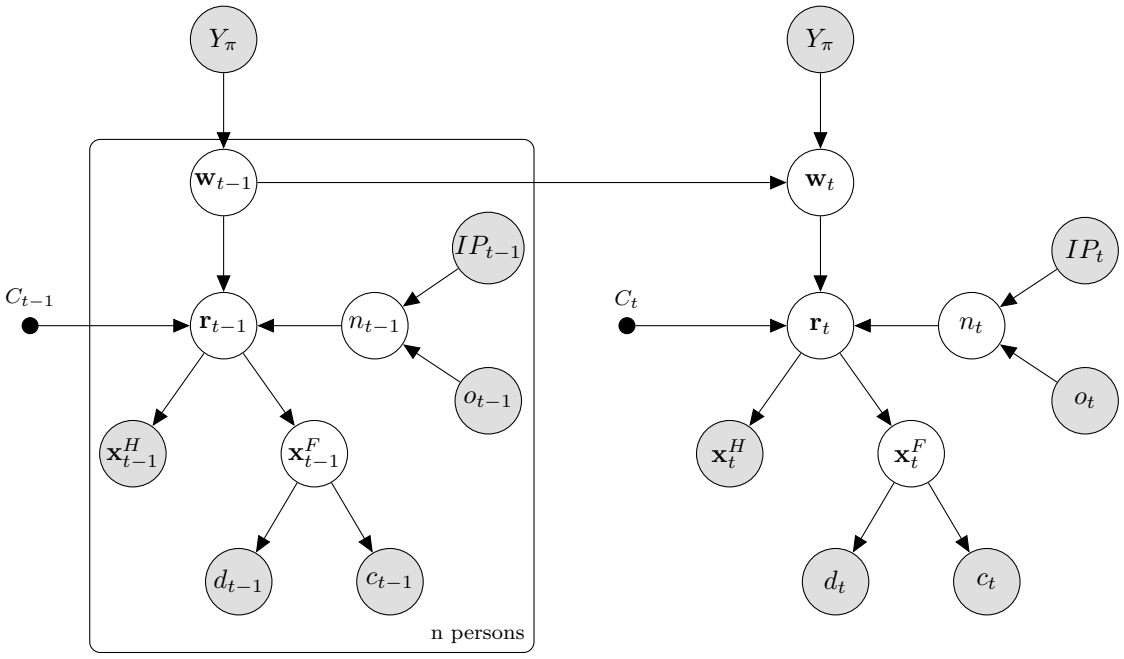


Figure 4.3: Dynamic Bayesian Network for multi-person tracking, shown for two successive time steps $t-1$ and t. The variables $\mathbf{w}$ (state vector), $\mathbf{r}$ (rectangle around a person), $\mathbf{x}^F$ (position of the feet) and σ_n (uncertainty about the position of the rectangle) are modelled by hidden variables. The variables IP, o, $\mathbf{x}^H$, d, c and Y_π are observed. The camera parameters C are constants. The state vector at time t depends on the state vectors of all n persons tracked at time $t-1$. See text for further details.

are defined individually for single persons or for all persons, as indicated by the subscripts i and j, respectively. According to the network structure of the DBN, the joint pdf of the involved variables can be factorised as follows:

$$\begin{aligned} & p\left(\mathbf{w}_{j=1\ldots n,t-1}, \mathbf{w}_{i,t}, \mathbf{r}_{i,t}, n_{i,t}, \mathbf{x}^F_{i,t}, \mathbf{x}^H_{i,t}, d_{i,t}, c_{i,t}, C_t, Y_\pi, o_{i,t}, IP_t\right) \\ \sim & p\left(\mathbf{w}_{i,t}|\mathbf{w}_{j=1\ldots n,t-1}\right) \cdot p\left(\mathbf{w}_{i,t}|Y_\pi\right) \cdot p\left(\mathbf{r}_{i,t}|\mathbf{w}_{i,t}, n_{i,t}, C_t\right) \cdot p\left(Y_\pi\right) \cdot p\left(\mathbf{x}^F_{i,t}|\mathbf{r}_{i,t}\right) \\ & \cdot p\left(\mathbf{x}^H_{i,t}|\mathbf{r}_{i,t}\right) \cdot p\left(n_{i,t}|o_{i,t}, IP_t\right) \cdot p\left(d_{i,t}|\mathbf{x}^F_{i,t}\right) \cdot p\left(c_{i,t}|\mathbf{x}^F_{i,t}\right) \cdot p\left(IP_t\right) \cdot p\left(o_{i,t}\right), \end{aligned} \tag{4.7}$$

where $\mathbf{w}_{j=1\ldots n,t-1}$ denotes the set of state vectors of all n persons being tracked in epoch $t-1$. In accordance with the network structure and with Equation 4.7 the joint probability can be written as the product of eleven probability distributions, which are explained in Table 4.1. The state vector of every person is conditioned on the previous state vector of the same person and on the state vectors of all other persons, as well as on the parameter of the ground plane. In this way, the state vector represents a first-order Markov-process, so that it temporally depends only on the preceding time step. The image position of the head and the reference point of the feet are conditioned on the parameters of the rectangle $\mathbf{r}$, which depends on the state vector, the camera orientation and the additional uncertainty about the position due to occlusions and prior knowledge about interesting places. The detector and classifier confidences are dependent on the reference point. To find a solution to the maximisation problem stated by Equation 4.6, message passing in form of Belief Propagation is applied as described in Section 4.5.

pdf	Description
$p\left(\mathbf{w}_{i,t}\middle\vert\mathbf{w}_{j=1\ldots n,t-1}\right)$	Probability that $\mathbf{w}_{i,t}$ is the state of person i given the states of all persons being tracked in the previous epoch
$p\left(\mathbf{w}_{i,t}\middle\vert Y_\pi\right)$	Probability that $\mathbf{w}_{i,t}$ is the state of person i given the parameter of the ground plane
$p\left(\mathbf{r}_{i,t}\middle\vert\mathbf{w}_{i,t}, n_{i,t}, C_t\right)$	Probability that $\mathbf{r}_{i,t}$ is the rectangle surrounding person i given that $\mathbf{w}_{i,t}$ is the state vector, variable $n_{i,t}$, and the orientation of the camera
$p(Y_\pi)$	Probability that Y_π is the distance of the ground plane from the camera
$p\left(\mathbf{x}_{i,t}^F\middle\vert\mathbf{r}_{i,t}\right)$	Probability that $\mathbf{x}_{i,t}^F$ is the reference point given that $\mathbf{r}_{i,t}$ is the surrounding rectangle
$p\left(\mathbf{x}_{i,t}^H\middle\vert\mathbf{r}_{i,t}\right)$	Probability that $\mathbf{x}_{i,t}^H$ is the image position of the head given that $\mathbf{r}_{i,t}$ is the surrounding rectangle
$p\left(n_{i,t}\middle\vert o_{i,t}, IP_t\right)$	Probability that $n_{i,t}$ is the uncertainty about the image position given the occlusion of person i and the prior knowledge about the scene
$p\left(d_{i,t}\middle\vert\mathbf{x}_{i,t}^F\right)$	Probability that the detector delivers $d_{i,t}$ given that $\mathbf{x}_{i,t}^F$ is the position of the reference point
$p\left(c_{i,t}\middle\vert\mathbf{x}_{i,t}^F\right)$	Probability that the classifier delivers $c_{i,t}$ given that $\mathbf{x}_{i,t}^F$ is the position of the reference point
$p(IP_t)$	Prior probability that a person is observed in the image
$p(o_{i,t})$	Probability that person i is occluded

Table 4.1: Probability distributions according to the factorisation represented by the DBN.

4.2 Observations

The observations are derived from three different sources of information, namely information valid for the entire scene (scene-specific knowledge), information valid for a single image that indicates the presence of *any* pedestrian (category-specific), and information valid for a *specific* person (instance-specific). The different observations are illustrated in Figure 4.4.

These observations complement each other in the determination of the desired state parameters:

- The scene-specific information describes regions in the scene that are likely to be passed by persons and thereby helps to suppress automatic pedestrian detections that are unlikely to be caused by persons. It is modelled in different ways for image sequences captured by static cameras and sequences captured by moving cameras.
- The category-specific information, which is given by the outcome of a generic person detector, highlights all regions in an image that are similar to a generic object model characteristic for persons. This information essentially contains information about the desired locations of persons that need to be incorporated into the filtering framework and is used to automatically initialise new trajectories. As the detector is designed to generalise over the entire class *pedestrian*, the resulting detections need to be associated to single instances of the pedestrian class based on additional information.
- The instance-specific information is based on a supervised classifier with one class for every

(a) Scene-specific knowledge indicating feasible regions for the occurrence of persons.

(b) Category-specific information about the presence of persons w.r.t. their heads.

(c) Category-specific information about the presence of persons w.r.t. their feet.

(d) Instance-specific knowledge about persons, here: localisation of the foremost person.

Figure 4.4: Illustration of the image-based observations and prior knowledge about the scene. Warmer colours represent higher confidences about about possible areas of occurrence (a) and actual presences of a person (b)–(d).

person that is tracked, and it is integrated into the proposed method in two ways: Firstly, the classifier computes similarity measures for the data association of detections to existing trajectories, and secondly, by classification in a sliding-window-based fashion, the classifier gives rise to an additional observation of the image position of individual pedestrians.

To derive geometrically accurate measurements of a person's location in an image is a challenging task due to the articulated motion of pedestrians and potential occlusions. The position of the head can typically be detected more reliably than the position of the feet, because the head undergoes less articulated motion and, thus, its observed projection in the image does not vary as much as that of the feet. Therefore, the position of the head is modelled as an observable variable, whereas the intersection point of the person with the ground plane is modelled as a hidden variable. Two complementary observations are related to the image-position of the feet: The confidence of a generic person detector about the presence of a person and the confidence of an instance-specific classifier about the presence of an individual person. The pdfs of the observations are depicted in Figure 4.4. The generation of these pdfs is described in the remainder of this section: The scene-specific information is explained in Section 4.2.1, the pedestrian detection is explained in Section 4.2.2, and the instance-specific classifier is described in Section 4.2.3.

4.2.1 Prior knowledge about the scene

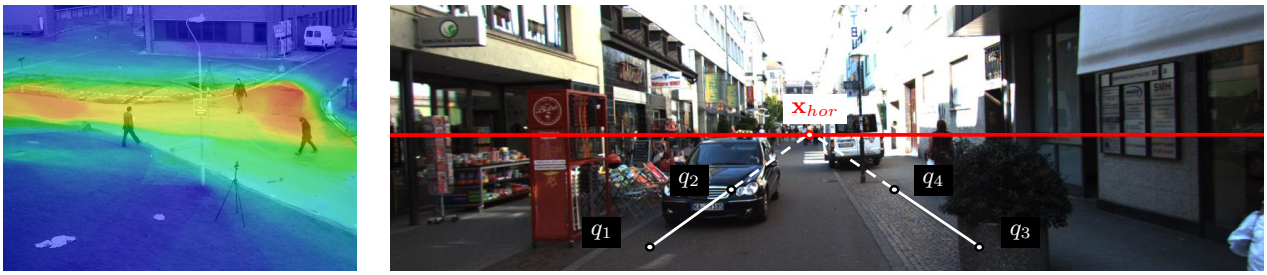
Prior knowledge about a scene is expected to help with the reduction of false positive detections, which are inherent in detection-based approaches for tracking. With respect to the potential application scenarios for this method, two different camera set-ups must be considered: Static cameras with constant exterior orientation (w.r.t. six degrees of freedom), and cameras mounted on moving platforms with variable exterior orientation. The first scenario can be found in surveillance applications, where the viewing angle is typically inclined towards the ground plane and the same area is observed all the time. As the orientation does not change, information about previous occurrences at certain image positions can be regarded as prior knowledge about future occurrences. The second scenario involves dynamic cameras, as given in the context of driver assistance systems and field robotics. To reduce the risk

of false positive detections in these scenarios, the position of the horizon in the image is modelled, so that detections can be validated by testing if the reference point of a detection lies below the horizon.

Static cameras. For the static camera set-up, prior knowledge about the scene is acquired from the distribution of frequently visited places in a supervised approach. To this end, a binary Random Forest classifier is trained with the image coordinates of the reference point as features and class assignments according to true and false positive detections obtained by a HOG/SVM detector (Dalal and Triggs, 2005) in a supervised training strategy. The training samples are split into positive and negative samples by validation with reference data, using an intersection-over-union score (IOU) threshold of 50%. By convention, the IOU of two rectangles $\mathbf{r}_1$ and $\mathbf{r}_2$ is computed as the ratio of the intersection area of these rectangles and the area of their union area, cf. Equation 4.8.

$$IOU\left(\mathbf{r}_1, \mathbf{r}_2\right) = \frac{area\left(\mathbf{r}_1 \cap \mathbf{r}_2\right)}{area\left(\mathbf{r}_1 \cup \mathbf{r}_2\right)} \tag{4.8}$$

For the assignment of probabilities to the image, every pixel is classified by the Random Forest using the image coordinates as features. Classification then delivers the probability for an image position to be occupied by a person. The probability distribution over IP is denoted $p(IP)$. If no training data are available for a specific scene, this pdf is set to be the uniform distribution. The distribution of frequently visited places learnt from a training sequence is shown in Figure 4.5(a), where reddish values indicate high, and blueish values low probabilities for the occurrence of persons.



(a) Static camera set-up: Interesting places learnt from pedestrian occurrences in a training stage.

(b) Dynamic camera set-up: Interesting places are defined as the region below the horizon, which is indicated by the red line. See text for details.

Figure 4.5: Prior knowledge about scenes with different camera set-ups.

Moving cameras. If the camera orientation changes, no constant values can be assigned to the pdf over the variable IP. In this context, the image row coordinate y_{hor} of the horizon is computed in every image, and every image position (x, y) is assigned a probability of the occurrence of pedestrians using the following rule:

$$p(IP) = \begin{cases} 1, & \text{if } y > y_{hor} \\ 0, & \text{if } y \leq y_{hor} \end{cases}. \tag{4.9}$$

The horizon is computed as the vanishing line of the ground plane, i.e., the line, on which two parallel lines on the ground plane intersect (Hartley and Zisserman, 2000). In practice, the vanishing line in the image is approximated as the horizontal line through the image point $\mathbf{x}_{hor}$, which is computed as the

intersection point of two parallel lines $\overline{Q_1Q_2}$ and $\overline{Q_3Q_4}$ on the ground plane, whose image projections $\overline{q_1q_2}$ and $\overline{q_3q_4}$ are depicted in Figure 4.5(b). When the roll angle of the camera is zero, as expected in the addressed scenarios, the estimated line corresponds to the real horizon.

4.2.2 Generic person detection

In this work, generic person detection is applied based on the sliding-window approach by Dalal and Triggs (2005) to find evidence about the presence of any pedestrian in an image. Therefore, an implementation of the HOG/SVM detector that is trained on the INRIA person dataset[1] is used. The results of detection are typically used as measurements for the update step of a recursive filter, which is done in the related work based on two assumptions: The first assumption considers the detection as representative for the object position in the image; the second assumption is that all measurements are equally accurate. In this work, it is argued that both assumptions are prone to be wrong. Firstly, a detection is easily misaligned due to mutual occlusions, articulated motion and other disturbing effects. In the presence of such effects, a detection cannot be relied on as an indicator of the object position. This is accounted for by modelling the reference point of a person as a hidden variable in the Dynamic Bayesian Network. Secondly, the accuracy of a measured position depends on the presence of these disturbing effects and assigning the same uncertainty to all detections ignores these effects. The different accuracies of person detections are accounted for by modelling the probability densities of the detector by kernel density estimation (kde). These steps are described in the remainder of this section in further detail.

As discussed in the related work chapter, sliding-window approaches typically deliver many positive detections near the true location of an object in the scale-space-representation of the image. To detect persons at different scales, the detection window is increased successively by a factor ρ_{det}. These detections are usually grouped to yield ideally one detection per object, which is also referred to as non-maximum suppression. To account for possible false positive detections, this work applies an additional validation step before the single-scale detections are grouped.

False positive reduction. To reduce the number of false-positive detection, a validation step is executed prior to the non-maximum suppression. The general work flow for the proposed strategy of false positive reduction and non-maximum suppression is illustrated in Figure 4.6. Given an input image (Figure 4.6(a)), primary detections are obtained by application of the HOG/SVM (cf. Figure 4.6(b)), which are then separated into validated and falsified detections (cf. Figure 4.6(c)) and grouped (cf. Figure 4.6(d)). The validation is carried out based on a comparison of the detections with an expected body height in 3D space, and the developed strategy is referred to as *3D false positive reduction* (3DFPR).

To find an optimal decision about the validity or invalidity of a detection, hypotheses tests are applied. Each detection is associated with the size and position of the rectangular detection window, defined as $\mathbf{r}_d = [x_d, y_d, w_d, h_d]^\top$, where ($x_d$,$y_d$) is the bottom centre point of the detection window, which is referred to as the reference point $\mathbf{x}_d$ of a detection. w_d and h_d are the width and height of the

[1] http://pascal.inrialpes.fr/data/human/ (accessed on September 2016)

detection window, respectively. The reference point of the detection is related to a unique position in space, which can be computed via the inverse collinearity equations with a fixed height component, i.e. the Y coordinate of the ground plane. The ratio $\frac{dst}{c}$ of the horizontal distance dst between the detection and the camera over the focal length c of the camera yields a scale factor s by which the observed height h_d of a detection is transformed to a metric scale:

$$H_d = h_d \cdot s. \tag{4.10}$$

Assuming a fixed standard deviation σ_d of a height in the image, the standard deviation σ_D of the corresponding height in metres is found using the scale factor s:

$$\sigma_D = \sigma_d \cdot s. \tag{4.11}$$

The height of a detection is represented by a normal distribution: $H_D \sim \mathcal{N}\left(\mu_D, \sigma_D^2\right)$. It is further assumed that the body height of all persons is normally distributed via $H_P \sim \mathcal{N}\left(\mu_P, \sigma_P^2\right)$. Every detection is evaluated using a statistical test of differences between two mean values of the form:

$$H_0 : \mu_D = \mu_P \Leftrightarrow \mu_D - \mu_P = 0, \tag{4.12}$$

$$H_A : \mu_D - \mu_P \neq 0. \tag{4.13}$$

The null-hypothesis H_0 states that the means of the detector's body height and the expected body height are equal, whereas the alternative hypothesis H_A assumes that they are different. The test statistic y_d is defined as:

$$y_d = \frac{|\mu_D - \mu_P|}{\sqrt{\sigma_D^2 + \sigma_P^2}}. \tag{4.14}$$

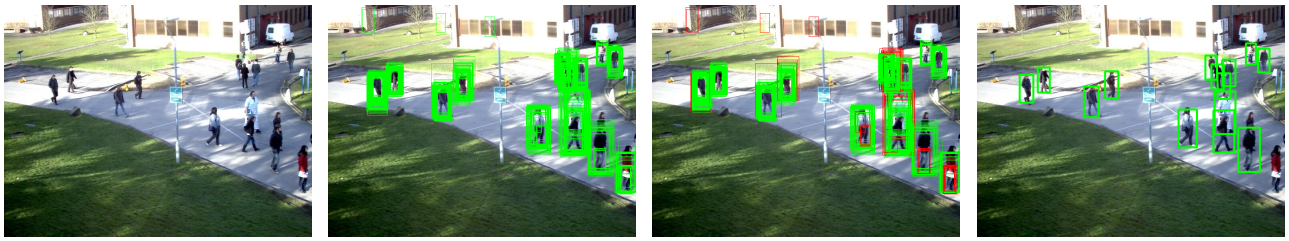
The limits $\pm y_{1-\frac{\alpha}{2}}$ of the acceptance area for the null-hypothesis are given by the cumulative distribution function of the standard normal distribution. The actual test is carried out under the following rules:

$$y_d \leq y_{1-\alpha} \quad \Rightarrow \quad \text{accept } H_0, \tag{4.15}$$

$$y_d > y_{1-\alpha} \quad \Rightarrow \quad \text{accept } H_A. \tag{4.16}$$

Note that, different from most of the related work that applies detection and tracking in the 2D image domain, the method takes advantage of the 3D geometry of the scene that is supposed to be known, and it is assumed that the number of false positives can be reduced in this way, as only detections with a plausible height survive.

Non-maximum suppression. The validated detections are grouped into disjoint sets of detections based on an equivalence criteria in terms of size and relative overlap based on an implementation available in OpenCV (Bradski and Kaehler, 2008). An equivalence threshold $\delta_{\mathbf{d}_1\mathbf{d}_2}$ of two detections



(a) Input image to be classified by the detector. (b) Primary detections achieved in the image scale-space. (c) Detections validated by the false positive reduction step. (d) Validated and grouped detections after non-maximum suppression.

Figure 4.6: Detection, false positive reduction and non-maximum suppression. In (c), green rectangles indicate accepted detections, red rectangles detections that are rejected.

$\mathbf{d}_1 = [x_1, y_1, w_1, h_1]^\top$ and $\mathbf{d}_2 = [x_2, y_2, w_2, h_2]^\top$ is determined using Equation 4.17,

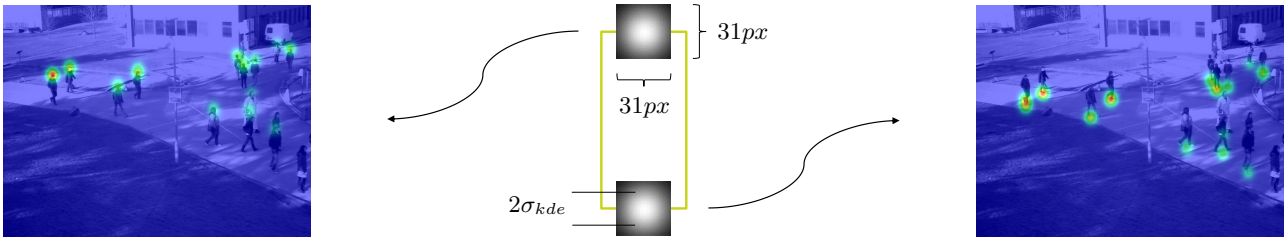
$$\delta_{\mathbf{d}_1\mathbf{d}_2} = \epsilon_{nms} \cdot \frac{min(w_1, w_2) + min(h_1, h_2)}{2}, \tag{4.17}$$

where the function $min(w_1, w_2)$ returns the width of the smaller rectangle and $min(h_1, h_2)$ its height, and ϵ_{nms} is a free parameter. Using smaller values of ϵ_{nms}, it is more likely that two detections are associated to different groups. Using a value of $\epsilon_{nms} = 0$, no grouping is applied at all. Two rectangles are assigned to the same group if all of the following conditions hold:

$$\begin{aligned} |x_1 - x_2| &\leq \delta_{\mathbf{d}_1\mathbf{d}_2} \\ |y_1 - y_2| &\leq \delta_{\mathbf{d}_1\mathbf{d}_2} \\ |(x_1 + w_1) - (x_2 + w_2)| &\leq \delta_{\mathbf{d}_1\mathbf{d}_2} \\ |(y_1 + h_1) - (y_2 + h_2)| &\leq \delta_{\mathbf{d}_1\mathbf{d}_2}. \end{aligned} \tag{4.18}$$

If less than a pre-defined number η_{nms} of single-scale detections is associated with a group, the group is discarded entirely. Otherwise, a final detection is generated from each group by computing the average position and size of the detections associated to that group. Each final detection is assigned a probability density, which is estimated by kernel density estimation based on the associated single-scale detections, as described in the following paragraph.

Detector confidences. To estimate the probability densities $p(\mathbf{x}_{i,t}^H|\mathbf{r}_{i,t})$ and $p(d_{i,t}|\mathbf{x}_{i,t}^F)$, the detector confidences of the generic person detector about the presence of persons in an image are modelled to be proportional to the number of the single-scale detections found by the sliding-window person detector that were validated in the non-maximum suppression step. Each single-scale detection is assumed to be found with a positional accuracy of σ_{kde}, which is assumed to be equal for the standard deviations in the image coordinates x and y, respectively. The distribution of the detector confidences is computed from the single-scale detections using a kernel density estimator with a constant Gaussian kernel with σ_{kde}, centred at every top centre position (to vote for the head) and bottom centre position (to vote for the feet) of all rectangles associated with the detection by the non-maximum suppression step.



(a) $p(\mathbf{x}_{i,t}^H|\mathbf{r}_{i,t})\forall i \in [\mathbf{T}]$ (b) Gaussian kernels of $31 \times 31px$ relative to a single-scale detection, illustrated by the green rectangle. (c) $p(d_{i,t}|\mathbf{x}_{i,t}^F)\forall i \in [\mathbf{T}]$

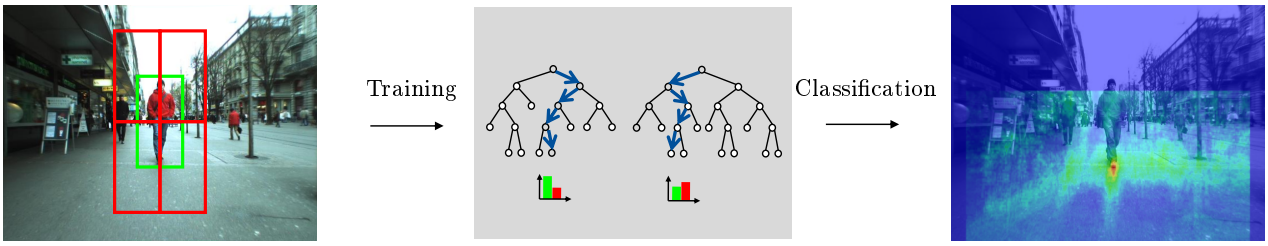
Figure 4.7: Estimation of the probability densities $p(\mathbf{x}_{i,t}^H|\mathbf{r}_{i,t})$ and $p(d_{i,t}|\mathbf{x}_{i,t}^F)$. Here, $[\mathbf{T}]$ is the index set of all existing and potentially new trajectories.

In Figure 4.7, the probability densities of all persons detected in an exemplary image are depicted. Every local maximum of the detector confidence corresponds to a grouped detection. To disambiguate these detections, a data association step must be applied, so that every detection is either associated with a person that is already tracked, or to a new trajectory. The final set of m grouped rectangles is denoted $\mathbf{D} = \{\mathbf{d}_1, ..., \mathbf{d}_k, ..., \mathbf{d}_m\}$, with $\mathbf{d}_k = [x_k, y_k, w_k, h_k]^\top$, and is considered as the input to the data association strategy of this method described in Section 4.4.

4.2.3 Instance-specific classification

In this section, the state-of-the-art in multi-person tracking-by-detection is extended by formulating a new classification approach capable of representing a variable set of tracked persons by a single classifier. This method accounts for the appearance of the pedestrians by training a Random Forest classifier based on the algorithm of Saffari et al. (2009), who made the classifier capable of being trained incrementally so that new training samples can be incorporated at runtime (cf. section 3.2). Despite of the multi-class nature of Random Forests, Saffari et al. (2009) only applied the classifier to single-class problems in the context of object tracking. Using a multi-class classifier in an object tracking application with a variable number of objects causes the necessity to change the number of classes at runtime and, thus, the need to update the class-statistics. In this chapter, a strategy for the application of the Online Random Forest (ORF) classifier to the multi-person tracking domain is described. In this application, every tracked person as well as the background is represented by a single class in the ORF.

Multi-person classification with variable number of classes. In this section, the multi-person classifier, that learns the appearance of multiple tracked persons, is described. Because the number of persons typically varies and training data are generally not available for specific persons, a new strategy for the extraction of training samples is required. To this end, the strategy applied here takes advantage of the recursive estimation results and generates training samples from the inferred pedestrian positions $\hat{\mathbf{r}}_{i,t}$ in the image, which are estimated in the way described in Section 4.5. Because training samples are rare, further positive training samples are taken from positions shifted by one pixel up, down, left and right from the reference point of $\hat{\mathbf{r}}_{i,t}$. Samples for the background class are taken from positions translated by half of the size of $\hat{\mathbf{r}}_{i,t}$ in the same four directions, as shown by the



(a) Training regions for negative samples (red rectangles) and a positive sample (green rectangle)

(b) Online Random Forest shown symbolically for two trees. Every leaf node in the trees stores the class labels of the training samples.

(c) Classifier confidence found by classification of every feasible position in the image.

Figure 4.8: Illustration of the classification model.

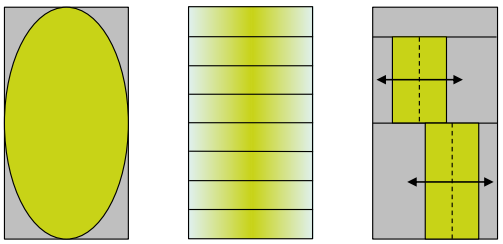
green and red rectangles indicating the training regions in Figure 4.8(a).

The appearance of pedestrians in an image sequence varies due to changing lighting conditions, viewing directions, and other effects. To leverage these effects, the adaptive learning strategy of Saffari et al. (2009) is applied. Because people may enter or leave a scene, the number of classes changes over time and the classifier needs to be re-initialised. Because the training samples are rare and in order to have an equal number of training samples for each class, the most recent samples obtained for every pedestrian and for the background are stored for a number of η_{que} time steps, where η_{que} is a free parameter. When the classifier is re-initialised, the Random Forest is trained using these samples, as illustrated in Figure 4.8(b). If new samples arrive, the oldest samples are discarded. In this way and by using the strategy for online training, the classifier adapts to the possibly changing appearance of a person over time.

Feature extraction. The way in which features are extracted from the training regions is described next. Because persons perform articulated movements, the actual area within a training region covered by a pedestrian varies. To extract features that only represent the person and not the background, the training regions are divided into regions from which features are extracted and background. Three different strategies for the subdivision of the training regions are investigated. To obtain feature representations of equal dimension for every sample, the training regions are normalised to a constant height of 48 pixels and width of 24 pixels. Figure 4.9 illustrates the proposed models.

The first model (*Ellipse*, ELL) takes all pixel values from an elliptic region defined on the basis of the training region. This way of dividing the training region into foreground and background is often found in the tracking literature, as, e.g., in (Schindler et al., 2010) and (Klinger et al., 2015). The semi-major axis of the ellipse corresponds to half of the height of the bounding rectangle and the semi-minor axis to half of the width of the rectangle. This model results in about 2700 features according to the number of pixels within the elliptic region and the number of input channels (three in this case).

The second model (*Stripes*, STR) divides the rectangle into horizontal stripes (the number of stripes η_{str} is a free parameter) and concatenates the weighted means and standard-deviations of each input channel and each row into the feature vector. Because the pixels along the central vertical line in



(a) ELL (b) STR (c) SYM

Figure 4.9: Different models for feature extraction from a surrounding rectangle: *Ellipse* (ELL), *Stripes* (STR) with Gaussian weighting function and *Symmetry axes* (SYM). Green areas indicate the expected position of persons within the rectangle, grey areas indicate background. The arrows in (c) indicate that the positions of the sub-regions in the SYM model are variable.

the training region are more likely to correspond to the foreground (person), weighting is applied to emphasise features close to this line. To this end, a Gaussian weighting function with the mean value according to the central pixel of each stripe and a constant standard-deviation of 4 pixels is applied. In this model the feature vector comprises 6 features for each stripe (mean and standard deviation for each input channel).

The third model (*Symmetry axes*, SYM) accumulates features around the body axes associated to the torso and lower body based on the Symmetry-Driven Accumulation of Local Features (Farenzena et al., 2010) as used for person re-identification. In this method, the bounding box of a person is divided into three parts associated to the head, torso and lower body, and the horizontal axes separating these regions as well as the vertical symmetry axes in each region are found automatically. For this application, the model by Farenzena et al. (2010) is simplified and only the regions associated to the torso and lower body are considered, while the horizontal axes are fixed, cf. Figure 4.9. In this model, the features are computed from windows within the bounding box having a fixed width of 12 pixels centred on the symmetry axes. The mean and standard deviation of each input channel within these regions are taken as features, resulting in a total number of 12 features.

The proposed models are alternative approaches for the definition of features. These alternatives will be compared in the experiments.

Classifier confidence. For the estimation of the probability density $p(c_i|\mathbf{x}_i^F)$ of a tracked person i in a current frame, the ORF classifies every possible position near the predicted feet position back-projected into the image and assigns it a confidence value of being the reference point of the regarded person. The result is modelled as the probability density $p(c_i|\mathbf{x}_i^F)$ and is shown for an example image for one tracked person (the foremost person in that scene) in Figure 4.8(c). In this context the classification is only applied within the image region inside the 3σ confidence ellipse of the predicted state. To account for low incidence angles of the image rays with the ground plane (which may lead to very narrow search spaces due to high aspect-ratios of the confidence ellipses), the actual search region is defined as a circular region around the predicted state with the radius equal to the semi-major axis of the 3σ confidence ellipse of the predicted state. This pdf will be used for the inference of the image position of the feet in combination with the detector confidence.

4.3 Temporal model

Traditionally, temporal models used for tracking in multi-person environments are realised by stand-alone filters applied to all persons independently. More recent work on temporal modelling is concerned with more complex motion models that predict a pedestrian state w.r.t. its own preceding state and that of other members of the scene. Modelling this so called *motion context* is motivated by the fact that, in a crowded scene, persons need to react to their environment when planning their motion and do not move in a way completely independent from other persons (Helbing and Molnár, 1995).

None of the papers described in the related work predicts the state with respect to all scene members under consideration of their uncertainties and estimates their interactions in a joint probabilistic model. In this chapter, a temporal model that comprises all of the desired properties is formulated. The goal will be met using Gaussian Process Regression (GPR). A new covariance function will be introduced that takes two trajectories as input and computes a covariance for the associated persons. The covariance is updated at every time step, which makes the model adaptive to the potentially changing degrees to which persons interact. As GPR models inherently account for noise of the input data and compute variances of the predicted variables, the model can be integrated without further adjustment into the recursive Bayesian estimation framework. Similar to Ellis et al. (2009) and Kim et al. (2011), this method models the velocity v_i of a person as the target variable of a GPR model. As opposed to the related work, the method takes the trajectories and velocities of all currently tracked pedestrians as input and estimates the interactions between all persons based on a new formulation of the covariance function at runtime. In this way, an explicit grouping of people is avoided and the proposed model will, thus, be referred to as *Implicit Motion Context* (IMC).

The IMC will be used for the prediction of the velocities as part of the state vector in the Dynamic Bayesian Network (cf. Section 4.1). Based on these velocities, the predictive distribution $p(\mathbf{w}_{t,i}|\mathbf{w}_{t-1,j=1...n})$ will be formulated in Section 4.5.

4.3.1 Implicit Motion Context

The proposed Implicit Motion Context model takes account of the idea that persons, moving in similar directions and being close to each other, mutually influence each other's movements. This idea is illustrated in Figure 4.10, where the person represented by the green trajectory interacts with the person represented by the blue trajectory, but not with the person represented by the grey trajectory.

Two separate Gaussian Process Regression models are defined for the prediction of the velocities v_X and v_Z in the directions parallel to the ground plane in object space. Because the models for the prediction of v_X and v_Z are equivalent, the model is only described for the general case of predicting the function value of a target variable v_i associated to person i. The velocities $\mathbf{v} = \{v_1, ..., v_n\}$ of all persons, including i, from the posterior states of the previous time step and the trajectories $\{\mathcal{T}_1, ..., \mathcal{T}_n\}$ of these persons are used as input variables. Each trajectory $\mathcal{T}_i$ is modelled as a time-ordered set of positions in input space with cardinality η_h. For the prediction of the state variables, the approach is to first predict the velocities for all persons at their input positions $\mathbf{X}_{i,t-1}$ by means of Gaussian Process Regression and then to append the distance covered in the time Δt with the estimated velocities to

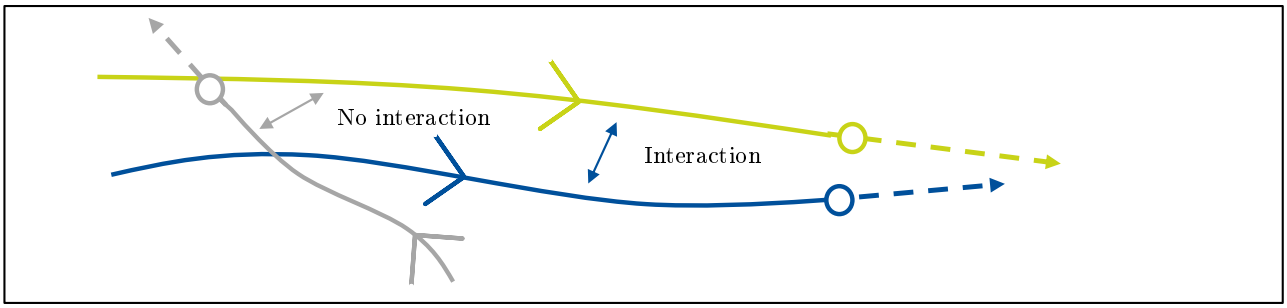


Figure 4.10: Sketch of the principles of Implicit Motion Context. The solid lines represent the trajectories of three persons in object space. The circles represent the current epoch. The dashed arrows indicate the posterior velocities at the previous time.

the posterior states of time $t-1$ in order to obtain the predicted positions at time t.

In accordance with the GPR model and as proposed in (Klinger et al., 2016), the velocity of a pedestrian is decomposed into a trend and a signal, and it is assumed that the signals of two pedestrians are correlated in case of interactions. By analogy to Equation 2.36, the predictive model for the velocity can be written in probabilistic form as a Gaussian distribution over the predicted velocity:

$$p(v_i|\mathbf{v}) \sim \mathcal{N}\left(\mathcal{GP}_\mu(\mathcal{T}_i, \mathbf{T}), \mathcal{GP}_\Sigma(\mathcal{T}_i, \mathbf{T})\right), \tag{4.19}$$

where the input variables $\mathbf{T} = \{(\mathcal{T}_1, v_1), ..., (\mathcal{T}_n, v_n)\}$ are given by the set of time-ordered tuples that consists of the trajectories and current velocity estimates of all n currently tracked pedestrians. The predicted velocity and covariance of a person i can be found using Equations 2.37 and 2.38. To define the Gaussian Process, the mean function and the covariance function need to be defined.

Mean function. Generally, the mean function of a Gaussian Process Regression model reflects the trend of the regression function. In terms of pedestrian tracking, a new mean function, which computes the average velocity $\bar{v}_i$ for each input trajectory $\mathbf{T}_i$, is defined as:

$$\bar{v}_i = E(v_i) = \left(\sum_{j=1...n} w\left(\mathcal{T}_i, \mathcal{T}_j\right)\right)^{-1} \sum_{j=1...n} w\left(\mathcal{T}_i, \mathcal{T}_j\right) v_j. \tag{4.20}$$

The function $w(\mathcal{T}_i, \mathcal{T}_j)$ is referred to as the angular function in this work. It takes into account the angular displacement α_{ij} of two trajectories. The angular function returns the cosine of the angular displacement if this displacement is smaller than a threshold θ_α, and otherwise zero:

$$w(\mathcal{T}_i, \mathcal{T}_j) = cos\left(\alpha_{ij}\right) \cdot \delta\left(\alpha_{ij} \leq \theta_\alpha\right) \tag{4.21}$$

Figure 4.11 shows the principle of the mean function for one person (associated to trajectory $\mathcal{T}_a$): The trend yields the weighted average (shown by the black solid arrow) computed from all velocities for which the angular displacement is smaller than the angular threshold θ_α. The area in which velocities of other persons are considered for the prediction of the velocity of person a is depicted in green. In this example, only the velocity of person b (represented by the blue trajectory) contributes to the

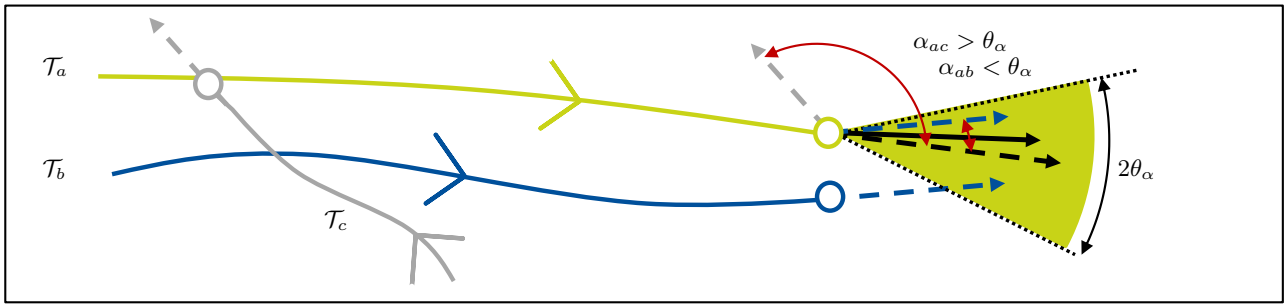


Figure 4.11: Mean function illustrated for trajectory $\mathcal{T}_a$, in the presence of two further trajectories $\mathcal{T}_b$ and $\mathcal{T}_c$. Due to the respective angular displacements (see red arrows), $\mathcal{T}_b$ contributes and $\mathcal{T}_c$ does not contribute to the estimated velocity of $\mathcal{T}_a$.

trend at the position of person a.

In this work, the distance between two persons is not accounted for by the mean function. This leads to a smoothing of trajectories even if no other persons are close to the regarded person. When a pedestrian is occluded in such situations and no measurements can be obtained, it is assumed that using the average motion, depending on the angular displacements only, leads to more realistic predictions.

Covariance function. The role of the covariance function is to express the similarity of two input variables. In the context of tracking, this work defines a new covariance function whose values are high for interacting persons. The covariance function is evaluated for every pair of pedestrians and the computed values are contained in the covariance matrix K, as illustrated by Figure 4.12. The function takes as input two trajectories $\mathcal{T}_i = [\mathbf{X}_{i,t-\eta_h}, ..., \mathbf{X}_{i,t-1}]^\top$ and $\mathcal{T}_j = [\mathbf{X}_{j,t-\eta_h}, ..., \mathbf{X}_{j,t-1}]^\top$ w.r.t. their current and η_h most recent positions in object space. The parameter η_h is the size of the temporal window that is regarded for computing the covariance; η_h is one free parameter of the system. It is assumed that the motion direction and the spatial distance of two pedestrians are representative for their interactions. Therefore, the function takes into account the angular displacement of the motion directions α_{ij} and the spatial distance d_{ij} between the current positions. The covariance function is defined as follows:

$$k(\mathcal{T}_i, \mathcal{T}_j) = w(\mathcal{T}_i, \mathcal{T}_j) \cdot \sigma_f^2 \cdot exp\left(-\frac{d_{ij}^2}{2l^2}\right) + \sigma_{n,i}^2 \cdot \delta(i = j), \tag{4.22}$$

where $d_{ij} = |\mathbf{X}_{i,t-1} - \mathbf{X}_{j,t-1}|$ is evaluated as the 2D distance between the current positions of persons i and j w.r.t. the posterior states at the previous epoch. The noise variance $\sigma_{n,i}^2$ reflects the uncertainty about the input velocities and is added to the diagonal elements of K. The input velocities are the posterior velocities from the previous time step and, thus, the noise variances are the posterior variances from the previous time step. α_{ij} is computed as the angle between the connecting straight lines of the first and last points of $\mathcal{T}_i$ and $\mathcal{T}_j$, respectively.

In Figure 4.12, the principle of the proposed covariance function is visualised based on an exemplary scene with three persons. The covariance function $k(\mathcal{T}_i, \mathcal{T}_j)$ is based on an exponential term that depends on the length scale parameter l and computes high covariances for two trajectories for which the spatial distance d_{ij} and the angular displacement α_{ij} are small. Depending on the outcome of the

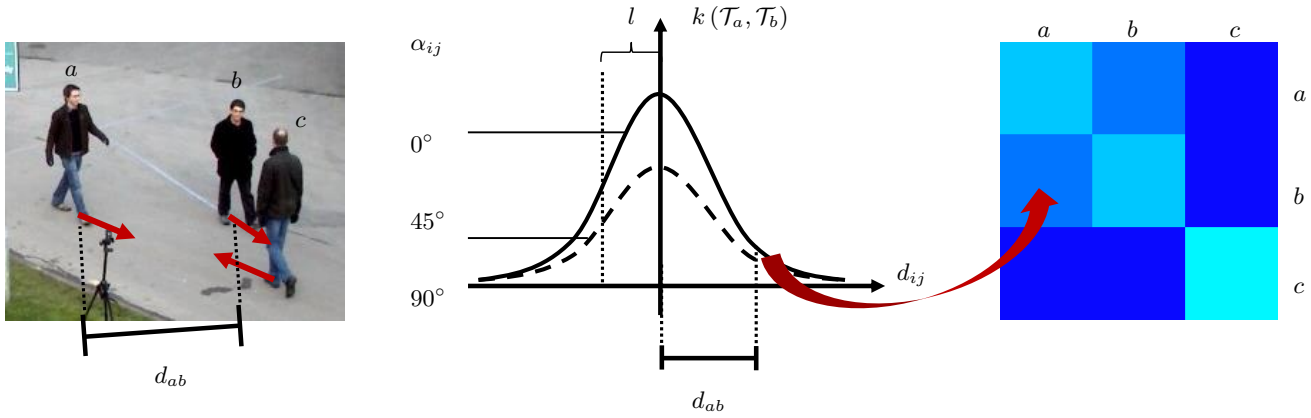


Figure 4.12: Illustration of the covariance function. On the left, an image with three persons a, b and c is shown. The red arrows indicate their motion directions and d_{ab} is the distance between persons a and b. In the middle, the covariance function for different angular displacements α_{ij} is shown. The covariance is computed based on the angular displacement and on the distance and is recorded in the covariance matrix, which referred to as K in the text and shown on the right. Brighter colours indicate higher covariances.

covariance function, every pair of trajectories is assigned a value in the covariance matrix K, as shown on the right side of the figure.

Having defined the covariance function in Equation 4.22, the covariance is computed for every pair of currently tracked pedestrians, whose covariances are expressed by the matrix K. The covariances between the observed and the unknown target variables are expressed by the vector K_i, and $K_{ii} = k(\mathcal{T}_i, \mathcal{T}_i)$ is the variance of the unknown target variable. Given the observed input data $\mathbf{T}$, the predicted values of the velocities, $\hat{v}_i$, and their variances $\hat{\sigma}^2_{vi}$, are found in accordance with Equations 2.37 and 2.38, respectively:

$$\hat{v}_i = \bar{v}_i + K_i K^{-1}(\mathbf{v} - E(\mathbf{v})), \tag{4.23}$$

$$\hat{\sigma}^2_{vi} = K_{ii} - K_i K^{-1} K_i^\top. \tag{4.24}$$

The Implicit Motion Context depends on four free parameters, the signal variance σ_f^2, the length-scale l, the length of the *history* η_h, and on θ_α, whose optimal values are determined by training in the experiments.

4.3.2 Mutual occlusions

In this section, the strategy for estimating mutual occlusions is explained. For every person, the occlusion o_i is computed from the overlap of the predicted surrounding rectangles of all persons tracked at time $t-1$. Based on the predicted velocities, the expected image reference point (x_i^{F+}, y_i^{F+}) and head position (x_i^{H+}, y_i^{H+}) of every person is computed via the measurement equations 4.1–4.4. The predicted surrounding rectangle of a person is defined as $\mathbf{r}_i^+ = [x_i^{F+}, y_i^{F+}, w_i^+, h_i^+]^\top$, where $h_i^+ = y_i^{F+} - y_i^{H+}$ is the height of the rectangle and w_i^+ is computed from h_i^+ based on the assumption that the aspect ratio of the surrounding rectangle is constant, i.e., equal to the aspect ratio of the initial detection.

The probability $p(o_{i,t})$ that person i is occluded at time t is defined as the degree to which person i is occluded by other persons:

$$p(o_{i,t}) = \frac{area\left(\mathbf{r}_i^+ \cap \left(union\left(\{\mathbf{r}_j^+\}\forall\{j \in [\mathbf{T} \setminus \mathcal{T}_i] | y_j > y_i\}\right)\right)\right)}{area(\mathbf{r}_i^+)}. \tag{4.25}$$

In Equation 4.25, the function $union(\cdot)$ computes the union of all predicted rectangles that are assumed to occlude person i, i.e. whose image row coordinates y_j are larger than that of person i. The function $area(\cdot)$ returns the size of a region in pixels. In this work, the occlusions are modelled as independent variables. A more detailed description of the joint probabilities would model the relationship between the occlusions and the state vector, whose values actually cause the presence or absence of the occlusions. This modelling would lead to loops in the graph structure, so that only approximate inference, e.g. by loopy belief propagation (Frey and MacKay, 1998), could be applied. Here, it is assumed that the feedback obtained from loopy belief propagation would not have a larger impact on the results, because the exact degree of occlusion is not as important as the vague estimation of whether or not a person is occluded at all and an approximate value for the occlusion.

4.4 Data association

To associate the verified and grouped detections given by the generic person detector (cf. chapter 4.2.2) to individual persons, a joint probabilistic data association strategy is used. To find the globally optimal solution to the association problem, linear programming is applied. This strategy leads to an optimal solution w.r.t. to the association weights computed by the affinity measures. In this section, a new strategy for the computation of the affinity measures is introduced, which is based on the prediction using the proposed motion model, as well as on the instance-specific classification strategy using the Online Random Forests.

Problem statement

At every time step, a set $\mathbf{D} = \{\mathbf{d}_1, ..., \mathbf{d}_m\}$ of m detections $\mathbf{d}_k = [x_k^F, y_k^F, w_k, h_k]^\top$, each characterised by the image coordinates of the reference point and the width w_k and height h_k of the surrounding rectangle in the image, and a set $\mathbf{T} = \{\mathcal{T}_1, ..., \mathcal{T}_n, \mathcal{T}_*\}$ of n existing trajectories and a potential new trajectory $\mathcal{T}_*$ is given. Under the constraint that each detection may only be assigned to zero or to one trajectory and that every existing trajectory may only be associated with zero or one detection, the data association problem can be formulated as an integer linear program with binary variables

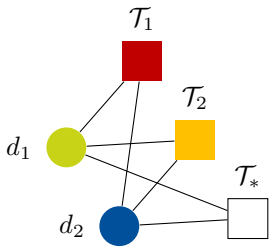


Figure 4.13: Bipartite association graph for two detections d_1 and d_2 and two trajectories $\mathcal{T}_1$ and $\mathcal{T}_2$. $\mathcal{T}_*$ represents a potential new trajectory.

(Dantzig, 1951):

$$\begin{aligned} \text{maximise} \sum_{k\in[\mathbf{D}]} w_i^k a_i^k \text{ subject to the constraints} \\ \sum_{i\in[\mathbf{T}]} a_i^k \leq 1 \ \forall k \in [\mathbf{D}], \\ \sum_{k\in[\mathbf{D}]} a_i^k \leq 1 \ \forall i \in [\mathbf{T} \setminus \mathcal{T}_*], \end{aligned} \tag{4.26}$$

where a_i^k is a binary indicator variable for the event that detection k is associated to trajectory i and w_i^k is the weight of that association. $[\mathbf{D}]$ denotes the set of detection indices and $[\mathbf{T}]$ denotes the set of trajectory indices. It is further required that the variable a_i^k has a lower bound of 0, which need not be stated explicitly since the a_i^k is defined as binary variable. For a toy example of an association problem with two detections and two trajectories, all feasible association events are visualised in Figure 4.13. The system of inequality constraints from Equation 4.26 for the situation depicted in Figure 4.13 is given in Equation 4.27.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_1^1 \\ a_1^2 \\ a_2^1 \\ a_2^2 \\ a_*^1 \\ a_*^2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{4.27}$$

Similarity measures

The weights w_i^k account for the spatial distance, as well as for the similarity in terms of appearance of the tracked objects and the detections. A high weight w_i^k favours the association of detection k to trajectory i. Let $\mathbf{r}_i^+ = [x_i^{F+}, y_i^{F+}, w_i^+, h_i^+]^\top$ denote the expected (predicted) reference point, width and height of object i in the image, the probability of the event that detection k is associated to object i w.r.t. the location and size of the detection is

$$p_d(a_i^k = 1) = \mathcal{N}(\mathbf{d}_k; \mathbf{r}_i^+, \Sigma_{\mathbf{rr},i}^+), \tag{4.28}$$

where $\mathcal{N}(\mathbf{d}_k; \mathbf{r}_i^+, \Sigma_{\mathbf{rr},i}^+)$ is a normal distribution evaluated for the distance between the image coordinates of the reference point and height of $\mathbf{d}_k$ and $\mathbf{r}_i^+$, respectively. $\Sigma_{\mathbf{rr},i}^+$ is the covariance matrix of the reference point and height of $\mathbf{r}_i^+$, given by variance propagation of the predicted position and height of person i into the image, using the Jacobian of the measurement equations 4.1–4.5. The similarity of a detection and the trajectories in terms of appearance is evaluated by using the Online Random Forest classifier. The classifier, which represents every person and the background by a class $\mathcal{C}_i$, delivers a posterior probability over all class labels. The probability of the event that detection k is associated to object i according to the classifier is modelled as the class-conditional probability of class $\mathcal{C}_i$:

$$p_c(a_i^k = 1) = p(id_k = \mathcal{C}_i | \mathbf{d}_k), \tag{4.29}$$

where $p(id_k = \mathcal{C}_i|\mathbf{d}_k)$ is the posterior probability that detection k belongs to object i, i.e. that the identity id_k of the detection is the class label C_i, given the detection window $\mathbf{d}_k$ that is classified. The weights w_i^k are defined as the product of the probabilities in Equations 4.28 and 4.29:

$$w_i^k = p_d(a_i^k = 1) \cdot p_c(a_i^k = 1). \tag{4.30}$$

The weight for the event of an association to $\mathcal{T}_*$, i.e., that a detection is not associated to any existing trajectory, is defined as the product of a constant term and the class-conditional probability of the background class $p_c(a_*^k = 1)$ found by classification of the detection window:

$$w_*^k = p(a_*^k = 1) = \mathcal{N}([x_k^F, y_k^F]^\top; [x_k^F, y_k^F]^\top, \Sigma_0) \cdot p_c(a_*^k = 1), \tag{4.31}$$

where $\mathcal{N}([x_k^F, y_k^F]^\top; [x_k^F, y_k^F]^\top, \Sigma_0)$ is a normal distribution over the reference point of the detection with covariance matrix $\Sigma_0 = diag(\sigma_{xF}^2, \sigma_{yF}^2)$, which accounts for the uncertainty of an initial position of a trajectory. Because the initialisation of a trajectory is carried out based on the results of pedestrian detection, a fixed value of σ_{kde} (cf. Section 4.2.2) is assigned to σ_{xF} and σ_{yF}, respectively.

Optimisation

The optimisation is carried out using the revised simplex method which is part of the Mixed Integer Linear Programming solver (Berkelaar et al., 2004). All associations, whose corresponding indicator variables have a value of 1 after the optimisation, are established. In case that a detection is associated to $\mathcal{T}_*$, the detection serves as initialisation of a new trajectory. If a trajectory is not associated with any detection after the optimisation, the trajectory is only continued by the temporal model at that time step.

Once a solution to the data association problem has been found, the single-frame detections can be combined with the results from the classifier to compute a position of the pedestrian's reference point in the image, which is then used for the determination of the pedestrian's position in object coordinates.

4.5 Recursive estimation

To find the optimal configuration of unknown variables, the Bayesian Network is transformed into a factor graph representation (Kschischang et al., 2001), as depicted in Figure 4.14, and message passing is applied. According to the factorisation shown by the factor graph, the joint pdf (cf. Equation 4.7) of all variables can be formulated as the product of all functions associated to the factor nodes:

$$\begin{aligned} & p\left(\mathbf{w}_{j=1\ldots n,t-1}, \mathbf{w}_{i,t}, \mathbf{r}_{i,t}, n_{i,t}, \mathbf{x}_{i,t}^{F}, \mathbf{x}_{i,t}^{H}, d_{i,t}, c_{i,t}, C_t, Y_\pi, o_{i,t}, IP_t\right) \\ \propto & f_1\left(\mathbf{w}_{i,t}, \mathbf{w}_{j=1\ldots n,t-1}\right) \cdot f_2\left(\mathbf{w}_{i,t} Y_\pi\right) \cdot f_3\left(\mathbf{r}_{i,t}, \mathbf{w}_{i,t}, n_{i,t}\right) \cdot f_4\left(Y_\pi\right) \cdot f_5\left(\mathbf{x}_{i,t}^{F}, \mathbf{r}_{i,t}\right) \cdot \\ & \cdot f_6\left(\mathbf{x}_{i,t}^{H}, \mathbf{r}_{i,t}\right) \cdot f_7\left(n_{i,t}, o_{i,t}, IP_t, \right) \cdot f_8\left(d_{i,t}, \mathbf{x}_{i,t}^{F}\right) \cdot f_9\left(c_{i,t}, \mathbf{x}_{i,t}^{F}\right) \cdot f_{10}\left(IP_t\right) \cdot f_{11}\left(o_{i,t}\right). \end{aligned} \tag{4.32}$$

In Equation 4.32, each factor corresponds to a pdf in the Dynamic Bayesian Network representation of the joint probability of all variables. Based on the factor graph representation, the procedure for performing inference can be derived, as described in the following paragraphs.

4.5.1 Inference

Because the graph does not contain cycles, inference on the graph is exact and is conducted using a message passing strategy in a way similar to the sum-product algorithm, see (Kschischang et al., 2001). Note, that inference using Belief Propagation with continuous variables usually requires a prior discretisation of the pdfs. In the presented work, the variables are either binary or Gaussian. The inference corresponds to the Kalman filtering strategy: Kalman filtering resembles the forward propagation over time of the Belief Propagation algorithm, whereas the Kalman smoothing equations realise the backward propagation of messages towards the leaf nodes. In this work, smoothing is not carried out, because it requires a closed system, which is not available when processing video streams online. This method sticks to the filtering strategy in the way that all messages are propagated to the root node variable, which is the system state, and that a full backward propagation is omitted. Backward propagation is only conducted down to the hidden variables defined in the observation space of time t, so that the image regions used for updating the classifier can be defined w.r.t. the best possible agreement of the state variables with all measurements and the temporal model.

The factor graph comprises ten factor nodes. Factor f_1 represents the predictive function of the recursive estimation framework. Factor f_2 represents the fictitious observation that pedestrians stand on the ground plane. Factor f_3 represents the update step of a Kalman Filter model, where the Extended Kalman Filter equations (Gelb, 1974) are used to linearise the measurement equations. Factor node f_4 represents the prior probability of the parameter describing the ground plane. In the forward propagation, the factor nodes f_5, f_6, f_8 and f_9 propagate messages in the observation space to define the belief about the image position of a person. Factors f_7, f_{10} and f_{11} propagate the belief about the accuracy of the image position to the variable n. The image position $\mathbf{r}$ and the additive measurement noise n are considered in the update step of the filter. Initially at every time step, all observed variables are assigned their pdfs as described in Sections 4.2 and 4.3.2. The Belief Propagation starts by passing messages from the leaf nodes of the factor graph in the upward direction. All message passing and belief update steps are given in Table 4.2.

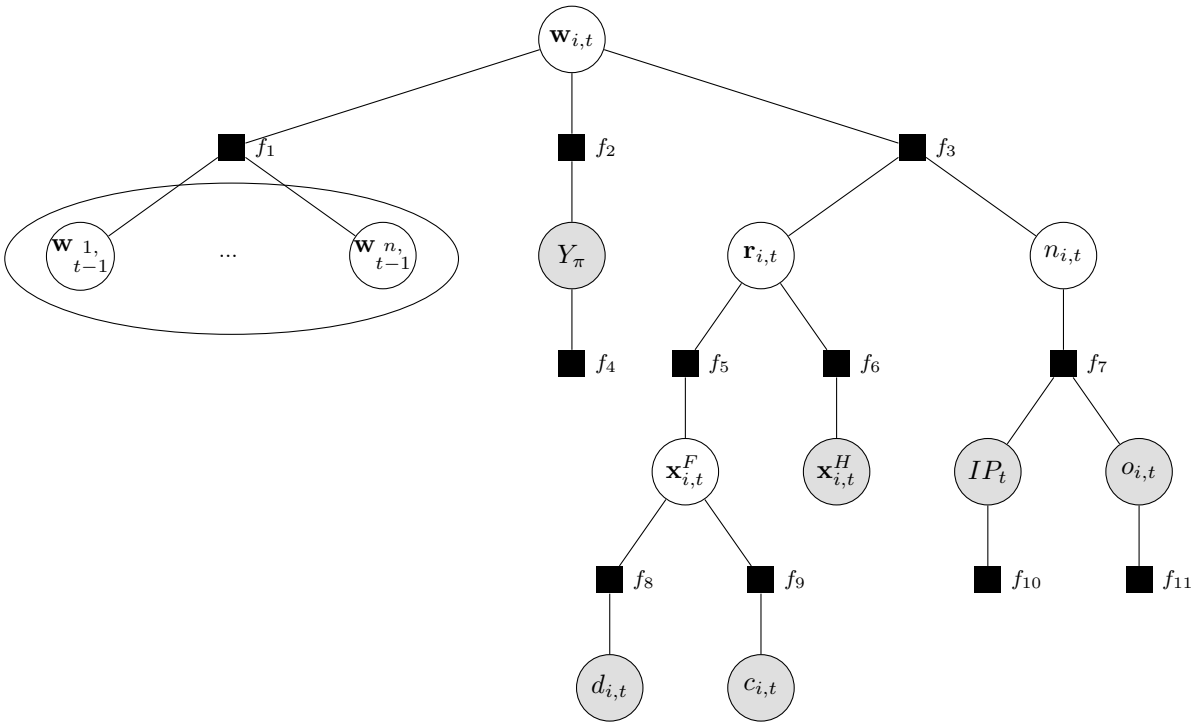


Figure 4.14: Factor graph representation of the model for multi-person tracking. The variable nodes represent the variables of the Dynamic Bayesian Network representation (cf. Figure 4.3). The factor nodes are represented by squares.

The backward recursion is only carried out down to the hidden variable representing the rectangle $\mathbf{r}$ around the corresponding person in the image. For visual support, the message passing steps are categorised into four blocks, according to their purpose in the model. These blocks are related to the temporal model, i.e. the prediction step of the recursive filter, the analysis of the measurements, the analysis of the measurement uncertainty w.r.t. to the occlusions and interesting places in the scene and the update step of the recursive filter. Details on the application of the message passing steps are given in the following paragraphs for each of the four sections, for the backward recursion and for the belief update. Because messages going out from nodes with only one neighbour, except for the receiving node, are equivalent to the incoming messages, the related steps are not discussed further.

Prediction. For the prediction, message passing steps 1 and 2 in Table 4.2 are carried out, based on the belief about the state vectors $\mathbf{w}_{j=1\ldots n,t-1}$ of all persons tracked at the previous epoch. For every person i, the pdf of the state vector $\mathbf{w}_{i,t}$ is modelled as a normal distribution

$$p(\mathbf{w}_{i,t}|\mathbf{w}_{j=1\ldots n,t-1}) = \mathcal{N}(\boldsymbol{\mu}^+_{\mathbf{w},t}, \Sigma^+_{\mathbf{ww},t}), \tag{4.33}$$

where the mean vector $\boldsymbol{\mu}^+_{\mathbf{w},t}$ is the expected state at time step t, $\Sigma^+_{\mathbf{ww},t}$ is its covariance matrix, and the predicted state $\mathbf{w}^+_t$ corresponds to the mean vector. The Implicit Motion Context is incorporated into the recursive filter by modelling the velocity components $v^+_{X,i}$ and $v^+_{Z,i}$ as target variables in two independent Gaussian Process Regression models.

Forward recursion	**Comment**
1: $m_{\mathbf{w}_{j,t-1}\to f_1}(\mathbf{w}_{j,t-1}) = 1 \; \forall j \in [\mathbf{T}]$	
2: $m_{f_1\to\mathbf{w}}(\mathbf{w}) = p\left(\mathbf{w}_{i,t}|\mathbf{w}_{j=1\ldots n,t-1}\right)$	Prediction of the state vector
3: $m_{d\to f_8}(d) = 1$	
4: $m_{c\to f_9}(c) = 1$	
5: $m_{f_8\to\mathbf{x}^F}(\mathbf{x}^F) = p(d|\mathbf{x}^F)$	Observation, cf. Sec. 4.2.2
6: $m_{f_9\to\mathbf{x}^F}(\mathbf{x}^F) = p(c|\mathbf{x}^F)$	Observation, cf. Sec. 4.2.3
7: $m_{\mathbf{x}^F\to f_5}(\mathbf{x}^F) = p(d|\mathbf{x}^F)\cdot p(c|\mathbf{x}^F)$	
8: $m_{\mathbf{x}^H\to f_6}(\mathbf{x}^H) = 1$	
9: $m_{f_5\to\mathbf{r}}(\mathbf{r}) = p(\mathbf{x}^F|\mathbf{r}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}^F}, \Sigma_{\mathbf{x}^F})$	
10: $m_{f_6\to\mathbf{r}}(\mathbf{r}) = p(\mathbf{x}^H|\mathbf{r}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}^H}, \Sigma_{\mathbf{x}^H})$	Observation, cf. Sec. 4.2.2
11: $m_{\mathbf{r}\to f_3}(\mathbf{r}) = m_{f_5\to\mathbf{r}}(\mathbf{r})\cdot m_{f_6\to\mathbf{r}}(\mathbf{r}) := \mathcal{N}(\boldsymbol{\mu}_{\mathbf{r}}, \Sigma_{\mathbf{rr}})$	
12: $m_{f_{10}\to IP}(IP) = p(IP)$	Observation, cf. Sec. 4.2.1
13: $m_{f_{11}\to o}(o) = p(o)$	Observation, cf. Sec. 4.3.2
14: $m_{IP\to f_7}(IP) = p(IP)$	
15: $m_{o\to f_7}(o) = p(o)$	
16: $m_{f_7\to n}(n) = p(n|IP,o)\cdot p(IP)\cdot p(o) := \mathcal{N}(0, \Sigma_{nn})$	
17: $m_{n\to f_3}(n) = m_{f_7\to n}(n)$	
18: $m_{f_4\to Y_\pi}(Y_\pi) = p(Y_\pi)$	
19: $m_{Y_\pi\to f_2}(Y_\pi) = p(Y_\pi)$	
20: $m_{f_2\to\mathbf{w}}(\mathbf{w}) = p(\mathbf{w}|Y_\pi)$	
21: $m_{f_3\to\mathbf{w}}(\mathbf{w}) = p(\mathbf{r}|\mathbf{w},n) := \mathcal{N}(\boldsymbol{\mu}_{\mathbf{r}}, \Sigma_{\mathbf{rr}} + \Sigma_{nn})$	Update of the state vector

Backward recursion	**Comment**
1: $m_{\mathbf{w}\to f_3}(\mathbf{w}) = 1$	
2: $m_{f_3\to\mathbf{r}}(\mathbf{r}) = p(\mathbf{r}|\hat{\mathbf{w}},n)$	Update of the image position

Belief update	**Comment**
$p(\mathbf{w}) = m_{f_1\to\mathbf{w}}(\mathbf{w})\cdot m_{f_2\to\mathbf{w}}(\mathbf{w})\cdot m_{f_3\to\mathbf{w}}(\mathbf{w})$	
$p(\mathbf{r}) = m_{f_3\to\mathbf{r}}(\mathbf{r})\cdot m_{f_5\to\mathbf{r}}(\mathbf{r})\cdot m_{f_6\to\mathbf{r}}(\mathbf{r})$	

Table 4.2: Message passing and belief update steps for the inference of the variables $\mathbf{w}_{i,t}$ and $\mathbf{r}_{i,t}$.

The prediction of the velocities is accomplished by computing the means of the Gaussian distributions associated to the velocities in accordance with Equation 4.23. The expected state vector $\mathbf{w}_t^+$ can then be written as $\mathbf{w}_t^+ = [X_t^+, Y_t^+, Z_t^+, H_t^+, v_{X,t}^+, v_{Z,t}^+]^\top$. The predicted positions X_t^+ and Z_t^+ are derived from the velocities and are computed relatively to the previous positions and the height Y_t^+

of the feet and H_t^+ of the head are assumed to be constant over time:

$$X_t^+ = X_{t-1} + v_{X,t}^+ \cdot \Delta t \tag{4.34}$$

$$Y_t^+ = Y_{t-1} \tag{4.35}$$

$$Z_t^+ = Z_{t-1} + v_{Z,t}^+ \cdot \Delta t \tag{4.36}$$

$$H_t^+ = H_{t-1} \tag{4.37}$$

$$v_{X,i,t}^+ = \bar{v}_{X,i,t} + K_{i,t}K_t^{-1}(\mathbf{v} - E(\mathbf{v})) \tag{4.38}$$

$$v_{Z,i,t}^+ = \bar{v}_{Z,i,t} + K_{i,t}K_t^{-1}(\mathbf{v} - E(\mathbf{v})). \tag{4.39}$$

The covariance matrix of the predicted state vector $\Sigma_{\mathbf{ww},t}^+$ is computed from the covariance matrix of the previous time step with the additive process noise Σ_p:

$$\Sigma_{\mathbf{ww},t}^+ = \Psi\Sigma_{\mathbf{ww},t-1}\Psi^\top + \Sigma_p, \tag{4.40}$$

where Ψ is the transition matrix that transforms the state vector from the previous time step to the current time step. Here, zero acceleration in the directions of X and Z and zero velocity in vertical direction is assumed for the parameters Y and H, so that the transition matrix is expressed via

$$\Psi = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.41}$$

Deviations from the assumption of constant velocity in v_X and v_Z and zero velocity for the parameters Y and H likely occur due to unforeseen accelerations (a_X,a_Z), e.g. a pedestrian stops at a traffic light, and velocities (v_Y,v_H), e.g. due to the changing gait of a person. It is assumed that the vector $\mathbf{u} = [a_X, v_Y, a_Z, v_H]^\top$ is normally distributed with expectation $E(\mathbf{u}) = \mathbf{0}$ and covariance $\Sigma_{\mathbf{uu}} = diag(\sigma_{aX}^2, \sigma_{vY}^2, \sigma_{aZ}^2, \sigma_{vH}^2)$. The uncertainty about these accelerations and velocities affects the uncertainty about the predicted state and is accounted for via the process noise covariance Σ_p, computed as

$$\Sigma_p = G\Sigma_{\mathbf{uu}}G^\top = \begin{bmatrix} \frac{\sigma_{a_X}^2\Delta t^4}{4} & 0 & 0 & 0 & \frac{\sigma_{a_X}^2\Delta t^3}{2} & 0 \\ 0 & \sigma_{v_Y}^2\Delta t^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sigma_{a_Z}^2\Delta t^4}{4} & 0 & 0 & \frac{\sigma_{a_Z}^2\Delta t^3}{2} \\ 0 & 0 & 0 & \sigma_{v_H}^2\Delta t^2 & 0 & 0 \\ \frac{\sigma_{a_X}^2\Delta t^3}{2} & 0 & 0 & 0 & \sigma_{a_X}^2\Delta t^2 & 0 \\ 0 & 0 & \frac{\sigma_{a_Z}^2\Delta t^3}{2} & 0 & 0 & \sigma_{a_Z}^2\Delta t^2 \end{bmatrix}. \tag{4.42}$$

The matrix G translates the covariance $\Sigma_{\mathbf{uu}}$ to the system state space and is defined as:

$$G = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & 0 & \Delta t \\ \Delta t & 0 & 0 & 0 \\ 0 & 0 & \Delta t & 0 \end{bmatrix}. \tag{4.43}$$

In this model, the accelerations a_X and a_Z are partly induced by interactions with other pedestrians, so that the variances

$$\sigma^2_{v_{X,i,t}} = K_{ii,t} - K_{i,t}K_t^{-1}K_{i,t}^\top \quad \text{and} \tag{4.44}$$

$$\sigma^2_{v_{Z,i,t}} = K_{ii,t} - K_{i,t}K_t^{-1}K_{i,t}^\top, \tag{4.45}$$

computed by the Gaussian Process Regression in accordance with Equation 4.24 reflect the uncertainties about the process noise, i.e., $\sigma^2_{aX}{=}\sigma^2_{v_{X,i,t}}\Delta t^{-2}$ and $\sigma^2_{aZ}{=}\sigma^2_{v_{Z,i,t}}\Delta t^{-2}$, where Δt is one frame.

Analysis of the measurements. In this step, the confidences $p(d_{i,t}|\mathbf{x}^F_{i,t})$ and $p(\mathbf{x}^H_{i,t}|\mathbf{r}_{i,t})$ of the detector and the confidence $p(c_{i,t}|\mathbf{x}^F_{i,t})$ of the classifier are computed for the current image frame using the methods proposed in Sections 4.2.2 and 4.2.3. After applying data association to the detections, the pdf $p(d_{i,t}|\mathbf{x}^F_{i,t})$ is modelled as the density computed by kernel density estimation based on all single-scale detections associated to the grouped detection that is assigned to trajectory i. According to the message passing step 7 in Table 4.2, the pdfs $p(d_{i,t}|\mathbf{x}^F_{i,t})$ and $p(c_{i,t}|\mathbf{x}^F_{i,t})$ are multiplied and the product is approximated by a Gaussian pdf, $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}^F}, \Sigma_{\mathbf{x}^F})$, whose mean $\boldsymbol{\mu}_{\mathbf{x}^F}$ and covariance $\Sigma_{\mathbf{x}^F}$ are computed as the weighted sample mean and covariance from the product of these distributions. The pdf $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}^F}, \Sigma_{\mathbf{x}^F})$ represents the current belief about the reference point. The belief about the position of the head $\mathbf{x}^H_{i,t}$ is represented by the Gaussian pdf $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}^H}, \Sigma_{\mathbf{x}^H})$, whose parameters are determined as the weighted sample mean $\boldsymbol{\mu}_{\mathbf{x}^H}$ and covariance $\Sigma_{\mathbf{x}^H}$ from the kernel density estimation of the head position. The belief about variable $\mathbf{r}_{i,t}$ is modelled by the Gaussian pdf $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{r}}, \Sigma_{\mathbf{rr}})$ with $\boldsymbol{\mu}_{\mathbf{r}} = [\boldsymbol{\mu}_{\mathbf{x}^F}, \boldsymbol{\mu}_{\mathbf{x}^H}]^\top$ and

$$\Sigma_{\mathbf{rr}} = \begin{bmatrix} \Sigma_{\mathbf{x}^F} & 0 \\ 0 & \Sigma_{\mathbf{x}^H} \end{bmatrix} = \begin{bmatrix} \sigma^2_{x^F} & \sigma_{x^F y^F} & 0 & 0 \\ \sigma_{x^F y^F} & \sigma^2_{y^F} & 0 & 0 \\ 0 & 0 & \sigma^2_{x^H} & \sigma_{x^H y^H} \\ 0 & 0 & \sigma_{x^H y^H} & \sigma^2_{y^H} \end{bmatrix}. \tag{4.46}$$

Analysis of the variable n. The message passing steps 12–17 are straight-forward. The variables IP_t and $o_{i,t}$ are set to their observed values, where IP_t is evaluated at the predicted reference point and the occlusion is evaluated using Equation 4.25. The conditional probability $p(n_{i,t}|IP_t, o_{i,t})$ is defined as zero-mean Gaussian pdf $p(n_{i,t}|IP_t, o_{i,t}){:=}\mathcal{N}(0, \Sigma_{nn})$, where $\Sigma_{nn}{=}IP_t \cdot o_{i,t} \cdot \rho_n \cdot \mathbf{I}_{44}$ and ρ_n is the measurement noise coefficient, which is a free parameter. $\mathbf{I}_{44}$ is the 4×4 identity matrix.

Update. The probability $p(Y_\pi)$ is evaluated using the message passing steps 18 and 19, cf. Table 4.2. This probability is modelled as a normal distribution $p(Y_\pi) \sim \mathcal{N}(\mu_\pi, \sigma_\pi^2)$. The mean value μ_π of this distribution is the given distance of the ground plane from the camera and σ_π^2 is the variance of that distance. The assignment of an uncertainty σ_π^2 to the distance Y_π accounts for potential deviations from the assumption of a horizontal terrain. The message sent from factor node f_2 to the state vector, step 20 in Table 4.2, is the probability $p(\mathbf{w}|Y_\pi)$ of the state vector given Y_π. The relationship between these variables is introduced via the fictitious observation that pedestrians stand on the ground plane, cf. Equation 4.5.

The message sent from factor node f_3 to the state vector according to step 21 in Table 4.2 is the pdf $p(\mathbf{r}|\mathbf{w}, n) := \mathcal{N}(\boldsymbol{\mu}_\mathbf{r}, \Sigma_{\mathbf{rr}} + \Sigma_{nn})$. It is assumed that the measurement noise is additive and Gaussian distributed. The belief update for the variable representing the state vector involves the multiplication of the messages $m_{f_2 \to \mathbf{w}}(\mathbf{w}) = p(\mathbf{w}|Y_\pi)$ and $m_{f_3 \to \mathbf{w}}(\mathbf{w}) = p(\mathbf{r}_{i,t}|\mathbf{w}_{i,t}, n)$. In this work, the product of these messages is modelled as a normal distribution:

$$p(\mathbf{r}_{i,t}|\mathbf{w}_{i,t}, n) \cdot p(\mathbf{w}|Y_\pi) = \mathcal{N}(\bar{\mathbf{r}}, \Sigma_{\mathbf{xx}}) = \mathcal{N}\left(\left[x^F, y^F, x^H, y^H, Y_\pi\right]^\top, \begin{bmatrix} \Sigma_{\mathbf{rr}} & 0 \\ 0 & \sigma_\pi^2 \end{bmatrix} + \begin{bmatrix} \Sigma_{nn} & 0 \\ 0 & 0 \end{bmatrix}\right), \tag{4.47}$$

whose mean vector $\bar{\mathbf{r}}$ includes the parameters of the surrounding rectangle and the ground plane, and whose covariance matrix $\Sigma_{\mathbf{xx}}$ accounts for the uncertainties about these parameters and for the additive measurement noise.

Finally, all incoming messages to the state variable $\mathbf{w}_{i,t}$, $m_{f_1 \to \mathbf{w}}(\mathbf{w})$, $m_{f_2 \to \mathbf{w}}(\mathbf{w})$ and $m_{f_3 \to \mathbf{w}}(\mathbf{w})$ can be evaluated. Because these messages are Gaussian pdfs, the update step for the state vector can be performed using the Extended Kalman Filter update equations. The update of the state vector is carried out by evaluating Equations 2.23 – 2.26:

$$\hat{\mathbf{w}}_{i,t} = \mathbf{w}^+ + K(\bar{\mathbf{r}}_{i,t} - \mathbf{r}_i^+), \tag{4.48}$$

where $\mathbf{r}_i^+ = [x^F(\mathbf{w}^+), y^F(\mathbf{w}^+), x^H(\mathbf{w}^+), y^H(\mathbf{w}^+), Y_\pi(\mathbf{w}^+)]^\top$ is the predicted state transformed to observation space by the (non-linear) measurement equations 4.1-4.5 and K is the Kalman Gain matrix,

$$K = \Sigma_{\mathbf{ww}}^+ M^\top (\Sigma_{\mathbf{xx},t} + M \Sigma_{\mathbf{ww}}^+ M^\top)^{-1}, \tag{4.49}$$

whereas M is the Jacobian of the measurement equations,

$$M = \begin{bmatrix} \frac{\partial x_F}{\partial X} & \frac{\partial x_F}{\partial Y} & \frac{\partial x_F}{\partial Z} & 0 & 0 & 0 \\ \frac{\partial y_F}{\partial X} & \frac{\partial y_F}{\partial Y} & \frac{\partial y_F}{\partial Z} & 0 & 0 & 0 \\ \frac{\partial x_H}{\partial X} & \frac{\partial x_H}{\partial Y} & \frac{\partial x_H}{\partial Z} & \frac{\partial x_H}{\partial H} & 0 & 0 \\ \frac{\partial y_H}{\partial X} & \frac{\partial y_H}{\partial Y} & \frac{\partial y_H}{\partial Z} & \frac{\partial y_H}{\partial H} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.50}$$

and $\Sigma_{\mathbf{ww},t} = \Sigma_{\mathbf{ww}}^+ - \mathrm{K}\mathrm{M}\Sigma_{\mathbf{ww}}^+$ is the covariance of the predicted state. Note that the last two columns with zeros in M indicate that no measurements for the velocities can be taken, because the measurements stem from single images. The estimation of velocities only derives from the temporal model.

In Equation 4.48, the term in brackets reflects the correspondence of the measurements with the predicted state, which is referred to as the innovation. When the predicted state coincides well with the measurements, the difference between these two entities is small. High values of the innovation indicate possible outliers in the measurements, given that the prediction is correct, or indicate wrong predictions, given that the measurements are correct. Either way, the source of the disagreement cannot be revealed, because no reference data are available at processing time. If the innovation is too high, it is decided to rely on the temporal model and to neglect the measurements. To this end, a hypothesis test is applied, were the null-hypothesis is formulated as

$$H_0 : \mathbf{i}_t = 0 \tag{4.51}$$

which is tested against the alternative hypothesis

$$H_A : \mathbf{i}_t \neq 0. \tag{4.52}$$

Given the predicted state and the measurement vector, the innovation $\bar{\mathbf{r}}_{i,t} - \mathbf{r}_i^+$ and its covariance $\Sigma_{\mathbf{ii},t}$ can be computed according to Equation 2.27. The test statistic is defined as:

$$\mathcal{X}_n^2 = \mathbf{i}_t^\top \cdot \Sigma_{\mathbf{ii},t}^{-1} \cdot \mathbf{i}_t \tag{4.53}$$

H_0 is accepted according to the following rule:

$$\mathcal{X}_n^2 \leq \gamma_{1-\alpha} \quad \Rightarrow \quad \text{accept } H_0, \tag{4.54}$$

$$\mathcal{X}_n^2 > \gamma_{1-\alpha} \quad \Rightarrow \quad \text{accept } H_A, \tag{4.55}$$

where $\gamma_{1-\alpha}$ is the $1-\alpha$ quantile of the $\mathcal{X}^2$ distribution. If H_0 is rejected, the predicted state is not updated.

Backward propagation. If the update step has been executed, the mean vector and covariance matrix of the corrected state are transformed back to the image domain using the measurement equations and the corresponding Jacobian according to the message passing step 2 in the backward iteration, cf. Table 4.2:

$$p(\mathbf{r}_i|\hat{\mathbf{w}}_i, n_i, C_t) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{r}, M\Sigma_{\mathbf{ww}}M^\top). \tag{4.56}$$

$\boldsymbol{\mu}_\mathbf{r}$ denotes the mean of the corrected image position $[x^F(\hat{\mathbf{w}}_i), y^F(\hat{\mathbf{w}}_i), x^H(\hat{\mathbf{w}}_i), y^H(\hat{\mathbf{w}}_i), Y_\pi(\hat{\mathbf{w}}_i)]^\top$, and $\Sigma_{\mathbf{ww}}$ its covariance matrix. Because the pdf in Equation 4.56 is Gaussian, its mean vector is the argument maximum of the distribution and its values $x^F(\hat{\mathbf{w}}_i)$, $y^F(\hat{\mathbf{w}}_i)$, $x^H(\hat{\mathbf{w}}_i)$ and $y^H(\hat{\mathbf{w}}_i)$ define the corrected image position of person i.

Finally, the Online Random Forest classifier is updated using new training samples taken from the new bounding rectangle. As described in section 4.2.3, additional training samples are taken from shifted positions of the rectangle and samples for the negative class are taken from background regions around the rectangle. If the null-hypothesis is rejected in favour of the alternative hypothesis 4.52, backtracking and updating of the classifier is omitted.

4.5.2 Initialisation and termination

New trajectories are initialised based on single-frame detections that are not associated with any existing trajectory yet. Every detection $\mathbf{d}_k$ that is not associated with an existing trajectory is considered as a hypothesis $h_k = [true, false]$ about the presence of a new object with image position $\mathbf{d}_k$. Each hypothesis is validated based on the detector confidence $d_{F,t}$ and prior scene knowledge IP, using a likelihood ratio test of the form

$$h_k = \begin{cases} true, & \text{if } \frac{p(h_k=true|\mathbf{d}_k)\cdot p(h_k=true|d_{F,t})}{p(h_k=false|\mathbf{d}_k)\cdot p(h_k=false|d_{F,t})} > \theta_{ini} \\ false, & \text{otherwise} \end{cases} . \tag{4.57}$$

In Equation 4.57, $p(h_k = true|d_{F,t})$ is the probability that the hypothesis is true given the detector confidence at that reference point and $p(h_k = true|\mathbf{d}_k)$ is the probability of the detection given the prior knowledge about the scene, evaluated at the position of the reference point of the detection. A new trajectory is initialised if the threshold θ_{ini}, which is a free parameter, is exceeded.

Tracking of a person is stopped if the predicted position of that person transformed to the image domain lies outside of the image, or if the trajectory is not updated for more than a predefined number θ_{abs} of frames, where θ_{abs} is a free parameter and is referred to as the *absence count threshold*. When a trajectory is terminated, the class of the Random Forest classifier that represents the corresponding person is eliminated from the queue of samples, and the classifier is trained anew with the remaining classes.

4.6 Discussion

In this section, the theoretical strengths and weaknesses of the proposed method are discussed to anticipate the results to be expected in the experimental section of this work.

Strengths and weaknesses

The expected strength of the approach lies in the joint evaluation of the different complementary observations and fixed entities in a probabilistic framework. Scene-specific knowledge is expected to reduce the number of false detections, category-specific object detection restricts the processing to areas in which the desired object class most likely appears and instance-specific classifiers highlight regions in the image that reflect appearance features associated to individuals. The multitude of observations that influence the estimated state variables satisfies the need for redundant information about the location of pedestrians in order to eliminate the influence of measurement errors. Consequently, a geometrically more accurate posterior position of pedestrians is expected.

The predictive function proposed on the basis of Implicit Motion Context and Gaussian Process Regression is expected to give more realistic state estimations, with the most beneficial impact expected for epochs where no measurements are available. Assuming that in such situations the predictions are correct, measurements can be associated more reliably to the trajectories than by using a stand-

alone filter for every pedestrian independently of the other pedestrians. In this way, one can expect a reduction in the number of tracking errors, such as trajectory continuations by alternating persons (i.e., identity switches). Data association is supported by the instance-specific classifier that, together with the motion model, defines the affinity measures. The Gaussian Process Regression model takes uncertainties of the input variables (i.e. the velocities of all tracked persons) into account. Hence, persons with more accurate state estimates automatically have a larger influence on a predicted velocity. During an occlusion, where the uncertainty increases due to the additive process noise, the prediction of the occluded person state progressively depends on the persons surrounding the occluded person.

The integration of the estimation of dynamic parameters and image-based variables allows for the evaluation of the agreement between the temporal model and the observations and to account for this agreement when updating the classifier. The prior knowledge from previous time steps allows to verify the detections based on deviations from instance-specific information: The height of a person, estimated as a part of the state vector, is updated over time based on instance-specific information provided by the observation model. In this way, the innovation test takes account of target specific information over time that helps to validate single frame detections and makes the tracking framework more robust against outliers in the measurements.

The approach requires the availability of the exterior and the interior parameters of the images. This requirement may be restrictive for arbitrary application scenarios, but, in the addressed applications, the requirement can easily be met. In practice, the cameras being used in the regarded application scenarios mostly have a fixed focal length so that the laboratory calibration can be regarded as constant. In case of moving camera platforms, the motivation of using the proposed method primarily lies in the context of assisted driving and robotics. Both applications typically involve a range of different sensors, including active ranging devices, global navigation satellite systems (GNSS), accelerometers, and car odometry. Alternatively, the self-localisation can be carried out by means of visual odometry (Geiger et al., 2011), but the absolute accuracy of the derived locations is generally not as high as the one obtained by using the active sensors.

The proposed method only uses an indirect approach to ensure that two or more persons cannot take the same position in space. This is realised by assigning detections to trajectories (typical for detection-based tracking in general), that is, no detection can be assigned to more than one trajectory. Due to the grouping of the rectangles, two detections cannot have the same position and size, and, thus, the update steps of the trajectories are always mutually exclusive. In the absence of measurements, however, when only the temporal model is used to continue the trajectories, there is no guarantee for mutual exclusion.

The classification strategy based on Online Random Forests is expected to improve the tracking results, because this strategy trains a multi-class person classifier with instance-specific appearance features. Different from the related work using only binary classifiers, the classification accuracy is expected to be superior, because the classifier can be learnt from the training samples representing all persons, while a binary classifier does not distinguish between the samples in the set of negative training data. Due to the ability of being updated at runtime, the classifier accounts for the gradually changing appearance of a target. However, the critical point is the derivation of training samples

for the classifier, which has to be done based on the tracking results. Because training is performed incrementally, samples derived from misplaced image positions lead to the learning of a non-optimal classifier, due to which the tracker is prone to be continued towards wrong positions in future recursions.

Assumptions

The proposed method depends on the validity of three basic assumptions:

1. It is assumed that the ground plane is horizontal and that the terrain is flat. The assumption enables the conversion of 2D image to 3D object coordinates. In practice, it may be easily violated in terrain with more complex topography. In case of mobile robotics and in autonomous driving applications, the viewing direction of the camera is nearly parallel to the ground plane, which leads to an unfavourable propagation of measurement errors to the state space domain. The susceptibility of the approach to deviations from the assumption of a flat world is one of its shortcomings. To account for violations of this assumption, the fictitious observation that keeps persons to the ground plane is modelled stochastically. Given that the assumption is fulfilled, the tracking in 3D is expected to improve the results of 2D tracking due to a more realistic model of motion and interactions.

2. The proposed method involves a decision for or against the update of the recursive filter with the inferred image positions. As described in Section 4.5.1, if the disagreement between measurement and prediction (as measured by the system innovation) is too high, the update is avoided and the trajectory is continued using the temporal model only. However, a very high innovation only implies an erroneous behaviour of the filter, the source of which cannot be located exactly. The assumption that the temporal model can be relied on, rather than the measurement, is motivated by the observation that in crowded environments measurements are prone to be inaccurate, whereas the temporal model based on Implicit Motion Context is designed to deal with such situations.

3. It is further assumed that the state variables in the Dynamic Bayesian Network have an underlying uni-modal Gaussian distribution. However, in the absence of measurements, the uni-modal distributions possibly do not model the belief about the state variables correctly, depending on the complexity of the scene. In the regarded applications, uni-modal state representations are expected to perform sufficiently well, as the expected time of occlusions is rather short. Alternative models, e.g. mixtures of Gaussian Processes (cf. Trautman et al., 2015) or particle-based approaches, might be relevant for more complex scenes, such as soccer matches, where persons perform abrupt manoeuvres.

Free parameters

The free parameters of the proposed method are summarized in Table 4.3. The parameters can be categorised into four groups, according to their associated module. The data association is parameter-free, except for the parameters of the Online Random Forest.

The detection model entails the parameters for the HOG/SVM detector, which are the SVM confidence threshold θ_{svm} and the coefficient of the detection window increase ρ_{det}, the parameters σ_d, μ_P and σ_P used for false positive reduction, the parameter σ_{kde}, which is the standard deviation assigned

Detection	
θ_{svm}	Threshold of the Support Vector Machine (SVM)
ρ_{det}	Coefficient of the detection window increase
σ_d	Standard-deviation of the height of a detection in pixels
μ_P	Expected value of the height of a person in metres
σ_P	Standard-deviation of the height of a person in metres
σ_{kde}	Standard-deviation of the Gauss-kernel in the kernel density estimation
ϵ_{nms}	Parameter of the grouping function used for non-maximum suppression
Classification	
η_{tre}	Number of trees
η_{dep}	Maximal depth of the trees
η_{tes}	Number of random tests per node
η_{sam}	Minimum number of samples required for splitting
η_{str}	Number of stripes to divide the image region of interest in
η_{que}	Time span to store the samples
Temporal model	
l	Length scale
σ_f^2	Signal variance
θ_α	Angular threshold
η_h	Number of time steps regarded for correlating the trajectories
Recursive filter	
θ_{ini}	Threshold for initialisation
θ_{abs}	Maximum absence count
ρ_n	Measurement noise coefficient

Table 4.3: Free parameters.

to the Gauss-kernel in the kernel density estimation, and the parameter ϵ_{nms} used for computing the equivalence criterion for two detections in the non-maximum suppression step. The parameters of the HOG descriptor are set to the values used in the implementation in the seminal work by Dalal and Triggs (2005).

The parameters for the classification strategy comprise the parameters of the Random Forest classifier, which are the number of trees, the maximum allowable depth of the trees, the number of random tests applied at each node and the minimum number of samples required for splitting. Two additional parameters are specific for the proposed strategy, namely the number of stripes in the STR-model and the size of the temporal window in which previous samples are stored.

The parameters related to the temporal model are the length scale, signal variance, angular threshold and length of the trajectory snippet.

For the recursive estimation, only three parameters are relevant. θ_{ini} is the threshold based on which is decided whether or not to initialise a new trajectories (cf. Section 4.5.2). θ_{abs} is the maximum number of time steps, before the tracking of a person, whose trajectory is not updated in successive frames, is finished. The measurement noise coefficient ρ_n controls the variances of the additive measurement noise added to the covariance matrix of the image position.

5 Experiments

In this chapter, the proposed method is evaluated with respect to its capability of multi-person localisation and tracking. After an introductory section on the used datasets and evaluation criteria, three specific aspects will be addressed, namely the sensitivity of the proposed method to the variation of single parameters and the determination of optimal parameters for the individual models, the impact of the individual components of the system with regard to the localisation and tracking capability, and a comparison of the developed system with the related work.

The datasets and evaluation criteria used for the experiments are introduced in Section 5.1. Two different kinds of image sequences are used: Sequences captured by static cameras, which are typically used for video surveillance and forensics, traffic control and sport-sciences, and sequences captured by cameras on moving platforms, as they are used in driver assistance systems and mobile robotics. Note that the proposed method was originally developed for the application to the first type of image sequences (Klinger et al., 2015, 2016). In this work, however, the method is also tested for the use with image sequences of the latter group, in order to assess its transferability and possible limitations.

In Section 5.2, the detection, classification and prediction components of the system are investigated in detail. That section analyses the sensitivity of these components to the variation of the parameters and determines the parameters that deliver the best results for each component. The detection with the proposed strategy for false positive reduction is compared with an out-of-the-box pedestrian detector and different parameters of the strategy are examined. For the classification strategy, different models for the aggregation of features taken from the rectangle surrounding a person and different colour spaces are tested. The Implicit Motion Context parameters are trained using the direct search approach by Hooke and Jeeves (1961) and the optimality of the prediction is evaluated based on the sequence of system innovations and compared to a stand-alone filtering approach. Finally, the free parameters of the recursive estimation framework are investigated.

In Section 5.3, the impact of every component of the system is analysed by omission of the respective component from the overall strategy. The analysis is supported by qualitative results that show the trajectories generated by all investigated model variants.

Finally, in Section 5.4, the tracking performance of the proposed method is investigated. This chapter includes an analysis of the localisation performance in terms of geometric accuracy, an analysis of the predictive function based on Implicit Motion Context. Finally, a comparative study based on the test datasets available in two different benchmarks reveals the superiority or inferiority of the proposed method with respect to related work.

5.1 Datasets and evaluation criteria

The datasets used for the experiments represent different application scenarios and, thus, differ in the variation of the camera orientation, viewing direction and in the complexity of the depicted scene in terms of the number of contained persons, the range of depth and the severity of mutual occlusions.

PETS 2009 and AVG-TownCentre

In the first class of image sequences, the images are captured by cameras with constant exterior orientation, mounted about 7 metres above the ground with an inclined viewing angle. The *Performance Evaluation of Tracking and Surveillance* (PETS) 2009 dataset (PETS, 2009) depicts a university campus (cf. Figure 5.1(a)). This dataset is divided into three datasets S1, S2 and S3, originally designed for person count and density estimation of crowds (S1), people tracking (S2) and event recognition (S3). In this work, only sequences from the S1 and S2 datasets are used. The S1 and S2 datasets comprise three image sequences L1, L2 and L3 each, where the size of the image sequences varies between about 200 and 800 frames. In these sequences, the difficulty levels in terms of crowd density vary between about two and 40 persons per scene. The image size is 768x576 pixels captured at 7 Hz. The AVG-TownCentre dataset from the *Active Vision Laboratory* (AVG) (Benfold and Reid, 2011) depicts a crowded pedestrian zone (cf. Figure 5.1(b)). The dataset contains one image sequence with 450 frames captured with a resolution of 1920x1080 pixels at 2.5 Hz. PETS09-S2L1, PETS09-S2L2 and the AVG-TownCentre dataset are part of the 3D Multiple Object Tracking benchmark (3DMOT, Leal-Taixé et al., 2015), which is used for the evaluation of the method in this work. In the 3DMOT benchmark, the PETS09-S2L1, and an additional dataset from Technische Universität Darmstadt (Andriluka et al., 2010), are available for training, the other datasets are held back for testing. Because the training data are not characteristic for the complexity of the test data, the PETS09-S1L1-1 dataset, consisting of 221 frames depicting up to about 30 persons at a time is used for training in this work[1]. The PETS09-S2L2 sequence involves a total of 42 pedestrian passings with 9641 single-frame annotations and the AVG-TownCentre sequence involves a total of 226 pedestrian passings with 7148 single-frame annotations. Due to the inclined viewing angle, the scale at which pedestrians are depicted is limited and mutual occlusions are only partial, depending on the density of the crowds. For all cameras, the parameters of the interior and exterior orientations are available together with the images.

KITTI

For the moving camera set-up, the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago) Object Tracking Evaluation 2012 dataset from the KITTI Vision Benchmark Suite (Geiger et al., 2012), is employed. In the KITTI dataset, the camera was mounted on a moving vehicle at a height of 1.65m above the ground. The dataset comprises 21 image sequences for training

[1] Reference data is available online (http://www.milanton.de/data/, accessed on September 2016). Previous publications that report results on that challenge, (Klinger et al., 2015, 2016), were trained on the training data of the 3DMOT benchmark only.


(a) PETS09-S2L2


(b) AVG-TownCentre


(c) Image from the KITTI training sequence 0016

Figure 5.1: Example images from the (a) PETS09-S2L2, (b) AVG-TownCentre and (c) KITTI dataset.

and 29 image sequences for testing (see Figure 5.1(c) for an example image). Stereo image data is available for all sequences. In this work, only the image data captured by the left camera is used. The KITTI dataset is designed for the evaluation of tracking different object classes (vehicles and pedestrians) and only five training sequences (set numbers 13, 15, 16, 17 and 19) and eleven sequences of the test dataset (set numbers 18–28) are characteristic for person tracking. Due to the nearly horizontal viewing direction at pedestrian height, the dataset comprises a large range of depth and severe mutual occlusions of the pedestrians. The exterior camera orientation was computed using the library for visual odometry of Geiger et al. (2011) prior to the actual processing.

Evaluation criteria

The evaluation metrics used in the literature address different criteria of the tracking results, which are related to the completeness and correctness of all automatic single-frame annotations (i.e., the posterior image position described by the rectangle surrounding a person) in total and per object, to logical errors in the detection-to-track assignment and to the geometrical accuracy of the generated results.

In accordance with the literature (Leal-Taixé et al., 2015), this work reports the numbers of false positive (FP) and false negative (FN) detections, and, based on these values, the number of false positive detections per image (FPPI), recall ($\frac{TP}{TP+FN}$) and precision ($\frac{TP}{TP+FP}$), where TP is the number of correct detections. These metrics assess the ability of the tracker to detect and localise all persons in all frames independently from the identity of the result. To account for the identity of the generated detections, the *mostly tracked* and *mostly lost* metrics (Li et al., 2009) are used. A person is considered as mostly tracked (MT) if it is tracked at least 80% of the time being present in consecutive images, and as mostly lost (ML) if it is tracked at most 20% of this time. Furthermore, the number of identity switches (IDS) and fragmentations of trajectories (FRAG) are reported. Finally, the *classification of events, activities and relationships* (CLEAR) metrics Multi Object Tracking Accuracy (MOTA) and Multi Object Tracking Precision (MOTP, Bernardin and Stiefelhagen, 2008) are reported. The MOTA metric is a combined measure of tracking errors, taking into account the numbers of FPs, FNs and IDS per frame over the sum of all reference annotations gt per frame,

$$\text{MOTA} = 1 - \frac{\sum_t FN_t + FP_t + IDS_t}{\sum_t gt_t}, \tag{5.1}$$

and is normalised to lie in a range of minus infinity to one (100%), where a value of one indicates

that no errors occurred. The MOTP metric reflects the positional accuracy of the results and is evaluated either in 2D as the average intersection-over-union score or as the average displacement from a reference position in 3D (Leal-Taixé et al., 2015), i.e.,

$$\text{MOTP} = 1 - \frac{\sum dist(\mathbf{X}, \mathbf{X}_{ref})}{\theta_{dist} \sum TP}, \tag{5.2}$$

where $dist(\mathbf{X}, \mathbf{X}_{ref})$ is the 3D distance between the estimated 3D position of the target and the reference position, and θ_{dist} is the acceptance threshold applied to the distance. By convention, this threshold is set to one metre. A value of MOTP= 1 (100%) corresponds to the best possible accuracy. Additional evaluation criteria are used in Section 5.2 to investigate the individual components of the model.

Settings of the parameters

An overview over the settings of the free parameters in the individual sections is given in Table 5.1, for their explanation see Table 4.3. Grey fields indicate that the respective parameter is modified within the associated section. Dashes indicate that the respective parameter has no impact on the experiments in that section. Fields with two values show the chosen values for the PETS and AVG datasets and for the KITTI datasets. If only one value is given, it is used for all datasets in the associated section.

Sec.	θ_{svm} [–]	ρ_{det} [–]	σ_d [px]	μ_P [m]	σ_P [m]	σ_{kde} [px]	ϵ_{nms} [–]	η_{tre} [–]	η_{dep} [–]	η_{tes} [–]	η_{sam} [–]	η_{str} [–]	η_{que} [s]	l [m]	σ_f [km/h]	θ_α [deg]	η_h [m]	θ_{ini} [–]	θ_{abs} [s]	ρ_n [px]
5.2.1			10	1.7	0.1	10	0.2	-	-	-	-	-	-	-	-	-	-	-	-	-
5.2.2	-	-	-	-	-	-	-							-	-	-	-	-	-	-
5.2.3	0.0/ 0.6	1.02/ 1.05	10	1.7	0.1	10	0.2	100	6	4	10	48	3					0.5	1	50
5.2.4	0.0/ 0.6	1.02/ 1.05	10	1.7	0.1	10	0.2	100	6	4	10	48	3	2.75	7.5	90	1	0.5		50
5.3	0.0/ 0.6	1.02/ 1.05	10	1.7	0.1	10	0.2	100	6	4	10	48	3	2.75	7.5	90	1	0.5	1/ 0.5	50
5.4.1	0.0/ 0.6	1.02/ 1.05	10	1.7	0.1	10	0.2	100	6	4	10	48	3	2.75	7.5	90	1	0.5	1/ 0.5	50
5.4.2	-	-	-	-	-	-	-	-	-	-	-	-	-	2.75	7.5	90	1	-	-	-
5.4.3	0.0/ 0.6	1.02/ 1.05	10	1.7	0.1	10	0.2	100	6	4	10	48	3	2.75	7.5	90	1	0.5	1/ 0.5	50

Table 5.1: Parameter settings. See Table 4.3 for a description of the parameters. When two values are given, the first one refers to the PETS and AVG datasets and the second refers to the KITTI dataset. Grey fields indicate that the respective parameter is trained in that section. Dashes mean that the parameter has no effect on the experiment in the respective section.

The covariance of the measurement noise is determined from the kernel density estimation, so that individual quality measures for the measurement vector are assigned automatically. For the covariance of the process noise, the accelerations in X and Z directions, are set based on the outcomes of the Gaussian Process Regression (cf. prediction step in Section 4.1). The expected velocities in Y and

H are set based on the expected deviation from the flat-world assumption to allow for an inclined ground plane and on the expected magnitude of change in a person's height due to gait and articulated movements. To account for the process noise, $\sigma_{vY} = 1mm/s$ and σ_{vH}=$10mm/s$ is set for the static camera set-ups, and $\sigma_{vY} = \sigma_{vH} = 10mm/s$ for the tracking in dynamic scenes, with the higher variance in Y to account for potentially changing terrain. σ_π is assigned a comparatively small value of $1mm$. The initial covariance of the filter state, $\Sigma_{ww,t=0}$, is assigned with σ_X=σ_Z=$0.3m$, σ_Y=$0.01m$, σ_H=$0.03m$ and σ_{vX}=σ_{vZ}=$0.1m/s$.

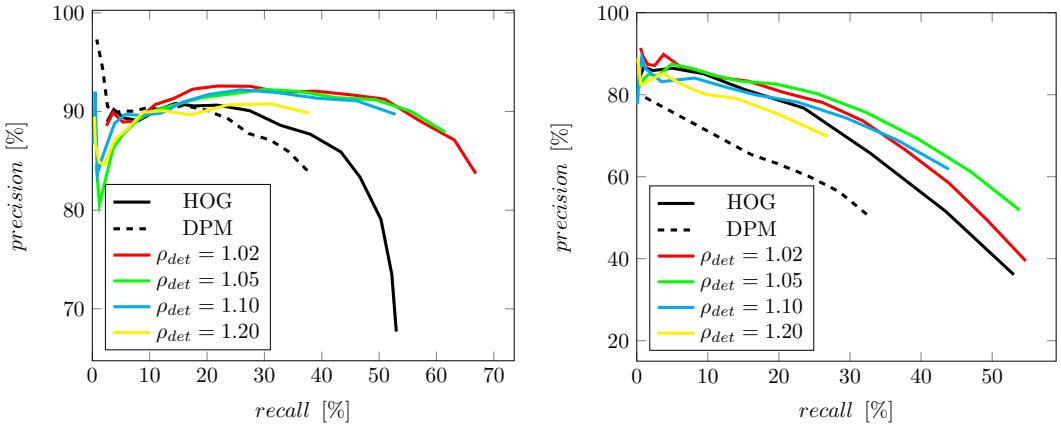
5.2 Sensitivity study and training

This chapter is divided into four parts following the grouping of the model parameters given in Table 4.3. In each part, the free parameters of the associated system component are investigated independently from the other components. As these groups form self-contained systems, the presumed independence between the parameters of different groups is a valid assumption.

5.2.1 Detector

The detection strategy depends on seven parameters. One parameter defines the acceptance threshold θ_{svm} for the distance of a sample from the SVM hyperplane, and another one is the coefficient of the detection window increase ρ_{det}, as part of the HOG/SVM detector. These parameters are investigated in this section. The other parameters of the detection model are kept constant throughout the experiments. For the false positive reduction, a normal distribution $H_D \sim \mathcal{N}\left(\mu_D, \sigma_D^2\right)$ is assumed for the detections, where the mean equals the projected size of a detection to object space and the standard-deviation σ_D is computed by variance propagation from the detection variance in image coordinates, for which a constant value of $\sigma_d = 10px$ is assumed. For pedestrians, a normal distribution $H_P \sim \mathcal{N}\left(\mu_P, \sigma_P^2\right)$ with $\mu_P = 1.7m$ and $\sigma_P = 0.1m$ is taken. For kernel-density estimation, a Gaussian kernel function with $\sigma_{kde} = 10px$ is also chosen heuristically for all experiments. For the statistical test used for false positive reduction, a probability of error of $\alpha = 5\%$ is assumed.

For the investigation of the parameters of the detector, the strategy proposed in Section 4.2.2 is applied to the training sequences PETS09-S1L1-1 and KITTI-0019, which comprises 1059 frames. For these sequences, the detection is applied to every image and the recall and precision of the detection results are averaged over the entire sequences. The results are reported in Figure 5.2, where the curves are generated by using the detection strategy at different values of the acceptance threshold θ_{svm}. The right-most point of every curve corresponds to a value of $\theta_{svm} = 0$. The experiment is executed for different values of the detection window increase ρ_{det}, as shown by the coloured curves. For comparison, results from a HOG detector (Dalal and Triggs, 2005), which this work is based on, and from a Deformable Part Model (DPM, Felzenszwalb et al., 2010) are shown as baselines. The increase of precision compared to the HOG and DPM detector is due to the fact that, if the underlying detector is applied permissively, high recall rates are achieved at the cost of many FPs, many of which can be rejected using the proposed strategy. For both datasets, the highest recall values are achieved using the proposed strategy with an acceptance threshold of 0.0 and a coefficient ρ_{det} of 1.02. For



(a) PETS09-S1L1-1 (b) KITTI-0019

Figure 5.2: Precision over recall. The coloured curves are generated at different values of the detection window increase ρ_{det}. The black curves are generated by the baseline detectors DPM and HOG.

the PETS09-S1L1-1 sequence, an increase of ρ_{det} to a value of 1.05 yields a higher precision at the cost of a lower recall. For the KITTI sequence, the same variation of coefficient ρ_{det} yields a higher precision at a similar recall value. Thus, the parameters $\rho_{det} = 1.02$ and $\rho_{det} = 1.05$ are chosen for the PETS 2009 and KITTI data in the remainder of the experiments, respectively. Because on the KITTI dataset, the precision decreases rapidly at increasing recall rates, an acceptance threshold of $\theta_{svm} = 0.6$, at which a recall of 0.32 and a precision of 0.76 is achieved, is taken for the KITTI dataset.

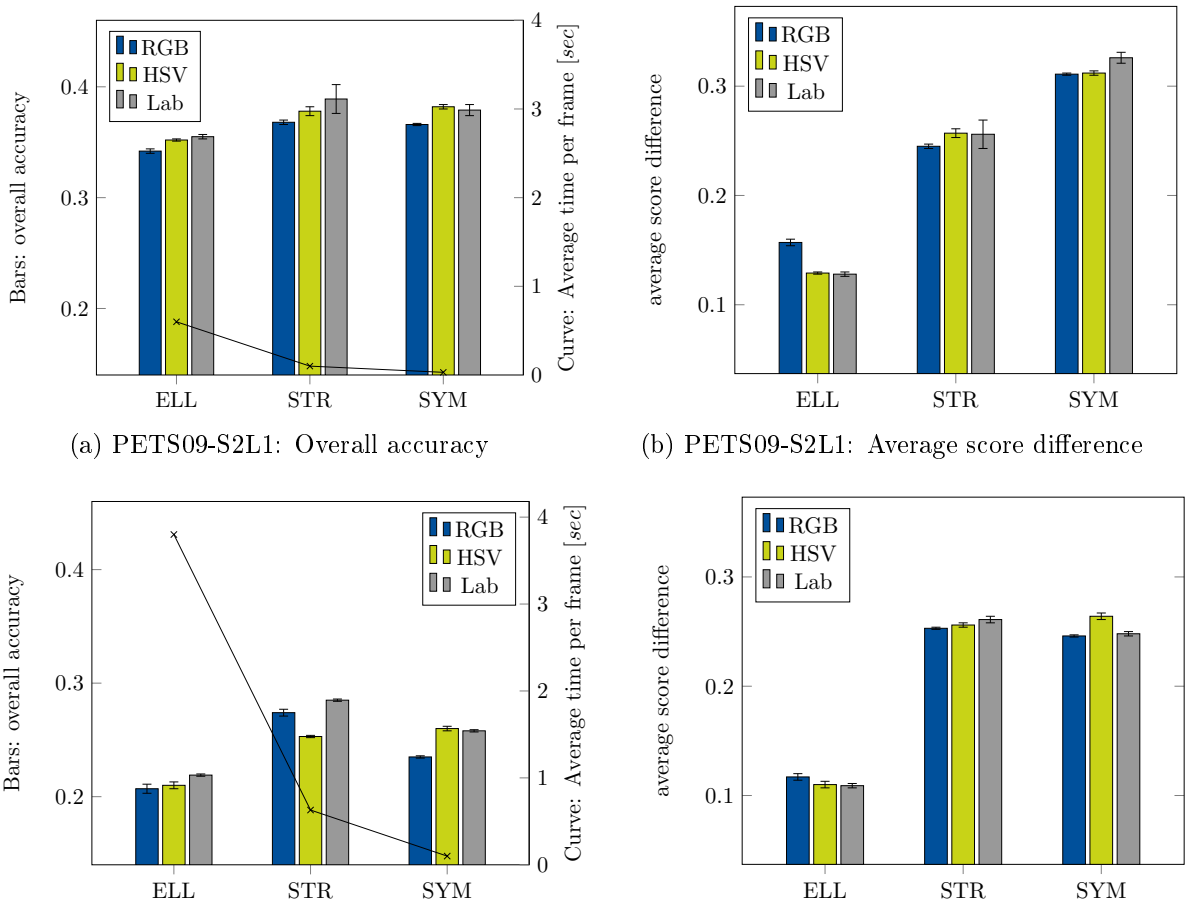
5.2.2 Classifier

In this section, the three different models for feature extraction proposed in Section 4.2.3 and the parameters of the classification strategy are analysed. Two different metrics are used here to assess the capability of the classifier to discriminate between individual persons: The overall accuracy, which reflects the percentage of correctly classified persons, and the average score difference (ASD), which is the difference between the classification score of all correctly classified persons from the class with the second highest score, averaged over all classifications. High overall accuracies are desirable to compute reliable weights in the data association. For localisation by classification, a large score difference is desirable to emphasise the region in the image associated to the desired target, while neglecting the influence of other persons on the measurements. Furthermore, the average runtime per frame is reported along with the classification metrics.

Model selection. In a first experiment, the classifier is applied using the model variants Ellipse (ELL), Stripes (STR) and Symmetry axes (SYM), as described in Section 4.2.3. The experiments in this section are carried out on two different image sequences from the PETS 2009 dataset with different numbers of persons passing the scene, the PETS09-S2L1 and PETS09-S1L1-1 sequence, which have different levels of complexity: The PETS09-S2L1 sequence never contains more than 8 persons at a time, while the PETS09-S1L1-1 sequence shows up to about 30 persons in one image. Because the

classification results are assumed to be independent from the camera orientation, these tests are not carried out on the KITTI dataset. Features are extracted from the RGB, HSV and Lab colour space. For this experiment the ORF is trained based on the annotations provided by the reference data, in order to avoid a degradation of the classification results caused by misaligned samples. In each frame, the rectangles provided by the reference data are classified before the classifier is updated using new samples from these rectangles. To account for the stochastics of the Random Forest classifier, each test was run five times and the means and standard deviations of all runs are presented here. The size of the feature vector is different in each model, with 2622 features for the ELL model, 288 features for the STR model, and 12 features for the SYM model.

Figures 5.3 (a) and (c) show the overall accuracy in form of bars with the error bars indicating the standard deviation of the overall accuracy, and Figure 5.3 (b) and (c) show the average score differences and standard deviations for the proposed models, for the proposed colour spaces and for both datasets.



(a) PETS09-S2L1: Overall accuracy

(b) PETS09-S2L1: Average score difference

(c) PETS09-S1L1-1: Overall accuracy

(d) PETS09-S1L1-1: Average score difference

Figure 5.3: Classification performance for different model and feature compositions. The *Ellipse* (ELL), *Stripes* (STR) and *Symmetry axes* (SYM) models are tested in combination with the colour spaces RGB, HSV and Lab. The bars show the achieved scores, the error bars show their standard-deviations. The curves in (a) and (c) show the average runtime per frame.

The highest overall accuracy is achieved using the STR model together with the features from the Lab colour space for both datasets. In terms of the average score difference, the SYM model with Lab features performs best on the PETS09-S2L1 dataset, whereas the SYM model with HSV features performs best for the PETS09-S1L1-1 dataset, closely followed by the STR model with Lab features. The ELL model performs worst in terms of all metrics with all colour spaces on both datasets. The runtime decreases with the number of samples used in each model, so that the ELL model is also the slowest due to the highest dimensionality of the feature space, whereas the SYM model is the fastest.

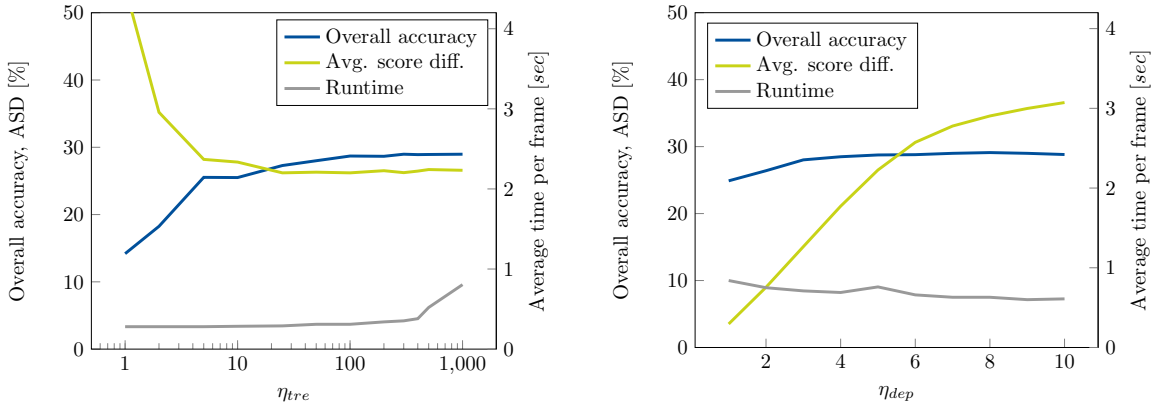
The absolute values show that in the best case (STR and Lab) 39% for PETS09-S2L1 and 28% for PETS09-S1L1-1 of the highest classification scores correspond to the desired target.

As the best overall accuracies are achieved using the STR model with Lab features, this feature extraction strategy is used for the remainder of the experiments.

Parameter selection. In a second experiment, the free parameters of the classification strategy (cf. Table 4.3) are investigated. Here, the experiments are carried out only on the PETS09-S1L1-1 sequence, which is among the most complex sequences from the PETS datasets and assumed to be representative for the other sequences, too. In Figure 5.4, the results for different parameter settings are shown. Independence between the variables is assumed and every parameter is varied along a predefined range. Default values for the parameters not modified during an experiment are set to $\eta_{tre} = 100$, $\eta_{dep} = 6$, $\eta_{tes} = 4$, $\eta_{sam} = 10$, $\eta_{str} = 48$, $\eta_{que} = 1s$. Figure 5.4 shows the overall accuracy and average score difference for different (a) number of trees, (b) maximal depths of the trees, (c) number of random tests considered at every node of each tree for splitting, (d) the minimum number of samples at each node required for splitting, (e) the number of stripes of the STR model, and (f) the time span samples are stored for re-initialisation of the classifier. The overall accuracy, ASD and processing time per frame are reported for every parameter.

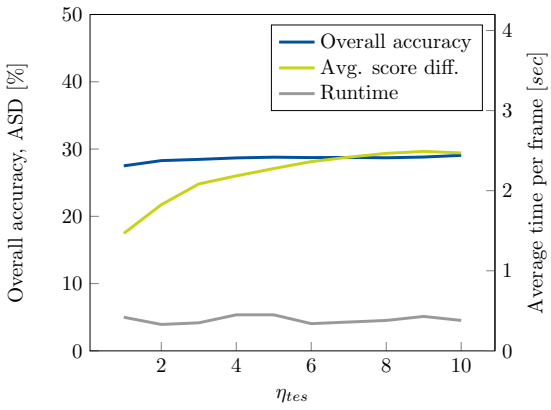
The number of trees affects both overall accuracy and ASD. The overall accuracy increases up to about 100 trees and then stagnates, while the ASD decreases rapidly down to a number of about 20 trees. The runtime increases rapidly as the number of trees exceeds 500. A value of $\eta_{tre} = 100$ is chosen for the remainder of the experiments. Varying the depth of the trees affects both metrics. The overall accuracy does not increase considerably for depth values larger than 3, whereas the ASD increases strictly monotonically up to a depth of 10. The depth of the trees should be high enough to build enough leave nodes to represent the available classes, while an overly deep tree makes the classifier prone to overfitting to the training data. A value of $\eta_{dep} = 6$ is chosen for the remaining experiments, which results in trees having up to 64 leaves, which is more than the expected number of classes in all test cases, so that every class (pedestrian) can be represented by at least one leave node in every tree. The number of random tests per node does not have a major impact on the overall accuracy, while the ASD increases strictly monotonically with growing parameter values and stagnates from approximately 8 tests. Hence, the value of 8 is chosen for the parameter η_{tes}. Parameter η_{sam} has no major impact on the overall accuracy for the experimental data either, whereas the ASD has a distinguished maximum at $\eta_{sam} = 6$, so that this value is appropriate for the remaining experiments. The number of stripes into which the bounding rectangle is divided affects both metrics in the parameter range between 1

and 24, after which both metrics stagnate. The value of 24 for parameter η_{str}, thus, is appropriate. The number of time steps for which samples are stored to re-train the classifier influences both metrics in the way that the overall accuracy is maximal at 20 and the ASD value stagnates at values larger than 20, so that $\eta_{que} = 20$ was chosen as optimal value for this application, which corresponds to a time span of about $3s$ at a frame rate of $7Hz$.

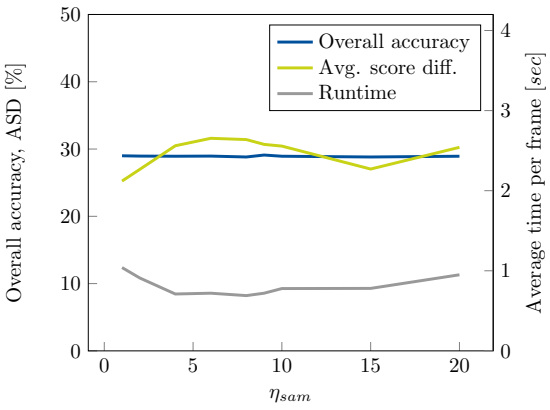


(a) Number of trees

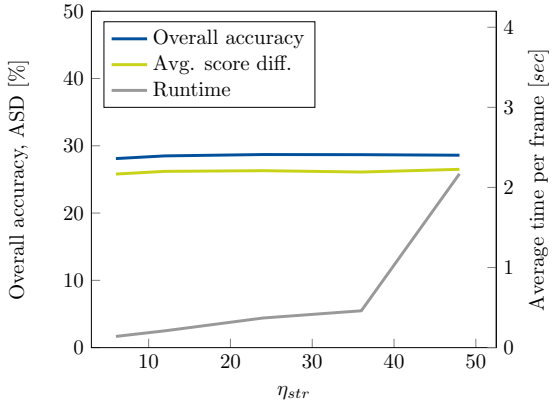
(b) Depth of the trees



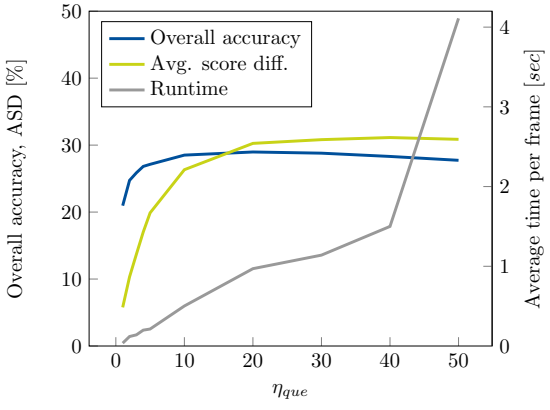
(c) Random tests per node



(d) Minimum number of samples for splitting



(e) Number of stripes



(f) Number of time steps to store samples

Figure 5.4: Classification performance for different parameter settings in the classification model.

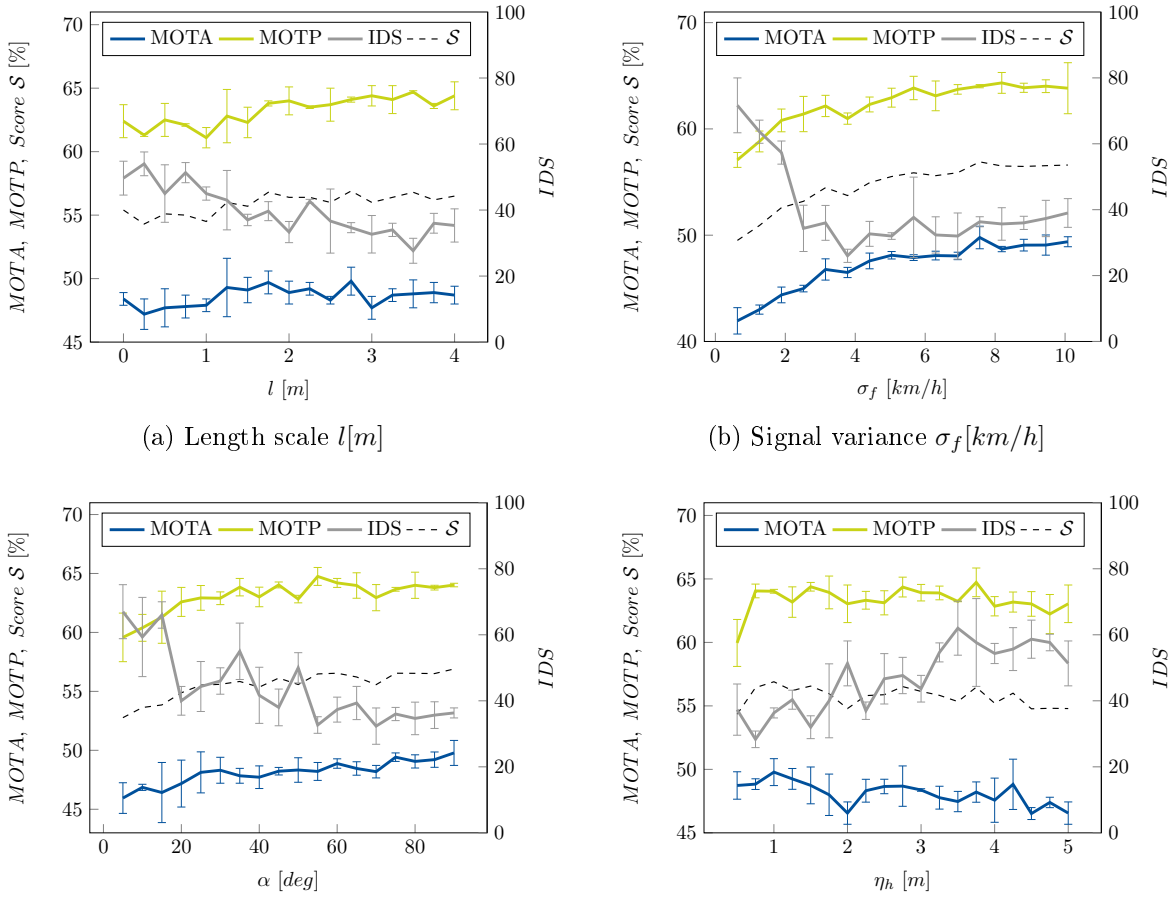
5.2.3 Temporal model

The temporal model, which involves the Implicit Motion Context (IMC), is guided by four parameters. The IMC parameters are the length-scale l, the signal variance σ_f^2, the angular threshold θ_α and the history h, which is the number of recent trajectory points used to compute the covariances and the trend. These parameters are learnt from the PETS09-S1L1-1 sequence using the direct search approach (Hooke and Jeeves, 1961). The initial values are $l = 2.0m$, $\sigma_f = 7.0km/h$, $\theta_\alpha = 90°$ and $\eta_h = 1m$. To find the parameters $\mathbf{p} = \{l, \sigma_f, \alpha_0, \eta_h\}$ that yield optimal results on the training data, the argument variables $\hat{\mathbf{p}}$ that solve the maximisation problem

$$\hat{\mathbf{p}} = \underset{\mathbf{p}}{\operatorname{argmax}} \mathcal{S}(\mathbf{p}), \tag{5.3}$$

with the score function $\mathcal{S}(\mathbf{p}) = (\mathrm{MOTA}(\mathbf{p}) + \mathrm{MOTP}(\mathbf{p}))/2$, are taken. Both metrics are considered equally important in the optimisation. Using the direct search approach one parameter is changed at a time, keeping the others fixed. To account for stochastic results, which appear, for instance, because of the Random Forest classifier, the processing with each parameter was executed three times and the results are averaged. The parameters yielding the best results are kept constant during the variation of the next parameter. This procedure was repeated for six iterations. In Figure 5.5, the results of the training after the final iteration are visualised. The Figure is divided into four parts, each showing the results achieved upon variation of one parameter. For each parameter the MOTA, MOTP, the number of ID switches, which is related to the MOTA metric, and the score function $\mathcal{S}$ are visualised. Only parameter values at fixed interval points are tested, as indicated by the error bars that show the 1σ intervals of the metrics. The parameters associated to the peaks of the score function (dashed line) are taken as optimal values.

The values of $l = 2.75m$, $\sigma_f = 7.5km/h$, $theta_\alpha = 90°$ and $\eta_h = 1m$ were determined to yield optimal results. The length-scale parameter indicates that interactions take place in a radius of about $3m$ around a person. The signal variance σ_f^2 controls the maximum range of velocities and limits the velocity estimates far from the input data. The value achieved by the training, thus, indicates that, when a person cannot be observed and no other persons contribute to the estimation of the velocities, it positively affects the result to assign a comparatively large value of $7.5km/h$ to σ_f. The angular threshold of 90° means that considering all persons moving with an angular displacement of at most 90° positively affects the tracking results. The length of the trajectories $\eta_h = 1m$ indicates that only the last metre of the trajectories contributes to the tracking; if longer parts of the trajectories are taken into account, sudden changes in the direction of motion do not affect the covariances of the trajectories, and the performance decreases. The determined values are used as parameters in the remainder of the experiments. Furthermore, it can also be concluded from the training that the method is not overly sensitive to these parameters. In terms of statistical significance, tested at an error probability of 5%, the difference between the lowest and the highest score achieved by variation of parameter l is significant. The same holds true for parameters σ_f and θ_α. Parameter η_h does not have any significant impact on the score function. However, the maximum value at $\eta_h = 1.0m$ and a comparatively low IDS rate justify the selection of that value for the remaining experiments.



(a) Length scale $l[m]$

(b) Signal variance $\sigma_f[km/h]$

(c) Angular threshold $\theta_\alpha[deg]$

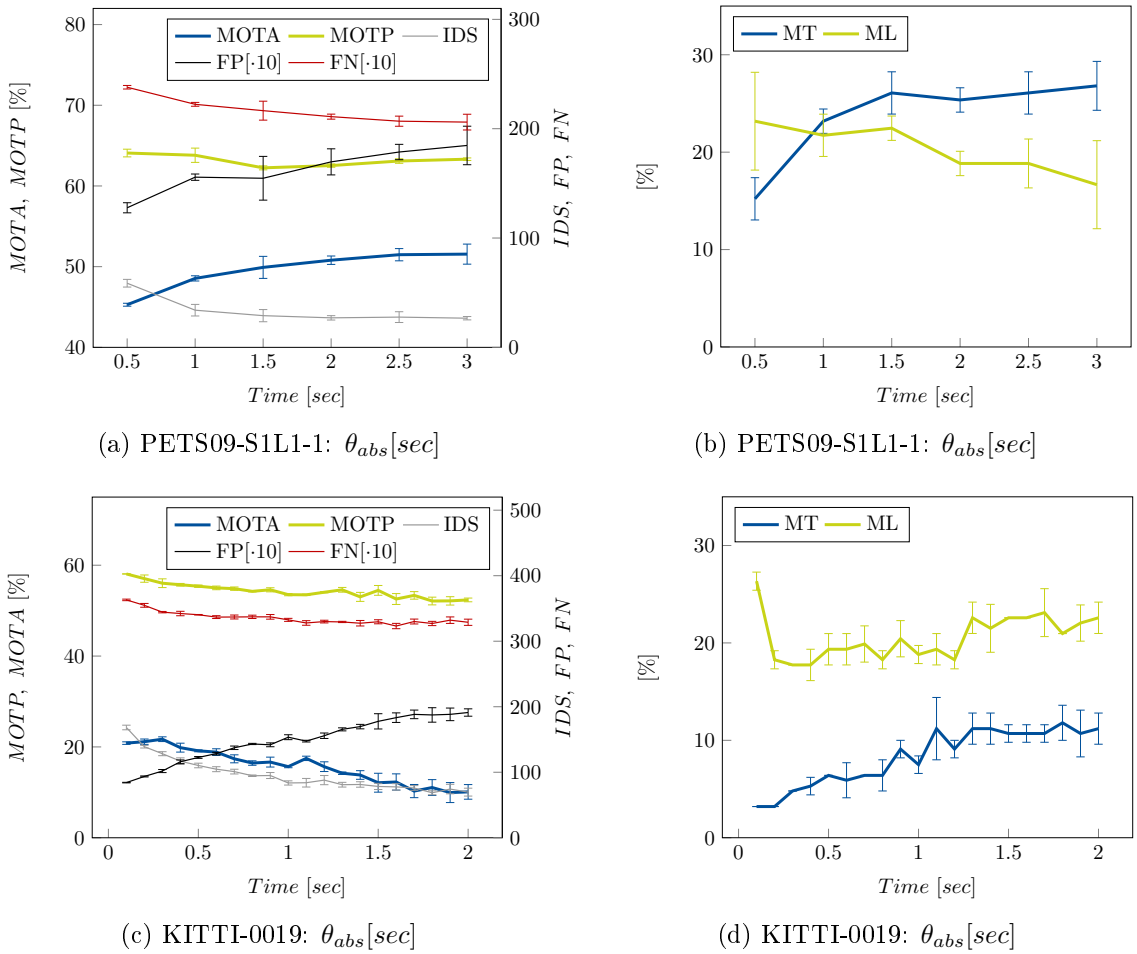
(d) Length of trajectories $\eta_h[m]$

Figure 5.5: Tracking performance for different parameter settings in the temporal model.

5.2.4 Recursive filter

In this section, the free parameters of the recursive estimation framework are addressed. The first parameter θ_{abs} controls the maximum time span to wait for the termination of a trajectory in the absence of measurements. The second parameter, the confidence threshold for the initialisation of new trajectories θ_{ini}, is set to a constant value of 0.5. These parameters control the initialisation and termination of the trajectories. The third parameter, ρ_n, which is referred to as the measurement noise coefficient, is set to a constant value of $50px$.

The parameter θ_{abs} is evaluated with respect to the metrics MOTA, MOTP, IDS, MT and ML. The scores in these metrics are plotted in Figure 5.6 (a) and (b) for PETS09-S1L1-1 and in Figure 5.6 (c) and (d) for KITTI-0019. The experiments show that the parameter θ_{abs} has a significant impact on the evaluation metric MOTA, which includes the number of identity switches (IDS), whereas the impact on the MOTP is not significant. The parameter controls the time to wait before terminating a trajectory. When tracking of a subject is finished after the first few frames where no measurements are obtained to update the trajectory, possible occlusions, which are inherent in the regarded scenes, cannot be bridged successfully. This is reflected in the number of mostly lost targets (ML): A target for which tracking is stopped after the first time a measurement cannot be obtained, e.g. during an



(a) PETS09-S1L1-1: $\theta_{abs}[sec]$

(b) PETS09-S1L1-1: $\theta_{abs}[sec]$

(c) KITTI-0019: $\theta_{abs}[sec]$

(d) KITTI-0019: $\theta_{abs}[sec]$

Figure 5.6: Variation of the parameter θ_{abs} for PETS09-S1L1-1 and KITTI-0019.

occlusion or due to other disturbing effects, is unlikely to be resumed for tracking while the disturbing effect persists. As the number of ML targets decreases, the number of mostly tracked targets increases. The IDS are also reduced with growing parameter values. Due to the persistent tracking of an object, when the absence count threshold is high, the class-statistics of the classifier are stored for a longer time and lead to more reliable data association over time. This comes at the cost of more false positive detections, which arise, if predicted trajectories drift away from the target. For the remaining experiments, a value of $\theta_{abs} = 1s$ is taken for the PETS dataset and $\theta_{abs} = 0.5s$ for the KITTI datasets, because these values yield reasonable trade-offs between the considered tracking metrics.

5.3 Model validation by ablation of its components

Having determined the parameters that yield optimal results on the training datasets, the aim is now to show the benefits of using the full model as proposed in this thesis empirically. To this end, the relevant building blocks of the proposed method are omitted from the method to demonstrate the impact of the corresponding component on the tracking and localisation ability of the tracker, as measured by the loss of performance in the relevant metrics. All metrics described in Section 5.1 are reported for the full model and for every modified version of the tracker, and the statistical significance

of any differences relative to the full model are indicated accordingly.

The tests are divided into two groups. In the first group, the initialisation of new trajectories is based on reference data, i.e. the initial position is given. In this way, errors related to omission or commission errors of the underlying detection strategy are avoided. In the second group of experiments, the initialisation is carried out based on automatic detection results. Different model variants, in which single system components are omitted, are tested within the first group. The omission of single components is assumed to affect the performance of the full model in the first group in the same way as in the second group. Thus, these models are not tested in the second group.

The first group of tests investigates the full model, referred to as model variant (a) *Full model with given initialisation* and four modified versions of the tracker. In variant (b) *Omission of* n, the analysis of the additive measurement noise estimated based on mutual occlusions and prior knowledge about the scene (cf. Section 4.5.1) is omitted. In variant (c) *No context*, the Implicit Motion Context is omitted from the system, so that all trajectories are continued by a temporal model with constant velocity assumption individually for all persons. In this way, this model variant resembles the predictive function applied in a standard Kalman Filter model. In model variant (d) *No classification*, the position of the feet is modelled as observation instead of a hidden variable and the position of the feet is measured by the detection model only (cf. Section 4.2.2), and in variant (e) *No detection* the positions of the feet are measured by the classification approach (cf. Section 4.2.3). The initialisation of the tracker with automatic initialisation is investigated in model variants (f) *Full model with automatic initialisation* and (g) *No 3DFPR*, where variant (f) represents the full and variant (g) omits the false positive reduction strategy (3DFPR) proposed in Section 4.2.2.

Experiments are conducted on the PETS09-S1L1-1 sequence and on the KITTI-0019 sequence. The results are reported in Tables 5.2 and 5.3, respectively. To account for stochastic results, induced for instance by the Random Forest classifier, every model variant is applied five times. For every metric, the mean value is listed in the tables. To compare the results between the full models (a) and (f) and the model variants, statistical significance tests (Student's t-tests) are applied, for which an error probability of 5% is assumed. The table cells highlighted in red indicate that the performance is significantly worse than the one for the full model, green indicates that the performance is significantly better than the full model, and white indicates no significant change. Exemplary images showing the results of every model variant are given in Figures 5.7 and 5.8, respectively.

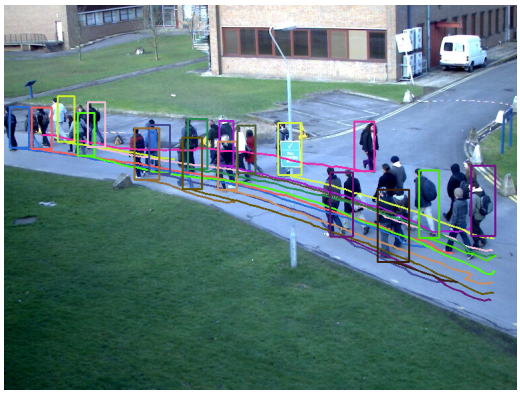
Table 5.2 shows that none of the other models (b) – (e) could improve any metric significantly for the PETS09-S1L1-1 sequence. In contrast, the full model achieves the highest scores in terms of precision, FPPI, FP, IDS and FRAG rates, and in the MOTP metric, though the differences are only significant for the MOTP metric in contrast to model variants (c) *No context* and (e) *No detection*. In variants (b) and (c) the performance decreased slightly in terms of precision, FPPI, FP, IDS, FRAG and MOTA. In variant (e), the performance deteriorates in terms of all metrics except for the number of mostly lost targets (ML). When only the detector is omitted from the model, the performance improves slightly in terms of recall, ML, FN and MOTA, though the differences are not significant. Collectively, these results reveal the benefits of using the proposed method. If either the estimation of the variable n or the motion context is omitted, the precision rate decreases slightly. The estimation of

Initialisation	Model	Recall [%]	Precision [%]	False Positives Per Image [-]	Mostly Tracked (MT) [%]	Mostly Lost (ML) [%]	False Positives (FP) [-]	False Negatives (FN) [-]	Identity Switches (IDS) [-]	Fragmentations (FRAG) [-]	Multi Obj. Tracking Accuracy (MOTA) [%]	Multi Obj. Tracking Precision (MOTP) [%]
Given	(a) Full model	52.9	**94.6**	**0.6**	24.3	22.6	**141**	2206	**34**	**126**	49.2	**65.0**
	(b) Omis. of n	52.9	93.7	0.8	24.3	23.0	166	2208	37	133	48.6	64.0
	(c) No context	53.7	93.5	0.8	**25.2**	20.0	175	2173	37	130	49.1	61.7
	(d) No classif.	**53.9**	94.5	0.7	23.9	**18.2**	164	**2161**	35	135	**50.0**	64.5
	(e) No det.	40.0	72.5	3.2	12.6	32.6	714	2815	69	163	23.3	40.3
Aut.	(f) Full model	**53.2**	**89.4**	**1.4**	**27.0**	**19.1**	**298**	**2193**	**26**	**124**	**46.3**	61.4
	(g) No 3DFPR	52.4	80.1	2.8	23.5	19.6	612	2230	70	169	37.9	**62.7**

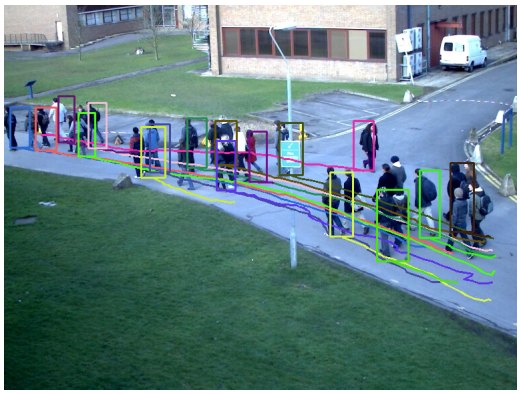
Table 5.2: Sensitivity study for the omission of single components of the method tested on the PETS09-S1L1-1 sequence. Best values are printed in bold. Grey cell indicate the baseline metrics of the full model, red cells indicate significant degradation compared to the full model, white cells indicate no significant change. See text for details.

n reduces the impact of the inferred image positions on the posterior state estimate in cases where the person is occluded or unlikely to occupy the estimated image position. If an occlusion occurs, parts of the occluded person can often be observed, nevertheless, but the reliability in the determined position decreases due to the limited visibility. The consideration of this effect, hence, improves the results. The motion context yields more reliable predictions in the absence of measurements, i.e. if a person is fully occluded or if the detection of that person fails. Using the stand-alone prediction as in variant (c) *No context*, the predicted positions are prone to drift away from the target in such situations, which causes false positive detections and, thus, affects the precision of the results. Note the zigzag course of the trajectories in Figures 5.7(b) and (c) and the rather smooth lines in Figure 5.7(a). Not using the classifier from the observation model, variant (d), decreases the performance in terms of fragmentations of the trajectories slightly. This effect can best be explained by the redundancy in the observations related to a specific target. Using different independent measurements of the target's position, gaps in the observations of a single component of the observation model can be bridged and the trajectory is terminated and re-initialised less frequently, which is reflected in the number of fragmentations. These fragmentations can be observed in Figures 5.7(d) and (e), where the generated trajectories often begin further away from the image border, where the persons actually appear.

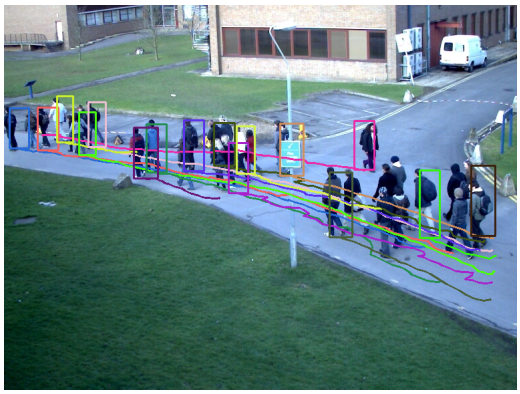
Using the automatic detection results for initialisation in variants (f) *Full model* and (g) *No 3DFPR*, the false positive rate increases compared to the initialisation based on reference data (cf. variant (a)), and the precision of the results, as well as the MOTA and MOTP metrics deteriorate. When the false positive reduction strategy is not used, comparing variant (g) *No 3DFPR* to variant (f) *Full model*, which uses that strategy, the performance decreases significantly in terms of the false positive rate, as expected, and in terms of precision, identity switches, fragmentation and MOTA. The incorporation of the false positive reduction in 3D, thus, improves the tracking considerably. As can be seen in Figure 5.7(g), the omission of the 3DFPR strategy leads to spurious trajectories initialised based on false



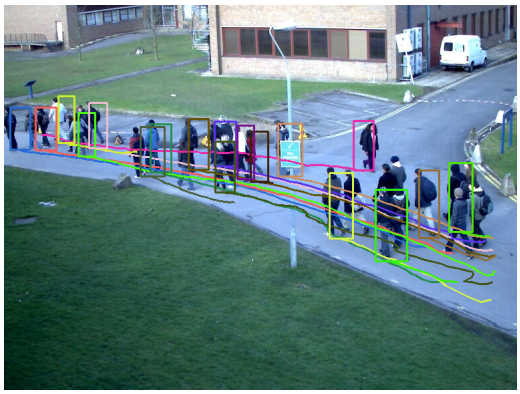
(a) Full model with given initialisation



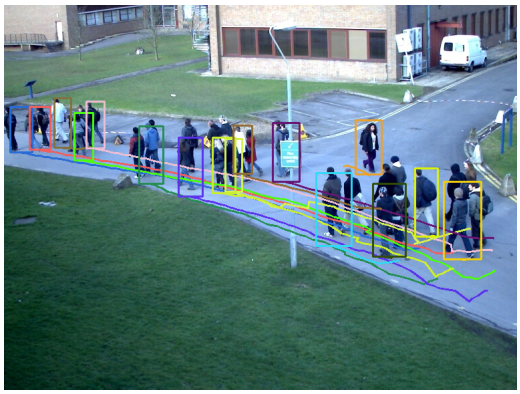
(b) Omission of n



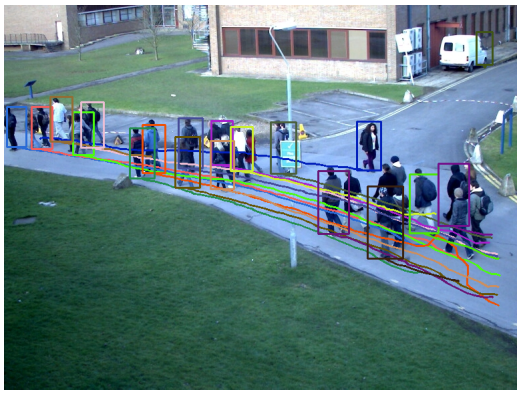
(c) No context



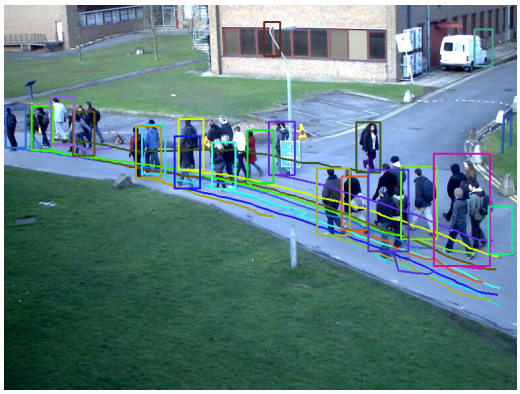
(d) No classification



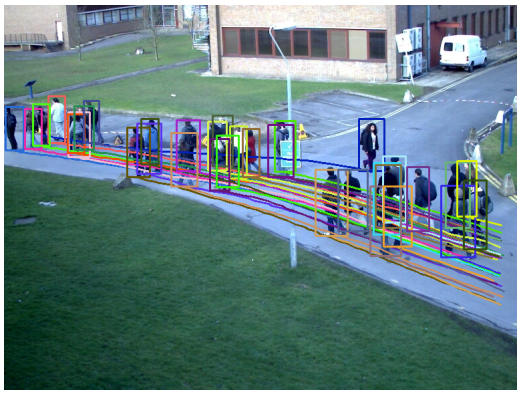
(e) No detection



(f) Full model with automatic initialisation



(g) No 3DFPR



(h) Reference

Figure 5.7: Qualitative results shown for frame number 94 of the PETS09-S1L1-1 sequence. The colours of the trajectories are chosen randomly.

Initialisation	Model	Recall [%]	Precision [%]	False Positives Per Image [-]	Mostly Tracked (MT) [%]	Mostly Lost (ML) [%]	False Positives (FP) [-]	False Negatives (FN) [-]	Identity Switches (IDS) [-]	Fragmentations (FRAG) [-]	Multi Obj. Tracking Accuracy (MOTA) [%]	Multi Obj. Tracking Precision (MOTP) [%]
Given	(a) Full model	41.9	71.1	0.9	**4.8**	**17.2**	999	3407	125	199	22.7	56.3
	(b) Omis. of n	38.2	**73.0**	**0.8**	2.2	26.9	**829**	3626	165	218	21.2	**58.0**
	(c) No context	**42.2**	71.6	0.9	**4.8**	18.8	985	**3387**	126	**199**	**23.3**	56.5
	(d) No classif.	41.7	71.1	0.9	3.2	21.0	992	3418	**124**	202	22.7	56.4
	(e) No det.	30.8	58.6	1.2	0.0	32.3	1277	4056	257	270	4.7	55.7
Aut.	(f) Full model	35.5	**52.5**	**1.8**	3.2	32.8	**1881**	3782	**102**	**156**	**1.7**	54.3
	(g) No 3DFPR	**38.6**	42.0	3.0	3.2	**27.4**	3121	**3601**	171	209	-17.6	**54.8**

Table 5.3: Sensitivity study for the omission of single components of the method tested on the KITTI-0019 training sequence. Best values are printed in bold. Grey cell indicate the baseline metrics of the full model, red cells indicate significant degradation, green cells significant improvement in the associated metric, compared to the full model, white cells indicate no significant change. See text for details.

positive detections on the façade of a building in the background, and on a group of persons on the right side of the image, which are avoided using the 3DFPR strategy as shown in Figure 5.7(f). The increase in the number of IDS in variant (g) compared to the full model (f) results from the spurious trajectories initialised based on false positive detections, which cannot be associated to any person persistently and, thus, pass over to nearby persons in the scene.

On the KITTI-0019 dataset, the results reported in Table 5.3 are not as clear as for the PETS data. Model variant (b) *Omission of n*, which leaves out the estimation of variable n based on mutual occlusions and prior scene knowledge, performs significantly worse in terms of recall, in terms of the number of mostly lost (ML) targets and FN and IDS, while the false positive rate is significantly lower than in variant (a) *Full model*. The omission of the Implicit Motion Context, variant (c) *No context*, has no significant impact on the performance on this dataset at all. This may be explained by the fact that persons observed in a pedestrian zone less frequently move in groups than on a campus (see results for the PETS dataset). As a consequence, the motion context does not assist the motion prediction, as the covariance function only takes into account persons close to each other. Not using the classifier for the observation model, variant (d), does not affect the results by any means, either. The omission of the detector, variant (e), leads to a decrease in all metrics, except for MOTP. Using the automatic detections for initialisation in variants (f) *Full model* and (g) *No 3DFPR*, the performance decreases in terms of almost all metrics, except for the number of IDS and FRAG, compared to the full model using the reference data for initialisation. Many tracking errors, thus, can be explained by wrong, i.e. missing or false positive detections. When the 3DFPR strategy is omitted from the full model, variant (g), the performance deteriorates in terms of precision, in the number of IDS and FRAG and in the MOTA metric. Here again, the use of 3D information for the validation of the detections improves the tracking results. However, the omission of that information also decreases the FN rate and, thus,

(a) Full model with given initialisation

(b) Omission of n

(c) No context

(d) No classification

(e) No detection

(f) Full model with automatic initialisation

(g) Without 3DFPR

(h) Reference

Figure 5.8: Qualitative results shown for frame number 602 of the KITTI-0019 sequence. The colours of the trajectories are chosen randomly.

the recall, which indicates that some persons that are actually present in the scene are detected, but not verified by the 3DFPR strategy.

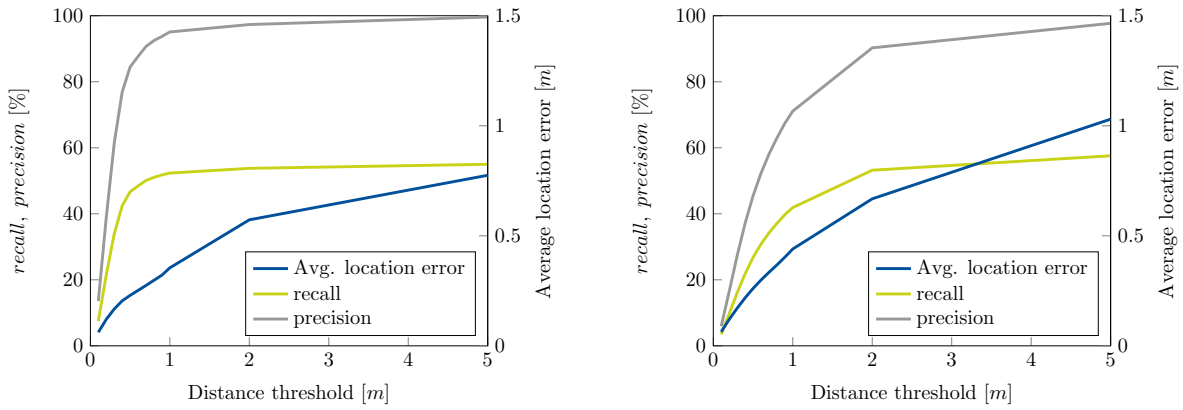
Figure 5.8 shows the results of all model variants for an exemplary image of the KITTI sequence. The qualitative results of the model variants (b) – (d) do not differ much from those of the full model (a). In variant (e) *No detection*, the trajectories are terminated and re-initialised more often, which can be seen by the rather short trajectories. When tracking is initialised based on the automatic detections, and the 3DFPR is omitted, a trajectory is initialised based on a detection with a wrong height on the left-most person in the scene and on the third person from the right, cf. Figure 5.8(g). When using the false positive reduction, the detections are aligned more accurately to the visual outlines of the persons (cf. Figure 5.8(f)).

5.4 Multi-person localisation and tracking evaluation

In this section, the actual localisation performance and the optimality of the recursive filter are investigated. The localisation accuracy is evaluated by the change in the tracking metrics in dependency of the acceptance threshold. The optimality of the filter is investigated exemplarily by comparison of the system innovations in a situation with simulated occlusions, with and without using the proposed model of Implicit Motion Context. Lastly, the method is compared to other methods from the related work based on two different benchmarks.

5.4.1 Localisation accuracy

To assess the localisation accuracy, the acceptance threshold for the validation of the estimated positions, i.e. the maximum allowable distance of an automatic annotation from the reference annotation in 3D, is varied. The achieved recall and precision and the average localisation error are plotted over the acceptance threshold in Figure 5.9 (a) for PETS09-S1L1-1 and in Figure 5.9 (b) for KITTI-0019. The plotted values are the average results of three times executing the tracking algorithm. As reflected by the precision curve, 95% of all automatic detections are correct with a maximum positional error of about $0.8m$ for the PETS data, whereas, for the KITTI sequence, 95% are only correct with at most $3m$ positional error. The average positional error of all correct assignments indicates that, within the range of an acceptance threshold of $1m$, as used as default threshold in all experiments, the average error amounts to $35cm$ for the PETS and to $44cm$ for the KITTI dataset. At a distance threshold of $0.5m$, the recall rate on the PETS dataset is almost saturated, whereas on the KITTI dataset, saturation only occurs for a threshold larger than $2m$. Below these values, the recall rates decrease rapidly.



(a) PETS09-S1L1-1 (b) KITTI-0019

Figure 5.9: Impact of the distance threshold on the PETS09-S1L1-1 and KITTI-0019 sequence.

5.4.2 Filter consistency

The optimality of a recursive filter can be evaluated based on the sequence of system innovations. If the innovation sequence is unbiased and white, the filter is free of systematic errors (Mehra, 1970). In this

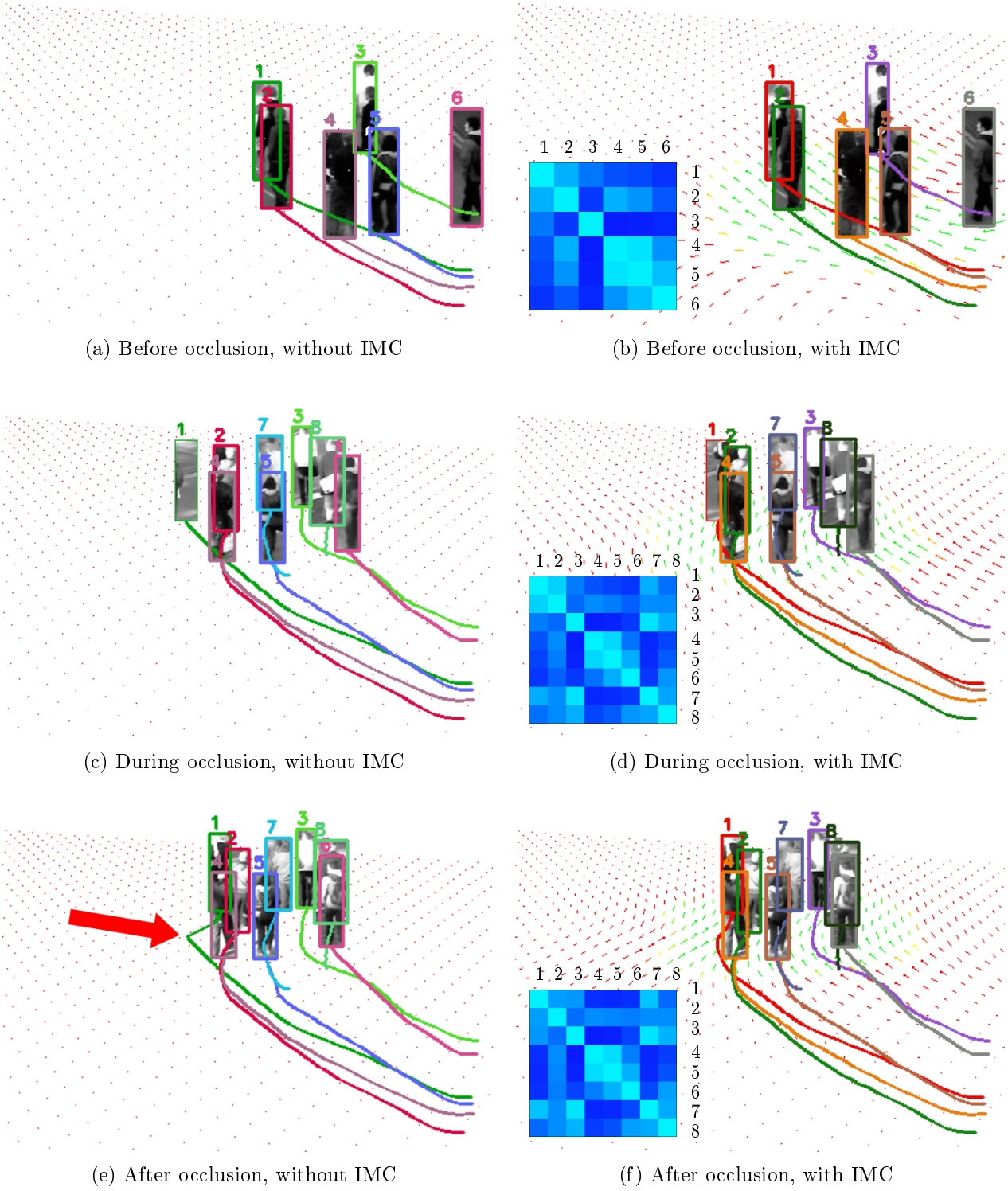
section, the whiteness property of the innovation sequence is evaluated exemplarily. The innovation depends both on the prediction, as well as on the measurements. In order to exclude errors related to the measurements and to be able to evaluate the predictive function, the manual annotations of pedestrians available in the reference data are taken as measurements, which are assumed to be correct. By looking at the sequence on system innovations, any bias or whiteness constraint violation can be attributed to the predictive function.

The innovations are observed for one person in a scenario in which six to eight persons are tracked. The prediction is carried out using a stand-alone filter and the IMC model, and measurements are taken from the reference data and are assumed to be correct. Person 1, for which the innovations are observed, is simulated to be occluded for 30 consecutive frames (frames 70-100). Figure 5.10 shows three images taken at time steps 70 (just before the occlusion), 99 (at the end of the occlusion) and 105 (shortly after the occlusion was resolved) for the tracking without IMC (left column) and with IMC (right column). For clarity, the background is removed in Figures 5.10 (a)–(f), and only the trajectories and rectangles around the tracked persons are shown. In the background, points in a discrete grid with a spacing $0.5m \times 0.5m$ in object coordinates are shown in Figures 5.10 (a), (c) and (e). In these figures, the motion context is not applied. In Figures 5.10 (b), (d) and (f), velocity vectors are shown in the background, indicating that velocities can be interpolated by means of Gaussian Process Regression at any position in object space, even if an area is occluded.

When the IMC is not used, the trajectory of person 1 drifts away from the person. Note the estimated bounding rectangle of person 1 only comprises background in Figure 5.10(c). After the end of the occlusion, a sharp bend appears in the estimated trajectory, as indicated by the arrow in Figure 5.10(e). When motion context is applied, correlations between trajectory 1 and other trajectories, e.g. 2 and 7, are detected by the IMC model (cf. Figure 5.10 (b)), by which the drift can be reduced (cf. Figure 5.10(d)) and the final trajectory has a smoother shape (cf. Figure 5.10(f)).

Figure 5.11 shows the sequences of innovations for variables x_F and y_F for the tracking of person 1 in Figure 5.10 without IMC (Figure 5.11(a)) and with IMC (Figure 5.11(b)). The time in which the tracked person is occluded is indicated by the grey-shaded area. Note the innovations right after the end of the occlusion at time 100, where the innovations are high for the stand-alone filter and lower for the IMC model.

The whiteness of an innovation sequence can be evaluated based on the autocorrelation of that sequence. Following the evaluation of Mehra (1970), the autocorrelation is evaluated for increasing time lags, so that the range and possible periodic effects of deviations from the whiteness assumption can be quantified. Figure 5.12 shows the autocorrelation of a sequence of system innovations for the person tracked with the ID number 1 in Figure 5.10. The left column of Figure 5.12 shows the autocorrelations of the innovations for variable x_f (a) and y_f (c) obtained by stand-alone Kalman filtering, while the right column shows the innovation of the respective variables obtained using the IMC model (Figures 5.12(b) and (d)). The lag is the time span between two samples of the sequence evaluated, so that small lags indicate the correlation between nearby time steps, and larger lags those of samples further apart. The horizontal lines mark the rejection regions for testing the autocorrelations to be equal to zero. Modelling the covariance of the autocorrelations by $\frac{1}{N}$, where N is the number



(a) Before occlusion, without IMC

(b) Before occlusion, with IMC

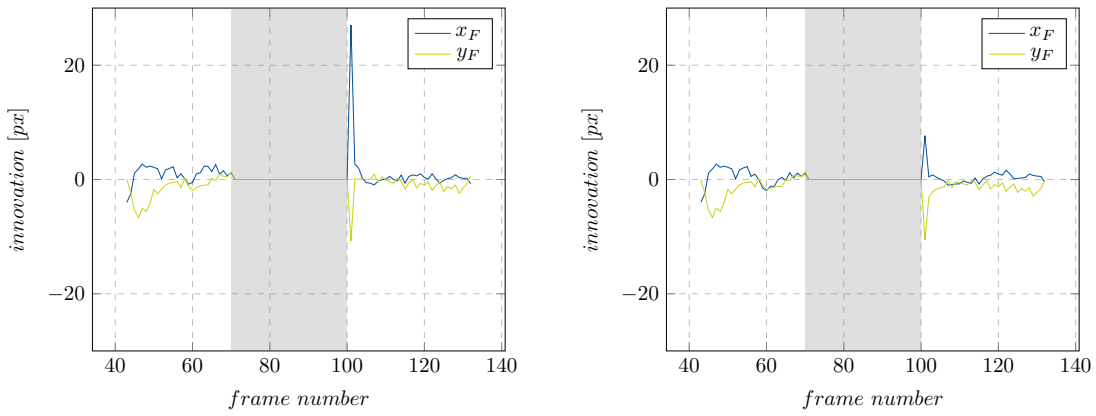
(c) During occlusion, without IMC

(d) During occlusion, with IMC

(e) After occlusion, without IMC

(f) After occlusion, with IMC

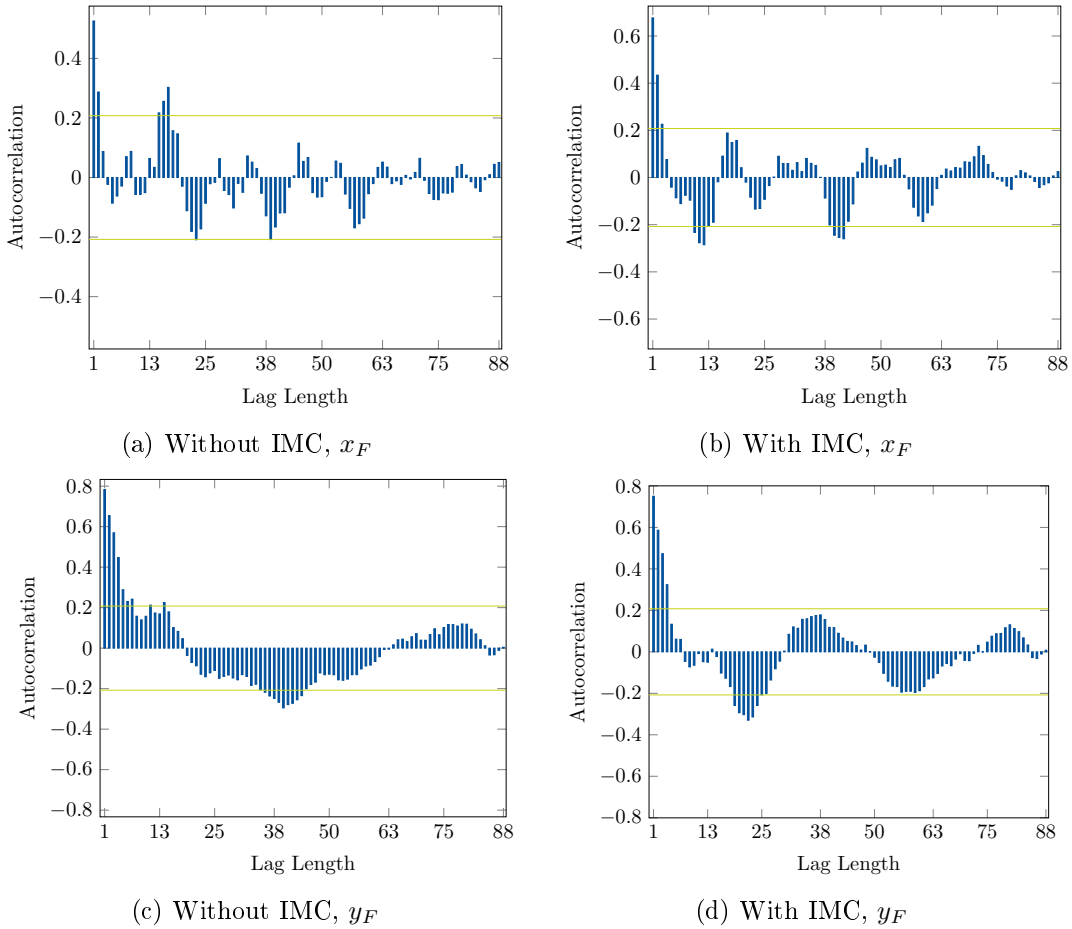
Figure 5.10: Qualitative comparison of the prediction with and without using the Implicit Motion Context (IMC) model. The Trajectories of six to eight persons are shown in random colours. The arrows in (b), (d) and (f) symbolise the interpolated velocities in a discrete grid of $0.5m \times 0.5m$. The variances of these velocities are indicated by the colours of the arrows, where red indicates high values and green indicates low values. The covariance matrices of the trajectories are shown in (b), (d) and (f). Note the sharp bend in the trajectory of person 1 in (e), as marked by the red arrow.



(a) Innovations, without IMC

(b) Innovations, with IMC

Figure 5.11: Sequence of system innovations for person 1 in Figure 5.10 with simulated occlusion marked as shaded area.



(a) Without IMC, x_F

(b) With IMC, x_F

(c) Without IMC, y_F

(d) With IMC, y_F

Figure 5.12: Autocorrelation of the sequence of system innovations for person 1 in Figure 5.10, without using Implicit Motion Context (IMC, (a) and (c)) and using IMC ((b) and (d)).

of samples, the confidence interval for an expected probability of error of $\alpha = 5\%$ can be modelled by $\pm 1.96/\sqrt{N}$ (Mehra, 1970). When the autocorrelation is zero, the sequence of innovations can be regarded as white noise. 95% of the samples are expected to lie within the area between these lines. For an optimal filter, the autocorrelations are spread randomly around zero.

In Figure 5.12, systematic deviations from zero can be observed for both models. These deviations indicate non-optimal performance of the predictive function, because the innovation sequence is not white. However, approximately 95% of the autocorrelation values lie within the expected interval for both models.

5.4.3 Benchmark results

Lastly, results achieved on the tracking benchmarks 3DMOT 2015 (Leal-Taixé et al., 2015) and KITTI Object Tracking Evaluation 2012 (Geiger et al., 2012) are reported in this section. The results on the test datasets from the 3DMOT challenge, PETS09-S2L2 and AVG-TownCentre are reported separately, whereas for the KITTI benchmark, the results achieved on the 28 test sequences are averaged. For the three datasets, the results are compared to the related work. For each method, the modus operandi (MO), the metrics MOTA and MOTP, MT and ML, IDS and FRAG, as well as the processing frequency in frames per second (FPS) is reported. MO indicates whether the processing of the method is conducted in 2D image or 3D object space, and whether the processing is capable of being applied on a frame-by-frame basis (online capability, on) or requires all frames at once (only offline capability, off).

3D Multi Object Tracking 3DMOT. The results of the presented work and those achieved by Pellegrini et al. (2009), Leal-Taixé et al. (2011), Klinger et al. (2015) and Klinger et al. (2016) are reported in Tables 5.4 and 5.5. In these methods, tracking is conducted in 3D object space, to which the 3DMOT challenge is dedicated. In (Klinger et al., 2015) and (Klinger et al., 2016) previous versions of the proposed tracking method are implemented. Both methods apply a greedy data association strategy as opposed to this work, which uses a joint probabilistic data association strategy. Implicit motion context is not used in (Klinger et al., 2015) and was introduced only in (Klinger et al., 2016). In addition to these results, the tables further report results from the 2DMOT challenge, where tracking is applied in 2D image space, and from which results on the same datasets are also available. As for all other methods, for the initialisation of new trajectories only the automatic detection results provided along with the benchmark dataset are used. Note that the evaluation in the 2DMOT challenge is also carried out in 2D, so that the precision of a result is evaluated based on the bounding box overlap (at least 50% overlap are required for acceptance) between the detection and the reference annotation, whereas in the 3DMOT challenge the 3D distance is decisive for the acceptance of the detection and by convention the threshold is one metre.

The related work reported in this section includes only a subset of the reported results of the 2DMOT challenge. Only methods with associated original publications are reported. These include the methods of Leal-Taixé et al. (2014), Wang and Fowlkes (2015) and Yoon et al. (2015), who exploit motion context in the image domain. A refinement of the position is applied in (Choi et al., 2013) by combining different observables and in (Milan et al., 2015) by joint segmentation and tracking of pedestrians. Continuous energy minimisation is applied in (Milan et al., 2014). Xiang et al. (2015), Yoon et al. (2015), Yang and Jia (2016), Yoon et al. (2016) and Milan et al. (2016) also apply tracking on a frame-by-frame level. Kim et al. (2015) apply data association based on Multi

Hypothesis Tracking, and similar to the proposed method, (Rezatofighi et al., 2015) also include a joint probabilistic data association approach for data association, which is established in an offline optimisation procedure. Like the proposed method, Wang et al. (2015a) also perform instance specific classification to define similarity measures for the data association. Convolutional neural networks are used in (Xiang et al., 2015) in the context of detection and in (Milan et al., 2016; Wang et al., 2016) and (Leal-Taixé et al., 2016) in the context of data association. The tracker of Wang et al. (2015a) performs best as measured by an average rank of 10 metrics in the 2DMOT challenge[2], while the best result by an online tracker is achieved by Xiang et al. (2015).

Within the 3DMOT challenge on the PETS09-S2L2 dataset, the proposed method performs best in terms of MOTA and third best in terms of MOTP. Previous versions of the proposed tracker, (Klinger et al., 2015) and (Klinger et al., 2016), show slightly higher MOTP scores (+0.2% and +3.4%). In terms of the MT and ML metrics, the proposed method tracks less targets persistently (−0.7%) than both previous versions of the tracker, but less objects are lost (−0.2% and −2.5%). A lower ML rate is only reported by (Pellegrini et al., 2009) (−2.2%), who also track considerably less targets persistently (−23.1%). In terms of IDS and FRAG, the proposed method performs best and second best, where (Klinger et al., 2016) presents less fragmentations (-15) but more identity switches (+16). The improved MOTA metric, compared to (Klinger et al., 2015) and (Klinger et al., 2016), can be explained by the way in which the data association is solved. In this work, a combinatorial problem is solved, assigning at most one detection to a trajectory, so that a global objective function is optimised. In (Klinger et al., 2015) and (Klinger et al., 2016), a greedy association scheme is used, assigning every single-scale detection to the trajectory with the closest predicted position. The greedy approach often leads to a suboptimal solution that becomes noticeable in cases a prediction deviates too much from the true state and where another nearby object can be assigned with detections actually stemming from a different person. Such errors often lead to a misalignment of the inferred positions that are counted as false positive detections. Regarding runtime, the proposed method performs ten times faster than the previous versions with about one frame per second on average. This is due to the improved runtime of the classification strategy in this work compared to (Klinger et al., 2015) and (Klinger et al., 2016). In these methods, the ellipse-model (ELL, cf. Section 4.2.3) was used, which was shown to perform slower than the stripes-model (STR, cf. Section 5.2.2), as used in this work.

A selection of the most recent papers reported in the 2DMOT challenge is given in the lower parts of Tables 5.4 and 5.5. Compared to these results, the proposed method provides the best MOTA metric, with +1.5% opposed to the best result achieved by Wang et al. (2015a) in 2D (see Table 5.4). The MOTP cannot be compared due to the different evaluation criteria. Pertaining to MT and ML, the proposed method reached the fourth place in terms of MT, following the previous versions of the tracker and (Wang et al., 2015a, −3.1%), and better in terms of ML than any method applying tracking online in 2D. The number of IDS is best and equal to (Rezatofighi et al., 2015), whose method performs inferior in terms of MOTA (−23.6%), MT (−16%) and ML (+14.4%). The number of fragmentations is higher than achieved by Milan et al. (2014), who in turn perform worse in terms of MOTA (−16.3%), MT (−16%), ML (+9.7%) and IDS (+11).

[2]https://motchallenge.net (accessed on September 2016)

Model	Modus operandi	**Multi Obj. Tracking Accuracy** (MOTA) [%]	**Multi Obj. Tracking Precision** (MOTP) [%]	**Mostly Tracked** (MT) [%]	**Mostly Lost** (ML) [%]	**Identity Switches** (IDS) [-]	**Fragmentations** (FRAG) [-]	**Frames Per Second** (FPS) [-]
This method	3D/on	**61.2**	63.4	27.9	4.6	**139**	181	1.0
Klinger et al. (2016)	3D/on	56.9	**66.8**	**28.6**	7.1	155	**166**	0.1
Klinger et al. (2015)	3D/on	57.6	63.6	**28.6**	4.8	231	245	0.1
Pellegrini et al. (2009)	3D/on	32.2	55.1	4.8	**2.4**	893	889	**83.5**
Leal-Taixé et al. (2011)	3D/off	41.3	55.7	7.1	16.7	243	271	8.4
Milan et al. (2016)	2D/on	38.3	71.6	9.5	14.3	320	417	**165.2**
Yoon et al. (2015)	2D/on	37.2	67.7	9.5	14.3	190	320	7.9
Yoon et al. (2016)	2D/on	44.6	69.3	7.1	14.3	175	289	6.8
Xiang et al. (2015)	2D/on	47.5	72.6	7.1	9.5	196	332	2.1
Yang and Jia (2016)	2D/on	43.1	69.4	9.5	11.9	158	412	5.9
Wang et al. (2015a)	2D/off	**59.7**	**74.4**	**31.0**	**4.8**	173	200	6.5
Kim et al. (2015)	2D/off	50.8	70.4	19.0	7.1	**142**	201	0.7
Choi (2015)	2D/off	53.4	70.5	14.3	9.5	**142**	208	11.5
Wang et al. (2016)	2D/off	49.6	70.7	11.9	11.9	192	218	1.7
Leal-Taixé et al. (2016)	2D/off	34.5	69.7	7.1	19.0	282	424	52.8
Rezatofighi et al. (2015)	2D/off	37.6	65.9	11.9	19.0	139	260	32.6
Leal-Taixé et al. (2014)	2D/off	46.6	67.6	9.5	14.3	238	264	1.4
Milan et al. (2015)	2D/off	46.1	70.6	26.2	16.7	211	211	0.2
Milan et al. (2014)	2D/off	44.9	70.2	11.9	14.3	150	**165**	1.1
Wang and Fowlkes (2015)	2D/off	41.5	70.5	7.1	16.7	212	249	41.3

Table 5.4: 3DMOT 2015 results for PETS09-S2L2. The results are grouped into methods tracking in 3D and in 2D. Best values are printed in bold.

Within the 3DMOT challenge on the AVG-TownCentre dataset, the proposed method performs best in terms of MOTA and MOTP. Regarding MT and ML, the method performs third best in terms of MT, following (Klinger et al., 2015, −5.5%) and (Klinger et al., 2016, −2.8%), and worst pertaining to ML, as opposed to (Pellegrini et al., 2009, +6.5%), who performs best in that metric. Regarding IDS and FRAG, the proposed method takes the third position in both metrics. Compared to the results reported in the 2DMOT challenge, the MOTA metric is outperformed by (Wang et al., 2015a) (+21.8%), who applies tracking in an offline setting, and by (Xiang et al., 2015, +3.2%). Considering MT, ML and IDS, the proposed method is outperformed by the same methods, whereas in terms of FRAG, the proposed method is superior to these methods. The IDS achieved in this work are higher than most of the related work in the 2DMOT challenge, though the achieved MOTA score is more than twice the size of the score of any of these methods, except for those by (Wang et al., 2015a) and (Xiang et al., 2015).

Model	Modus operandi	**Multi Obj. Tracking Accuracy** (MOTA) [%]	**Multi Obj. Tracking Precision** (MOTP) [%]	**Mostly Tracked** (MT) [%]	**Mostly Lost** (ML) [%]	**Identity Switches** (IDS) [-]	**Fragmentations** (FRAG) [-]	**Frames Per Second** (FPS) [-]
This method	3D/on	**46.3**	**58.9**	23.3	23.3	171	182	1.0
Klinger et al. (2016)	3D/on	41.1	55.0	26.1	20.8	**112**	181	0.1
Klinger et al. (2015)	3D/on	42.4	57.1	**28.8**	20.4	149	**173**	0.1
Pellegrini et al. (2009)	3D/on	15.2	51.4	7.1	**16.8**	945	797	**83.5**
Leal-Taixé et al. (2011)	3D/off	28.7	51.9	15.0	22.6	277	330	8.4
Milan et al. (2016)	2D/on	13.4	68.8	3.5	41.2	299	414	**165.2**
Yoon et al. (2015)	2D/on	5.5	66.9	0.9	59.7	74	171	7.9
Yoon et al. (2016)	2D/on	29.3	69.6	15.0	42.9	88	233	6.8
Xiang et al. (2015)	2D/on	49.5	70.1	38.9	15.5	121	297	2.1
Yang and Jia (2016)	2D/on	25.3	70.3	15.0	39.4	68	223	5.9
Wang et al. (2015a)	2D/off	**66.1**	**72.9**	**47.8**	**14.6**	159	198	6.5
Kim et al. (2015)	2D/off	27.1	70.4	17.3	44.2	74	165	0.7
Choi (2015)	2D/off	31.6	70.1	11.1	36.3	146	233	11.5
Wang et al. (2016)	2D/off	31.3	69.5	16.8	33.2	137	246	1.7
Leal-Taixé et al. (2016)	2D/off	19.3	69.0	4.4	44.7	142	289	52.8
Rezatofighi et al. (2015)	2D/off	18.3	66.8	4.4	63.3	**23**	108	32.6
Leal-Taixé et al. (2014)	2D/off	11.9	70.3	0.9	69.9	74	**75**	1.4
Milan et al. (2015)	2D/off	3.3	69.3	0.9	86.3	151	108	0.2
Milan et al. (2014)	2D/off	-2.6	68.9	5.8	54.0	186	232	1.1
Wang and Fowlkes (2015)	2D/off	14.7	70.1	2.7	61.5	123	141	41.3

Table 5.5: 3DMOT 2015 results for AVG-TownCentre. The results are grouped into methods tracking in 3D and in 2D. Best values are printed in bold.

KITTI Object Tracking Evaluation. In the KITTI Object Tracking Evaluation, the assessment of the results is carried out in image space, so that an automatic detection counts as correct if an intersection-over-union score threshold of 50% is exceeded. The results of the proposed method and those from the related work are given in Table 5.6. From the related work only the results according to the original publications are reported here. The competing methods include the works by Yoon et al. (2015, 2016), Xiang et al. (2015), Milan et al. (2014), Choi (2015), and Wang and Fowlkes (2015). All these methods apply tracking in image space. Only the proposed method applies tracking in 3D.

On the KITTI Object Tracking Evaluation benchmark, the proposed method performs worst in terms of MOTA, MOTP, MT and ML. According to the number of identity switches, this work performs second best, and best according to the number of fragmentations. The superiority in terms of IDS and FRAG, however, must be put into perspective of the low MOTA score, which is due to a high number of false negative detections, so that the risk of IDS and FRAG is inherently lower. The most top-ranked scores are achieved by Xiang et al. (2015).

Model	**Modus operandi**	**Multi Obj. Tracking Accuracy** (MOTA) [%]	**Multi Obj. Tracking Precision** (MOTP) [%]	**Mostly Tracked** (MT) [%]	**Mostly Lost** (ML) [%]	**Identity Switches** (IDS) [-]	**Fragmentations** (FRAG) [-]	**Frames Per Second** (FPS) [-]
This method	3D/on	11.97	66.59	4.81	51.89	24	**564**	1.0
Yoon et al. (2016)	2D/on	26.02	68.45	9.62	47.08	**16**	724	20
Yoon et al. (2015)	2D/on	25.47	68.06	13.06	47.42	81	692	**100**
Xiang et al. (2015)	2D/on	**35.91**	**70.36**	**23.02**	**27.84**	88	830	11.1
Milan et al. (2014)	2D/off	18.18	68.48	8.93	51.89	96	610	11.1
Choi (2015)	2D/off	25.55	67.75	17.53	42.61	34	800	11.1
Wang and Fowlkes (2015)	2D/off	23.37	67.38	12.03	45.02	72	825	16.7

Table 5.6: KITTI Object Tracking Evaluation results. Best values are printed in bold.

6 Discussion of the results

In this chapter, the experimental results are discussed in two steps. Firstly, the method is analysed with regard to the strengths and weaknesses of its central building blocks. For each component, the sensitivity to the variation of the free parameters and the impact of that component to the overall system performance are evaluated. Secondly, the generated trajectories are analysed critically.

6.1 Method evaluation

Generic object detection

The proposed detection strategy improves the output of a state-of-the-art pedestrian detector (Dalal and Triggs, 2005) using a strategy for non-maximum suppression based of 3D information. The experimental results given in Chapter 5.2.1 show that the precision of the detections at a recall rate of about 50% can be improved by about 20% for the investigated datasets. This improvement increases the quality of the final trajectories considerably, as validated in the impact study (cf. models (f) and (g) in Chapter 5.3). However, the detection recall rates in Section 5.2.1 only lie in a range of at most 55%–70%, so that approximately every third person is missed in the PETS datasets and every second person in the KITTI dataset. Because the proposed detection strategy relies on a single pedestrian detector, omission errors committed by the detector cannot be corrected. Hence, the combination of different sources of information about the presence of pedestrians is a promising direction of future work.

Prior knowledge about the scene

The prior knowledge about the scene is used for the validation of detections for new trajectory initialisations and for the inference about the image position of persons. For the latter, the prior knowledge is considered together with information about mutual occlusions for the inference of the variable n, which has the function of increasing the measurement noise if a person is either occluded or unlikely to be observed at a specific position. In model variant (c) in Section 5.3, the omission of the variable n from the system was investigated and it was shown by the decrease in tracking performance that this variable carries valuable information for the trajectory update. However, a separate study on the importance of the occlusion model and the prior knowledge was not carried out and should be addressed in future work. For the scenarios with static camera orientation, the prior knowledge about the scene is learnt prior to the actual processing and may be outdated when new image sequences are processed. Consequently, an incorporation of that information as a hidden variable of the graphical model is a

suggestion for future work as well. In this work, the information was learnt using a Random Forest classifier, which is also available for online training, so that the approach can be readily transferred to the online domain.

Instance specific classification

The Online Random Forest classifier proposed in this work, whose paramters are investigated in Chapter 5.2.2, yields an overall accuracy of only up to 30 to 40%, depending on the number of classes (persons, between about 8 and 30 in the test cases). These values are low when compared to state-of-the-art classification accuracies, e.g. in remote sensing applications. The low accuracy is due to the similarity in the visual appearance of different persons. The online classifier is designed to adapt to the changing appearance, but training samples are always rare at the beginning of the tracking process. In comparison to related work from the re-identification community (e.g., Farenzena et al., 2010), however, the overall accuracy meets the state of the art. Furthermore, no decisions within the tracking framework are based solely on the classifier and for both data association and localisation, where the classifier is applied, even a classification score lower than the highest rank influences the tracking results positively (cf. model (e) in Chapter 5.3).

Temporal model

The temporal model, which comprises the Implicit Motion Context is investigated with respect to the impact of the free parameters in Section 5.2.3, the impact of that model on the overall performance of the tracker in Section 5.3 and the optimality in terms of whiteness of the system innovations in Section 5.4.2. The parameters were shown to have weak, yet measurable influence on the tracking performance. Parameters should, thus, be selected with caution. Training is applied on a single dataset under the assumption that the learnt parameters are transferable to other scenes. This assumption is not verified, but it is justified by the fact that the motion parameters are modelled in a common 3D object space, where all persons reside. However, due to different situations (e.g., pedestrian zones vs. campuses), different motion behaviour is imaginable, so that a training of the parameters on further datasets may be reasonable. By the analysis of the IMC model, the decrease of performance when IMC is not used was shown empirically in Section 5.3 for the PETS dataset, whereas the model does not have a significant influence on the results achieved for the KITTI data. As the transferability of the parameters was not investigated, it cannot be concluded whether different parameters have an impact on the results. In future work, the aspect of transferability of the learnt parameters should, thus, be addressed.

It was further shown empirically that the IMC model improves the predictive function in a multi-person tracking environment in Section 5.4.2. For a controlled environment with measurements that can be assumed to be correct, the sequence of system innovations in the update model of the Dynamic Bayesian Network is shown not to be biased by the IMC model, while, during an occlusion, which was simulated to take place over approximately 4 seconds, the predictive model yields more plausible results and reduces the disagreement between the prediction and the measurements when the occlusion

is over. Using the covariance function of the IMC model, correlations between the observed persons are determined automatically, so that the trajectory of the occluded person is continued with the aid of the state estimates of the surrounding persons.

The autocorrelation of the system innovations further shows that neither the stand-alone filter nor the IMC model perform strictly optimally in terms of an unbiased and white state prediction. To cope with this effect, higher order motion models that loosen the assumption of zero-acceleration should be investigated in future work.

Data association

The joint probabilistic data association (JPDA) strategy applied in this work combines location and appearance-based measures of similarity. The data association strategy is not addressed explicitly in the experiments, but both central components of the data association strategy, the temporal model and the instance-specific classifier, are investigated as discussed above. Furthermore, previous versions of the proposed method (Klinger et al., 2015, 2016), which use a greedy scheme for data association, are compared to the proposed method in Section 5.4.3. The greedy approach often leads to a suboptimal solution which becomes noticeable in cases where a prediction deviates too much from the true state, and where another nearby object can be assigned to detections actually stemming from a different person. Such errors lead to a misalignment of the measurements so that the posterior estimates of the positions might not be matched with the reference data and are counted as false positive detections instead. Using the JPDA strategy as shown in this work, a combinatorial problem is solved, which assigns a maximum of one detection to a trajectory.

Recursive estimation

The way in which the available information is combined in the Dynamic Bayesian Network is investigated by the experiments in Section 5.3. The handling of multiple persons for the predictive model was discussed separately above. The proposed Dynamic Bayesian Network models the joint probability of all variables used in this work. Different models representing the same variables are imaginable, depending on the category (observed vs. unknown) each variable is associated to. For instance, the occlusion and the prior knowledge about the scene are introduced as observations, but could be modelled as unknowns likewise. The design of the model is justified by the experimental results in Section 5.3. The omission of any component from the model either reduces the performance in the majority of the evaluation metrics or leaves the performance unaffected. Because inference in this work is carried out by means of belief propagation on a tree-structured graph and the essential steps of the recursive filter are carried out based on Gaussian Process Regression and an Extended Kalman Filter model, the inferred variables yield the optimal solutions at every time step.

6.2 Evaluation of the trajectories

The aim of the developed approach for localisation and tracking of multiple persons was the improvement of the reliability and precision of the generated trajectories. The ability to achieve these goals is assessed by the experiments conducted in Section 5.4.

The localisation accuracy is addressed in Section 5.4.1. It was shown that 95% of all correct detections have a positional displacement from the reference smaller than $0.8m$ for the PETS and smaller than $3m$ for the KITTI dataset. As measured by the recall rates at different acceptance thresholds, almost all detections the tracker is capable to achieve are made with an accuracy of about $0.5m$ on the PETS and $2m$ on the KITTI dataset. In other words, if all automatic detections are considered to be correct (i.e. if the acceptance threshold is sufficiently high to judge all annotations as correct), the positional accuracy is about $0.5m$ and $2m$, respectively. This accuracy for the scenarios with moving cameras gives insight to the limitations of the proposed method. At an expected maximum positional offset of automatic annotation results of $1m$ (cf. acceptance criteria in the 3DMOT challenge), only 71% of all feasible detections are obtained. This indicates that the tracking in these scenarios with the proposed method is not precise. Especially in the context of driver assistance systems, where the exact localisation of pedestrians is essential, the achieved accuracy is not sufficient. However, the localisation accuracy evaluated in 2D (cf. Table 5.6) is similar to the values achieved by other state-of-the-art methods.

As measured by the MOTP metric in Section 5.3, the average positional displacement within an acceptance radius of $1m$ amounts to $0.35m$ for the PETS dataset and to $0.44m$ for the KITTI dataset. These differences are partly due to the different viewing directions of the cameras, which are less inclined in the KITTI dataset, so that the same positional offset of an image measurements yields larger errors in world coordinates for the KITTI dataset than for the PETS dataset. The geometric accuracy reached in the 3DMOT benchmark, however, is superior to the related work, and could be improved by about 5 to 10 cm.

The completeness and correctness of the detections are assessed once for the single-frame detection (cf. Section 5.2.1) and for the tracking results (cf. Section 5.3). The single-frame detection achieves the highest recall score of 66% with a precision of 84% for the PETS and a recall of 54% at 52% precision for the KITTI dataset. The recall rates achieved on the same sequences after tracking decreases about 14% and 12%, while the precision is increased about 11% and 19%, respectively. The recall achieved by the tracker is limited by the recall achieved at the single-frame detection stage. As the final recall rates of about 50% and 40% indicate that about half of all persons in all images are not detected at all, these results encourage the use of different detection strategies in place or in addition to the used HOG/SVM detector. The proposed strategies for false positive reduction and estimation of the probability densities enable the application of other sliding-window-based detectors. The Dynamic Bayesian Network, which joins different sources of information, can readily be extended to incorporate further detectors.

The persistence of the tracking of a person is measured by the MT and ML metrics. These are reported in Sections 5.2.4, 5.3 and 5.4.3. According to these metrics, one of four persons in the PETS

dataset is tracked most of the time while being present in the image sequence, whereas for the KITTI dataset, only one of twenty persons is tracked persistently. In the addressed scenarios with constant camera orientation, many persons are occluded for larger periods of time. If the occlusions last longer than the system waits before terminating a trajectory, the occluded target is not tracked further and unlikely to be re-initialised as long as the occlusion persists. These values reveal yet another weakness of the generated trajectories: Though the completeness of the trajectories meets the state of the art in scenes with constant camera orientation, the number of lost targets is too high to be applicable for autonomous technologies.

Logical errors in the trajectory continuation of multiple persons are further measured by the number of identity switches and fragmentations. In these metrics the trajectories generated by the proposed method again perform as well as the state-of-the-art on average, and outperform the related work on the PETS dataset. In this respect, the reliability of the trajectories generated within a multi-person environment has seen substantiated improvements compared to the state-of-the-art.

The direct comparison of the achieved results to those of other trackers from the related work on the KITTI benchmark shows that the proposed method performs worst among all methods listed in the benchmark. It is noteworthy that all these methods apply tracking in 2D image space, which is opposed to the proposed tracking approach in 3D object space. Despite substantiated success of this method for the static camera constellations, the conversion of image coordinates to object space under the flat world assumption and very low incidence angles of the image rays with the ground plane entail an unfavourable propagation of errors from image-based observations to the state variables in 3D. The estimated height of a person in object space is computed based on the estimated scale factor that projects the person into the image. That factor depends on the estimated distance of the person on the ground plane from the camera. If the terrain in inclined, the flat world assumption is broken, and the estimated distance and, thus, the 3D height is wrong. The estimated height in 3D is evaluated at the initialisation of new trajectories, and at the false positive reduction and data association. If the initial height as part of the state vector is wrong, predictions of the height that are evaluated in the data association step are prone to result in unrealistic similarity measures. As a consequence, current detections may not be associated to existing trajectories and the tracking is corrupted. If the false positive reduction fails, the recall rate of the single-frame detection decreases and trajectories are prone not to be updated at all. Furthermore, if the camera orientation, which is assumed to be known, is imprecise, the data association is prone to yield wrong results. In this work, the camera orientation was only estimated by means of visual odometry and the quality of the results was not analysed. Lastly, on the KITTI dataset, the recall and precision values of the detection model are considerably worse than those achieved on the PETS dataset. This is due to fact that the size of persons in the KITTI-images varies considerably, whereas the underlying HOG/SVM detector is only trained on pedestrian images with a fixed height of 96 pixels (cf. Section 4.2.2).

A possible remedy to the effect of the flat world assumption violation is the combination of the monocular tracking with 3D information, e.g. in form of a digital terrain model or depth information extracted from stereoscopic imagery (Schindler et al., 2010; Geiger et al., 2011). In order to account for the low recall and precision values of the detection results, a training of the HOG/SVM detector based on images specific for the KITTI dataset is expedient.

7 Conclusions and future work

The aim of this work was the improvement of the reliability and geometric accuracy of trajectories generated by tracking in monocular image sequences. The research goal was approached by presenting four scientific contributions that (a) investigate a new probabilistic framework for the joint localisation and tracking of multiple persons, (b) present a new strategy for instance-specific classification of multiple persons, (c) achieve more realistic state predictions for interacting pedestrians and (d) describe a new model for the assignment of similarity measures for data association based on the new models of motion and appearance.

Conclusions

The Dynamic Bayesian Network proposed and investigated for the task of localisation and tracking provides a framework for statistical inference of the unknown variables, i.e. the state parameters and the image position of the tracked persons. Optimal decisions about the parameters are taken by inference using belief propagation. Because the underlying graph structure does not contain loops, exact inference can be conducted by forward and backward propagation of the belief about the variables.

A key advantage of the proposed framework is that, despite the joint inference of the unknown variables, the model is modular and expandable. This enables, for instance, weak components as the underlying detector to be replaced by better detection methods in the future. Also, different detection and classification strategies can be integrated into the model in addition to the existing components, without requiring the entire model to be reconstructed from scratch.

By setting the task of trajectory generation into a probabilistic framework, another key advantage of the method arises. The assignment of probabilities to the desired variables enables the quantification of uncertainties about the variables. This, in turn, offers a way to evaluate the results using statistical tests, which lend themselves to the self-diagnosis of the system, which is carried out in the proposed method by applying statistical tests for outlier detection in the measurements based on the system innovations. Set in the context of online recursive estimation frameworks, this is an important step towards the integrity of the system.

The generated trajectories are evaluated under different aspects. Regarding the desired property of a high reliability in the correctness of trajectory continuation, i.e. the consistency of tracking a specific target within the presence of others, the results reveal a considerable reduction in the number of tracking errors in scenes with constant camera orientation and a flat terrain. Compared to the related work on the tracking in image sequences from static cameras, the number of identity switches and fragmentations is reduced while preserving the completeness of the overall detections. By comparison

of the full model with trimmed versions of the tracker, the increase of multi-object tracking accuracy can be dedicated to the redundancy in the measurements derived from the image.

Regarding the geometric accuracy of the generated trajectories, the results achieved by this tracker outperform the results achieved by other researchers in the 3DMOT challenge by about 5 to 10cm, as measured by the average displacement from the reference annotations of the persons. By comparison of the full model with trimmed versions of the tracker, the increase of geometric accuracy can be mainly dedicated to the improved motion model, while the incorporation of additional observations about the image positions has no clear impact on the geometric accuracy. Having shown that the geometric accuracy is significantly lower when omitting the detection from the observation model, it can be concluded that the classifier used for the localisation performs poorly. A promising prospect for future developments is, thus, either improving the classification strategy or adding further observations to the tracking framework.

The method presented in this work was originally designed for the tracking in scenes with static camera orientation and flat terrain (Klinger et al., 2015, 2016). The application to scenes with dynamic camera orientation was carried out to demonstrate the transferability of the approach to diverse application scenarios on the one hand and to outline the limitations of that approach on the other hand. The tracking performance on the image sequences from dynamic platforms reveals the limitations of the proposed method. In comparison to the related work using the KITTI benchmark, in which all other methods perform tracking in 2D, the tracking by the proposed method in 3D performs worst. Here, the major difference lies in the significant increase in omission and commission errors at the assignment of automatic detections. Training of a pedestrian detector on this specific dataset can be expected to improve these results.

Future work

With respect to the conclusions drawn from this work, the proposed method for multi-person localisation and tracking paves the way for several aspects relevant in future research. Both the positive and the negative aspects of the achieved results motivate future investigations of this line of research.

The substantial improvements in the geometric accuracy by using the Implicit Motion Context calls for further investigations especially on image sequences from moving camera platforms, in order to transfer the benefits from using that model to more generic situations for tracking. Especially due to the need for highly accurate pedestrian trajectories in the context of autonomous driving and robot navigation, the use of Implicit Motion Context is a promising approach. To this end, new models to measure the interactions between pedestrians should be investigated, in order to find correlations in the trajectories of pedestrian that do not move in crowds.

Because the recall rate of the pedestrian detection on the KITTI image sequences was rather low, the application of different approaches to the pedestrian detection is expected to yield clear improvements of the overall results. The proposed method for false positive reduction and density estimation can be applied to other sliding-window-based detectors. Bottom-up approaches for pedestrian detection, such as the Deformable Part Model (Felzenszwalb et al., 2010) or Implicit Shape Models (Leibe et al.,

2008) yield probability densities right away, and, thus, can be integrated into the Dynamic Bayesian network immediately.

A further drawback of the proposed method is the flat-world assumption made for the conversion of image to world coordinates. A more sophisticated model of the scene can be estimated by simultaneous localisation and mapping (SLAM), which is applicable both to monocular (Davison et al., 2007; Engel et al., 2014) and to stereoscopic image sequences (Engel et al., 2015). Integrating the localisation of pedestrians and 3D scene reconstruction in a joint probabilistic model was already shown to improve tracking by Schindler et al. (2010) using stereoscopic image sequences, so that the refinement of the proposed method by the 3D modelling of the scene is proposed to be applied in future work.

The performance of the recursive estimation framework was investigated mainly under the aspects of multi-person localisation and tracking performance. The statistical properties of the recursive filter, i.e. the validation of the assigned process noise and the measurement noise covariances, were not further investigated. An investigation of these properties of the tracker would improve the reliability and integrity of the method from the statistical point of view, as further statistical tests could be applied to detect outliers in the measurements.

Lastly, the applicability of the proposed method to real-time systems is inhibited by the average processing rate of 1 Hz. Several components of the system enable parallelisation and/or implementation on a GPU. The detection and the classification parts are the slowest components of the system. An available GPU implementation of the HOG detector (Prisacariu and Reid, 2009) yields about ten times faster processing rates than the CPU-based implementation used in this work. Also, the modularity of the Random Forests enables parallelisation and further improvements in the runtime.

Bibliography

Andriluka, M., Roth, S. and Schiele, B., 2008. People- tracking-by-detection and people-detection-by-tracking. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Andriluka, M., Roth, S. and Schiele, B., 2010. Monocular 3d pose estimation and tracking by detection. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 623–630.

Arora, C. and Globerson, A., 2013. Higher order matching for consistent multiple target tracking. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 177–184.

Ben Shitrit, H., Berclaz, J., Fleuret, F. and Fua, P., 2014. Multi-commodity network flow for tracking multiple people. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (8), pp. 1614–1627.

Benfold, B. and Reid, I., 2011. Stable multi-target tracking in real-time surveillance video. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3457–3464.

Berclaz, J., Fleuret, F. and Fua, P., 2009. Multiple object tracking using flow linear programming. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), pp. 1–8.

Berkelaar, M., Eikland, K. and Notebaert, P., 2004. lpsolve: Open source (mixed-integer) linear programming system. Eindhoven U. of Technology, http://lpsolve.sourceforge.net/5.5/, accessed on September 2016.

Bernardin, K. and Stiefelhagen, R., 2008. Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing, 2008 (1), pp. 1–10.

Bishop, C., 2006. Pattern recognition and machine learning. Springer, New York.

Bradski, G. and Kaehler, A., 2008. Learning OpenCV: Computer vision with the OpenCV library. O'Reilly, Sebastopol, CA.

Brau, E., Guan, J., Simek, K., Pero, L., Dawson, C. and Barnard, K., 2013. Bayesian 3d tracking from monocular video. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 3368–3375.

Breiman, L., 2001. Random forests. Machine learning, 45 (1), pp. 5–32.

Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E. and Van Gool, L., 2011. Online multi-person tracking-by-detection from a single, uncalibrated camera. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (9), pp. 1820–1833.

Čehovin, L., Leonardis, A. and Kristan, M., 2015. Visual object tracking performance measures revisited. arXiv:1502.05803.

Choi, W., 2015. Near-online multi-target tracking with aggregated local flow descriptor. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 3029–3037.

Choi, W., Pantofaru, C. and Savarese, S., 2013. A general framework for tracking multiple people from a moving camera. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (7), pp. 1577–1591.

Collins, R. T., 2012. Multitarget data association with higher-order motion models. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1744–1751.

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

Dai, Q. and Hoiem, D., 2012. Learning to localize detected objects. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3322–3329.

Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 886–893.

Dantzig, G. B., 1951. Maximization of a linear function of variables subject to linear inequalities. In: Activity Analysis of Production and Allocation. Ed: T.C. Koopmans. Jon Wiley & Sons, New York, pp. 347–358.

Davison, A. J., Reid, I. D., Molton, N. D. and Stasse, O., 2007. MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (6), pp. 1052–1067.

Deutscher, J., Blake, A. and Reid, I., 2000. Articulated body motion capture by annealed particle filtering. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 126–133.

Dollár, P., Appel, R., Belongie, S. and Perona, P., 2014. Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (8), pp. 1532–1545.

Dollár, P., Wojek, C., Schiele, B. and Perona, P., 2011. Pedestrian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (4), pp. 743–761.

Elgammal, A., Harwood, D. and Davis, L., 2000. Non-parametric model for background subtraction. In: Proc. of the European Conference on Computer Vision (ECCV), pp. 751–767.

Ellis, D., Sommerlade, E. and Reid, I., 2009. Modelling pedestrian trajectory patterns with gaussian processes. In: Proc. of the ICCV Workshop on Visual Surveillance, pp. 1229–1234.

Engel, J., Schöps, T. and Cremers, D., 2014. Lsd-slam: Large-scale direct monocular slam. In: Proc. of the European Conference on Computer Vision (ECCV), pp. 834–849.

Engel, J., Stückler, J. and Cremers, D., 2015. Large-scale direct SLAM with stereo cameras. In: Proc. of the IEEE and RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1935–1942.

Enzweiler, M., Eigenstetter, A., Schiele, B. and Gavrila, D. M., 2010. Multi-cue pedestrian classification with partial occlusion handling. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 990–997.

Ess, A., Leibe, B., Schindler, K. and van Gool, L., 2008. A mobile vision system for robust multi-person tracking. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A., 2010. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88 (2), pp. 303–338.

Farenzena, M., Bazzani, L., Perina, A., Murino, V. and Cristani, M., 2010. Person re-identification by symmetry-driven accumulation of local features. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2360–2367.

Felzenszwalb, P., Girshick, R., McAllester, D. and Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (9), pp. 1627–1645.

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

Frey, B. J. and MacKay, D. J. C., 1998. A revolution: Belief propagation in graphs with cycles. In: Proc. of Advances in Neural Information Processing Systems (NIPS), pp. 479–485.

Gall, J. and Lempitsky, V., 2013. Class-specific hough forests for object detection. In: A. Criminisi and J. Shotton. (Eds.), Decision Forests for Computer Vision and Medical Image Analysis, pp. 143–157.

Gauss, C. F., 1809. Theoria motus corporum coelestium in sectionibus conicis solem ambientium. Reprinted translation by C. H. Davis: Theory of motion of the heavenly bodies moving about the sun in conic sections. Dover Publications Inc., New York, 1963.

Gavrila, D. and Munder, S., 2007. Multi-cue pedestrian detection and tracking from a moving vehicle. International Journal of Computer Vision, 73 (1), pp. 41–59.

Ge, W., Collins, R. T. and Ruback, B., 2009. Automatically detecting the small group structure of a crowd. In: Proc. IEEE Workshop on Applications of Computer Vision (WACV), pp. 1–8.

Geiger, A., Lenz, P. and Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3354–3361.

Geiger, A., Ziegler, J. and Stiller, C., 2011. Stereoscan: Dense 3d reconstruction in real-time. In: Proc. of the IEEE Intelligent Vehicles Symposium (IV), pp. 963–968.

Gelb, A., 1974. Applied optimal estimation. MIT press, Cambridge, Massachusetts.

Godec, M., Roth, P. M. and Bischof, H., 2011. Hough-based tracking of non-rigid objects. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 81–88.

Hartley, R. and Zisserman, A., 2000. Multiple view geometry. Cambridge University Press, Cambridge, UK.

Helbing, D. and Molnár, P., 1995. Social force model for pedestrian dynamics. Physical review E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 51 (5), pp. 4282–4286.

Hinz, S. and Schmidt, F., 2015. Personentracking in Luftbildsequenzen. W. Freeden, R. Rummel (Hrsg.), Handbuch der Geodäsie, Band Photogrammetrie und Fernerkundung (C. Heipke, Hrsg.). Springer Reference Naturwissenschaften, DOI 10.1007978-3-662-46900-2_51-1.

Hoiem, D., Efros, A. A. and Hebert, M., 2008. Putting objects in perspective. International Journal of Computer Vision, 80 (1), pp. 3–15.

Hooke, R. and Jeeves, T. A., 1961. Direct search solution of numerical and statistical problems. Journal of the Association for Computing Machinery (JACM), 8 (2), pp. 212–229.

Isard, M. and Blake, A., 1996. Contour tracking by stochastic propagation of conditional density. In: Proc. of the European Conference on Computer Vision (ECCV), pp. 343–356.

Jaakkola, T. S. and Haussler, D., 1999. Probabilistic kernel regression models. In: D. Heckerman and J. Whittaker (Eds.), Proc. of the 7th Workshop on Artificial Intelligence and Statistics (AISTATS). Morgan Kaufmann, San Francisco, CA.

Jiang, H., Fels, S. and Little, J. J., 2007. A linear programming approach for multiple object tracking. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Julier, S. J. and Uhlmann, J. K., 1997. New extension of the Kalman filter to nonlinear systems. In: Proc. of the International Symposium of Aerospace/Defence, Sensing, Simulation and Controls, pp. 182–193.

Kalman, R. E., 1960. A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82 (1), pp. 35–45.

Keller, C. G., Hermes, C. and Gavrila, D. M., 2011. Will the pedestrian cross? Probabilistic path prediction based on learned motion features. In: Proc. of the DAGM Symposium for Pattern Recognition, pp. 386–395.

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

Kim, C., Li, F., Ciptadi, A. and Rehg, J. M., 2015. Multiple hypothesis tracking revisited. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 4696–4704.

Kim, K. and Davis, L., 2011. Object detection and tracking for intelligent video surveillance. Multimedia Analysis, Processing and Communications, pp. 265–288.

Kim, K., Lee, D. and Essa, I., 2011. Gaussian process regression flow for analysis of motion trajectories. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 1164–1171.

Kim, K., Lee, D. and Essa, I., 2012. Detecting regions of interest in dynamic scenes with camera motions. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1258–1265.

Klinger, T., Rottensteiner, F. and Heipke, C., 2014. Pedestrian recognition and localisation in image sequences as bayesian inference. In: Proc. of the Computer Vision Winter Workshop (CVWW), pp. 51-58.

Klinger, T., Rottensteiner, F. and Heipke, C., 2015. Probabilistic multi-person tracking using dynamic Bayes networks. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume II-3/W5, pp. 435-442.

Klinger, T., Rottensteiner, F. and Heipke, C., 2016. A Gaussian Process based multi-person interaction model. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume II-3/W3, pp. 271-277.

Ko, J. and Fox, D., 2009. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. Autonomous Robots, 27(1), pp. 75–90.

Krige, D. G., 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. Journal of Chemical, Metallurgical, and Mining Society of South Africa, 94(3), pp. 95–112.

Kschischang, F. R., Frey, B. J. and Loeliger, H.-A., 2001. Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory, 47 (2), pp. 498–519.

Kuhn, H. W., 1955. The Hungarian method for the assignment problem. Naval research logistics quarterly, 2 (1-2), pp. 83–97.

Leal-Taixé, L., Canton-Ferrer, C. and Schindler, K., 2016. Learning by tracking: Siamese CNN for robust target association. arXiv:1604.07866.

Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B. and Savarese, S., 2014. Learning an image-based motion context for multiple people tracking. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3542–3549.

Leal-Taixé, L., Milan, A., Reid, I., Roth, S. and Schindler, K., 2015. MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942.

Leal-Taixé, L., Pons-Moll, G. and Rosenhahn, B., 2011. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: Proc. of the ICCV Workshop on Modeling, Simulation and Visual Analysis of large Crowds, pp. 120–127.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. Nature, 521 (7553), pp. 436–444.

Leibe, B., Leonardis, A. and Schiele, B., 2008. Robust object detection with interleaved categorization and segmentation. International Journal of Computer Vision, 77 (1), pp. 259–289.

Leibe, B., Seemann, E. and Schiele, B., 2005. Pedestrian detection in crowded scenes. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 878–885.

Li, Y., Huang, C. and Nevatia, R., 2009. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2953–2960.

Luber, M., Stork, J. A., Tipaldi, G. D. and Arras, K. O., 2010. People tracking with human motion predictions from social forces. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp. 464–469.

Ma, C., Huang, J.-B., Yang, X. and Yang, M.-H., 2015. Hierarchical convolutional features for visual tracking. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 3074–3082.

Mehra, R., 1970. On the identification of variances and adaptive Kalman filtering. IEEE Transactions on Automatic Control, 15 (2), pp. 175–184.

Meuter, M., Iurgel, U., Park, S.-B. and Kummert, A., 2008. The unscented Kalman filter for pedestrian tracking from a moving host. In: IEEE Intelligent Vehicles Symposium, pp. 37–42.

Milan, A., Leal-Taixé, L., Schindler, K. and Reid, I., 2015. Joint tracking and segmentation of multiple targets. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5397–5406.

Milan, A., Rezatofighi, S. H., Dick, A., Schindler, K. and Reid, I., 2016. Online multi-target tracking using recurrent neural networks. arXiv:1604.03635.

Milan, A., Roth, S. and Schindler, K., 2014. Continuous energy minimization for multi-target tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (1), pp. 58–72.

Moritz, H., 1973. Least-Squares Collocation. Deutsche Geodätische Kommission (DGK), Series A, Nr. 75, Munich, Germany.

Murphy, K. P., 2002. Dynamic bayesian networks: representation, inference and learning. PhD thesis, University of California, Berkeley.

Oh, S., Russell, S. and Sastry, S., 2004. Markov chain monte carlo data association for general multiple-target tracking problems. In: Proc. IEEE Conference on Decision and Control, 1, pp. 735–742.

Okuma, K., Taleghani, A., Freitas, N., Little, J. and Lowe, D., 2004. A boosted particle filter: Multitarget detection and tracking. In: Proc. of the European Conference on Computer Vision (ECCV), pp. 28–39.

Ommer, B., Mader, T. and Buhmann, J. M., 2009. Seeing the objects behind the dots: Recognition in videos from a moving camera. International Journal of Computer Vision, 83 (1), pp. 57–71.

Ouyang, W. and Wang, X., 2013. Single-pedestrian detection aided by multi-pedestrian detection. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3198–3205.

Oza, N. C., 2005. Online bagging and boosting. In: Proc. of the IEEE International Conference on Systems, Man and Cybernetics, 3, pp. 2340–2345.

Pearl, J., 1988. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Mateo, CA.

Pellegrini, S., Ess, A. and Van Gool, L., 2010. Improving data association by joint modeling of pedestrian trajectories and groupings. In: Proc. of the European Conference on Computer Vision (ECCV), pp. 452–465.

Pellegrini, S., Ess, A., Schindler, K. and Van Gool, L., 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 261–268.

Perera, A. A., Srinivas, C., Hoogs, A., Brooksby, G. and Hu, W., 2006. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 666–673.

PETS, 2009. Pets 2009 benchmark data, http://www.cvg.reading.ac.uk/pets2009/a.html, accessed on September 2016.

Pirsiavash, H., Ramanan, D. and Fowlkes, C. C., 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1201–1208.

Possegger, H., Mauthner, T., Roth, P. and Bischof, H., 2014. Occlusion geodesics for online multi-object tracking. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1306–1313.

Prince, S., 2012. Computer Vision: Models Learning and Inference. Cambridge University Press, Cambridge, UK.

Prisacariu, V. and Reid, I., 2009. FastHOG-a real-time GPU implementation of HOG. Technical Report 2310/09, Department of Engineering Science, Oxford University, Oxford, UK.

Rabiner, L. R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77 (2), pp. 257–286.

Ramanan, D., 2007. Using segmentation to verify object hypotheses. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Rasmussen, C. E., 2006. Gaussian processes for machine learning. The MIT press, Cambridge, Massachusetts.

Reid, D. B., 1979. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control, 24 (6), pp. 843–854.

Rezatofighi, S. H., Milan, A., Zhang, Z., Shi, Q., Dick, A. and Reid, I., 2015. Joint probabilistic data association revisited. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 3047–3055.

Rujikietgumjorn, S. and Collins, R., 2013. Optimized pedestrian detection for multiple and occluded people. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3690–3697.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M. and Edwards, D. D., 2003. Artificial intelligence: a modern approach. Prentice-Hall, Englewood Cliffs, NJ.

Saffari, A., Leistner, C., Santner, J., Godec, M. and Bischof, H., 2009. On-line random forests. In: Proc. of the ICCV Workshop on on-line learning for Computer Vision, pp. 1393–1400.

Schindler, K., Ess, A., Leibe, B. and Van Gool, L., 2010. Automatic detection and tracking of pedestrians from a moving stereo rig. ISPRS Journal of Photogrammetry and Remote Sensing, 65 (6), pp. 523–537.

Scovanner, P. and Tappen, M. F., 2009. Learning pedestrian dynamics from the real world. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 381–388.

Shi, J. and Tomasi, C., 1994. Good features to track. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 593–600.

Shu, G., Dehghan, A. and Shah, M., 2013. Improving an object detector and extracting regions using superpixels. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3721–3727.

Shu, G., Dehghan, A., Oreifej, O., Hand, E. and Shah, M., 2012. Part-based multiple-person tracking with partial occlusion handling. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1815–1821.

Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A. and Shah, M., 2014. Visual tracking: An experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (7), pp. 1442–1468.

Solera, F., Calderara, S. and Cucchiara, R., 2015. Learning to divide and conquer for online multi-target tracking. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 4373–4381.

Stauffer, C. and Grimson, W., 2000. Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (8), pp. 747 –757.

Storms, P. P. and Spieksma, F. C., 2003. An lp-based algorithm for the data association problem in multitarget tracking. Computers & Operations Research, 30 (7), pp. 1067–1085.

Tran, A. and Manzanera, A., 2015. A versatile object tracking algorithm combining particle filter and generalised hough transform. In: International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 105–110.

Trautman, P., Ma, J., Murray, R. M. and Krause, A., 2015. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. The International Journal of Robotics Research, 34 (3), pp. 335–356.

Urtasun, R., Fleet, D. J. and Fua, P., 2006. 3D people tracking with Gaussian process dynamical models. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 238–245.

Viola, P. and Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 511–518.

Wang, B., Wang, G., Chan, K. L. and Wang, L., 2015a. Tracklet association by online target-specific metric learning and coherent dynamics estimation. arXiv:1511.06654.

Wang, B., Wang, L., Shuai, B., Zuo, Z., Liu, T., Luk Chan, K. and Wang, G., 2016. Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. In: Proc. of the CVPR workshop on DeepVision: Deep Learning in Computer Vision, pp. 1–8.

Wang, L., Ouyang, W., Wang, X. and Lu, H., 2015b. Visual tracking with fully convolutional networks. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 3119–3127.

Wang, S. and Fowlkes, C., 2015. Learning optimal parameters for multi-target tracking. In: Proc. of the British Machine Vision Conference (BMVC), pp. 1–13.

Wojek, C., Walk, S. and Schiele, B., 2009. Multi-cue onboard pedestrian detection. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 794–801.

Wu, B. and Nevatia, R., 2007. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. International Journal of Computer Vision, 75 (2), pp. 247–266.

Xiang, Y., Alahi, A. and Savarese, S., 2015. Learning to track: Online multi-object tracking by decision making. In: Proc. of the IEEE International Conference on Computer Vision (ICCV), pp. 4705–4713.

Yamaguchi, K., Berg, A. C., Ortiz, L. E. and Berg, T. L., 2011. Who are you with and where are you going? In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1345–1352.

Yang, B. and Nevatia, R., 2012. An online learned CRF model for multi-target tracking. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2034–2041.

Yang, M. and Jia, Y., 2016. Temporal dynamic appearance modeling for online multi-person tracking. arXiv:1510.02906.

Yoon, J. H., Yang, M.-H., Lim, J. and Yoon, K.-J., 2015. Bayesian multi-object tracking using motion context from multiple objects. In: Proc. IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 33–40.

Yoon, J., Lee, C.-R., Yang, M.-H. and Yoon, K.-J., 2016. Online multi-object tracking via structural constraint event aggregation. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1392–1400.

Zhang, L. and van der Maaten, L., 2013. Structure preserving object tracking. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1838–1845.

Zhang, L., Li, Y. and Nevatia, R., 2008a. Global data association for multi-object tracking using network flows. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Zhang, T., Liu, S., Ahuja, N., Yang, M.-H. and Ghanem, B., 2015. Robust visual tracking via consistent low-rank sparse learning. International Journal of Computer Vision, 111 (2), pp. 171–190.

Zhang, Z., Hu, Y., Chan, S. and Chia, L.-T., 2008b. Motion context: A new representation for human action recognition. In: Proc. of the European Conference on Computer Vision (ECCV), pp. 817–829.

Zhao, T. and Nevatia, R., 2004a. Tracking multiple humans in complex situations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (9), pp. 1208–1221.

Zhao, T. and Nevatia, R., 2004b. Tracking multiple humans in crowded environment. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 406–413.

Zia, M. Z., Stark, M., Schiele, B. and Schindler, K., 2013. Detailed 3d representations for object recognition and modeling. IEEE transactions on Pattern Analysis and Machine Intelligence, 35 (11), pp. 2608–2623.

Acknowledgements

I want to express my grateful thanks to all persons who contributed to the success of this dissertation in one way or another.

First and foremost I thank my advisor Christian Heipke for his longstanding trust and support throughout my studies and my doctorate. He gave me the freedom to develop my own research ideas and his advise and assistance whenever it was needed.

Furthermore, I want to express my gratitude to Franz Rottensteiner for acting as referee of this dissertation. I benefited a lot from many insightful discussions with him about my research. I also thank Stefan Hinz and Steffen Schön for acting as referees of this dissertation and Ingo Neumann for chairing the examination committee.

Moreover, I thank all my friends and colleagues from the Institute of Photogrammetry and GeoInformation for the nice time and the warm and friendly working atmosphere. I enjoyed the encouraging and most memorable time in the office together with Moritz and Jakob. I also thank Joachim for his friendship and inspiring discussions in all these years. Furthermore, I thank Torge for the proofreading of the English text of my dissertation.

I express my cordial thanks to my parents, brothers and sister for their loving support and faith throughout my life. I would particularly like to thank my parents Margrit and Rolf for their holistic support in every respect. Finally, I thank my dear girlfriend Anna for her unconditional love and understanding during restless times.

Curriculum Vitae

Personal Information

Name	Tobias Klinger
Date of birth	23.11.1984 in Göttingen, Germany

Work Experience

Dec. 2010 – Jan. 2017	Institute of Photogrammetry and GeoInformation Leibniz Universität Hannover *Research Assistant*
Feb. 2013 – June 2013	Computer Vision Lab Pontifical Catholic University of Rio de Janeiro *Guest Researcher*
May 2010 – Nov. 2010	Geo++ GmbH, Garbsen *Technical Support Engineer*

Education

2009 – 2010	Alfred Wegener Institute *Diploma thesis*
2007 – 2008	Course Geodesy and Geoinformatics Technical University of Valencia, Spain *Erasmus program*
2004 – 2010	Course Geodesy and Geoinformatics Leibniz Universität Hannover *Diploma*
1997 – 2004	High School