Video-based Pedestrian Intention Recognition and Path Prediction for Advanced Driver Assistance Systems

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Andreas Schulz

aus Rastatt

Tag der mündlichen Prüfung:	8. November 2016
Erster Gutachter:	Prof. Dr.-Ing. Rainer Stiefelhagen
Zweiter Gutachter:	Prof. Dr.-Ing. Johann Marius Zöllner

Zusammenfassung

Fortgeschrittene Fahrerassistenzsysteme (*FAS*) spielen eine sehr wichtige Rolle in zukünftigen Fahrzeugen um die Sicherheit für den Fahrer, der Fahrgäste und ungeschützte Verkehrsteilnehmer wie Fußgänger und Radfahrer zu erhöhen. Diese Art von Systemen versucht in begrenztem Rahmen, Zusammenstöße in gefährlichen Situationen mit einem unaufmerksamen Fahrer und Fußgänger durch das Auslösen einer automatischen Notbremsung zu vermeiden. Aufgrund der hohen Variabilität an Fußgängerbewegungsmustern werden bestehende Systeme in einer konservativen Art und Weise konzipiert, um durch eine Restriktion auf beherrschbare Umgebungen mögliche Fehlauslöseraten drastisch zu reduzieren, wie z.B. in Szenarien in denen Fußgänger plötzlich anhalten und dadurch die Situation deeskalieren. Um dieses Problem zu überwinden, stellt eine zuverlässige Fußgängerabsichtserkennung und Pfadvorhersage einen großen Wert dar.

In dieser Arbeit wird die gesamte Ablaufkette eines Stereo-Video basierten Systems zur Intentionsschätzung und Pfadvorhersage von Fußgängern beschrieben, welches in einer späteren Funktionsentscheidung für eine automatische Notbremsung verwendet wird.

Im ersten von drei Hauptbestandteilen wird ein Echtzeit-Verfahren vorgeschlagen, das in niedrig aufgelösten Bildern aus komplexen und hoch dynamischen Innerstadt-Szenarien versucht, die Köpfe von Fußgängern zu lokalisieren und deren Pose zu schätzen. Einzelbildbasierte Schätzungen werden aus den Wahrscheinlichkeitsausgaben von acht angelernten Kopfposen-spezifischen Detektoren abgeleitet, die im Bildbereich eines Fußgängerkandidaten angewendet werden. Weitere Robustheit in der Kopflokalisierung wird durch Hinzunahme von Stereo-Tiefeninformation erreicht. Darüber hinaus werden die Kopfpositionen und deren Pose über die Zeit durch die Implementierung eines Partikelfilters geglättet.

Für die Intentionsschätzung von Fußgängern wird die Verwendung eines robusten und leistungsstarken Ansatzes des Maschinellen Lernens in unterschiedlichen Szenarien untersucht. Dieser Ansatz ist in der Lage, für Zeitreihen von Beobachtungen, die inneren Unterstrukturen einer bestimmten Absichtsklasse zu modellieren und zusätzlich die extrinsische Dynamik zwischen unterschiedlichen Absichtsklassen zu erfassen. Das Verfahren integriert bedeutsame extrahierte Merkmale aus der Fußgängerdynamik sowie Kontextinformationen mithilfe der menschlichen Kopfpose.

Zum Schluss wird ein Verfahren zur Pfadvorhersage vorgestellt, welches die Prädiktionsschritte eines Filters für multiple Bewegungsmodelle für einen Zeithorizont von ungefähr

einer Sekunde durch Einbeziehung der geschätzten Fußgängerabsichten steuert. Durch Hilfestellungen für den Filter das geeignete Bewegungsmodell zu wählen, kann der resultierende Pfadprädiktionsfehler um ein signifikantes Maß reduziert werden. Eine Vielzahl von Szenarien wird behandelt, einschließlich seitlich querender oder anhaltender Fußgänger oder Personen, die zunächst entlang des Bürgersteigs gehen aber dann plötzlich in Richtung der Fahrbahn einbiegen.

Abstract

Advanced driver assistance systems (*ADAS*) play a very important role in manufacturing future vehicles to enhance safety for human drivers, passengers and vulnerable road users like pedestrians and cyclists. These systems try to avoid collisions in dangerous situations involving an inattentive driver and pedestrian by triggering an autonomous emergency braking. Due to the high variability in pedestrian movement behavior, existing systems are designed in a conservative way by decreasing benefit in order to reduce false activation rates in scenarios where pedestrians are suddenly stopping and deescalating the situation. To overcome this problem, a reliable pedestrian intention recognition and path prediction are of great value.

This work presents the overall processing chain of a stereo-video based system for pedestrian intention recognition and path prediction in daily traffic scenarios to be integrated into a function for automatic emergency braking.

As one of three major parts, first, a real-time method is proposed that tries to localize pedestrian heads and to estimate their poses in low resolution gray value images including complex highly dynamic inner-city scenarios. Single frame based estimates are derived using confidence outputs of eight trained head pose classifiers applied at the image region of a pedestrian candidate. Further robustness in head localization is achieved by incorporating stereo depth information. Furthermore, head positions and head poses are tracked over time by implementing a particle filter.

For the task of intention recognition, the use of a robust and highly performing machine learning approach is investigated in different scenarios. This approach is able to model the intrinsic sub-structure of a specific intention class while additionally capturing extrinsic dynamics between different intention classes in time-series. Powerful features are integrated namely pedestrian dynamics by means of estimated lateral and longitudinal velocity components resulting from a pedestrian detection and tracking system as well as the pedestrian's awareness of an oncoming vehicle indicated by the human head pose.

Finally, a method for path prediction is developed that controls the prediction steps of a multiple motion-model filter for a time horizon of approximately one second by incorporating the estimated pedestrian intentions. By helping the filter to choose the appropriate motion model, the resulting path prediction error can be reduced by a significant amount. A wide range of scenarios is addressed including lateral crossing or stopping pedestrians or pedestrians that initially are walking along the sidewalks but then suddenly bend in towards the road.

Acknowledgments

I would like to thank all those who contributed to this thesis. First of all, my greatest thanks go to my doctoral supervisor Prof. Dr.-Ing. Rainer Stiefelhagen from the Institute for Anthropomatics and Robotics at the Karlsruhe Institute of Technology. His support in both, scientific and administrative issues was the key to the successful conclusion of this thesis. I also would like to thank Prof. Dr.-Ing. Johann Marius Zöllner for taking over responsibility as co-referee.

Dr. Thomas Heger and Dr. Klaus-Günther Fleischer deserve special thanks for giving me the opportunity to work on future innovative products in the field of video-based Advanced Driver Assistance Systems within their groups at the Robert Bosch GmbH in Leonberg. I express my thanks to Prof. Dr. Edgar Seemann, who supported me during the first months of my PhD. His advices built the basic parts of this work.

Further thanks go to my PhD colleagues at the Robert Bosch GmbH Thomas, Johannes, Marc, Susanne, Florian, and Stephan for their valuable inputs during interesting discussions. Many thanks go to Dr. Alexander Schick and PD Dr. Marco Huber for their help with proof reading.

My deepest thanks go to my parents Hildegard Uhrig and Reinhard Schulz. You enabled and encouraged my professional career and supported me during challenging times. I thank my brothers Mathias and Benedikt and my sister Judith for helping me to maintain focus on the successful conclusion of my thesis. At the end, I want to express special thanks to my girlfriend Filomena. You were the daily driver that helped me to keep on track. No matter how difficult the times, I could always count on your love and support.

Erklärung

Ich versichere wahrheitsgemäß, die Dissertation bis auf die dort angegebenen Hilfen selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und als kenntlich gemacht zu haben, was aus Arbeiten anderer und eigenen Veröffentlichungen unverändert oder mit Änderungen entnommen wurde.

Karlsruhe, 12. März 2017

Contents

List of Figures

List of Tables

1 Introduction

During the past years, worldwide commissions for transport and road traffic are increasingly focusing on the reduction of the number of pedestrian injuries and fatalities due to road accidents. First measures for infrastructure, e.g. signalizing pedestrian areas and crosswalks, safety barriers and speed limits [WHO, 2013], and early active and passive safety systems for vehicles [Lawrence et al., 2004] resulted in a significant reduction of the total number of fatalities [Yannis et al., 2015]. But nevertheless, in 2013, 22% of all fatalities in road accidents within Europe still were pedestrians, see Figure 1.1.

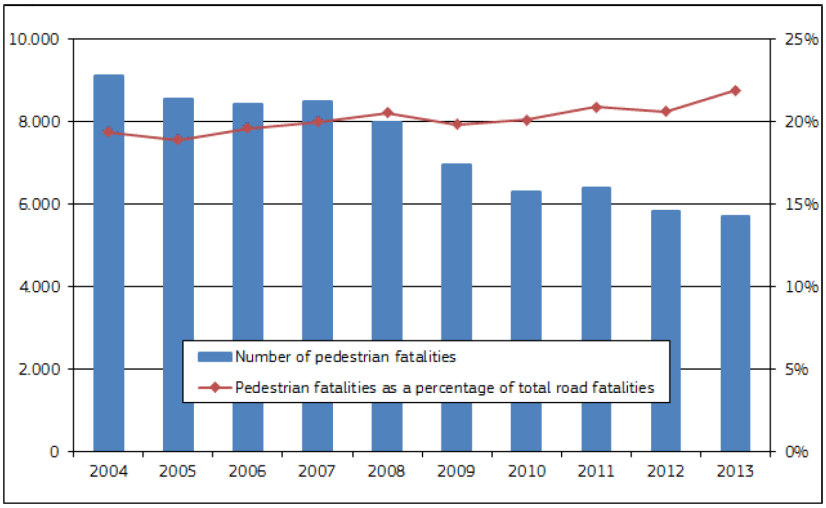


Figure 1.1: Number of pedestrian fatalities and percentage of all road fatalities within EU, 2004-2013, [Yannis et al., 2015].

Additionally, in the U.S., the number of pedestrian fatalities of about 4,500 corresponding to 15% of all fatalities [NHTSA, 2016] still remains critically stable.
In this context, advanced driver assistance systems (*ADAS*) play an important role in manufacturing future vehicles to enhance safety for human drivers, passengers, and vulnerable road users such as pedestrians and cyclists. There are several reasons, namely increasing computational power of embedded platforms at low cost or emerging technologies in the sector of intelligent sensors like video, radar, laser or ultrasonic. Especially, the static improving performance of video-based pedestrian detection resulted in first commercial active pedestrian protection systems available for a wide range of vehicles, including Mercedes Benz, Volvo, or Volkswagen. Those kinds of systems try to avoid collisions to a limited extent in dangerous situations involving an inattentive driver and pedestrians by warning

the driver or triggering an automatic emergency braking (*AEB*), cf. [Euro NCAP, 2016]. Recently, new growing up companies from the Silicon Valley like Tesla Motors or Google integrate these kinds of technologies with additional extensions to manufacture completely self-driving vehicles realizing a fusion of multiple redundant sensors.

1.1 Motivation and Challenges

Most of those systems are already able to reliably detect pedestrians. However, for a higher benefit in collision avoidance or preventive actions one needs to get deeper information about the underlying scene and the pedestrian's future behavior. One of the most challenging task is to interpret situations correctly, where pedestrians are laterally approaching the roadside. Due to the ability of having a high variability of movement patterns, pedestrians can change their walking direction within a short time period or suddenly start or stop. In the latter case of an abruptly stopping pedestrian, triggering an autonomous braking would be a false reaction, which could result in consequential damages involving the driver, passengers, and other traffic participants. Therefore, existing systems are designed in a very conservative way by limiting the system's availability to a small set of controllable scenarios in order to reduce potential false activations (*FAs*) and thus, to meet the requirements of automotive safety integrity levels (*ASIL*), cf. [ISO 26262-1:2011, 2011]. To overcome this limitation, a reliable pedestrian intention recognition and path prediction is of a great value. In this context the human head pose plays an important role as it gives a hint of the pedestrian's awareness of an approaching vehicle. A pedestrian facing the vehicle may be aware of it and thus, is less likely to cause a critical situation in comparison to pedestrians just walking towards the road without observing the environment. A following derived intention recognition helps improving situation interpretation and therefore, to avoid system false activations and at the same time keeping the benefit at a sufficiently high level.

As one can expect, a lot of difficult scenarios and reactions have to be considered and figured out. [Schmidt and Färber, 2009] provide dominant features and their significance for pedestrian intention recognition, see Figure 1.2. Three quarter of the most powerful features, i.e., head/leg movement, dynamics, and characteristics, are related to the pedestrian itself. The remaining quarter includes aspects depending on the environment and context. This motivates to focus on pedestrian intention recognition, where most of these features are derived from internal data provided by an available stereo-video based driver assistance system. [Oxley et al., 1995] propose a study on head movement behavior for older and younger people during road crossings. In their experiments it turns out, that crossing pedestrians spend a dominant amount of time (>45%) in facing the nearside traffic, i.e., oncoming vehicles, while approaching the curbside. On average 1.4 to 1.5 dominant head turnings can be observed during that time. [Hamaoka et al., 2013] present similar results on head turning rates especially when pedestrians are already very close to the road. As a result thereof, the head pose is the important hint (most dominant feature in [Schmidt and Färber, 2009]) on

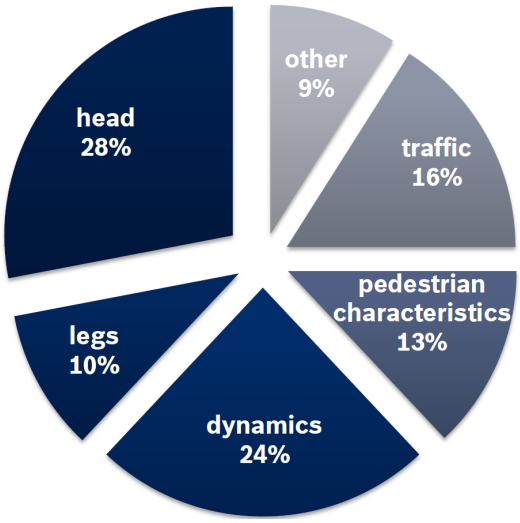


Figure 1.2: Feature categories for pedestrian intention recognition, [Schmidt and Färber, 2009]. *Head* (focusing, left/right turnings), *Legs* (foot liftings on the street), *Dynamics* (running, walking, standing still), *pedestrian characteristics* (upper body movement, distance to curb, age), *Traffic* (traffic density, speed of oncoming vehicles), *other* (zebra crossing, mother with child, group behavior)

what the pedestrian will do next. In the context of ADAS, the head pose indicates the pedestrians' awareness of the vehicle existence but it also helps to predict the intended movement direction of the pedestrian for a further use within tracking methods or risk assessment. For the task of head pose estimation in video-based driver assistance systems using a vehicle mounted camera, one has to deal with low resolution, gray-value images, as well as highly dynamic variations in illumination and complex environment. Furthermore, operation should be possible at daytime, nighttime and under different weather conditions. Such a system also needs to be robust to large variations in pedestrian and head appearances, and it needs to guarantee a high degree of reliability.

From algorithmic point of view, the following problems arise. Because of the low resolution images and the wide head pan angle range required for such a system, head pose estimation based on facial feature detection, as for example in [Gee and Cipolla, 1994, Horprasert et al., 1996, Vatahska et al., 2007], will not be suitable. The lack of color information also prevents the use of color histograms, cf. the works of [Benfold and Reid, 2008, Benfold and Reid, 2009, Canton-Ferrer et al., 2007, Canton-Ferrer et al., 2008, Siriteerakul et al., 2010] or [Robertson and Reid, 2006] for skin color detection. The main challenges in the context of this work are person bodies appearing on very low resolutions in inner-city scenarios, i.e., head sizes down to approximately 6×6 pixels in order to estimate the pedestrians' intention as soon as possible, see Figure 1.3. Additionally, heads strongly vary in their appearances, e.g. for people wearing hats or umbrellas, with or without beard or having long or short hair. For intention recognition itself, one has to find a way to integrate all necessary information within a model that is robust against people dependent variations. For instance, elder pedestrians are normally walking slower than younger ones. Some people turn their heads faster or slower than others or even do face the approaching vehicle but still cross the road.

One of the main critical aspects for current systems realizing an AEB on pedestrians is the predicted lateral impact point between approaching vehicle and pedestrian. Already a prediction error larger than 30cm can make the difference of having a false activation or saving a pedestrian life. Therefore, current systems restrict the range for the predicted impact point relevant for a brake intervention till a desired FA-rate is achieved over a larger endurance run while still guaranteeing a certain level of benefit. Without any further information as given by an estimated pedestrian intention recognition, the error in path prediction hardly lies below the desired threshold, especially for an abruptly stopping pedestrian. The arising question is then how to integrate pedestrian intentions in order to achieve better results. Also the path prediction result still has to show acceptable performance in case of an uncertain intention recognition.
Besides the functional requirements, algorithms have to operate in real-time with the computational power of hardware that is used for automotive systems. In general, this is restricted in processor performance and on board memory, which prevents to application of more complex algorithms.
Another system requirement is a high degree of availability in terms of functional range. Figure 1.3 shows a typical brake characteristic of an AEB system. Visualized is the object distance relative to the ego vehicle, at which an automatic braking action has to be triggered, over the ego vehicle's speed.

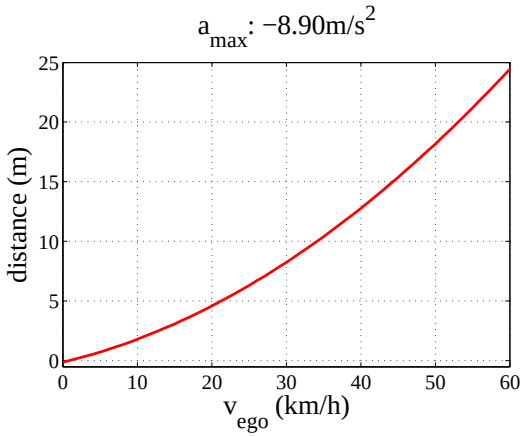


Figure 1.3: Typical brake characteristic curve for AEB systems. Object distance at which an AEB with a maximum deceleration of -8.9m/s^2 has to be triggered over ego vehicle speed.

For the addressed inner city scenarios, where vehicles are moving with a speed up to 60km/h, it is mandatory to reliably detect objects, more specifically pedestrians, and provide all necessary information for a later function reaction up to 25m. This means, that an initial object hypothesis already has to be generated at higher distances of up to 30m approximately, in order to be able to perform object plausibility checks and derive temporal information for a safe function decision. Later experiments will focus on system performance within this range.

1.2 Contribution

This work presents a system for pedestrian intention recognition and path prediction in city traffic scenarios. The scope of application is a video-based advanced driver assistance system for forward collision warning and collision avoidance of a driving car with other vulnerable road users (*VRU*) such as pedestrians. The group of VRUs is a risky class because lack of attention by the driver and the pedestrian could result in a collision. Pedestrian candidates are detected by a stereo-video system mounted behind the windshield of an approaching vehicle. Motivated by scientific evidence about pedestrian behavior in city traffic situations, [Hamaoka et al., 2013, Oxley et al., 1995, Schmidt and Färber, 2009], intention decisions will be made by two dominant factors. Firstly, the pedestrian dynamics by means of predicted lateral and longitudinal velocity components as a result of a non-linear dynamical system. Secondly, the pedestrian's awareness of an oncoming vehicle with the help of observing the human head pose. Figure 1.4 shows a rough overview of the presented system.

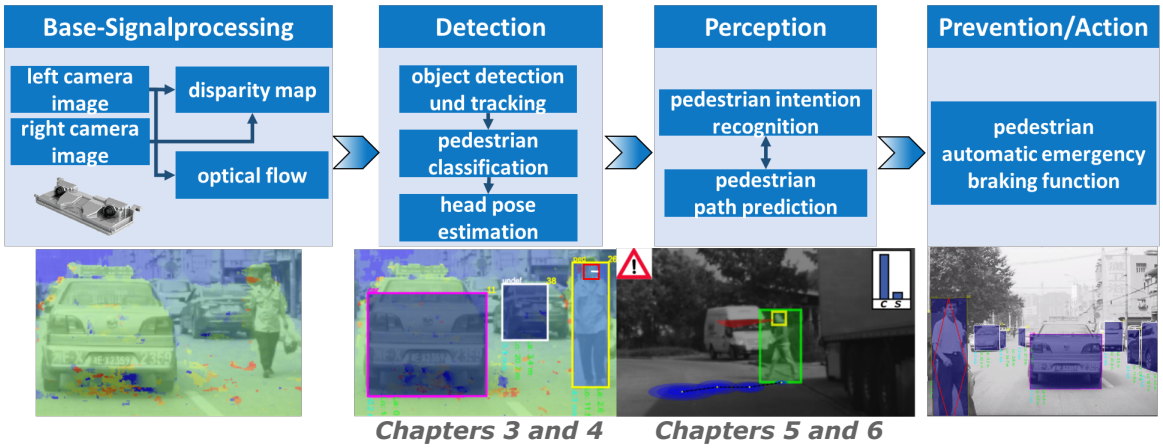


Figure 1.4: System Overview. Pedestrians are detected and tracked by a stereo-video based generic obstacle detection and tracking approach and verified by an image intensity-value based classifier. For each pedestrian candidate the head orientation will be estimated from the image data. The previous extracted information serves as input for an iterative intention recognition and path prediction.

The overall processing pipeline of the system starts with the estimation of the pedestrian's head pose and ends with a concrete prediction of the pedestrian's future position and intention that can be incorporated into a function for automatic emergency braking. Similar to recent contributions to pedestrian intention recognition and the closely related field of pedestrian path prediction, this work uses the outputs from an existing system for video-based pedestrian detection. The main contributions of this work are:

1. A method for video-based pedestrian head localization and head pose estimation in low-resolution images taken from a stereo-video camera.

2. A system for pedestrian intention recognition using the information of the human head pose and additional features.

3. Integration of the pedestrian intention into an improved path prediction to be used for a later decision on forward collision warning (*FCW*) or automatic emergency braking.

Pedestrian Head Pose Estimation

In the first part of this work, a real-time method is proposed, that realizes a combined pedestrian head localization and head pose estimation. The underlying resources are gray value images taken from a vehicle mounted camera. These images contain complex highly dynamic inner-city scenarios, where pedestrian heads appear at very low resolution. In a first step a single frame based approach is developed, where the confidence outputs of eight trained head pose classifiers applied to the image region of a pedestrian candidate are processed to result in a head location and head pose estimate. Furthermore, head positions and head poses are tracked over time by implementing a particle filter. This leads to a more robust, more reliable and more efficient pedestrian head pose estimation. In addition, through the nature of the proposed method it is possible to estimate continuous head pan angles instead of determining only one of eight discrete head pose classes, cf. [Orozco et al., 2009, Siriteerakul et al., 2010]. By incorporating stereo depth information, head localization and head pose estimation results are improved in two ways. First, an additional stereo-based head localization achieves more robustness in scenarios with highly structured background. Second, estimates on the 3D object movement direction serve as prior knowledge for expected human head poses. The presented approach is applied on top of an overall system for stereo-vision based pedestrian detection and tracking and will serve as input for a later pedestrian intention recognition and path prediction. In comparison, recent approaches mainly rely on a pedestrian orientation estimation to get an idea of the pedestrian intention, cf. [Gandhi and Trivedi, 2008, Shimizu and Poggio, 2004, Enzweiler and Gavrila, 2010]. However, the pedestrian's movement direction does not indicate the real pedestrian's awareness of an approaching vehicle a priori. Hence, with the head pose estimation, a more reliable long-term intention recognition for the upcoming few seconds can be derived.

Pedestrian Intention Recognition

In this work, the use of a robust and highly performing machine learning approach, namely Latent-Dynamic Conditional Random Fields [Morency et al., 2007], for the task of intention recognition in different scenarios is investigated. This approach is able to model the intrinsic sub-structure of a specific intention class while additionally capturing extrinsic dynamics between different intention classes in time-series. The method integrates features extracted from pedestrian dynamics as well as contextual information using the human head pose. An additional benefit of this model is the fact that it can work with time series of arbitrary

length, where additional confidence about pedestrian intentions can be retrieved by temporal integration. By the nature of the model, it is possible to capture dynamical changes for single features, such as head turnings or different movement behaviors. Different connections of features can be learned automatically from training data but nevertheless, expert knowledge still can be brought in when designing the structure of the model. Compared to recent approaches in literature, besides stopping and crossing pedestrians, a wider range of scenarios is addressed including pedestrians that initially walk along the sidewalks but then turn towards the road. The proposed latent-dynamic discriminative model applied on extracted time-series is then able to integrate all the previous mentioned features including pedestrian dynamics and pedestrian awareness, and to learn inner connections within a specific type of scenario and external correlations between different types of scenarios. The output of the presented approach will be integrated into a later system for pedestrian path prediction, but could also be used for controlling the switching states of the Switching Linear Dynamical System (*SLDS*) presented in [Kooij et al., 2014a].

Pedestrian Path Prediction

General object tracking algorithms basically consist of two steps, *Prediction* and *Update*. For prediction the object state is propagated using suitable motion models, resulting in an object state hypothesis for the next time instance. During the update step new sensor measurements are incorporated to correct the previously performed prediction. The underlying object tracking method consists of the well-known Interacting Multiple Model Filter (*IMM*), as this filter is able to deal with multiple motion models, which turned out to be practicable for the task of pedestrian tracking. Path prediction can be performed by applying the prediction steps multiple times for a small time horizon (< 1 second) without incorporating new measurements. In this context, a method is developed that controls the filter prediction steps by incorporating the estimated pedestrian intentions. In doing so, the filter is supported to choose the appropriate motion model and thus, the resulting path prediction error can be reduced by a significant amount.

1.3 Outline

The remainder of this work is organized as follows. Chapter 2 presents an overview of related work in the field of video-based pedestrian detection, stereo-video based object detection and tracking and focuses on human head pose estimation in low resolution images as well as on pedestrian intention recognition and path prediction. The approach for pedestrian head pose estimation developed in this work is presented in Chapter 3. Chapter 4 demonstrates the benefits of using stereo vision for a more robust and reliable head localization and head pose estimation. The second major contribution of this work on pedestrian intention recognition is presented in Chapter 5, where the head pose estimation is integrated as a powerful feature.

Chapter 6 finally combines pedestrian intention recognition and tracking for path prediction within a controlled framework. The overall approach is embedded into an existing stereo-video system for pedestrian forward collision warning and automatic emergency braking. The work concludes with a discussion in Chapter 7.

2 Related Literature

This chapter gives an overview of related contributions. Since the substantial part of this work is based on the reliable detection and tracking of pedestrians in video sequences, related literature to this topic will be presented here. Furthermore, methods for 3D object detection using stereo video sensor data follows. As presented in Chapter 1, one core part of this work deals with the estimation of pedestrian head poses, which directly has impact on an intention recognition in a subsequent step. The latter in turn is used in order to achieve an improved path prediction to increase the benefit of automatic emergency braking systems. For all three contribution topics current related literature is presented. This chapter is structured as follows. Section 2.1 presents an overview of current approaches for image-based pedestrian recognition. General methods for 3D object detection and tracking using stereo-vision will be introduced in Section 2.2. In Section 2.3 pursuing approaches for detection of human heads and head pose estimation are introduced. Section 2.4 and 2.5 finally deal with the intention recognition of pedestrians and followed path prediction.

2.1 Monocular Pedestrian Detection

There has been done a lot of research in the field of video-based pedestrian detection during the past decade. Especially in the field of ADAS this task is very hard to handle due to a wide range of challenging scenarios, i.e., high variations in pedestrian appearance in highly dynamic scenes. A broad overview of the research area on pedestrian detection for driver assistance systems is given by [Gandhi and Trivedi, 2007]. Here, besides video-based approaches, further methods using different sensor technologies like radar or laser are summarized. For the main subject of video-based pedestrian detection, [Enzweiler and Gavrila, 2009, Dollár et al., 2012b] and [Benenson et al., 2014] provide detailed evaluations of state-of-the-art approaches. Currently, so-called discriminative methods distinguished themselves. Here, one tries to learn decision functions based on extracted features using a huge amount of labeled training examples, which allows a separation of pedestrians from other image contents. Available methods basically differ in the type of extracted features, the fusion of multiple features and the use of machine learning methods for determination of the separating plane. One of the most powerful features is given by the Histogram of oriented Gradients (*HoG*), which is robust against changes in scale and illumination and initially was introduced by [Dalal and

Triggs, 2005]. Variations thereof are presented and used in [Dalal et al., 2006, Jia and Zhang, 2007, Zhu et al., 2006, Wang et al., 2009, Maji et al., 2008, Dollár et al., 2010, Benenson et al., 2012b]. The basic idea is to represent the pedestrian appearance and shape in the image by a distribution of edge gradients. For this, a selected image patch is divided into overlapping blocks consisting of multiple cells. Image gradients contribute to a histogram per cell based on their orientation. The final feature vector is obtained by concatenating the normalized orientation histograms with respect to each block. [Wang et al., 2009] and [Zhu et al., 2006] describe HoG similar features, which are calculated using integral histograms and therefore significantly reduce computational effort compared to the conventional feature in order to achieve real-time capability. Other approaches for pedestrian recognition use illumination invariant local binary pattern (*LBP*) descriptors [Mu et al., 2008, Wang et al., 2009]. LBP features in their original form represent the 3×3 neighborhood of a pixel using a bit-string by comparing the intensity values of all pixels in the neighborhood with the one for the center pixel and hence are very fast to compute.

For discriminative methods different types of classifiers are used. In this context, Support Vector Machines (*SVM*) are widely used in both linear, [Dalal and Triggs, 2005, Dalal et al., 2006, Enzweiler and Gavrila, 2011, Wojek et al., 2009], and non-linear variants, see [Papageorgiou and Poggio, 1999, Dalal and Triggs, 2005, Maji et al., 2008]. Other popular classifiers include neural networks, [Gavrila and Munder, 2007, Enzweiler et al., 2010, Enzweiler and Gavrila, 2010], Adaptive Boosting (*AdaBoost*), [Dollár et al., 2010, Wojek et al., 2009] and cascades of boosted classifiers, [Tuzel et al., 2008, Viola et al., 2003, Zhu et al., 2006, Wang et al., 2009, Jia and Zhang, 2007]. In general, Adaboost classifiers as presented by [Freund and Schapire, 1997, Freund and Schapire, 1999] help to select the most powerful features from an over-complete set of features, like Haar wavelets, cf. [Viola and Jones, 2001a].

Recent methods combine multiple features using shape-, texture- and color information in the form of integral channels in order to achieve higher detection rates, see [Dollár et al., 2009a, Dollár et al., 2010] and [Benenson et al., 2012b]. Most of the publications concerning automotive applications, use a fusion of several cues from depth, motion and intensity together with object detection and classification techniques. [Gavrila and Munder, 2007, Keller et al., 2011a, Keller et al., 2011b, Schneider and Gavrila, 2013] for instance use stereo information, shape and texture based features as well as tracking approaches. Similar approaches are also used by [Bajracharya et al., 2009, Wojek et al., 2009, Walk et al., 2010, Enzweiler et al., 2010, Rohrbach et al., 2009]. Other methods directly make the use of stereo depth information in order to restrict the search space for a following pedestrian classification, cf. [Bertozzi et al., 2005, Bajracharya et al., 2009, Enzweiler et al., 2010, Enzweiler et al., 2012, Keller et al., 2011b]. Related to this topic, [Llorca et al., 2012] present a survey on stereo region of interest (*ROI*) selection. This significantly reduces the run-time for monocular pedestrian detection resulting in real-time systems and helps to reduce the false positive rate while keeping a suitable detection rate, e.g., see [Enzweiler and Gavrila, 2011].

Further improvements can be achieved by incorporating scene context [Hoiem et al., 2006, Torralba, 2003] or contextual information from neighboring regions on detector level [Ding and Xiao, 2012, Dollár et al., 2012a].

2.2 Stereo-Vision based Object Detection and Tracking

Recent approaches for autonomous driving use the benefits provided by a stereo vision sensor. The idea is to extract 3D depth measurements in terms of disparities calculated by matching images regions from the left and right camera images. There already exists a wide range of stereo algorithms that differ in run-time, density and accuracy, see [Geiger et al., 2012] just like standard block matching or semi global matching. These algorithms mostly differ in the way how to describe image areas in order to easily find the best matching areas in both images in addition to smoothness assumptions for estimated disparities within a pixel neighborhood.

Most of existing methods for stereo-based object recognition try to divide the given disparity map into clusters or regions of interest belonging to raised obstacles, cf. [Labayrade et al., 2002, Leibe et al., 2007, Ess et al., 2007, Leibe et al., 2008, Ess et al., 2009, Nedevschi et al., 2009]. Some methods first try to extract the ground plane in order to restrict search space for vertical oriented column-wise features, also known as *Stixels*, cf. [Labayrade et al., 2002, Badino et al., 2009, Pfeiffer and Franke, 2011]. For real-time applications, [Benenson et al., 2011, Benenson et al., 2012a] propose very fast versions for Stixel calculation without the necessity of a computed disparity map in a pre-processing step. Other methods, for instance [Ess et al., 2007, Ess et al., 2009], try to simultaneously infer the ground plane and the set of valid pedestrian hypotheses using a Bayesian Network for temporal stability and accuracy. Concatenating similar segments in terms of similar depth estimates together results in larger object hypothesis or clusters. These generic obstacles are then tracked with a sequence of images using Kalman Filters, cf. [Bar-Shalom et al., 2002, Blackman and Popoli, 1999, Li and Jilkov, 2001, Li and Jilkov, 2003, Li and Jilkov, 2005], or particle filters, [Arulampalam et al., 2002, Isard and Blake, 1998], to get smooth, stable and more accurate estimates. Additionally, further unobservable information about the object state like velocity, acceleration or collision risk with respect to the ego vehicle can be obtained out of the filter. A further processing step involves object classification to separate vehicles, pedestrians or other classes from objects generated on remaining infrastructure, [Leibe et al., 2007, Keller et al., 2011b, Enzweiler et al., 2012, Ess et al., 2009, Gavrila and Munder, 2007].

2.3 Human Head Localization and Head Pose Estimation on low Resolution Images

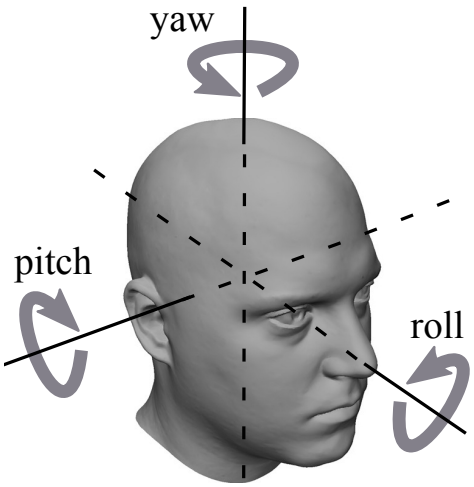


Figure 2.1: Headpose coordinate system.

As this work focuses on the head localization and head pose estimation problem for pedestrians, related literature dealing with both problems is discussed here. Moreover, this section introduces some approaches for pedestrian orientation estimation due to their application similarity in the field of driver assistance systems.

The task of human head pose estimation mostly has to be divided into the two major parts, namely head localization or head detection and the actual head pose estimation given a head candidate. This work differentiates between head localization and head or face detection. Head localization is a specific type of head detection, where the number of heads in a given image is already known. In comparison to face detection, which usually deals with frontal poses of human heads, head localization tries to localize the head in any given pose. Nevertheless, the approach of this work is largely related to the problem of face detection in general. One of the most related contribution in this field is the work of [Viola and Jones, 2001a], where a cascade of boosting classifiers based on Haar-like features is used to detect faces in gray-value images. Due to the cascaded structure of the classifiers and the fast feature computation using integral images, face detection can be performed in real-time. [Jones and Viola, 2003] generalize their method to detect faces appearing in multiple views and heads under various poses. Another major approach for face detection is the use of the Modified Census Transform (*MCT*) presented by [Fröba and Ernst, 2004] to describe the neighborhood around pixel positions by binary structures similar to the LBP feature. Features based on the MCT will be combined in a similar way to [Viola and Jones, 2001a] by training a cascaded boosting classifier. Recently, [Zheng et al., 2010] suggested an extended 12-bit MCT feature set for low resolution face detection in color images. Others like [Roy and Marcel, 2009] propose the *Haar-LBP* feature, a hybrid between Haar-like features and

Local Binary Patterns, to achieve an illumination-invariant face detection.

Head pose estimation approaches are usually application driven. [Murphy-Chutorian and Trivedi, 2009] present a survey of existing head pose estimation approaches for different scenarios and variable kinds of input data. Depending on the application, head pose estimation methods vary in their requirements on input data, accuracy, and run-time. Applications in human machine interaction, for example addressed by [Vatahska et al., 2007], require more precise pose measures related to rotations around the main axes of a head located coordinate system (*pan/yaw*, *tilt/pitch*, *roll*), see Figure 2.1. There, the input data mostly contains high resolution head images and sometimes depth information using stereo or multiple camera systems, cf. [Fanelli et al., 2011, Martin et al., 2014, Seemann et al., 2004, Voit et al., 2008, Voit and Stiefelhagen, 2009]. These approaches mostly consider the range from frontal face poses to side views of heads (-90°,...,90°). Recent literature about head pose estimation for driver assistance systems mostly considers the determination of the driver's head pose in the vehicle interior for health monitoring and intention recognition, cf. [Bär et al., 2012, Murphy-Chutorian et al., 2007] and [Martin et al., 2014]. In contrast, this work focuses on pedestrian head pose estimation using a vehicle mounted camera. For this task, also addressed by [Flohr et al., 2014, Rehder et al., 2014], or in the field surveillance as presented in [Benfold and Reid, 2009, Hirata et al., 2008, Orozco et al., 2009, Siriteerakul et al., 2010, Chen et al., 2011, Chen and Odobez, 2012, Robertson and Reid, 2006], one has to deal with low resolution images under harsh conditions in order to provide head pose estimates covering the full head pan angle. Fortunately, there are usually weaker demands such as estimating a discrete head pose class and larger pose steps without the necessity to give estimates on tilt and roll angles. Regarding the camera setup and available information, many approaches use a network of two or more calibrated cameras, cf. [Bäuml et al., 2010, Canton-Ferrer et al., 2008, Niese et al., 2006, Zhang et al., 2007, Seemann et al., 2004, Fanelli et al., 2011]. Nevertheless, some approaches are based on conventional 2D image signals, see [Ba and Odobez, 2004, Benfold and Reid, 2008, Benfold and Reid, 2009, Hirata et al., 2008, Orozco et al., 2009, Siriteerakul et al., 2010, Robertson and Reid, 2006].

To estimate the head pose, different algorithms have been presented. Recent methods make use of the Iterative Closest Point algorithm (*ICP*, [Chen and Medioni, 1992]) to find rigid transformations between a pre-defined 3D head model and sequentially extracted point clouds from consumer depth cameras to achieve highly accurate angle measurements, cf. [Bär et al., 2012, Martin et al., 2014]. However, high resolution images are required in addition to a precise head position and head pose initialization. For the problem in this work, a so-called *detector array*-based approach is chosen, where a series of head detectors each assigned to a specific head pose is trained. To estimate the head pose, a discrete pose class will be determined based on the detector with the greatest support over a given image region. Related work on this can be found in [Jones and Viola, 2003, Rowley et al., 1998, Flohr et al., 2014, Zhang et al., 2007]. One advantage of this type of methods is, that it can be easily used for head localization as well by explicitly learning the separation from background regions

during the training process. In contrast, other methods have to work with regions including head candidates, that have to be generated in advance using generally trained head detectors or foreground-background segmentation as an output of complex environment models. Another benefit of the chosen approach is the possibility to make use of head detectors, that are well suited for evaluation on low resolution images due to their integrated features which can work on lowest pixel level as presented in [Fröba and Ernst, 2004] and [Viola and Jones, 2004]. Some works estimate the head pose of a given head candidate by measuring the similarity to discrete appearance templates, cf. [Orozco et al., 2009], while others try to detect facial features and then estimate the head pose according to the geometrical positions of these features, cf. [Gee and Cipolla, 1994, Horprasert et al., 1996, Stiefelhagen et al., 1996] and [Vatahska et al., 2007]. [Benfold and Reid, 2008] and [Benfold and Reid, 2009] use *Ferns* to determine skin and non-skin segments of the head, based on color information. In a second step they estimate the head pose to be one of eight pose classes.

The following two methods are presented in more detail as they serve as a baseline for later performance evaluation in this work. The method of [Orozco et al., 2009] builds upon a similarity distance between a given head sample and a set of pre-computed mean head pose templates. Motivated by the *Kullback Leibler divergence*, which was originally introduced in [Kullback and Leibler, 1951] to compare two probability density functions, they define *Kullback Leibler coefficients* δ_{KL} to measure the pixel-wise similarity of a given head image with all head pose templates. Defining the intensity values of the mean template corresponding to head pose class c and the input sample at pixel position (i, j) by $m^c_{i,j}$ and $n_{i,j}$, respectively, the divergence coefficient can be expressed as

$$\delta_{KL}(m^c_{i,j}||n_{i,j}) = \max_{RGB} \left\{ m^c_{i,j} \left(\log \frac{m^c_{i,j}}{n_{i,j}} \right) \right\}. \tag{2.1}$$

The final feature descriptor is a 2D similarity distance weighting map $\{x_{i,j}\}_{i,j}$, where each pixel position (i, j) is determined by taking the maximum KL divergence coefficient at that position over all head pose templates, i.e.,

$$x_{i,j} = \max_{c \in \{1,\dots,8\}} \delta_{KL}(m^c_{i,j}||n_{i,j}). \tag{2.2}$$

Figure 2.2 shows the mean pose templates for their eight head pose clusters and some input head samples with their corresponding descriptor.

For the final classifiers they compute the introduced descriptors on a set of training samples and train a multi-class *one-vs.-all* SVM. In their experiments they show better results in terms of higher classification rates compared to [Robertson and Reid, 2006].

The second method to compare with was introduced by [Siriteerakul et al., 2010]. Here, the basic idea is, that on very low resolution images one has to directly work on pixel level. The feature they propose, namely non-local Intensity Difference Feature (*iDF*) compares the

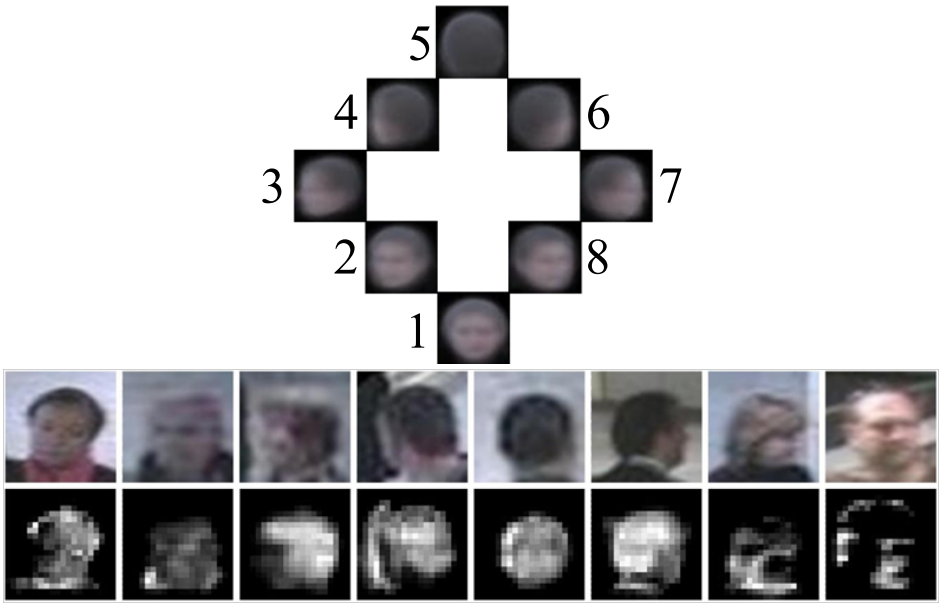


Figure 2.2: Method proposed by [Orozco et al., 2009]. Upper image: Head pose templates for each of eight discrete head pose classes using mean intensity image over training data. Lower Image Calculated Descriptor on a set of test samples.

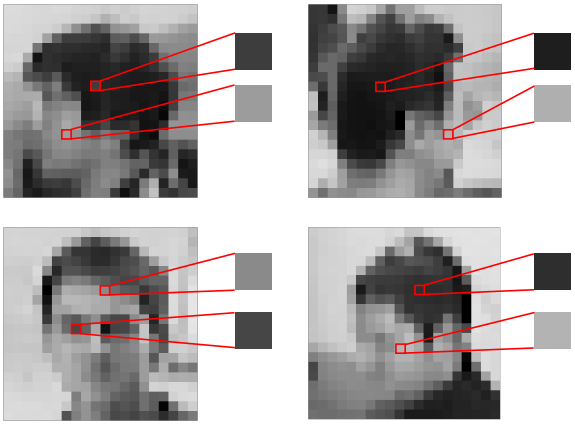


Figure 2.3: The method of [Siriteerakul et al., 2010]. Non-local intensity differences features are calculated at multiple image positions to train a classifier.

intensity values of two arbitrarily chosen pixel positions $\mathbf{p}_1$ and $\mathbf{p}_2$ as follows.

$$iDF(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} 0 & \text{if } I(\mathbf{p}_1) = I(\mathbf{p}_2) \\ 1 & \text{if } I(\mathbf{p}_1) > I(\mathbf{p}_2) \\ 2 & \text{if } I(\mathbf{p}_1) < I(\mathbf{p}_2) \end{cases}, \tag{2.3}$$

where $I(\mathbf{p})$ denotes the image intensity value at pixel position $\mathbf{p}$. Figure 2.3 shows different choices of pixel pairs for some input test samples. For training, 10000 random pixel pairs are chosen within a 20×20 image patch. Considering eight head pose classes, the iDF is calculated for a set of training samples to train a multi-class one-vs.-all SVM. After several training loops, choosing a random subset of pixel pairs to be considered and varying the sam-

ples taken for training and validation, they choose the best performing classifier at the end. Additionally, they present a non-local Color Difference Feature (*cDF*) using color images, which further improves classification rates in combination with the iDF.
Within the context of pedestrian intention recognition for driver assistance systems, approaches on pedestrian orientation estimation are presented. From the application point of view, in video driver assistance systems the head pose is used as a cue for pedestrian intention recognition. Some works discuss the pedestrian body pose as an alternative measure of a pedestrian's intention and gaze in order to improve tracking algorithms, cf. [Enzweiler and Gavrila, 2010, Gandhi and Trivedi, 2008, Shimizu and Poggio, 2004]. Most of them assume already detected pedestrians and try to estimate their orientations, see [Gandhi and Trivedi, 2008, Shimizu and Poggio, 2004]. Others integrate the pedestrian detection and orientation estimation within a probabilistic mixture of experts framework to achieve better results, cf. [Enzweiler and Gavrila, 2010]. [Shimizu and Poggio, 2004] use Haar-like features to train *one-vs.-all SVMs*, while [Gandhi and Trivedi, 2008] train *one-vs.-one* SVMs using Histograms of oriented Gradients from [Dalal and Triggs, 2005]. [Enzweiler and Gavrila, 2010] also use HoG features combined with support vector machines, as well as adaptive local receptive fields (*LRF*) in combination with a multilayer neural network. Evaluations on their test data show the positive effect of the integrated detection and orientation estimation in comparison to the results of [Gandhi and Trivedi, 2008, Shimizu and Poggio, 2004]. Current methods in the field of surveillance or driver assistance systems take advantage of the fact, that the observed head pose is dependent on the human body pose and vice versa. [Chen et al., 2011, Flohr et al., 2014] give estimates for both head- and human body pose within a joint probabilistic framework, taking into account anatomical constraints and detector uncertainties.

2.4 Pedestrian Intention Recognition

This section lists the main recent contributions on pedestrian intention recognition. Pedestrian behavior and intention in daily traffic situation has been studied for years from various perspectives like crash studies, crowd behavior and optimizing infrastructure to enhance pedestrian safety. In this context, many approaches from psychological aspect are based on statistical models, which are determined on the basis of collected data, see [Oxley et al., 1995, Schmidt and Färber, 2009, Fugger et al., 2001, Avineri et al., 2012, Evans and Norman, 2003, Kadari and Vedagiri, 2013, Bernhoft and Carstensen, 2008]. To take environmental aspects into account, [Hamaoka et al., 2013] analyze pedestrian head turning behaviors at crosswalks regarding the best point of warning for inattentive pedestrians. In the field of intelligent vehicles, one attempts to derive potential intentions of a pedestrian using sensor data like video sequences for instance. One related subject area is the field of action recognition for human machine interaction, where a humanoid robot shall recognize a certain kind of human actions, gestures or facial expressions in order to react by offering support or ex-

ecuting commands. [Turaga et al., 2008] present a survey on this field. One basic idea is the use of the so-called motion history feature, where extracted human shapes are aggregated over a sequence of images to learn a template for one particular action. A derivation of this approach is used by [Köhler et al., 2012] and [Köhler et al., 2013] in the automotive field for early detection of a starting action for an initially standing pedestrian at the curbside. This only involves an infrastructure-based sensor setup, i.e., no on-board vehicle mounted camera. For a given pedestrian track, they calculate a motion contour image-based HoG-like descriptor. The differentiation of a standing pedestrian and a potential starting action is trained using the motion-contour images in combination with a linear support vector machine (SVM). [Schindler and Van Gool, 2008] extract textural information using Gabor filters and motion information using optical flow for a given action snippet to train linear one-vs.-all classifiers for a pre-defined set of actions. [Laptev et al., 2008] propose the use of local space-time features using HoG and Histograms of oriented Flow (*HoF*) within a cuboid around extracted Harris interest points, space-time pyramids and multichannel non-linear SVMs for human action recognition in movies. [Matikainen et al., 2009] choose the standard bag-of-words approach to set up a dictionary of trajectory words (trajectons) from a clustered set of trajectory snippets including temporally extracted features. A given sample trajectory is represented by a normalized histogram of accumulated trajecton indices assigned to each snippet. This histogram is used to classify action categories using a trained linear SVM. [Matikainen et al., 2009] show an overall better classification performance compared to the method of [Laptev et al., 2008].

In the research area of intention recognition in comparison to action recognition one handles with the prediction of a future action and not the actual recognition thereof to perform preventive actions. Nevertheless, the step from action to intention recognition is a small one. [Wakim et al., 2004] formulate a Markov chain for the four discrete states "standing still", "walking", "jogging" and "running". Model transitions are influenced by successive pedestrian states within a trajectory including estimates for the pedestrian's velocity magnitude and direction. Initial state and transition probability distributions are statistically determined from various studies on human gait pattern analysis. During simulations, the actual motion state is used to predict the potential impact of a crossing pedestrian and an approaching vehicle. [Furuhashi and Yamada, 2011] extract HoG descriptors for a set of pedestrian postures in a sequences and principal component analysis (*PCA*) to learn a model for crossing pedestrians using a k-means classifier. [Kitani et al., 2012] use physical scene features (e.g., parking cars, buildings,...) within a static environment using semantic scene labeling from noisy visual input in combination with optimal control theory to infer future actions of pedestrians. For their system it is assumed, that people are fully aware of their environment in order to account for preferences to move around certain regions like sidewalks. [Tamura et al., 2012] define three types of pedestrian intentions, namely *free walk*, *avoid* and *follow*, to formulate sub-goals, which are used to enhance the social force model of [Helbing and Molnár, 1995] for pedestrian behavior analysis.

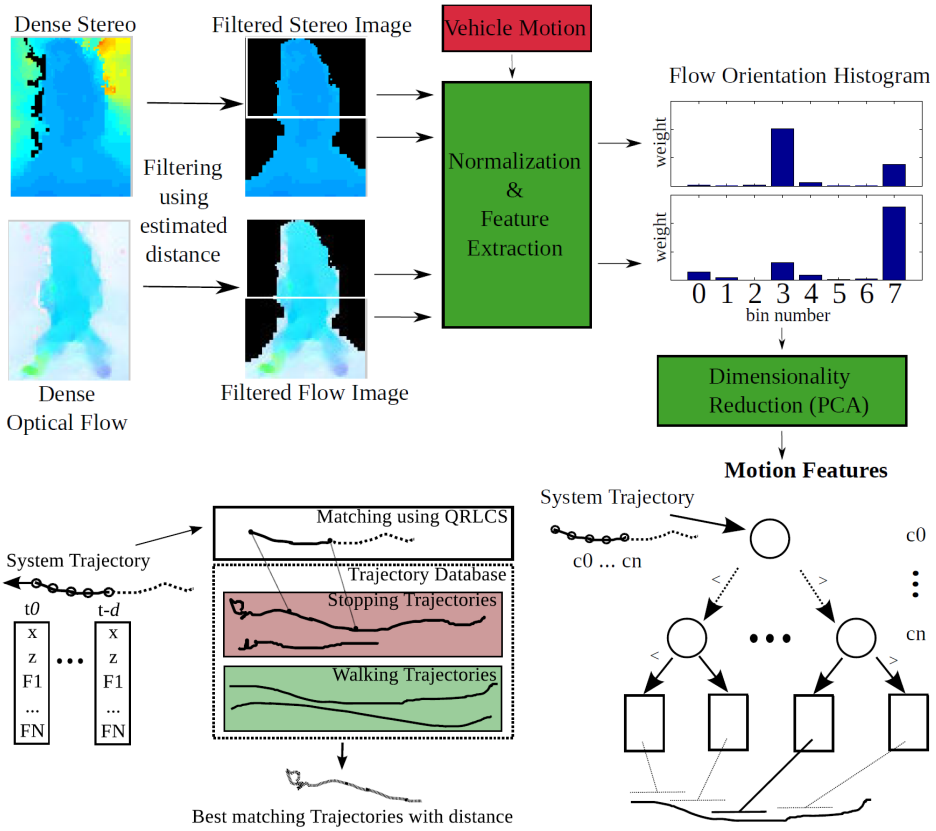


Figure 2.4: Method for pedestrian intention recognition proposed by [Keller et al., 2011c]. For a given pedestrian candidate valid disparity and flow entries are filtered in a first step. A parallax flow field is computed by normalizing flow vectors with disparity and subtracting the influence of ego-motion. The resulting flow vectors contribute to a motion histogram over the upper and lower pedestrian body. PCA achieves dimensionality reduction. Histograms are extracted over time to result in a database of trajectories for stopping and crossing scenarios. For a given sample, the scenario given by the best matching trajectory from the database is chosen, performing an intelligent probabilistic tree search for efficiency.

Recent approaches for pedestrian intention recognition build upon an existing system for video-based pedestrian detection and mainly focus onto the situation of lateral approaching pedestrians. As accident statistics show, this covers the main scenario of accidents involving vehicles and pedestrians, cf. [Marchal et al., 2003]. [Keller and Gavrila, 2014] for example, propose two non-linear, higher order Markov models to estimate whether an approaching pedestrian will cross the street or stop at the curbside. First a Probabilistic Hierarchical Trajectory Matching (*PHTM*) is used to match an actual observation of a pedestrian track with a database of trajectory snippets. Using the information of future locations and pedestrian behavior from the best matching snippets future pedestrian motion and hence intention is extrapolated. In addition, a Gaussian Process Dynamical Model (*GPDM*) which models the dense flow for walking and stopping motions is suggested to predict future flow fields. Both suggested models integrate features that capture pedestrian positions and dynamics by means of dense optical flow. See Figure 2.4 for a basic overview of the PHTM approach. All these

models only try to access features from pedestrian moving dynamics but do not take the underlying scene context into account. [Bonnin et al., 2014] propose a generic context based model to predict crossing behaviors of pedestrians. Multiple models are combined within a *Context Model Tree* capturing general inner-city scenarios, as well as the special context of zebra crossings. Contextual features, e.g. distance/time to curbstone or distance/time to zebra crossing, are extracted with a stereo camera.

A common part of most of these methods is the use of machine learning approaches to classify a series of extracted observations. At this point, the choice of the right method can have significant impact on later performance. [Morency et al., 2007] investigate the use of Latent-dynamic Conditional Random Fields (*LDCRF*) as an extension of conventional Conditional Random Fields (*CRF*, [Lafferty et al., 2001]) for labeled time series of observations by adding a layer of hidden latent states. These hidden state variables can model the intrinsic sub-structure of a specific class label and capture extrinsic dynamics between different classes. Furthermore, LDCRFs proved to outperform typical CRF models, the well-known Hidden Markov Models (*HMM*) and conventional machine learning algorithms like Support Vector Machines (*SVMs*) in the field of gesture recognition. Figure 2.5 shows a simplified version of a LDCRF.

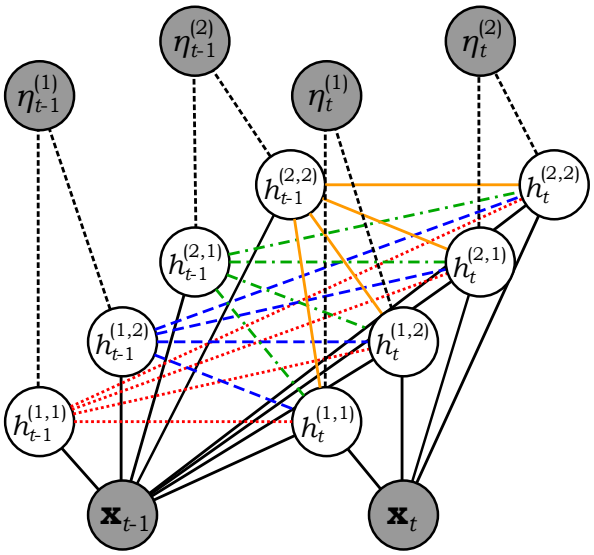


Figure 2.5: LDCRF for a 2-class problem with 2 hidden states per class label. Observables are displayed as shaded nodes. Connections between latent hidden states are colored. The hidden states $\mathbf{h}_j^{(i,1)}$ and $\mathbf{h}_j^{(i,2)}$ model the intrinsic structure for class label $\eta_j^{(i)}$, $i \in \{1, 2\}$, while the connections between $\mathbf{h}_j^{(1,k)}$ and $\mathbf{h}_j^{(2,k)}$, $k \in \{1, 2\}$, model the extrinsic relations between the class labels $\eta_j^{(1)}$ and $\eta_j^{(2)}$.

2.5 Pedestrian Path Prediction Systems

The field of path prediction is closely related to pedestrian behavior or intention recognition. For some works the difference is only the definition of a discrete intention in form of

a goal location depending on the predicted pedestrian path. Therefore, the previously introduced works of [Wakim et al., 2004, Kitani et al., 2012] or [Tamura et al., 2012] are suited for path prediction as well, as internally predicted future pedestrian positions are available. Most of these methods build upon agent models, where an observer tries to get an idea about the agent's intention, it's preferences to move around certain regions or behavior for collision avoidance with other agents or the observer itself. [Bandyopadhyay et al., 2013], for instance, propose a method for motion planning taking into account uncertain intention estimates for an interacting agent (pedestrian). For each of a finite set of potential intentions they construct a motion model based on social force for later combination within a *Mixed Observability Markov Decision Process* framework. The hidden intention thereby is the agent's goal location, which cannot directly be extracted by the observer (robot vehicle). Related to the social force model presented by [Helbing and Molnár, 1995], [Pellegrini et al., 2009] propose a *Linear Trajectory Avoidance* model for short-term path prediction, that uses the expected point of closest approach to foreshadow and minimize potential collisions of agents. [Chen et al., 2008] form trajectories including estimates for pedestrian position, velocity and heading angle to learn *Motion Pattern* clusters for a multi-level prediction model. Within these levels long-, middle- and short-term predictions are carried out depending on the matching quality with a trained Motion Pattern. In driver assistance, most recent contributions on pedestrian intention recognition and the closely related field of pedestrian path prediction build upon an existing system for video-based pedestrian detection, see Section 2.1 for a survey. The focus of these approaches is to address the situation of lateral approaching pedestrians. The earlier introduced method of [Keller and Gavrila, 2014] for example is also used for path prediction. For their PHTM approach (Fig. 2.4), future pedestrian states can be easily extrapolated by forwarding the next steps using the best matching trajectory from the database. In addition, the proposed GPDM models the dense flow in order to predict future flow fields and thereby lateral velocity, which in turn is used for path prediction. [Quintero et al., 2014] follow up the GPDM approach in combination with 3D body language by means of joints and body parts determined using point clouds extracted from a stereo vision sensor. A GPDM is used for dimensionality reduction and learning pedestrian dynamics within a latent space for efficient path prediction over a time horizon of 0.8 seconds. [Goldhammer et al., 2014] focuses on pedestrian path prediction in static traffic scenes. Motion trajectories including velocity estimates using pedestrian head tracking are extracted and approximated by polynomials for better generalization. Path prediction is carried out by training a Multilayer Perceptron (*MLP*) Artificial Neural Network (*ANN*), which is able to predict a continuous trajectory for a time horizon of 2.5 seconds. For path prediction in dynamic scenarios, [Schneider and Gavrila, 2013] analyze the usability of different linear and non-linear dynamical systems involving Kalman filters (KF) and Interacting Multiple Models (*IMM*) to predict future pedestrian positions by propagating pedestrian states for a small time slot of 1.9 seconds. All these models only try to access features from pedestrian moving dynamics but do not take the underlying context into account. Initially, [Kooij et al., 2014a] present a Dynamic Bayesian

Network (*DBN*) on top of Switching Linear Dynamic System (*SLDS*) for path prediction, where they integrate contextual information using latent information from pedestrian awareness, the pedestrian position with respect to the curbside and the criticality of the underlying situation. To account for inattentive pedestrians they extract features from the human head pose inspired by the work of [Hamaoka et al., 2013]. Further, the expected point of closest approach, presented by [Pellegrini et al., 2009], serves as a measure for criticality. At the end an existing system for curb stone detection is used to determine, whether a pedestrian is at the curbside or still too far away to state a risk.

3 Monocular Pedestrian Head Localization and Head Pose Estimation

This chapter describes one essential part of this work for pedestrian intention recognition and path prediction. From related literature ([Schmidt and Färber, 2009, Oxley et al., 1995, Hamaoka et al., 2013]) it is known, that an essential indicator for future behavior of pedestrians in road traffic situations is given by the line of sight or head pose respectively. The human head position and head pose can be captured by a video sensor and analyzed using image processing algorithms. Thematically, one finds oneself in the field of head pose estimation in very low resolution images, which constitutes a very challenging task in modern image processing and analysis [Benfold and Reid, 2009, Robertson and Reid, 2006, Flohr et al., 2014, Orozco et al., 2009]. In this work a so-called *detector-array* approach will be developed, cf. [Murphy-Chutorian and Trivedi, 2009], that initially is working on single images only. Here, a series of head pose detectors each assigned to a specific head pose class is trained with the addition of annotated training data. The benefit of the chosen approach is the possibility to make use of head detectors that are well suited for evaluation on low resolution images due to their integrated features, which can work on lowest pixel level as presented in [Fröba and Ernst, 2004] and [Viola and Jones, 2004].
The later application is based on video sequences. Therefore, the use of a tracking based approach turns out to be well suited in order to stabilize the single-frame based estimates over time. For tracking a particle filter is deployed, which states a good choice in the research area of interest.
The remainder of this chapter is structured as follows. Section 3.1 deals with the approach for human head pose estimation based on single low resolution images. The application on video sequences together with the development of a tracking method follows in Section 3.2. Section 3.3 presents different types of datasets on which the developed approach gets evaluated. Experiments in Section 3.4 show a detailed performance analysis as well as comparisons with state-of-the-art approaches from literature. Finally, a conclusion is given by the end of this chapter in Section 3.5.

3.1 Pedestrian Head Pose Estimation in Single Images

This work proposes an integrated head localization and head pose estimation approach for very low resolution images. The method builds its decisions based on the confidence values provided by head pose associated classifiers. These classifiers follow state-of-the-art object detection approaches, each of which is trained separately under a modified one-vs.-all framework to achieve a high classification rate and ensure a stable training process. Assuming a pre-defined pedestrian hypothesis given in form of a image bounding box, the provided algorithm searches in the upper region followed by a head localization and head pose estimation block. The head pose of interest is set to be one of eight discrete pose classes distributed over the full head pan angle of 360°. Figure 3.1 shows an overview of the proposed system.

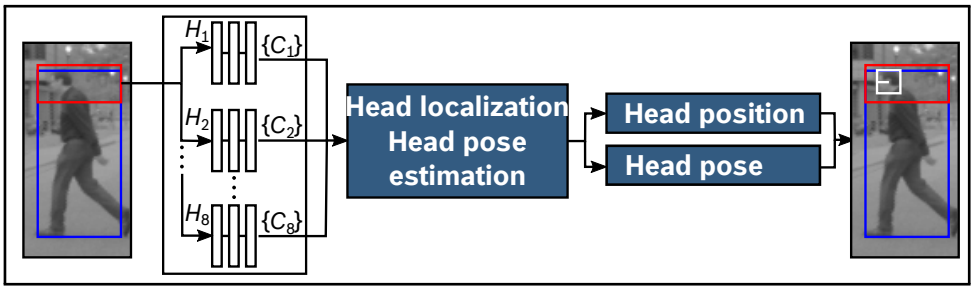


Figure 3.1: Signal processing chain for single frame based approach. Different head pose associated classifiers H_m, $m = 1...8$, evaluate all possible windows in the head search area (red) of a pedestrian detection (blue). The out-coming confidence values $\mathcal{C}_m$ are used to perform the head localization and pose estimation

The structure of this section is as follows. The Sections 3.1.1 to 3.1.3 explain suitable features for head detection and general object classification approaches. A modified one-vs.-all training procedure is applied in Section 3.1.4 to train the eight head pose classifiers. Each classifier consists of a cascade of boosting classifiers and produces a classification confidence, when a test sample is evaluated. Given a pedestrian detection, the head pose classifiers are searching for head correspondences in a specific area at the top of the pedestrian image area using a sliding window technique at different scales, see Section 3.1.5. This results in eight confidence values per search location, which are normalized in a following step in Section 3.1.6 to be comparable among each other. At the end in Section 3.1.7, the head localization and head pose estimation is achieved by comparing the confidence values for all evaluated windows at different scales, assuming the presence of exactly on head per pedestrian hypothesis.

3.1.1 Feature Extraction for Head Detection and Head Pose Estimation

In order to perform accurate and precise classification, discriminative and informative features have to be extracted from the image to represent the same in the training and classification procedure. Among other image features used in literature, e.g. in [Siriteerakul et al., 2010, Orozco et al., 2009] and [Robertson and Reid, 2006], two types of features turned out to have a high discriminative power for the task of face recognition namely Haar-like features presented in [Viola and Jones, 2001a, Viola and Jones, 2001b, Viola et al., 2003] and local structure features using the modified Census Transform (*MCT*) introduced by [Fröba and Ernst, 2004]. Both types of features recently were the center of many object detection studies due to their classification performance at very low computational costs. This section continues with a detailed description of both feature types.

Haar-like Features

As working with image intensities is computationally expensive and not efficient, [Papageorgiou and Poggio, 1999] discussed working with an alternative feature set instead of the conventional image intensities. This feature set considers the difference between the sums of gray values in adjacent rectangular areas. Due to similarities to the Haar Wavelet proposed by [Haar, 1911], those features earned their name. Haar-like features show a very good performance representing images for classification purposes. These features were used for pedestrian detection by [Papageorgiou and Poggio, 1999] and for face detection by [Viola and Jones, 2001a]. Different structures of Haar-like features can be used as seen in Figure 3.2. The feature value is the difference between the pixel intensity value sums in the shaded areas and the clear areas. In [Viola and Jones, 2001a] two, three, and four types of rectangle features are used for object detection. $45°$ tilted features, such as in Figure 3.2b and Figure 3.2d, were later presented by [Lienhart and Maydt, 2002].

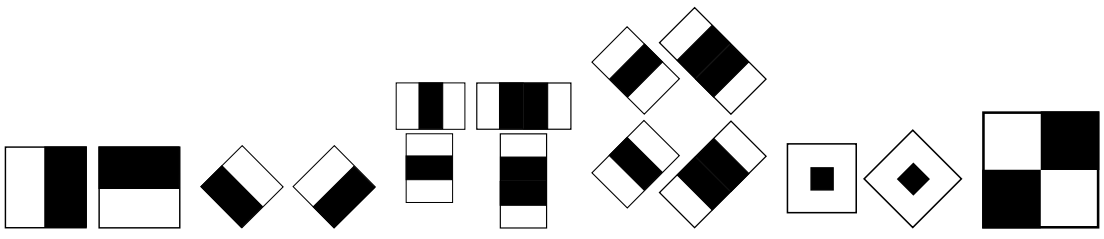
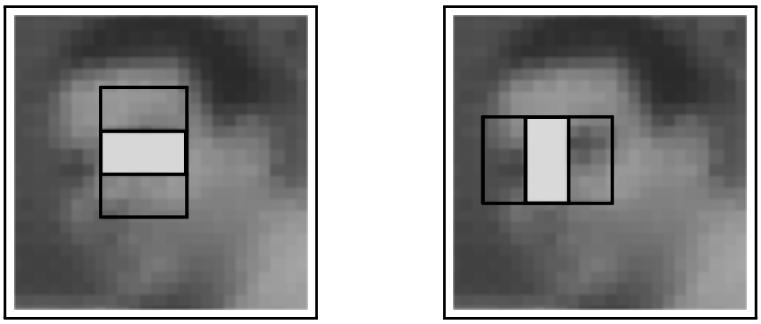


Figure 3.2: Haar-like features: the feature value is the difference between the intensity values sums between the black and white areas. a) edge features. b) Tilted edge features. c) Line features. d) Tilted line features. e) Center-surround features. f) Diagonal feature.



(a) (b)

Figure 3.3: a) Haar-Like Feature representing intensity variation around the eye area. b) Haar-Like Feature representing intensity variation in the nose-eyes area.

The Haar-like features represent certain patterns of intensity variation in certain positions of the image. For example, in face detection some of these features may represent the intensity variation caused by the darker area in the eye region, or the hair line, cf. Figure 3.3.
One of the main advantages of Haar-like features is their computational efficiency. Using the so-called *integral image* presented by [Crow, 1984], the sum of pixel values in a rectangle can be calculated by referring to only four pixel values, independent of the rectangle's position or size. With the original input image I, the integral image $I_\Sigma(u,v)$ at a pixel position (u,v) can be calculated as

$$I_\Sigma(u,v) = \sum_{i \leq u} \sum_{j \leq v} I(i,j) \tag{3.1}$$

This computational efficiency extends to calculating multiple rectangle features. For example, a two rectangle feature, as shown in Figure 3.4, can be calculated using only six references in the integral image. Assuming r_1 and r_2 to be the sum of intensity values in the first and second rectangle, and every corner notation in the figure represents the value of the corner pixel in the corresponding integral image (i.e., the sum of all pixel values above and to the left of it), the Haar-like feature value (HF) can be simply calculated as

$$\text{HF} = r_2 - r_1 = D - C - E + F - (C - B - F + A) = D - 2C - E + 2F + B - A. \tag{3.2}$$

Local Structure Features based on the Modified Census Transform

Searching for an image transformation that represents image structures independently from illumination variations, [Fröba and Ernst, 2004] propose the modified Census Transform (MCT) to create illumination invariant local structure features for face detection tasks. Here the MCT feature properties are presented. The features represent 3×3 local kernels describing certain structures just as oriented edges, lines or points. The structures are binary coded so that every structural pattern is represented by a separate binary string, see Figure 3.5.

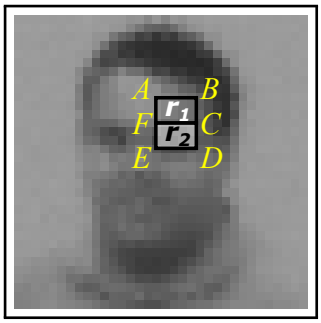


Figure 3.4: Two rectangle Haar-like feature calculation. Here the corner notations A, B, C, D, E and F represent the corner pixel values in the corresponding integral image. r_1 and r_2 represents the sum of all pixel values within the two rectangles. The feature value is the difference between r_1 and r_2, as in equation (3.2).

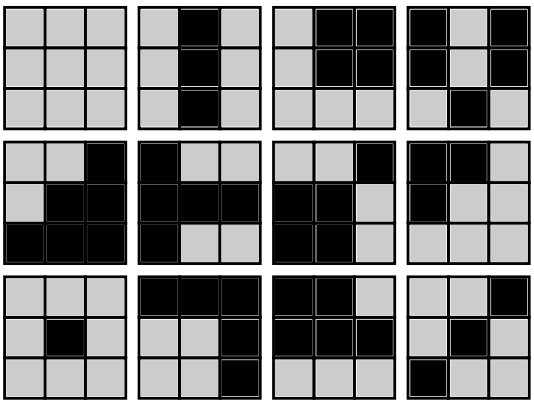
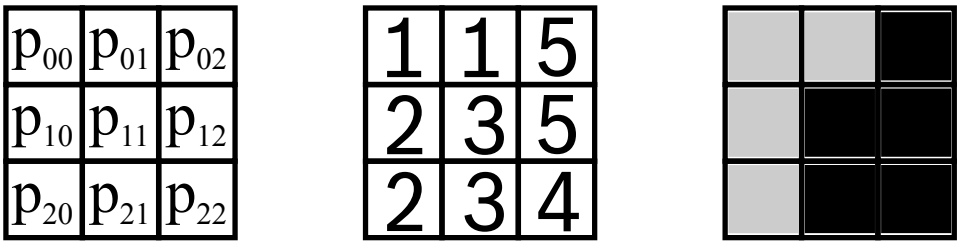


Figure 3.5: Examples of 3×3 local structures: binary coded 3×3 local structure leads to 2^9 possible structures that can represent lines, edges, points, etc.

In total, 2^9 different binary structures can be generated.[1] For face detection at each position of an image including a visible head, the most fitting structure has to be found. For this task, the modified Census Transform is used. To find the most fitting structure at a certain position within an image, [Zabih and Woodfill, 1994] used the conventional Census Transform. However, this only describes 2^8 = 256 of the 511 structures. [Fröba and Ernst, 2004] present the modified Census Transform in an effort to represent all 511 possible structures. To calculate the modified Census Transform let the position (pixel) of interest be $\mathbf{p}_c$ and $N(\mathbf{p}_c)$ a local spatial neighborhood around $\mathbf{p}_c$, that includes $\mathbf{p}_c$. $I(\mathbf{p})$ is the intensity of pixel $\mathbf{p}$, and the mean intensity value of the neighborhood $N(\mathbf{p}_c)$ is noted as $\hat{I}(\mathbf{p}_c)$. ζ is a comparison function such that

$$\zeta(a,b) = \begin{cases} 1, \text{ if } a < b,\ a,b \in \mathbb{N}_0 \\ 0, \text{ else} \end{cases} . \tag{3.3}$$

[1] To be more accurate: As information obtained by the structure with all 0 elements and the structure with all elements equal to 1 is redundant, the total number of structures is 2^9-1 = 511.



(a) (b) (c)

Figure 3.6: Modified Census Transform: (a) 3×3 structure with the central Pixel $\mathbf{p}_c = \mathbf{p}_{11}$. (b) The intensity values in the neighborhood $N(\mathbf{p}_c)$. The mean intensity value can be calculated to $\hat{I}(\mathbf{p}_c) = 2.89$. (c) The resulting MCT structure by applying equation 3.4

.

Figure 3.7: Input images and their corresponding transformed images using MCT. The calculated feature indices are mapped to gray values. The illumination changes between the two input images (left side) show no impact on the MCT result (right side).

Let $\bigotimes$ be the concatenation operation, the modified Census Transform at $\mathbf{p}_c$ can be written as

$$\Gamma(\mathbf{p}_c) := \bigotimes_{\mathbf{q} \in N(\mathbf{p}_c)} \zeta(\hat{I}(\mathbf{p}_c), I(\mathbf{q})) \tag{3.4}$$

In other words, each pixel's intensity in a neighborhood is compared to the mean intensity value of that neighborhood. In the transformed neighborhood, the value of this pixel is set to 1, if it is higher than the mean intensity value or to zero otherwise. Figure 3.6 shows the MCT structure calculation for an exemplary 3×3 image patch. The result is an index or bit-array representing one structure out of the 511 possible structures that this position belongs to. [Fröba and Ernst, 2004] report the illumination invariance of the modified Census Transform which is directly inherited to the extracted local structure features. Figure 3.7 shows this effect on an exemplary face image. To transform the result indices into a more mean-

ingful statistical form for learning algorithms, a look-up table is created for each position (pixel) of a given image patch. These look-up tables contain 511 slots, each representing one of the possible structures. The value in each slot represents the number of times the associated structure occurred while evaluating the training data including a various amount of face images. Therefore, in the face detector training procedure a weight for each slot in the look-up table at each pixel position is assigned. Based on these weights, every pixel in a test image will be assigned to a certain class. For the detailed training procedure please refer to [Fröba and Ernst, 2004]. It must be mentioned that around the same time the MCT algorithm was proposed, a similar approach was presented by [Jin et al., 2004] called *Improved Local Binary Pattern* (ILBP) with only minor differences compared to MCT, but as MCT, represents neighborhood structures around a pixel position.
The next section describes the integration of the above introduced features into machine learning algorithms that are able to separate regions including human heads from regions belonging to background.

3.1.2 Machine Learning Algorithms for Head Pose Detectors

Machine learning algorithms try to give machines the ability to make decisions based on experiences. These algorithms can be categorized into supervised and unsupervised techniques. Unsupervised algorithms try to find the way data is organized into classes, and is trained given only unlabeled data. Supervised learning tries to build a function that classifies unknown data based on the knowledge gain from labeled training data. In this work supervised learning is used. Learning algorithms can be biologically inspired algorithms. Here, one of the most famous algorithms are the Neural Networks [Vapnik, 1995]. Neural Networks try to model the decision making within the human neural system. Other learning algorithms are built on statistical basis such as Support Vector Machines [Vapnik, 1995] and Boosting [Freund and Schapire, 1997]. These algorithms not only try to reach perfect classification for the training data as in Neural Networks, but also try to minimize the generalization error leading to better performance for unknown data. The Boosting classifier in particular is the main learning algorithm used in this work, as it is receiving a big amount of attention in computer vision research and proved to be one of the most promising learning algorithms. One reason for that is its capability to successively select the most powerful features out of a large pool of relevant features during training. This section will start with a generalized notation of the learning problem. Then, the following topics are presented: Decision Trees, a basic statistical learning algorithm, Boosting algorithms, the basic learning algorithm used in this work, Cascade Classifiers, a bundle of classifiers leading to higher efficiency and accuracy, and Winnow Classification, a learning technique for high dimensional training data.

Notations for Binary Classification Problems

Before presenting the different learning algorithms used in this work, a common set of notations is presented. For binary classification problems the training set $\mathfrak{T}$ contains m pairs of the form

$$\{\mathbf{x}_i, y_i\}, i = 1, \ldots, m, \tag{3.5}$$

where $\mathbf{x}_i$ is a real vector of dimension d,

$$\mathbf{x}_i := \left(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(d)}\right)' \in \mathbb{R}^d, \tag{3.6}$$

and $y_i \in \{-1, 1\}$ the corresponding class assignment, where 1 and -1 represent the positive and negative class label, respectively. The binary classification function generated in a supervised learning algorithm by the training set $\mathfrak{T}$ is denoted by f, and for each input vector $\mathbf{x} \in \mathbb{R}^d$ assigns a class label $y = f(\mathbf{x}) \in \{-1, 1\}$, i.e.,

$$f : \mathbf{x} \mapsto f(\mathbf{x}), \mathbb{R}^d \to \{-1, 1\}. \tag{3.7}$$

The notations presented here are occasionally repeated, in order to provide clear and fast understanding of the algorithms being discussed.

Decision Trees

Decision trees are simple classifiers that predict the corresponding class of a given data sample. They analyze a given labeled training data set to be able, after training, to predict an unknown data sample's class. Decision Trees were introduced by [Breiman et al., 1984] as the Classification and Regression Tree (*CART*) method. This algorithm produces a decision tree based on a training set $\mathfrak{T}$, which is aiming to recursively separate the training set into subsets of higher purity, whose elements ideally belong to only one class. [Breiman et al., 1984] present a detailed explanation of the decision tree training procedure. Given an unknown sample $\mathbf{x} \in \mathbb{R}^d$, the decision tree can make a classification decision based on the tree leaf that the given sample reaches after a series of threshold comparisons in the nodes leading to that leaf. The run time complexity of a decision tree with a maximum depth of $M \leq d$ to classify an input sample $\mathbf{x} \in \mathbb{R}^d$ equals to $\mathcal{O}(M)$, where $\mathcal{O}$ is the Landau notation that represents the upper limit of a function complexity. The decision trees have high computational efficiency and can be implemented easily. Therefore, they are a common candidate to be the basic structure in more complex classifiers such as Random Trees or Boosting.

Boosting

Boosting is a supervised machine learning algorithm based on the idea of creating a strong classifier from a set of weak classifiers. This idea was discussed by [Kearns, 1988], where a weak classifier is a classifier that produces results slightly better than pure guessing, and

a strong classifier is a classifier that produces results of arbitrary accuracy. Weak classifiers can take many forms. Most used weak classifiers are decision stumps (feature value thresholding) and decision trees. In general, boosting algorithms consist of a weighted sum of iteratively learned weak classifiers. When a new weak classifier is added, the training data is reweighed such that misclassified samples gain weight while correctly classified samples lose weight. This way, the next weak classifier will concentrate more on the misclassified samples. Different Boosting algorithms vary mainly in the way, on how the training data and weak classifiers are weighted. The most popular boosting algorithm is Adaboost, formulated by [Freund and Schapire, 1997] and [Freund and Schapire, 1999].

Discrete Adaboost The Discrete Adaboost algorithm proposed by [Freund and Schapire, 1997] and [Freund and Schapire, 1999] has been the center of many studies. The input to the algorithm consists of a set $\mathfrak{T}$ of training samples (descriptive vectors) associated with their respective labels $\{\mathbf{x}_1, y_1\}, \ldots, \{\mathbf{x}_m, y_m\}$. Based on these vector-label combinations, a classification function $f(\mathbf{x})$ is trained in order to return a class label for a test sample $\mathbf{x}$. During training, all training samples are equally weighted before the first iteration. In each iteration t a weak classifier h_t is trained with output domain $\{-1, 1\}$. Based on the training error value ϵ, the training samples are re-weighted. With higher weights for misclassified samples, the classifier in the next iteration will concentrate more on these misclassified samples. The weak classifiers are also weighted based on their accuracy. The final output $f(\mathbf{x})$ is simply the sign of the weighted weak classifiers' sum.

Real Adaboost As the weak classifiers in discrete Adaboost can only return a binary decision, this drives the weak classifiers to make strict decisions even in ambiguous situations and thus reduce the accuracy of the decision confidence. The *Real Adaboost* as described in [Friedman et al., 2000] deals with this problem by providing a non-binary weak classifier decision. During training, in each iteration t, a class probability estimate p_t is set based on the weighted training data. This estimate represents the probability of the sample belonging to the positive class and is used to calculate the weak classifier function h_t. The training sample weights are updated according to the weak classifiers' decisions as they provide a sense of confidence. The final classifier $f(\mathbf{x})$ is the sign of the weak classifiers' sum. Here, the weak classifiers are not weighted as they provide a confidence measure (weight) embedded in their output.

Gentle Adaboost In practice, the number of samples available for training is limited and many data outliers exist in the training data. In these cases, the Real Adaboost algorithm may lead to very high outputs of weak classifiers, which will cause training problems because of its effect on the weight updates, as well as the final hypotheses. Gentle Adaboost avoids the problem of high-valued weak classifiers output by limiting the domain of their output to the interval $[-1, 1]$. The Gentle Adaboost algorithm is introduced in [Friedman et al., 2000].

Instead of fitting a probability estimate on the training data and from this estimate calculate the weak classifier h_t as in Real Adaboost, the weak classifier in Gentle Adaboost is directly calculated by fitting itself with weighted least-squares to the training data. The weak classifier can be seen as $h_t(\mathbf{x}) = P_w(y = 1|\mathbf{x}) - P_w(y = -1|\mathbf{x})$ as described in [Friedman et al., 2000]. This will limit the weak classifier output to $[-1, 1]$.
In summary, different methods of Adaboost classification can be used with different types of weak classifiers and different types of features. They are also simple to implement and less prone to over-fitting than most of other boosting classifiers. Classifiers based on Adaboost are used in various areas such as pedestrian detection in combination with Histogram of Oriented Gradients (HoG) features by [Dalal and Triggs, 2005] and by [Viola and Jones, 2001a] in combination with Haar-like features for face detection. Discrete Adaboost provides a less accurate confidence measure for the classification results. In Real Adaboost, this classification confidence can be calculated more accurately, but the algorithm faces problems dealing with real training data in practice (limited data set and outliers). Gentle Adaboost tolerates the imperfections in the training data and, as Real Adaboost, provides an accurate classification confidence measure.

Cascade Classifier

Using a single classifier for object classification may lead to unoptimized solutions. For once, the classifier may achieve a high correct classification rate but usually combined with a high error rate as well. This results from the limited amount of training samples (especially negative ones) included in the training procedure due to learning machine limitations. Also, using a single classifier is not an optimized solution in terms of computational efficiency and therefore not suitable for real-time applications. In a single classifier case, all the samples to be classified, whether simple or complicated, will require the same high computational effort. A technique known as Cascade Classifier aims to achieve a high detection rate with relatively small error combined with a significant improvement of the computational efficiency. This section presents the Cascade Classifier structure, its functionality, training and feature selection technique. The algorithm presented here is based on Boosting classifiers in a similar manner to [Viola and Jones, 2001a], although, different types of base classifiers can be used as stages in a cascaded structure.

Cascade functionality The Cascade Classifier, as used by [Viola and Jones, 2001a, Viola and Jones, 2001b] and [Viola and Jones, 2004] in combination with Haar-like features consists of stages represented by boosting classifiers in a cascaded arrangement. As the possibilities of negative samples are much higher and diverse in general, the first stages of the cascade must reject as many negative samples as possible, while keeping a high hit rate. To achieve a fast performing classifier, the number of features evaluated in each stage should be as small as possible. Usually, the majority of the negative samples can be rejected using a

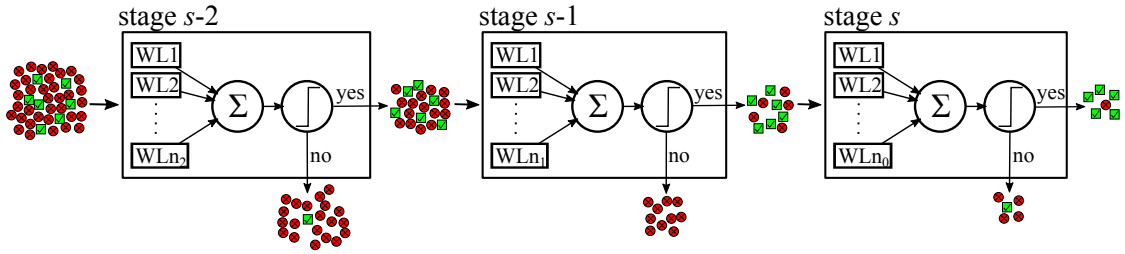


Figure 3.8: Cascade of Boosting Classifiers structure. The green squares and red circles represents positive and negative samples to be classified. The first stages reject most of the negative samples. These rejected samples will not be further processed by subsequent stages. Ideally, the initial stages will consider a smaller number of features, therefore, will be less complicated. The number of weak classifiers in each stage n_2, n_1 and n_0 is related to the number of considered features and the computational complexity of the stage. Usually in this case $n_2 < n_1 < n_0$.

small number of features, this means, they will be rejected in the first stages without a large computational effort. A small number of more complicated negative samples will advance in the cascade to more complex stages that use a larger number of features and thus are more computationally demanding, see Figure 3.8. In more details, each stage classifier predicts the sample class (positive or negative). If the sample is predicted as negative it will not be further processed by the next stages and will not consume additional computational effort. Only the samples that pass a certain stage will be further classified by the next one. Therefore, the number of processed samples will get smaller and smaller as they advance in the cascade stages, and only the sample that passes all the stages will be classified as a positive.

Cascaded feature selection Using Haar-like features and local structure features based on MCT (Sec. 3.1.1) results in a big number of features most of which are irrelevant for recognizing particular objects. In a cascade classification structure, the minimum number of features must be chosen to achieve the required performance in each stage. To achieve that, a feature selection technique must be used to determine the most informative and distinctive features for each stage. In the cascade of boosting classifiers presented in [Viola and Jones, 2001a], feature selection in each stage is integrated within the boosting algorithm. [Viola and Jones, 2001a] restrict the boosting classifiers to use only one feature per weak classifier. Therefore, measuring the error of all the weak classifiers under the current sample weights provides a quality measure for all single features. The feature that produces the weak classifier with the lowest error is chosen to be the next weak classifier added to the final classification function.

Cascade Classifier Training As mentioned before, the cascade classifier consists of stages of simpler classifiers. Here, the stage training procedure of the cascade of boosting classifiers is presented. Then, the relation between the subsequent stages during training is discussed.

Stage Training: For simplicity, the boosted cascade stage training algorithm for Discrete Adaboost is described here as used in [Viola and Jones, 2001a]. Knowing that Real Adaboost or Gentle Adaboost algorithms differ from the Discrete Adaboost only by the weak classifier formulation and the training sample weight update rule, it is clear that these boosting techniques can be easily placed within the described approach. In [Viola and Jones, 2001a], the approach starts by initializing the weights for the training samples so that the positive sample weights and the negative sample weights are equal per class and sum up to one. From all the one-feature based possible weak classifiers and in each iteration, the algorithm selects the weak classifier with the lowest training error. After the weak classifier selection step, the training samples are re-weighted, so that the correctly classified ones get lower weights while the wrongly classified ones get higher weights. Therefore, the weak classifier (feature) selected in the next iteration will concentrate on the miss-classified samples.

Cascade Training: Cascade classifier training is a tradeoff procedure between computational efficiency and classification performance. The tradeoff is made between the number of cascade stages, the number of features in each stage, and the stage training performance thresholds. In practice, this optimization problem is very hard to be solved. For a cascade classifier, if a detection rate, a false positive rate, and the number of cascade stages were decided, knowing that each stage in the cascade reduces the detection rate and the false positive rate, the stage minimum detection rate and maximum false positive rate can be calculated. If D is the cascade classifier detection rate and F its false positive rate, d_s the minimum stage detection rate, f_s the maximum false positive rate for each stage and S the number of cascade stages then

$$D = \prod_{s=1}^{S} d_s, \; F = \prod_{s=1}^{S} f_s. \tag{3.8}$$

It must be mentioned that in order to train a cascade classifier, two sets of data are required. One is called the training set and is used for the separate training of each stage. The other set is called the validation set and is used to measure the performance of each stage during training, as well as the whole cascade performance. The use of the validation set will be clear in the following paragraphs. Before starting the cascade classifier training, some parameters must be set. These parameters are the maximum false positive rate per stage f_s, the minimum detection rate per stage d_s, the cascade minimum detection rate D, the cascade maximum false positive rate F and the maximum number of stages S. In each stage and after every iteration (adding a new weak classifier), the performance of the stage is measured by classifying on the validation set. If the stage training conditions of f_s and d_s are reached, the stage training is terminated. Otherwise, a new weak classifier (feature) is added to the stage until it meets the minimum detection rate d_s and maximum false positive rate f_s. After the training of each stage, the overall performance of the cascade classifier is measured over the validation data set. If the overall minimum detection rate D and the maximum false pos-

itive rate F conditions are met, the cascade training is terminated. Otherwise, a new stage is added to the cascade until the maximum number of stages S is reached. Instead of assigning a maximum false positive rate f_s and minimum detection rate d_s for each stage, some works assign a fixed number of features or weak classifiers for each stage. This is done for example in the work of [Fröba and Ernst, 2004] for face detection.

Classification Confidence: In order to get information about the classification decision certainty, a confidence measure of the classifier decisions must be acquired. Calculating the confidence of a single classifier is relatively simple compared to the case where the decision of a cascaded classifier is considered. As the cascade of boosting classifiers is the main type of classifiers used in this work, the confidence measurements are going to be presented for this classifier's framework. None the less, dealing with a cascade of a different type of base classifiers can be easily adapted to the confidence calculations presented here. For a single boosting classifier, whether it is a standalone classifier or a stage in a cascade classifier, the final hypothesis is a weighted sum of the weak classifiers' results. A weak classifier result represents a measure of its decision confidence. The classification confidence of a single boosting classifier is the difference between the weighted sum of the weak classifiers and the decision threshold. Assuming a Gentle Adaboost algorithm with a final hypothesis, that is

$$H(\mathbf{x}) = \operatorname{sgn}\left(\sum_{t=1}^{T} h_t(\mathbf{x})\right). \tag{3.9}$$

Then, the confidence $c(\mathbf{x})$ of this classifier decision can be calculated as

$$c(\mathbf{x}) = \left(\sum_{t=1}^{T} h_t(\mathbf{x})\right) - \alpha, \tag{3.10}$$

with a decision threshold α. For cascade classifiers, different techniques are used to calculate the classification confidence. In the work of [Horton et al., 2007] the cascade confidence is calculated by summing up the confidences of the separate stages one by one while stepping through the cascade and stop when a stage returns a negative confidence. Another approach would be to give each stage a confidence weight, incrementing while stepping towards the last stage. These weights will be summed up for all successfully passed stages, plus the confidence of the last passed stage as calculated in equation (3.10). In this work, the classification confidence is calculated by summing up the confidence values of the separate stages until the last stage. In case the last stage is not reached or passed successfully, the confidence value is set to zero. Let $\mathcal{C}_c(\mathbf{x})$ be the cascade classifier decision confidence for a classified sample $\mathbf{x}$, $c_s(\mathbf{x})$ the s-th stage classification confidence of the sample $\mathbf{x}$ and S the total number of cascade stages. Then the cascade classification confidence is calculated as

$$\mathcal{C}_c(\mathbf{x}) = \begin{cases} \sum_{s=1}^{S} c_s(\mathbf{x}), & \forall s \in \{1, 2, \ldots, S\} : c_s(\mathbf{x}) > 0 \\ 0, & \exists s \in \{1, 2, \ldots, S\} : c_s(\mathbf{x}) \leq 0 \end{cases}. \tag{3.11}$$

This approach for cascade classifier confidence calculation assures, that only samples passing all the cascade stages are considered as potential object candidates. Based on different classification confidence values, the head localization and head pose estimation decisions are made later on.

Winnow Classifier

Similar to other machine learning algorithms, the Winnow classifier creates a decision hyperplane separating different classes given labeled training data. The main advantage of this algorithm is, that it performs well in high dimensional spaces, where many dimensions are irrelevant. Winnow gains its properties from using a multiplicative weight update rule. The Winnow algorithm presented here is similar to the one presented in [Littlestone, 1988] and used by [Fröba and Ernst, 2004]. It is assumed that each sample $\mathbf{x}_i \in \mathbb{R}^d_{\geq 0}$ belongs to a class y_i, $y_i \in \{-1, 1\}$ and $W = \{w_1, \ldots, w_d\}$ is a weighting set for the descriptor vector elements (features). All weights in W are initially set to 1. A threshold θ is introduced such that the learned winnow classifier decision is

$$f(\mathbf{x}) = \begin{cases} 1, & \text{if } \sum_{k=1}^{d} w_k x_i^{(k)} > \theta \\ -1, & \text{if } \sum_{k=1}^{d} w_k x_i^{(k)} \leq \theta \end{cases}. \tag{3.12}$$

This classification model goes under an iterative weights update. In each Iteration, the weights are changed only if the classifier result was wrong. Even then, only the weights associated with non-zero features are updated. Assuming the weight update rates to be α and β, where $\alpha > 1$ and $0 < \beta < 1$. In each iteration, a new training sample $\mathbf{x}_i$ is classified and depending on the result, the weights are updated for all $k = 1, \ldots, d$ in the following way.
Case 0: Classifier is correct: no weight update
Case 1: Classifier prediction is 1 (positive) but correct response is -1 (negative):

$$w_k = \begin{cases} \beta w_k, & \text{if } x_i^{(k)} > 0 \\ w_k, & \text{if } x_i^{(k)} = 0 \end{cases} \tag{3.13}$$

Case 2: Classifier prediction is -1 (negative) but correct response is 1 (positive):

$$w_k = \begin{cases} \alpha w_k, & \text{if } x_i^{(k)} > 0 \\ w_k, & \text{if } x_i^{(k)} = 0 \end{cases} \tag{3.14}$$

3.1.3 Multi-class Classification

Head pose estimation as discussed here is a multi-class classification problem that aims to map continuous head poses to a number of discrete head pose classes. Multi-class classification can be built on a direct method just like the nearest neighbor approach [Runkler,

2012], where an unknown sample is given the class label of the nearest training sample in the feature space, a space where each sample is represented by a point depending on the feature's value. Further direct methods are for example decision trees, random forests or neural networks [Vapnik, 1995]. Other algorithms are based on a number of binary classifiers such as the *one-vs.-all* classifier or the *one-vs.-one* multi-class classifier, which in literature also appears as *all-vs.-all* classifier. A one-vs.-all classifier is a classification framework built to separate N classes, and hence consists of N binary classifiers. Each one is trained to separate samples of one class from all the other classes' samples. [Rifkin and Klautau, 2004] proved that the one-vs.-all classifier is as accurate as the more sophisticated and computational intense methods of [Weston et al., 1998] and [Crammer and Singer, 2002], who consider a direct approach by training one single classifier. The idea of the one-vs.-one classifier is to train multiple classifiers separating one from another of N classes. In total this results in $N(N-1)/2$ trained classifiers. Given a test sample, a majority vote between all classifier class outputs provides the final class decision. [Debnath et al., 2004] present an efficient one-vs.-one SVM without the necessity to train all $N(N-1)/2$ hyperplanes.
The method used in this work for multi-class classification (head pose estimation) is similar to the one-vs.-all classifier as it is based on binary classifiers, but with some differences. First, as this system tries to combine head localization and head pose estimation, the classifier needs not only to decide on the head pose class but also if a current area of interest in the image includes a person head or background. On the other hand, the head movement is continuous and so is the pose of the head. Therefore discrete decisions for discrete poses are hard to make and do not provide full information about the actual head pose. A modified one-vs.-all classification framework is described in details in Section 3.1.4. The structure of the multi-class decision making and derivation of a head pose class probability used in this work are discussed in details in Section 3.1.7.

3.1.4 Head Pose Classifier Training Procedure

In this work the continuous full head pan angle range is mapped into a discrete set of eight head pose classes. Therefore, discretization problems generating the training data may appear in separating border elements of neighboring classes. In other words, one is facing the problem of a training set with fuzzy borders between the different classes, see Figure 3.9. A conventional one-vs.-all training (Sec. 3.1.3), where a classifier for a certain class is trained against all the other classes, may lead to an unstable training process, especially when using outlier-sensitive boosting algorithms. To prevent this problem a modified one-vs.-all training procedure is proposed, where a special head pose classifier is trained against all the other poses, except of its direct neighboring poses. Experiments show, that this yields more stable classifiers and higher classification rates. In order to separate image parts including pedestrian heads from background, areas around the head and background images are additionally included into the negative set. Figure 3.10 shows some examples for real-world training data.

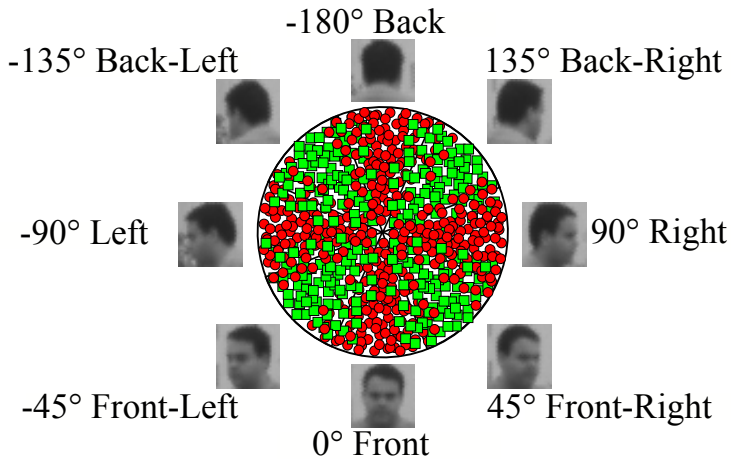


Figure 3.9: Distribution of the manually labeled head samples with fuzzy borders between different head pose classes

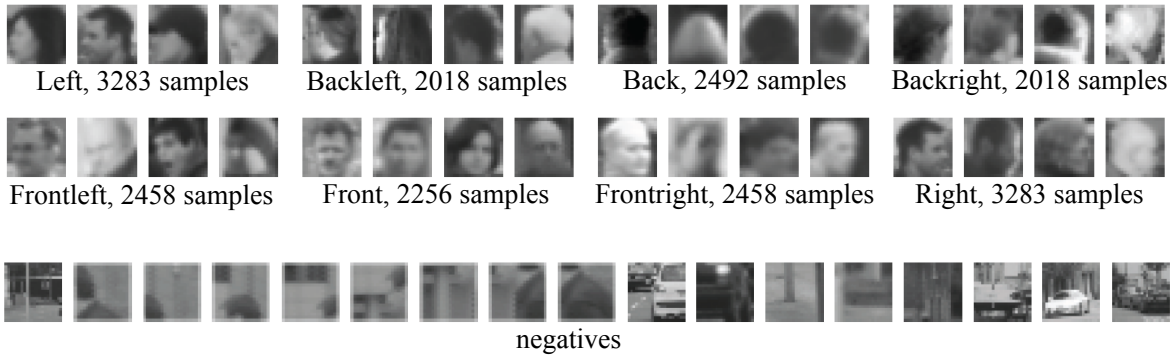
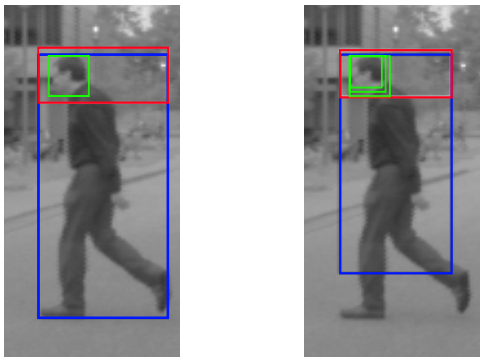


Figure 3.10: Examples of the training data for eight pedestrian head pose classes and negative samples

The number of samples in each pose associated training set differs due to the distribution over pedestrian appearances in driver assistance scenarios. For each discrete head pose class two types of classifiers are trained namely boosting cascades integrating Haar-like features and local structure features using the MCT respectively, see Section 3.1.1 and Section 3.1.2. This results in eight head pose associated classifiers per feature type. The detailed setup for the trained classifiers will be explained in later experiments in Section 3.4.

3.1.5 Selection of Head Search Domain

Within the overall system (Fig. 3.1), detected pedestrian candidates are the input for the head localization and head pose estimation system developed here. The proposed method will search for the pedestrian head in the upper part of a given pedestrian hypothesis by scanning the same using the eight trained head pose classifiers mentioned in Section 3.1.4. On average, the head represents approximately 1/8 of the total human height, [Bogin and Varela-Silva, 2010]. If a perfectly aligned pedestrian detection is assumed, the mentioned value would be fine. In practice however, pedestrian detectors sometimes will output a pedestrian detection box that is not perfectly aligned with the pedestrian appearance in the image. This problem



(a) (b)

Figure 3.11: Pedestrian detection (blue), head search domain (red) and head detector search windows (green). The ideal height for the head search domain of 1/8 of the pedestrian's height (a), does not hold for imperfect aligned pedestrian detection hypothesizes (b). This has to be considered when defining the head search area.

is handled by enlarging the head search region. Resulting from previous experiments, a head search domain is considered having the same width as the pedestrian detection box but choosing a height which is 20% of the pedestrian box's height. This area is scanned at multiple scales and image positions to detect heads at different sizes and locations. For this, an image pyramid is calculated including the scaled head search areas with scaling factors varying from 0.7 to 1.0. An example is given in Figure 3.11.

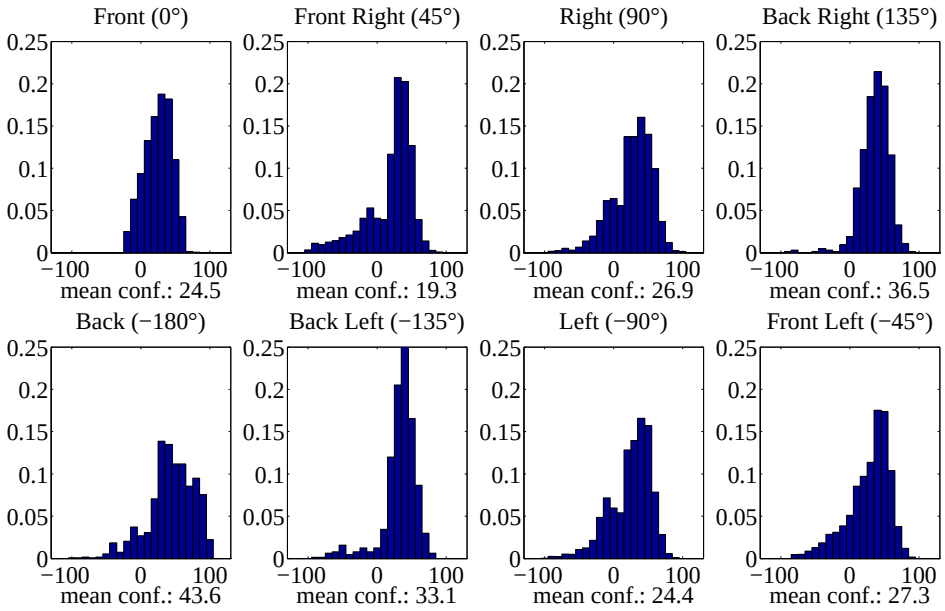
3.1.6 Calculation and Normalization of Confidence Values

Classifying a certain region of an input image using a sliding window technique results in eight confidence values per search position (Sec. 3.1.2). These values are used later in a comparative manner to determine the localization and the pose of the pedestrian's head. Since the eight pose classifiers are trained independently, the different confidence values are also independent, thus will have different ranges. To bring the different outcomes to a comparable level, confidence value normalization is performed. For this, each head pose classifier first is evaluated on its training data to compute the mean output confidence value $\bar{\mathcal{C}}_c$. To normalize the different classifiers' confidence value ranges, the mean confidence values $\bar{\mathcal{C}}_c$, $c = 1, \ldots, 8$, need to be equalized. This can be achieved by multiplying these values with normalization coefficients α_c such that $\alpha_c \bar{\mathcal{C}}_c = \alpha_\zeta \bar{\mathcal{C}}_\zeta$, $\forall \zeta \neq c$. Normalization with respect to the median of all mean confidence values results in

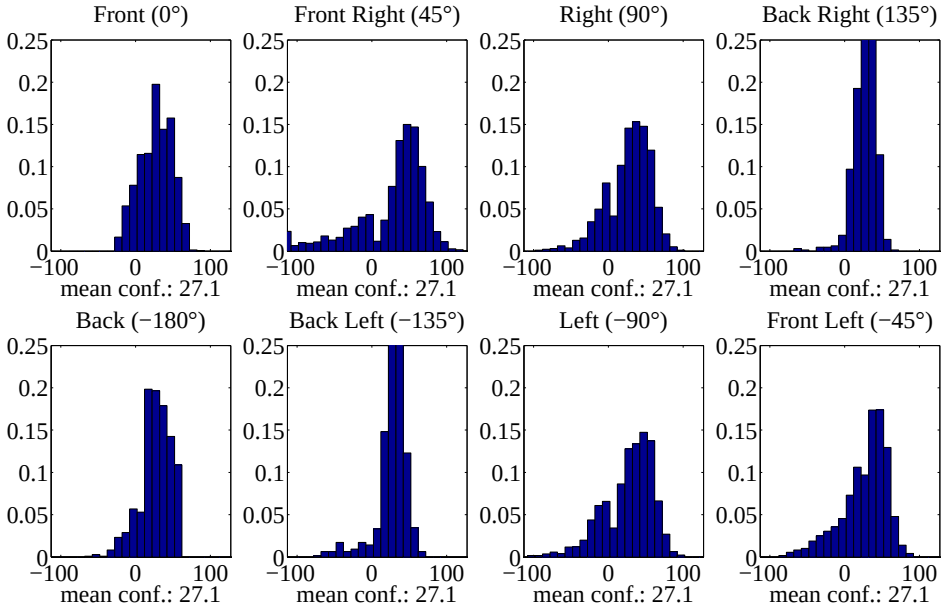
$$\alpha_c = \underset{\zeta=1,\ldots,8}{\text{median}}\{\bar{\mathcal{C}}_\zeta\}/\bar{\mathcal{C}}_c,\ c = 1, \ldots, 8. \tag{3.15}$$

The outputs $\mathcal{C}_c(\mathbf{p})$ of the different classifiers at a special image position $\mathbf{p} = (u, v)'$ are then normalized by the corresponding coefficient α_c, i.e.,

$$\mathcal{C}_{c,\text{norm}}(\mathbf{p}) = \alpha_c \mathcal{C}_c(\mathbf{p}). \tag{3.16}$$



(a) original confidence values



(b) normalized confidence values

Figure 3.12: Distribution of confidence values for considered head pose classifiers over training data. (a) Original values with different ranges of values. (b) Normalized values with a common range of values.

Figure 3.12 shows the initial distribution of confidence values over the training set for different head pose detectors and their corresponding distributions after normalization. While the ranges of values differ significantly for different classifiers, the normalized confidence values lie in the same range. In the following normalized confidence values are assumed ignoring the additional label.

3.1.7 Head Localization and Head Pose Estimation

The search for the head within a given pedestrian hypothesis is limited to the defined head search domain introduced in Section 3.1.5. This area will be scanned by a sliding window at multiple scales, where each image position $\mathbf{p}_h$ is evaluated by all different head pose classifiers. As a result one gets eight normalized confidence values per search position. A Bayesian decision is considered to assign the sample $\mathbf{x}$ the position $\hat{\mathbf{p}}_h$ and head pose class $\hat{\theta}$ with the highest a posteriori probability, that is

$$(\hat{\mathbf{p}}_h, \hat{\theta}) = \underset{\mathbf{p}_h, j \in \{1,\ldots,8\}}{\arg\max} P(\mathbf{p}_h, \theta_j | \mathbf{x}), \tag{3.17}$$

with

$$P(\mathbf{p}_h, \theta_j | \mathbf{x}) \approx \frac{\mathcal{C}_j(\mathbf{p}_h)}{\sum_{k=1}^{8} \mathcal{C}_k(\mathbf{p}_h)}, \; \mathbf{p}_h \subset \mathbf{x}, \tag{3.18}$$

where $\mathcal{C}_j$, $j = 1, \ldots, 8$ represent normalized classifier confidence values (Sec. 3.1.6) and $\theta_j \in \{-180, -135, \ldots, 180\}$ the j-th head pose class's associated head pan angle. Therefore, the head localization and head pose estimation is performed by simply taking that particular head position and head pose class, which scores the maximum confidence over all head pose classifiers, positions, and scales within the head search area. Hence, the classifier outputs are used twice, for head localization and estimation of the head pose.
The proposed single-frame based approach for combined pedestrian head localization and head pose estimation is used as an initialization step for a following head pose tracking presented in the next section.

3.2 Head Pose Estimation for Video Sequences

Section 3.1 describes the estimation of pedestrian head poses at very low resolution in single images. One drawback can be, that uncertainties of the head pose classifiers in one single frame can directly have impact on the overall performance or later function decisions. Additionally, lacks of detections in case of occlusions cannot be handled. As the underlying data sources are video-sequences, the idea to implement a head pose tracking arises immediately in order to overcome the above mentioned problems. The basic content of this section can be summarized as follows. The presented method extends the single frame based approach as presented in Section 3.1 by including head pose tracking over time. This leads to a more robust, more reliable and more efficient pedestrian head pose estimation. Additionally through the nature of the new approach it is possible to estimate continuous head pan angles instead of determining only one of eight discrete head pose classes (cf. [Robertson and Reid, 2006, Orozco et al., 2009, Siriteerakul et al., 2010]). An overview for the tracking method can be seen in Figure 3.13.

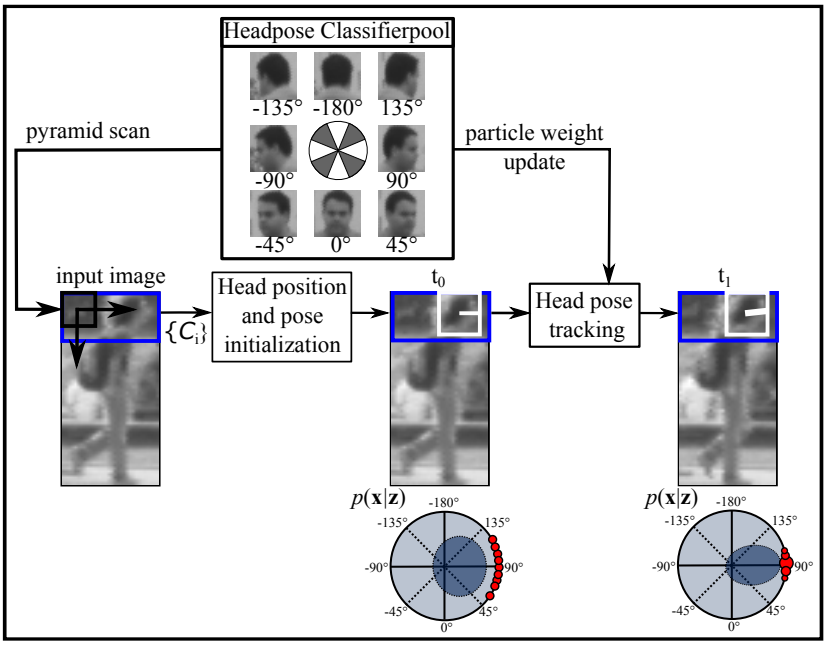


Figure 3.13: Head pose tracking system overview. The upper part (blue) of the pedestrian input image will be scanned by a set of head pose classifiers. Depending on the resulting confidence values $\{C_i\}$ the head position and head pose get initialized. Additionally, tracking over time is realized using a particle filter, where the head pose classifiers are used once again to update the particle weights.

Assuming a detected pedestrian, the position and the pose of its head are initialized by the single frame based approach presented in Section 3.1. In brief, a head search area will be defined to be the upper part of the detected pedestrian. This area will be scanned by eight head pose associated classifiers resulting in a set of confidences about potential head candidates per search location. A consideration of these confidence values results in a head position and head pose prediction. This will be the input for the tracking module, where a particle filter is implemented capturing the head position and head pose over time. The final head pan angle will be a continuous estimate within the range from -180° to 180°. The presented method can be applied easily on top of an existing pedestrian detection system and used as input for later pedestrian intention recognition (Chap. 5) or path prediction (Chap. 6).
The remaining part of this section is organized as follows. Section 3.2.1 presents the theoretical background for Bayesian tracking together with a general formulation of the particle filtering approach. Section 3.2.3 focuses on the initialization of the head position and its pose. In Section 3.2.4 the initialized person head pose will then be tracked, applying the principles of Section 3.2.1.

3.2.1 General Formulation of Bayesian Tracking

Consider the discrete-time system model

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \boldsymbol{\nu}_{k-1}) \tag{3.19}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{n}_k), \tag{3.20}$$

where $\mathbf{x}_k \in \mathcal{X}$ is the system state vector belonging to a state space $\mathcal{X} \subseteq \mathbb{R}^d$ and has to be estimated in time $k = 1, 2, \ldots$. The vector $\mathbf{z}_k \in \mathcal{Z}$ represents the observations obtained at time instance k with $\mathcal{Z}$ defining the observation space. $\boldsymbol{\nu}_k$ and $\mathbf{n}_k$ are mutually independent stochastic processes following an arbitrary distribution and encoding the system noise and the observation noise, respectively. Both mappings f_k (state evolution model) and h_k (observation function) are assumed to be nonlinear, in general. Since the system and the observations are stochastic, only the probability density function (*pdf*) $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ of the state $\mathbf{x}_k$ can be determined, given all past and current observations $\mathbf{z}_{1:k} := \{\mathbf{z}_1, ..., \mathbf{z}_k\}$. Hence, in Bayesian tracking the goal of the state estimation problem is to determine the conditional (posterior) pdf $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ at each time step k. Assuming a starting state prior distribution $p(\mathbf{x}_0)$ is known, the posterior pdf can be recursively determined according to the prediction and update steps.

1. Prediction (prior):

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}. \tag{3.21}$$

2. Update (posterior):

$$p(\mathbf{x}_k|\mathbf{z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \tag{3.22}$$

$$= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k}. \tag{3.23}$$

$p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the state transition pdf, which is defined by the system equation (3.19), and $p(\mathbf{z}_k|\mathbf{x}_k)$ is the observation likelihood function defined by the measurement model in equation (3.20). Equation (3.23) represents Bayes' rule. For some applications of Bayesian tracking no analytic solution exists, especially, when trying to solve the integral in the denominator of equation (3.23). To handle this fact, particle filters provide an opportunity to approximate the optimal Bayesian solution, see [Arulampalam et al., 2002].

3.2.2 Introduction to Particle Filtering

A state space trajectory $\mathbf{x}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k\} \subset \mathcal{X}$ is defined, which represents the evolution of the state $\mathbf{x} \in \mathcal{X}$ during the time interval from 0 to k. To circumvent the integration that is necessary for the evaluation of the denominator of the right-hand side in equation (3.23), in particle filtering (PF), the posterior pdf $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ is approximated by a set of random support samples (*particles*) $\{\mathbf{x}_{0:k}^{(i)}, i = 1, ..., N\}$ with associated weights $\{w_k^{(i)}, i = 1, ..., N\}$. The weights are normalized such that $\sum_i w_k^{(i)} = 1$.

The posterior state pdf at k is approximated as

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)}), \tag{3.24}$$

with δ denoting the Dirac delta function. Each particle with the respective weight $\{\mathbf{x}_{0:k}^{(i)},\ w_k^{(i)}\}$ can be considered as a hypothesis that the real state trajectory is $\mathbf{x}_{0:k}^{(i)}$ with a certainty of $w_k^{(i)}$. This hypothesis is updated in two steps.

1. State update: Randomly draw N new samples $\{\mathbf{x}_{0:k+1}^{(i)},\ i = 1, ...,\ N\}$ by applying the state evolution model (eq. (3.19)) to the previous state $\mathbf{x}_k^{(i)},\ i = 1, ...,\ N$
2. Observation update: Update the weights $w_{k+1}^{(i)},\ i = 1, ...,\ N$ with respect to the likelihood function of the current observation.

It can be shown that if $N \to \infty$, equation (3.24) approaches the true posterior pdf $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. In both steps, the principle of Sampling Importance Re-sampling (*SIR*) plays a crucial role to drive particles across time, cf. [Arulampalam et al., 2002].

Sampling Importance Re-sampling (SIR) Filter

For the task of approximating the pdf $p(\mathbf{x})$, which is difficult to sample from, it is supposed, that a test pdf $\pi(\mathbf{x}) \propto p(\mathbf{x})$ exists, which can be evaluated for a given $\mathbf{x}$. Let $\mathbf{x}^{(i)} \sim q(\mathbf{x}),\ i = 1, ...,\ N$, be samples that are drawn (sampled) from another pdf $q(\mathbf{x})$, which is called importance density or proposal distribution. Then, an approximation of the pdf $p(\mathbf{x})$ is given by

$$p(\mathbf{x}) \approx \sum_{i=1}^{N} w^{(i)} \delta(\mathbf{x} - \mathbf{x}^{(i)}),\ \text{with}\ w^{(i)} \propto \frac{\pi(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}\ \text{and}\ \sum_{i=1}^{N} w^{(i)} = 1, \tag{3.25}$$

where $w^{(i)}$ is the normalized weight of the ith sample. Given this principle, the weights in equation (3.24) are defined to be

$$w_k^{(i)} \propto \frac{p(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k})}, \tag{3.26}$$

where the proposal distribution $q(\mathbf{x})$ can be arbitrarily chosen. Nevertheless, the choice of $q(\mathbf{x})$ is a relevant step. [Arulampalam et al., 2002] show, that if $q(\cdot)$ is chosen to factorize as

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}), \tag{3.27}$$

then the weights are recursively determined by

$$w_k^{(i)} \approx w_{k-1}^{(i)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}. \tag{3.28}$$

Algorithm 1 Sampling Importance Re-sampling (SIR) Particle Filter Algorithm

$[\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N, \hat{\mathbf{x}}_k] = \text{SIR}[\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N, \mathbf{z}_k]$
  for $i = 1 : N$ **do**
    draw $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$
    update particle weight $w_k^{(i)}$ according to eq. (3.31)
  end for
  calculate sum over weights $s = \sum_{j=1}^N w_k^{(j)}$
  for $i = 1 : N$ **do**
    normalize weight, $w_k^{(i)} = w_k^{(i)}/s$
  end for
  calculate next state hypothesis $\hat{\mathbf{x}}_k$ according to eq. (3.32)
  re-sample particles, see [Arulampalam et al., 2002]

This expression can be easily evaluated for a given triple of $\mathbf{x}_{k-1}^{(i)}$, $\mathbf{x}_k^{(i)}$ and $\mathbf{z}_k$ since it contains the known measurement, the state model in the numerator and the user-defined proposal distribution q in the denominator. A frequently used proposal distribution is the transition prior, i.e.,

$$q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) = p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}), \tag{3.29}$$

see [Arulampalam et al., 2002]. Given this assumption, the weight update process can be expressed by the recursion

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)}). \tag{3.30}$$

It has been shown by [Doucet et al., 1998], that the variance of the weights can only increase over time. This means in general, that after a few iterations all but one weight will be close to zero, which is called the *degeneracy problem*. To prevent this, particles are regularly re-sampled, i.e., particles with small weights are eliminated, and new particles are created at or around those having larger weights. There exist several efficient re-sampling algorithms of computational complexity $\mathcal{O}(N)$, that typically map the newly created particles to existing ones having high weights, such as *systematic re-sampling* in [Arulampalam et al., 2002], that is later used in this work. Re-sampling at every time step k, thus re-initializing the weights to $w_{k-1}^{(i)} = 1/N,\ i = 1, ..., N$ will further simplify equation (3.30) to

$$w_k^{(i)} \propto p(\mathbf{z}_k|\mathbf{x}_k^{(i)}). \tag{3.31}$$

The track's new state can then be determined by the conditional expectation value of the current posterior $p(\mathbf{x}_k|\mathbf{z}_k)$,

$$\hat{\mathbf{x}}_k = \mathbb{E}(\mathbf{x}_k|\mathbf{z}_k) = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_k) d\mathbf{x}_k \approx \int \mathbf{x}_k \sum_{i=1}^N w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) d\mathbf{x}_k = \sum_{i=1}^N w_k^{(i)} \mathbf{x}_k^{(i)}. \tag{3.32}$$

In this work,the SIR particle filtering method is implemented as described by Algorithm 1.

Besides SIR, [Arulampalam et al., 2002] propose two further formulations for a particle filter namely auxiliary sampling importance re-sampling (*ASIR*) and the regularized particle filter (*RPF*), that mainly differ in the choice of the importance density. Both methods prove to outperform standard SIR in various applications. Nevertheless, SIR is leading to acceptable performance for the scope of this work with the advantage of having a simpler formulation and therefore being less computational expensive. In the following sections, the mentioned basic theory is applied to the head pose estimation problem, where one seeks to give estimates for the observation likelihood function $p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$ based on head pose detector confidence values.

3.2.3 Head Position and Head Pose Initialization

In order to track the pedestrian's head position and head pose, both have to be initialized first. This can be achieved by the single frame based approach presented in Section 3.1. There, eight different head pose classifiers are trained, associated to eight discrete head pose classes. These classes, each having a range of 45°, cover the full head pan angle range of 360°, see Figure 3.9. Given a pedestrian hypothesis (bounding box) B, the so-called *head search area* $A_{\text{search}} \subset B$ is defined to be the upper part of the person's bounding box. A_{search} will be scanned at multiple scales by the trained head pose associated classifiers, resulting in several classifier confidence values per search position. Therefore, A_{search} can be described as a finite set of triples including all image positions (u, v) at a particular scale s, that need to be evaluated. I.e., $A_{\text{search}} = \{(u^{(1)}, v^{(1)}, s^{(1)})', \ldots, (u^{(N)}, v^{(N)}, s^{(N)})'\}$. The final position and pose of the pedestrian's head is determined performing a Bayesian decision, where the posterior probability of having a head at a particular position and pose, given the head search area will be maximized. The posterior probability will be approximated by the confidence outputs of the head pose classifiers, that are evaluated restricted to the area A_{search}. Classifier outputs are obtained from boosting cascades integrating Haar-like, cf. [Viola and Jones, 2001a], or local structure features using MCT, cf. [Fröba and Ernst, 2004], for each of the head pose classes within a modified *one-vs.-all* framework. While testing, the track initialization is given as mentioned previously, thus taking that particular position and head pose, which holds the highest confidence over the whole head search area. Defining the state space vector $\mathbf{x} := (u, v, s, \theta)'$ capturing the image coordinates (u, v) of the head's position, the scale s of the detection pyramid level and the associated discrete head pan angle $\theta \in \mathbb{O} = \{-180°, -135°, \ldots, 135°\}$, the initial state is estimated by

$$\hat{\mathbf{x}}_0 = \underset{\boldsymbol{\xi} \in A_{\text{search}}, o \in \mathbb{O}}{\arg\max} \; P(\mathbf{x} = (\boldsymbol{\xi}, o)' | A_{\text{search}}), \tag{3.33}$$

with

$$P(\mathbf{x} = (\boldsymbol{\xi}, o)' | A_{\text{search}}) = P\left\{(u, v, s)' = \boldsymbol{\xi}, \theta = o | A_{\text{search}}\right\} \approx \mathcal{C}_o(\boldsymbol{\xi}). \tag{3.34}$$

$\mathcal{C}_o(\boldsymbol{\xi})$ is the confidence value resulting from evaluating the head pose classifier corresponding to the discrete head pose class $o \in \Theta$ at the image position and scale, which are comprised by $\boldsymbol{\xi}$. To ensure the comparability of the confidence values, that are resulting from different independent classifiers, and to represent a probability measure, each original value is mapped to the range $[0, 1]$ by a sigmoid function $f(\mathcal{C}) = 1/(1 + e^{-(a\mathcal{C}+b)})$. The parameters a and b are determined by a logistic regression, cf. [Mehta and Patel, 1995], independently for each classifier using the confidence values from a validation set. The state $\hat{\mathbf{x}}_0$ is the input for the head pose tracking part using particle filtering discussed in the next Section 3.2.4.

3.2.4 Particle Filtering for Head Pose Tracking

In order to achieve robust head pose estimation results over time, the initialized head position and head pose given in Section 3.2.3 are tracked by a particle filter algorithm presented in Section 3.2.2. This section describes the basic parts for the design of the particle filter used within this work namely state evolution model and the state observation model. One problem arises, when modeling the system and measurement noise with respect to the tracked head pose state. A frequently used candidate for system noise modeling is the normal distribution. This however is not suited for periodic angular signals given by the head pose. An equivalent of the normal distribution for the circular domain is given by the *von Mises* distribution, which is presented in the following.

The von Mises Distribution

The von Mises probability density function for the angle θ is given by

$$\mathcal{M}(\theta; \mu, \kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)}, \tag{3.35}$$

where $I_0(k)$ is the modified Bessel function of order 0. The parameters μ and $1/\kappa$ are analogous to μ and σ^2 (mean and variance) in the normal distribution. I.e., μ is a measure for location (the distribution is clustered around μ), and κ is a measure for concentration. For κ equal to zero the von Mises distribution reduces to a circular uniform distribution, while for larger κ, the distribution becomes very concentrated around the angle μ, i.e., approaching a normal distribution in θ with mean μ and variance $1/\kappa$.

State Evolution Model

As potential state propagation models one could think of complex motion models that capture human head turnings. In practice, however, much simpler models turn out to give sufficiently high accuracy especially for head pose tracking at very low resolution.

In this work, the state and generated particles are propagated by an evolution model, which is assumed to be constant except of an additive random noise term using a Gaussian for position and scale, and a von Mises distribution for the head pose, respectively. To be more specific, the head movement is modeled by an additional uncertainty on the current state assuming no significant change within two time instances. The state evolution model is defined as follows,

$$\mathbf{x}_k := \begin{pmatrix} u_k \\ v_k \\ s_k \\ \theta_k \end{pmatrix} = \begin{pmatrix} u_{k-1} \\ v_{k-1} \\ s_{k-1} \\ \theta_{k-1} \end{pmatrix} + \boldsymbol{\nu}_{k-1}. \tag{3.36}$$

$\boldsymbol{\nu}_{k-1}$ is a zero mean joint distribution comprising a normal distribution for uncertainty propagation of the state's image position and scale and a von Mises distribution modeling the head pose evolution process noise, i.e.,

$$\boldsymbol{\nu}_{k-1} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_u & 0 & 0 \\ 0 & \sigma_v & 0 \\ 0 & 0 & \sigma_s \end{pmatrix} \right) \otimes \mathcal{M}\left(\theta; 0, \kappa_\theta\right) \tag{3.37}$$

with uncertainty parameters σ_u, σ_v, σ_s and κ_θ. All process noise parameters get attuned for the later application through multiple simulations using a validation data set. Compared to Section 3.2.3, the head pose state θ_k is now a continuous value within the interval $[-180°, 180°)$.

State Observation Model

Tracking starts by randomly drawing N samples $\mathbf{x}_k^{(i)} = (u_k^{(i)}, v_k^{(i)}, s_k^{(i)}, \theta_k^{(i)})'$ around the initial head position $\mathbf{x}_0$. The particle weights are initialized to be equal, summing up to one. The measurement vector $\mathbf{z}_k \in \mathbb{R}^4$ will include estimates for position, head size and discrete head pose class, depending on the head pose classifier outputs. As the head pose detectors only give a discrete head pose class as measurement output, for the state observation model the step from the continuous to discrete domain with regard to the actual observed discrete head pose class $\Theta \in \mathbb{O} = \{-180°, \ldots, 135°\}$ has to be performed. Therefore, the observation likelihood function (eq. (3.23)) at each particle state for time instance k is defined as

$$p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) = \sum_{o \in \mathbb{O}} p(\mathbf{z}_k|\Theta = o, \mathbf{x}_k^{(i)}) p(\Theta = o|\mathbf{x}_k^{(i)}) \tag{3.38}$$

$$= \sum_{o \in \mathbb{O}} p(\mathbf{z}_k|\Theta = o, u_k^{(i)}, v_k^{(i)}, s_k^{(i)}) p(\Theta = o|\theta_k^{(i)}), \tag{3.39}$$

using the *law of total probability*. The inner term $p(\Theta = o|\theta_k^{(i)})$ models the probabilistic relation between the actual head pose $\theta_k^{(i)}$ and one of the discrete head pose class realizations

$o \in \mathbb{O}$ and can be calculated using *Bayes' Theorem* by

$$p(\Theta = o|\theta_k^{(i)}) = \frac{p(\theta_k^{(i)}|\Theta = o)P(\Theta = o)}{\sum_{o\in\mathbb{O}} p(\theta_k^{(i)}|\Theta = o)P(\Theta = o)}. \tag{3.40}$$

For a particular o, $p(\theta_k^{(i)}|\Theta = o)$ is modeled by a von Mises distribution centered at the value of $\mu_o \in [-180, 180)$ with the concentration parameter κ_o, i.e.,

$$p(\theta_k^{(i)}|\Theta = o) \sim \mathcal{M}(\theta; \mu_o, \kappa_o). \tag{3.41}$$

$p(\theta_k^{(i)}|\Theta = o)$ can also be interpreted as the distribution for the class dependent discretization error.
The prior distribution $p(\Theta)$ is modeled to be uniform, thus

$$P(\Theta = o) = 1/8, \; \forall o \in \mathbb{O}. \tag{3.42}$$

The first inner term in equation (3.39) is now the measurement likelihood with respect to a discrete head pose class $o \in \mathbb{O}$. This is now estimated by the detection confidence output for the head pose classifier corresponding to the head pose class o evaluated at the particle's head position similar to equation (3.34), hence

$$p(\mathbf{z}_k|\Theta = o; u_k^{(i)}, v_k^{(i)}, s_k^{(i)}) \approx \mathcal{C}_o(u^{(i)}, v^{(i)}, s^{(i)}). \tag{3.43}$$

After re-sampling and propagating with respect to the evolution model, the particle weights are updated according to equation (3.31) and the likelihood observation function from equation (3.39), followed with the estimation of the next state applying equation (3.32).

3.3 Datasets

This section describes the datasets that are being used throughout the experiments in Section 3.4. The presented approach gets evaluated on three kinds of datasets varying in their application background and proposed ground truth information. Each dataset is described in detail by means of statistics over samples per head pose class and head size occurrences.

3.3.1 The CLEAR Dataset

One possible choice for a dataset is the CLEAR dataset, which was generated for the *CLEAR* workshop in 2008, cf. [Stiefelhagen et al., 2008, Voit et al., 2008]. The data consists of multiple video sequences including 15 different persons recorded with four cameras mounted at the upper corners of a smart room. During recording, all persons were equipped with a magnetic motion sensor to measure their ground truth head orientation (pan, tilt and roll)

Figure 3.14: Example images recorded with camera 1 for all 15 person datasets contained within the CLEAR dataset.

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
4796	4794	4555	4923	4789	4923	4555	4794

Table 3.1: Number of samples per head pose class for CLEAR training data.

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
2322	2460	2300	2464	2406	2464	2300	2460

Table 3.2: Number of samples per head pose class for CLEAR evaluation data.

relative to a fixed grid, which was aligned with the smart room's coordinate system. Using the camera calibration parameters, the measured 3D head pan angle can be transformed into each camera frame and then used for image based head pose evaluation. The 15 person datasets are divided into a development/training and evaluation set. The training set consist of the video sequences for the person 6 to 15 while the sequences containing person 1 to 5 are used for evaluation. Example images from all 15 CLEAR person datasets are shown in Figure 3.14. In this work, training of head pose detectors will be performed on the complete training set (4 cameras). Head samples are additionally mirrored about the vertical axes and added to the corresponding opposite head pose class sets. Later experiments on cropped head patches only (Sec. 3.4.2), run on the evaluation data using all camera images as well including mirroring of samples. The Tables 3.1 and 3.2 show the distribution of head samples over the considered eight discrete head pose classes for the development/training and evaluation set, respectively, using the labeled images of all four cameras. The combined system for pedestrian head localization and head pose estimation introduced in Section 3.1 and 3.2 however will be evaluated only on the sequences recorded with camera 1. The resulting sample distribution restricted to camera 1 is displayed in Table 3.3.

In later experiments, the performance of the developed algorithm for pedestrian head pose

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
299	381	267	271	248	244	347	340

Table 3.3: Number of samples per head pose class for CLEAR evaluation data using camera 1 only.

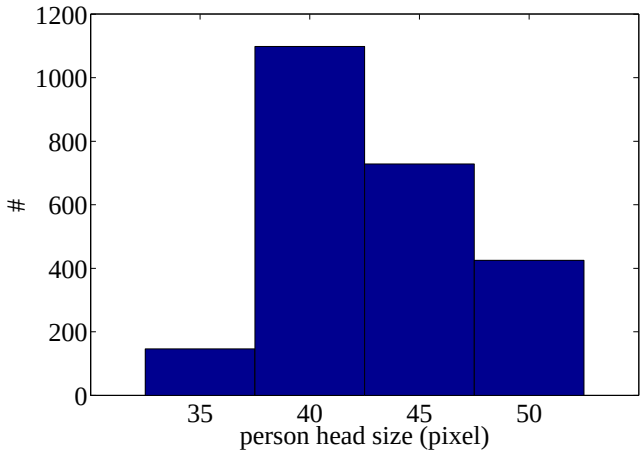


Figure 3.15: Distribution over person head sizes (pixel) for the CLEAR evaluation dataset (person 1 to 5) considering the images from camera 1 only.

estimation is evaluated depending on the samples' head size, that is defined to be the maximum value over of the head width and head height in the image plane. Together with camera calibration parameters the minimum head size with acceptable performance indicates the maximum detection range of the developed system. A distribution over the person head sizes in the image for the CLEAR evaluation set (person 1 to 5) restricted to camera 1 is shown in Figure 3.15. Mainly, head sizes lie within the range from 34 to 50 pixels.

3.3.2 The CAVIAR Dataset

Another dataset including labeled persons, person heads and head pan angles is given by the CAVIAR data, cf. [CAVIAR, 2004]. This dataset consists of several sequences of static images including moving persons in a shopping mall taken from a surveillance camera. Persons in the images appear at very low resolution, similar to the conditions in driver assistance scenarios. Therefore, the dataset is well suited for evaluation in this work. The CAVIAR dataset is used for evaluation only, i.e., there is no separation of data into training and testing subsets. Head pose labels are available for the sequences *ShopAssistant1cor*, *ThreePastShop1cor*, *OneShopOneWait2cor* and *ThreePastShop2cor*. Figure 3.16 displays some screenshots for these sequences. For later evaluation, only fully visible persons having a bounding box height of at least 40 pixels (head size $\approx 6\times6$ pixels) are considered. For lower resolution meaningful results cannot be expected. Table 3.4 shows the resulting distribution of head pose samples over the considered eight discrete head pose classes. Figure 3.17 shows the distribution of person head sizes in the image for all evaluated samples from the CAVIAR

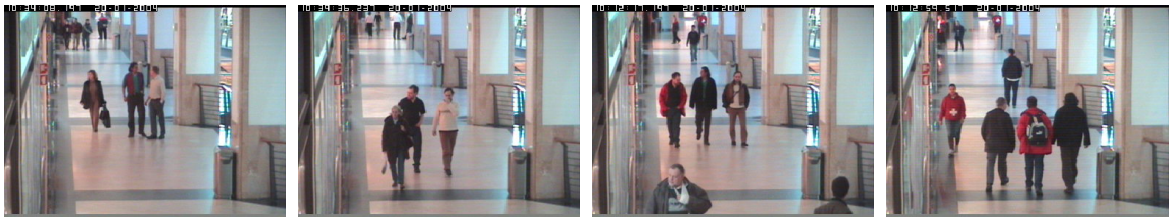


Figure 3.16: Example images from the [CAVIAR, 2004] dataset.

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
7378	1416	2214	171	3075	470	2386	3239

Table 3.4: Number of samples per head pose class for CAVIAR dataset.

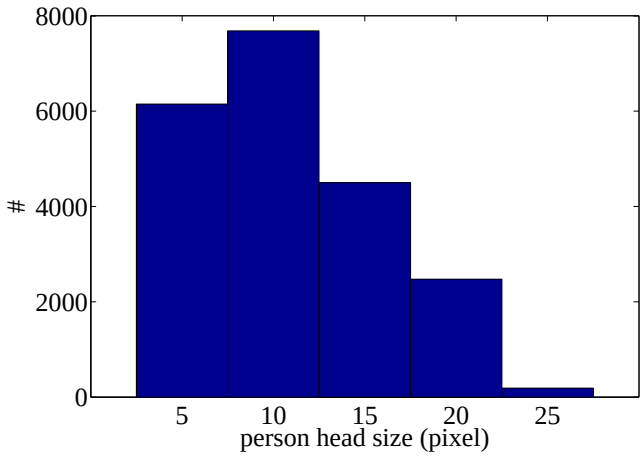


Figure 3.17: Distribution over person head sizes (pixel) for the CAVIAR dataset.

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
14756	4655	4600	641	6150	641	4600	4655

Table 3.5: Number of samples per head pose class for increased CAVIAR dataset involving a mirrored version for each sample.

dataset. Human heads are mainly appearing at resolutions of 6×6 pixels and not at resolutions higher than 30×30 pixels. Similar to the CLEAR dataset, for the head patch only evaluation (Sec. 3.4.2) samples are additionally mirrored about the vertical axis resulting in the increased evaluation set given by Table 3.5.

3.3.3 The Bosch Inner-City Dataset

Finally, as this is the main focus of the presented work's contribution, results on real world data taken from a moving camera in inner-city scenarios are presented. The inner-city dataset includes labeled pedestrian heads in video sequences containing high dynamic gray value images. For training and validation purposes, the dataset is divided into two parts. Head pose labels are provided with discretization steps of 45°. Similar to the CLEAR training

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
2256	2458	3283	2018	2492	2018	3283	2458

Table 3.6: Number of samples per head pose class for the Bosch Inner-City training data.

Figure 3.18: Collection of test samples of the Bosch Inner-City dataset.

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
1796	823	3716	953	1299	676	1563	365

Table 3.7: Number of samples per head pose class for the Bosch Inner-City evaluation data.

set, all training samples are mirrored about the vertical axes and added to the corresponding opposite head pose class sets. Table 3.6 shows the resulting distribution of head pose samples over the considered eight discrete head pose classes on the training set, which is later used to train multiple head pose detectors. For evaluation, samples are collected from 24 inner city video sequences with 300 Frames each. The test data consists of 243 person tracks including 11191 labeled pedestrian images in total. Some of the test images can be seen in Figure 3.18. The number of samples per discrete head pose class for the evaluation data is given in Table 3.7. Figure 3.19 shows the distribution of person head sizes in the image for all evaluated samples from the Bosch Inner-City dataset. Pedestrian heads are mainly appearing at resolutions of around 6×6 pixels and a maximum resolution of 60×60 pixels.

Dataset Summary The most important facts over all three evaluation datasets presented in this section are summarized in Table 3.8.

3.4 Experiments

This section will present a performance evaluation for the introduced pedestrian head pose estimation approach. As it is difficult to retrieve labeled data from inner-city scenarios including pedestrian head pose measurements, this work additionally makes the use of public

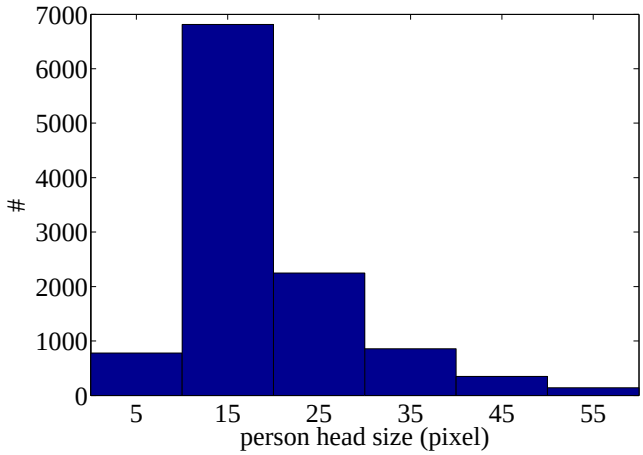


Figure 3.19: Distribution over pedestrian head sizes (pixel) for the Bosch Inner-City evaluation dataset.

	# Persons	# Samples	Min. head size	Label acc. θ	Scenario
CLEAR	5	2397 (cam 1)	34×34 pxls.	0.1°	Smart room
CAVIAR	22	20349	6×6 pxls.	10 °	Surveillance
Bosch	243	11191	6×6 pxls.	45 °	Inner-city

Table 3.8: Overview of the three evaluation datasets (CLEAR/CAVIAR/Bosch) presented in Section 3.3. Number of different persons, total number of samples, minimum head size, head pose label accuracy and scenario type are displayed.

available datasets from different environments such as data from surveillance or smart rooms presented in Section 3.3. It has to be pointed out, that the goal is not to outperform state-of-the-art methods working well in these indoor scenarios, rather than getting a feeling for the developed system's performance. In this work, the general interest is in a fast and robust method (depending on the later application), that is easy to apply to an overall pedestrian protection system. In the following, the two suitable public available datasets, namely *CLEAR* and *CAVIAR*, will be evaluated together with the Bosch Inner-City dataset, taken from a vehicle mounted camera.

First, the experimental setup for the chosen datasets is explained in Section 3.4.1. In a second step, a comparison with approaches from literature is carried out (Sec. 3.4.2). Furthermore, the evaluation for the single-frame based approach and the head pose tracking is presented in Section 3.4.3 and Section 3.4.4.

3.4.1 Experimental Setup

The proposed method integrates classifiers based on different kinds of features, which proved to be very powerful in the fields of pedestrian- and face-detection as well as for head pose estimation problems. First, boosting cascades of *Haar-like* features are trained as proposed in [Viola and Jones, 2001a, Lienhart and Maydt, 2002]. The second type of classifiers are based on local structure features calculated using the modified Census Transform, cf. [Fröba

and Ernst, 2004]. These features proved to outperform the Haar-like features concerning the problem of face detection. Depending on the underlying evaluation data, a different system setup under various aspects is performed. First, there is a difference on the training set that is used to learn the head pose classifiers and secondly the tracking system parameters like, for instance, the system and observation noise are adapted separately for each dataset.

CLEAR Setup

For evaluation on the CLEAR data, the partial datasets for person 6 till 15 are used for training of the head pose detectors. Therefore, a modified one-vs.-all framework is realized as explained in Section 3.1.4 using the training data given in Table 3.1. Training samples are scaled to a common size of 20×20 pixels. The mean training samples for each of the eight head pose classes are shown in Figure 3.20.

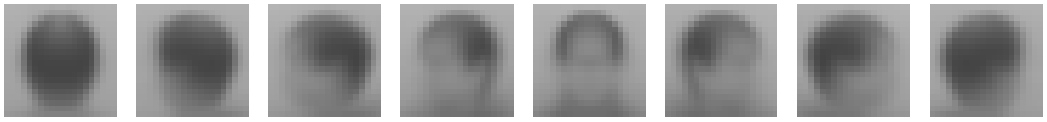


Figure 3.20: Mean pedestrian head pose templates used to train eight head pose classifiers.

MCT-based cascades consists of 6 stages with the maximum allowed feature locations per stage to be {20,80,140,200,400} and a final Winnow classifier, see [Fröba and Ernst, 2004]. Boosting cascades integrating Haar-like features are trained with 8 stages consisting of strong Gentle Adaboost classifiers incorporating decision trees as weak learners (cf. Sec. 3.1.2). The overall system is evaluated using the images contained in the person datasets 1 to 5. As reference, only images taken from camera 1 are used instead of considering the different views from the additional cameras (2,3,4), which would further improve the head pose estimation result, see [Voit et al., 2008]. As the presented method receives the input of a pre-detected pedestrian bounding box, rectangular candidate regions are generated by additionally labeling the full person bodies. The resulting bounding boxes will then be the system input.

In a further step, the system noise model parameters (eq. (3.37)) for the particle filter introduced in Section 3.2.4 have to be estimated. For the chosen design (eq. (3.36)), the process noise parameters are mainly influenced by the change in head position, size (scale), and head pose between to time instances. Hence, the different parameters can be estimated by building up statistics for these changes using the ground truth head label information. This procedure is exemplary shown in Figure 3.21 to determine the process noise parameter κ_θ for the head pose state θ. Analogously, the same procedure can be applied for the remaining parameters σ_u, σ_v and σ_s. Including heuristics through expert knowledge the parameters get slightly adapted in order to achieve best performance on the training sets. The resulting parameters are given in Table 3.9. For the observation model in equation (3.39), the step from con-

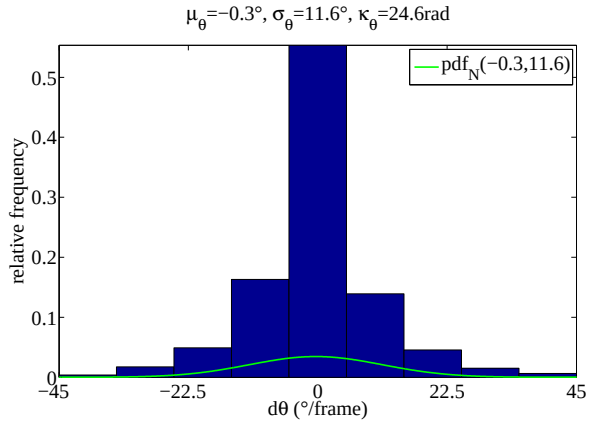


Figure 3.21: Process noise parameter estimation for the head pose state θ on the CLEAR dataset. The probability density function (pdf) of a normal distribution gets fitted to the ground truth frame-wise angular changes for the pan angle θ. The value of the estimate for the standard deviation $\sigma = 11.6$ is directly related to the dispersion value $\kappa_\theta = 24.6$ for the von Mises distribution.

σ_u	σ_v	σ_s	κ_θ
4.0	2.0	0.15	24.6

Table 3.9: Process noise parameters for head pose tracking determined for CLEAR dataset

tinuous to discrete state space with respect to the head pose state θ results in modeling the conditional probability of having a continuous head pose θ given the discrete head pose class observation Θ (eq. (3.41)). This probability is modeled to be a von Mises distribution, where the model parameters μ_o and κ_o, $o \in \{-180, \ldots, 135\}$ have to be estimated using the ground truth head pose labels, see [Fisher, 1996]. Figure 3.22 shows the resulting distributions in the circular domain.

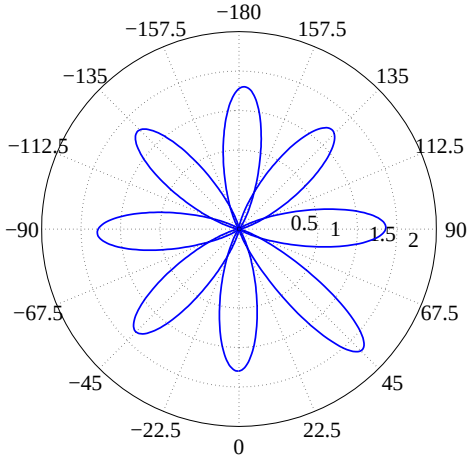


Figure 3.22: Estimated models for dependency of continuous head pose state estimate θ and observed discrete head pose class $\Theta = o$, $o \in \{-180, \ldots, 135\}$ (eq. (3.41)) on the CLEAR dataset.

μ_{-180}	μ_{-135}	μ_{-90}	μ_{-45}	μ_0	μ_{45}	μ_{90}	μ_{135}
177.8°	-133.9°	-88.5°	-45.2°	-0.5°	45.6°	90.7°	136.8°
κ_{-180}	κ_{-135}	κ_{-90}	κ_{-45}	κ_0	κ_{45}	κ_{90}	κ_{135}
20.7	20.2	20.5	21.6	20.6	30.2	22.0	18.9

Table 3.10: Estimated model parameters for dependency of continuous head pose state estimate θ and observed discrete head pose class $\Theta = o$, $o \in \{-180, \ldots, 135\}$ on CLEAR data.

In Table 3.10 the estimated model parameters related to all discrete head pose classes are displayed.

CAVIAR Setup

For evaluation on the CAVIAR dataset the same head pose detectors trained on the CLEAR data are used, so there is no additional adapted training. As the input pedestrian bounding box for the developed method the labeled rectangular region from ground truth is used by adding additional noise at the label position and label dimensions to simulate a realistic output from a pedestrian detection system. For this, a uniform noise of up to 10% of the label boxes' height is added to the height and the center of the resulting bounding box.
Again, the particle filter system noise model parameters (eq. (3.37)) have to be adapted for the CAVIAR dataset. Similar to the CLEAR data, the different parameters are estimated by building up statistics for the observed state changes within two time instances using the ground truth head label information. Figure 3.23 shows the procedure to determine the process noise parameter κ_θ for the head pose state θ.

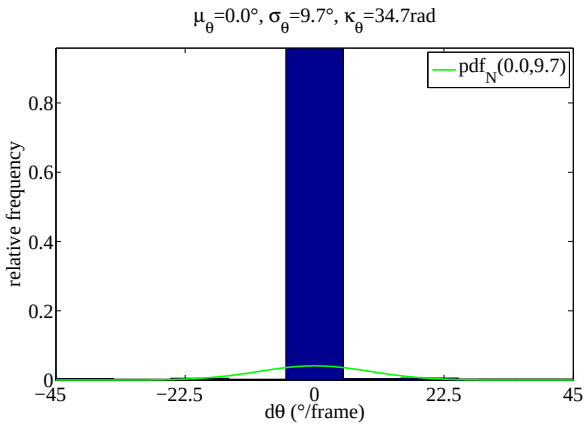


Figure 3.23: Process noise parameter estimation on the CAVIAR dataset for the head pose state θ. The probability density function (pdf) of a normal distribution gets fitted to the ground truth frame-wise angular changes for the pan angle θ. The value of the estimate for the standard deviation $\sigma_\theta = 9.7°$ is directly related to the dispersion value $\kappa_\theta = 34.7$ for the von Mises distribution.

σ_u	σ_v	σ_s	κ_θ
4.0	2.0	0.15	34.7

Table 3.11: Process noise parameters for head pose tracking determined for the CAVIAR dataset.

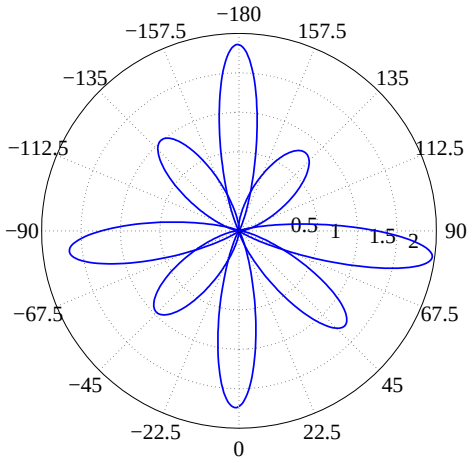


Figure 3.24: Estimated models for dependency of continuous head pose state estimate θ and observed discrete head pose class $\Theta = o$, $o \in \{-180, \ldots, 135\}$ (eq. (3.41)) on the CAVIAR dataset.

Together with adaptations through expert heuristics the final parameters are displayed in Table 3.11. For the observation model, the von Mises distribution to model the conditional probability of having a continuous head pose state θ given the discrete head pose class observation Θ (eq. (3.41)) is estimated using the ground truth head pose labels in the same way as it is done for the CLEAR dataset. The results are visualized in the circular domain in Figure 3.22. The exact estimated model parameters related to all discrete head pose classes are listed in Table 3.12. It has to be pointed out, that in case of the CAVIAR dataset, tracking parameter optimization is performed on the data itself. This might lead to a slight imaginary performance gain when using the developed tracking approach.

μ_{-180}	μ_{-135}	μ_{-90}	μ_{-45}	μ_0	μ_{45}	μ_{90}	μ_{135}
-179.6°	-139.3°	-83.1°	-45.5°	-1.1°	48.2°	82.3°	139.5°
κ_{-180}	κ_{-135}	κ_{-90}	κ_{-45}	κ_0	κ_{45}	κ_{90}	κ_{135}
35.2	14.7	29.7	13.8	31.8	20.6	38.6	10.9

Table 3.12: Estimated model parameters for dependency of continuous head pose state estimate θ and observed discrete head pose class $\Theta = o$, $o \in \{-180, \ldots, 135\}$ on the CAVIAR dataset.

Bosch Setup

In comparison to the CAVIAR data head pose detectors are retrained in a modified one-vs.-all fashion using cropped pedestrian head images from the Bosch Inner-City dataset, see Section 3.3.3. First, all training samples are scaled to the common size of 20×20 pixels. The mean training samples for each of the eight head pose classes are shown in Figure 3.25. In order to be more robust against false positive detections, MCT-based cas-

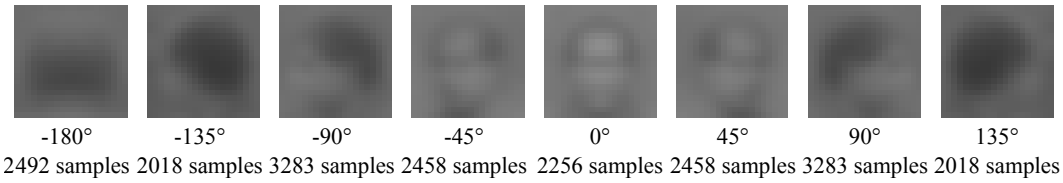


Figure 3.25: Mean pedestrian head pose templates used to train eight head pose classifiers on the Bosch Inner-City dataset.

cades consists of 10 stages with the maximum allowed feature locations per stage to be {20,40,80,120,160,200,220,280,400} and a final Winnow classifier, see [Fröba and Ernst, 2004]. The number of stages for boosting cascades integrating Haar-like features is increased to be 14. Again, each stage consists of strong Gentle Adaboost classifiers incorporating decision trees as weak learners (cf. Sec. 3.1.2). Similar to the CAVIAR data, labeled pedestrian bounding boxes serve as input for the developed approach. These bounding boxes are perturbed in center and height by an additive uniform noise of up to 10% of the boxes' height.
For later real world application, the system noise model parameters (eq. (3.37)) for the particle filter introduced in Section 3.2.4 are updated. Therefore, the set of training sequences is used. The built statistics for determination of the process noise parameter κ_θ using the ground truth head label information is shown in Figure 3.21.
The remaining estimated parameter values for σ_u, σ_v and σ_s are mentioned in Table 3.13.

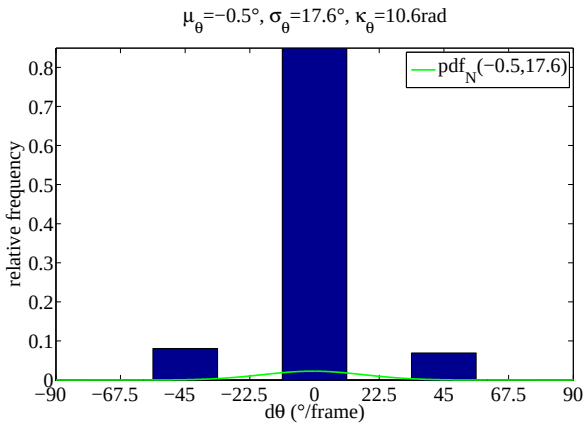


Figure 3.26: Process noise parameter estimation on the Bosch Inner-City dataset for the head pose state θ. The probability density function (pdf) of a normal distribution gets fitted to the ground truth frame-wise angular changes for the pan angle θ. The value of the estimate for the standard deviation $\sigma = 17.6$ is directly related to the dispersion value $\kappa_\theta = 10.6$ for the von Mises distribution.

σ_u	σ_v	σ_s	κ_θ
5.0	2.0	0.15	10.6

Table 3.13: Process noise parameters for head pose tracking determined for the Bosch Inner-City dataset.

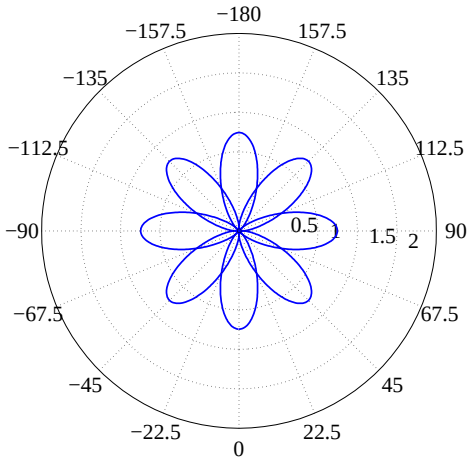


Figure 3.27: Estimated models for dependency of continuous head pose state estimate θ and observed discrete head pose class $\Theta = o$, $o \in \{-180, \ldots, 135\}$ (eq. (3.41)) on the Bosch Inner-City dataset.

When trying to determine the parameters for the observation model on the real world data one problem arises. For the real world data, labels are only available for a discretization step of 45°. Hence, estimated von Mises distribution parameters using ground truth labels in order to model the conditional probability of having a continuous head pose state θ given the discrete head pose class observation Θ (eq. (3.41)) can only result in peaked Dirac-Delta-distributions located at the discretization centers (discrete head pose classes). To overcome this problem, the affected model parameters are determined using expert knowledge and the results from the other presented datasets (CLEAR and CAVIAR). This results in the von Mises probability density functions displayed in Figure 3.27. The chosen parameters can be seen in Table 3.14.

μ_{-180}	μ_{-135}	μ_{-90}	μ_{-45}	μ_0	μ_{45}	μ_{90}	μ_{135}
-180.0°	-135.0°	-90.0°	-45.0°	0.0°	45.0°	90.0°	135.0°
κ_{-180}	κ_{-135}	κ_{-90}	κ_{-45}	κ_0	κ_{45}	κ_{90}	κ_{135}
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0

Table 3.14: Chosen model parameters for dependency of continuous head pose state estimate θ and observed discrete head pose class $\Theta = o$, $o \in \{-180, \ldots, 135\}$ on the Bosch Inner-City dataset. Parameter values are mainly set by expert heuristics due to the granularity of head pose labels.

3.4.2 Comparison with State-of-the-Art Methods

As a first part of experiments, the performance of the developed head pose estimation approach is compared to other state-of-the-art approaches from literature. As already mentioned in Section 2.3, there are several approaches relevant for the topic of head pose estimation in low resolution images. These methods and the one presented in this work mainly differ in the methodology to extract human heads from an image. Nevertheless, the subsequent step of estimating the head pose given a potential head region is quite similar throughout the methods. For this reason, the performance of the single trained head pose classifiers in this work is compared to similar approaches from literature based on extracted head images. In particular, the method of [Orozco et al., 2009] and [Siriteerakul et al., 2010] will be used for comparison here, as the basic ideas can easily be reimplemented while achieving good results in later applications. For evaluation of Orozco's method, mean head pose templates are derived from the CLEAR training set (Sec. 3.3.1) based on the origin intensity images at first. Additionally, the use of image gradients values is investigated within later experiments for comparison. As this work deals with gray-value images only, raw gray intensity values are considered instead of all RGB color channels. I.e., for this work, the Kullback Leibler coefficients that need to be calculated for Orozco's approach will not make use of color information. The mean head pose templates for intensity and image gradient are visualized in Figure 3.28. Figure 3.29 shows the calculated descriptors (eq. (2.2)) for a set of selected head

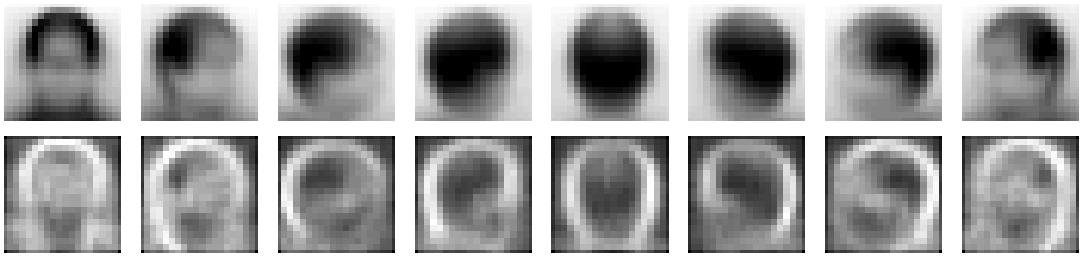


Figure 3.28: The mean head pose templates to calculate the descriptor proposed by [Orozco et al., 2009] computed for the CLEAR dataset. Upper row: intensity based only. Lower row: Image gradient based

samples taken from different head pose classes. To train the classifiers including the Non-local Intensity Difference Features (*iDF*, eq. (2.3)) proposed by [Siriteerakul et al., 2010] 10000 randomly selected out of 79800 pixel pairs are defined for a 20×20 image patch in multiple training rounds to achieve maximum performance.

Another type of features that turned out to be powerful in object recognition are the well-known Histograms of oriented Gradients (*HoG*) developed by [Dalal and Triggs, 2005]. Similar to [Voit and Stiefelhagen, 2009], HoG features are extracted for a 21×21 image patch having one block, 3×3 cells including 7×7 pixels and 8 orientation bins per cell histogram, taking the image gradient sign of orientation into account. Figure 3.30 shows the calculated HoG descriptor for a set of samples from the training set associated with different head pose classes. Two types of classifiers are used to determine the predicted head pose,

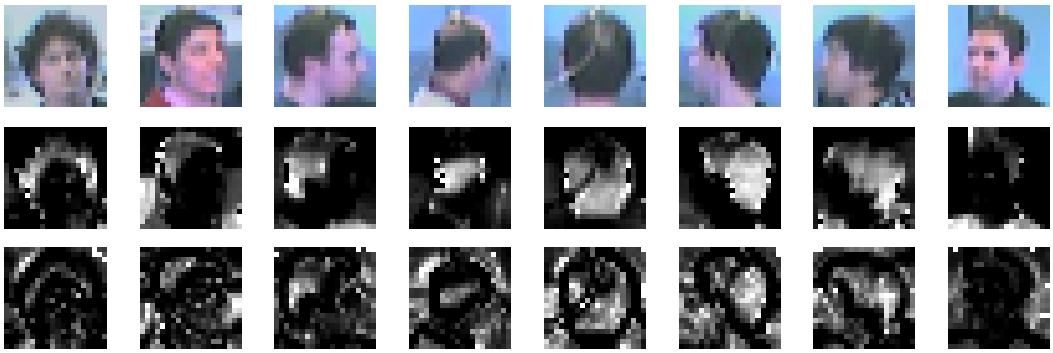


Figure 3.29: Descriptor from [Orozco et al., 2009] calculated for different samples associated with 8 discrete head pose classes. Upper row: Input images. Middle row: original descriptor using image intensity values. Lower row: Computed Descriptor based on image gradients.

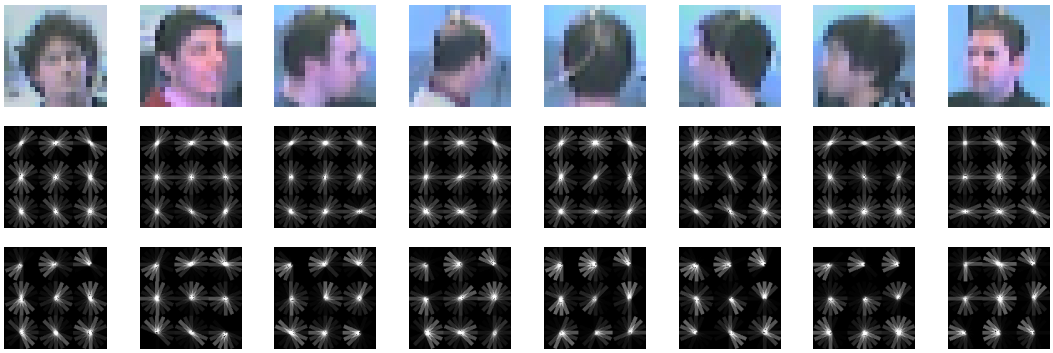


Figure 3.30: HoG descriptor from [Dalal and Triggs, 2005] calculated for different samples associated with 8 separated head pose classes. Upper row: Input images. Middle row: Original HoG settings with computed orientation angles in $[0, \pi)$. Lower row: Setup used in this work with computed orientation angles in $[0, 2\pi)$.

given a feature vector from the above mentioned methods, namely Support Vector Machines (*SVM*) and Random Forests (*RF*). As an extension to multiclass problems, both, one-vs.-one and one-vs.-all classifiers are trained, see Section 3.1.3. Training and evaluation will be performed on the CLEAR dataset (Sec. 3.3.1), where for both cases, image patches containing heads corresponding to one of the eight head pose clusters are generated at first. Additionally, head patches from the CAVIAR data (Sec. 3.3.2) are cropped and divided into eight classes resulting in a total amount of 20349 samples (40698 incl. mirroring) for evaluation. Again, as for the later real world application one has to deal with gray-value images, three channel color images will be converted to single channel images for both datasets.

The upper row of Figure 3.31 shows the averaged support vectors resulting from the trained one-vs.-all SVM multiclass classifier on CLEAR using the reimplemented method of [Orozco et al., 2009]. For all eight head pose classes, the learned appearance changes between regions where one would expect hair (lighter) and parts of the face where skin is visible (darker areas) can be observed. The lower row of Figure 3.31 displays the most distinctive features learned using a one-vs.-all RF classifier in order to differentiate one class from all other classes. The 50 most discriminative pixel pairs of Siriteerakul's method, that are trained us-

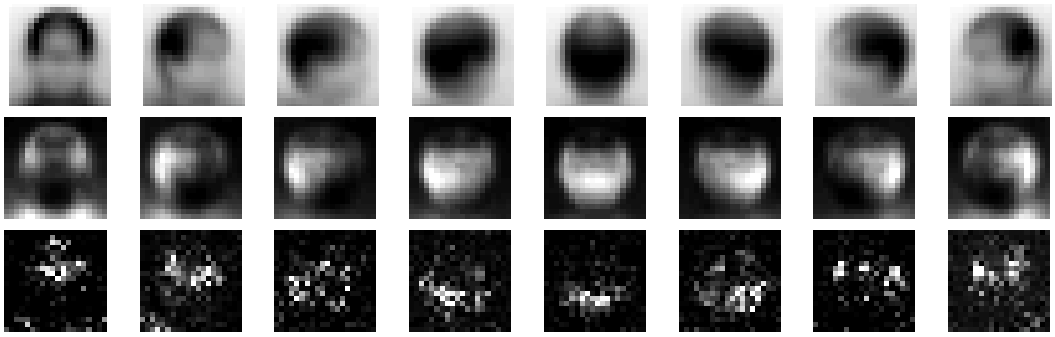


Figure 3.31: Most discriminative features learned with different classifiers. Upper row: Mean head pose templates. Middle row: Averaged support vectors for a trained one-vs.-all multiclass SVM using the method of [Orozco et al., 2009]. Lower row: most distinctive features learned with a one-vs.-all multiclass Random Forest.

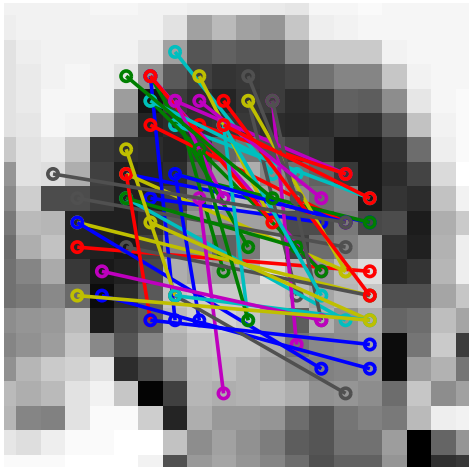


Figure 3.32: iDF classifier from [Siriteerakul et al., 2010]: Overall 50 most powerful pixel pairs learned with a one-vs.-one multiclass Random Forest.

ing a one-vs.-one multiclass RF classifier are shown in Figure 3.32. Most of the pixel pairs compare intensity values for lighter skin parts with darker areas belonging to hair or eye regions. A similar output is observed, when training a one-vs.-all multiclass RF classifier incorporating intensity difference features, see Figure 3.33.

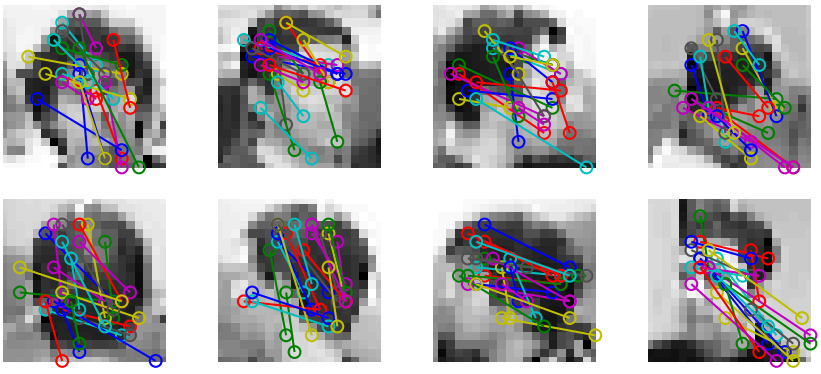


Figure 3.33: iDF classifier from [Siriteerakul et al., 2010]: 20 most powerful pixel pairs per head pose class for a one-vs.-all multiclass Random Forest.

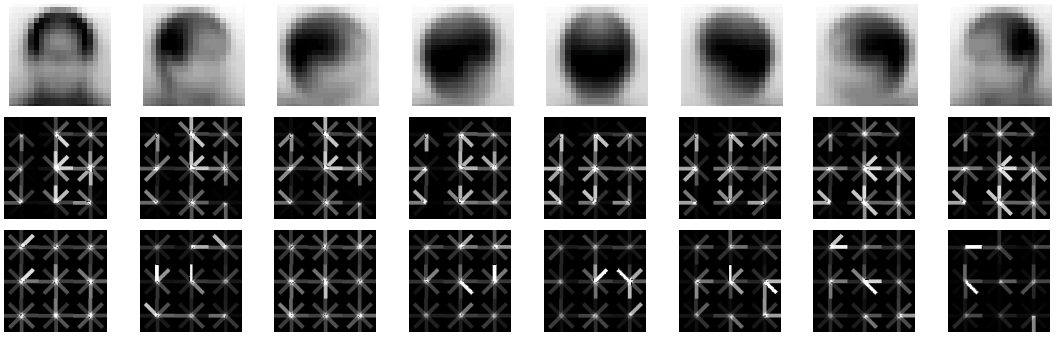


Figure 3.34: Most discriminative HoG features [Dalal and Triggs, 2005] learned with different classifiers. Upper row: Mean head pose templates. Middle row: Averaged support vectors for a trained one-vs.-all multiclass SVM. Lower row: most discriminative features learned with a one-vs.-all multiclass Random Forest.

Finally, Figure 3.34 shows learned distinctive features using different classifiers in combination with HoG features. In the upper row, the head pose class dependent averaged support vectors for a one-vs.-all multiclass SVM are displayed. Larger gradients normally appear at the head contours as well as at transitions from hair to skin areas. The most informative HoG features learned with the one-vs.-all RF classifier can be seen in the lower row of Figure 3.34. In a next step, evaluation results are presented using the following performance measures. Overall classification rates based on the eight considered head pose classes are calculated stating the ratio between correct classified head samples and all available samples in the evaluation set. A good head pose classifier should therefore have a classification rate which is significantly higher than pure guessing, i.e., higher than 12.5%. Additionally, classification rates are presented, where a predicted head pose class is assumed to be correct, if the corresponding ground truth head pose class is the same or one of the direct neighboring head pose classes. The modified classification rate gets the suffix "$\oplus$". This rate expresses the potential confusion between neighboring classes and gives an idea about the error made by discretization. A third performance measure is given by the mean angular error in head pose estimation by comparing the ground truth head pan angle with the pan angle associated to the predicted head pose class. Evaluation is performed separately on the CLEAR and CAVIAR evaluation sets. During experiments no significant difference in performance could be obtained when training a one-vs.-one or one-vs.-all classifier, respectively. Therefore, only the best one-vs.-one classifier results are presented. For comparison, the trained boosting cascades integrating Haar-like features and local structure features using MCT (Sec. 3.4.1) are subject to the patch-wise evaluation as well. Table 3.15 shows the concrete numbers for the above introduced performance measures using the different combinations of features and classifiers for head pose estimation. Figure 3.35 gives a graphical idea about single classifier performance. As expected, performance for all classifiers decreases, when evaluating the CAVIAR data instead of CLEAR due to huge number of head samples in very low resolution (Fig. 3.17). Additionally, head pose estimation performance on CAVIAR data suffers from trained classifiers generalizing from the CLEAR training set, cf. Section 3.4.1.

	classification rate (%)		classification rate ⊕ (%)		Mean angle error (°)	
	CLEAR	CAVIAR	CLEAR	CAVIAR	CLEAR	CAVIAR
Orozco + SVM	53.2/49.8**	31.3/22.9**	85.9	55.6	28.8	58.8
Orozco + RF*	69.1/67.5**	28.7/28.2**	92.8	61.1	17.5	48.2
iDF + SVM	64.0	31.2	90.2	60.7	21.7	52.9
iDF + RF*	69.2	27.5	92.2	62.0	18.1	51.8
HOG + SVM*	71.1	35.0	91.0	71.6	19.3	49.4
HOG + RF*	**72.3**	**35.4**	**94.4**	**73.1**	**15.1**	**43.3**
MCT-cascade*	69.8	32.7	92.6	70.5	19.8	51.2
Haar-cascade*	68.9	28.2	91.3	68.1	21.3	53.7

Table 3.15: Classification rates and mean angular errors for different methods and datasets (CLEAR, CAVIAR). State-of-the art approaches (Orozco SVM, iDF SVM) and contributions of this work (*). Additionally Orozco's descriptor was evaluated using mean gradient image templates (**).

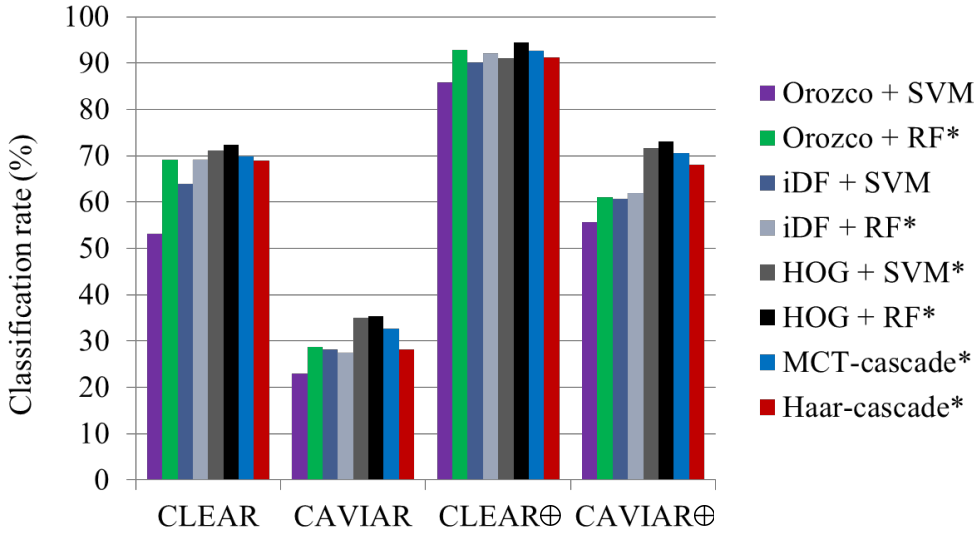


Figure 3.35: Comparison of different feature and classifier combinations on different datasets (CLEAR, CAVIAR).

Comparing the original classification rates with the modified ones (⊕), a rate increase of approximately 20% for CLEAR data and more than 30% in the CAVIAR case can be noticed meaning, that there is high confusion between neighboring classes but no significant conspicuous behavior for the different classifiers. One interesting fact is observed when using Orozco's feature. The trained RF classifier achieves nearly 16% higher classification rates for the CLEAR data compared to a trained SVM as proposed by [Orozco et al., 2009]. Descriptors calculated using mean templates based on image gradients (** in Table 3.15) are performing worse than the original proposed version based on image intensity values. Another surprising point is the discriminative power of the conventional HoG feature especially on very low resolution. Using the HoG/SVM or HoG/RF for the CAVIAR data leads to a

performance gain of more than 5% compared to all other feature and classifier combinations. The performance of the trained MCT- and Haar boosting cascades ranges from slightly worse to comparable with regard to all other classifiers. However, it has to be mentioned, that the negative sets for the MCT- and Haar-detector training additionally include areas around the heads and background images (Sec. 3.1.4), which slightly reduces the class separation efficiency. The head pose estimation developed in this chapter will later be integrated into an overall real-time application for pedestrian intention recognition and path prediction. Keeping in mind, that head pose classifiers are evaluated multiple times within the pedestrian head search area, single classifier runtime should be sufficiently low in order to retain overall system latency. For this, Table 3.16 shows the averages classifier runtimes in milliseconds on a Intel® Core™ i7-3740QM CPU @2.7 GHz for a image patch wise evaluation.

	Overall	**Feature computation**	**Classification**
Orozco + SVM	0.06 (100.0%)	0.04 (69.9%)	0.02 (30.1%)
Orozco + RF	0.11 (100.0%)	0.03 (31.6%)	0.08 (68.4%)
iDF + SVM	0.32 (100.0%)	0.03 (9.3%)	0.29 (90.7%)
iDF + RF	0.21 (100.0%)	0.03 (8.2%)	0.18 (91.8%)
HOG + SVM	0.06 (100.0%)	0.05 (87.1%)	0.01 (12.9%)
HOG + RF	0.25 (100.0%)	0.08 (30.7%)	0.17 (69.3%)
MCT-cascade	0.15 (100.0%)	0.13 (85.5%)	0.02 (14.5%)
Haar-cascade	0.19 (100.0%)	0.02 (9.6%)	0.17 (90.4%)

Table 3.16: Averaged runtimes (*ms*) per sample for patch-wise head pose estimation. Overall runtime for a 20×20 head patch and its fractional parts actually consumed for feature computation and classification.

The Orozco SVM and HoG SVM multi-class classifiers consume comparatively low runtime due to a lower dimension of the input feature vectors. The iDF-based classifiers are most computational expensive due to the very high number of pixel pairs that have to be compared for calculation of the descriptor. However, head pose estimation with the iDF feature only achieves comparable results with other methods, see Table 3.15. With $0.15ms$ and $0.19ms$ the mean evaluation runtimes for the proposed MCT- and Haar-cascades lie in between the above two extrema. In addition to the performance evaluation results from Table 3.15, especially the MCT-based classifier represent a good choice for further use within the later application. Finally, one can conclude, that using the trained head pose detectors (MCT/Haar) developed in this work show no drawback in head pose estimation performance compared to state of the art approaches by means of comparable accuracy and runtime consumption but also having the benefit to directly cope for the head localization problem (Sec. 3.1.7).

3.4.3 Single-Frame Head Localization and Head Pose Estimation Performance on Inner-City data

In this section, the developed approach for combined pedestrian head localization and head pose estimation in single images (Sec. 3.1) is evaluated at first. Due to the later application, the method has to deal with head images at very low resolution. Concerning this point, the integrated classifiers are evaluated on test samples with different resolutions in order to get the dependency between system performance and image resolution, which is directly related to the distance of a pedestrian with respect to an approaching vehicle. Obviously, the effort for the later overall system is to detect the pedestrian's intention as soon as possible. Thus, for evaluation, pedestrian detections with a maximum height of 140 pixels (head size $\approx 20 \times 20$ pixels) and a minimum height of 50 pixels (head size $\approx 7 \times 7$ pixels) are considered. As reference the Bosch Inner-City dataset presented in Section 3.3.3 is taken into account. Here, the two reference sizes of head appearances in the images are used, namely 20×20 pixels and 7×7 which corresponds to a distance of approximately $10m$ and $25m$ respectively. For evaluation, all 11191 pedestrian samples from the test set are down- or upscaled to the above mentioned reference sizes. Every sample is evaluated separately by the method proposed in Section 3.1 in order to derive the corresponding head position and head pose. An image pyramid with a scaling factor of 1.1 including 2 levels around the original scale is calculated for the head search area extracted from a pedestrian candidate. Each pyramid level is scanned with a step size of 2 pixels resulting in a list of normalized confidence values.

In a first step, the performance for head detection/localization is analyzed. Being aware of the fact, that the presence of a head within a pedestrian hypothesis is already assumed, it is possible to achieve very high head localization rates in general by ignoring false positive head detections having a low confidence value. A head localization is assumed to be correct, if cover and overlap (see [Leibe et al., 2005]) of annotated ground truth data and detection hypothesis result in values greater than a predefined threshold. During experiments, the value 0.7 was found to be appropriate in order to guaranty an adequate accuracy for later head pose estimation. Figure 3.36 shows, amongst others, the head localization rates for different classifiers and the two mentioned head resolutions.

All configurations achieve very good correct localization rates, reaching over 92% even when evaluating on the lowest considered head size of 7×7 pixels. The local structure features based on MCT seem to perform slightly better for higher resolutions than the Haar-like features, resulting in a correct localization rate of nearly 98% for head sizes of 20×20 pixels. For smaller resolutions the performance of the MCT-based features breaks down and is comparable to Haar-like features. This can be explained with the loss of pixel-wise structures within a 7×7 pixels patch in comparison to a 20×20 pixels head image whereas the Haar cascade mostly considers larger areas of sums over pixel intensities. Figure 3.37 shows some samples of miss-localized heads at different resolutions. Most of these errors occur, when the structure of the background has a high similarity to the head appearance.

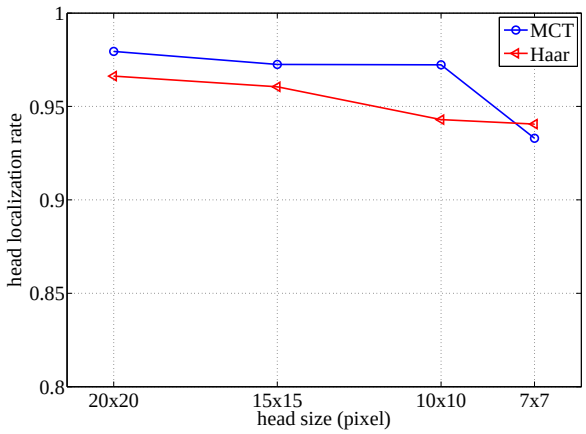


Figure 3.36: Correct head localization rates for different pedestrian heights in image pixels. The corresponding head sizes cover a range from 20×20 pixels to 7×7 pixels.

Figure 3.37: Samples at different image resolutions, where the head localization for the single-frame based method (Sec. 3.1) using MCT-based features fails.

The correct estimation of the head pose is a more difficult task than the localization itself. As the upper part of a pedestrian detection is scanned with eight different head pose classifiers, it is likely, that at least one of those results in a very high confidence at the correct head position. For head pose estimation the head pose class related to that particular classifier would be outputted. However this could probably be a fail decision concerning the difficulties of the fuzzy multi-class problem on low resolution images (Sec. 3.1.4). To get an idea about inter-class decision errors, results are therefore represented using confusion matrices as proposed in [Enzweiler and Gavrila, 2010, Orozco et al., 2009] and [Siriteerakul et al., 2010]. Here, rows correspond to the ground truth head pose class, while columns outline the system's predicted head pose. Higher values concentrated along the diagonals are related to a better performance for a multi-class classification algorithm. Figure 3.38 shows the confusion matrices for the head pose estimation system when integrating MCT- or Haar-like boosting cascade classifiers at different resolutions. All configurations achieve satisfying correct decision rates, even when evaluation is performed on very low resolution images (7×7 pixel head size). The MCT-based approach reaches an overall correct decision rate over 60% for head sizes of 20×20 pixels, where the use of Haar-like features achieves 49%. The best accuracy up to 76% is reached by the left and right head pose MCT-classifiers. For small head sizes at 7×7 pixels, the overall performance breaks down to a correct decision rate of 42% for the MCT-classifier and 44% for the classifier based on Haar-like features. The highest confusion between neighboring classes happens for the frontal MCT head pose

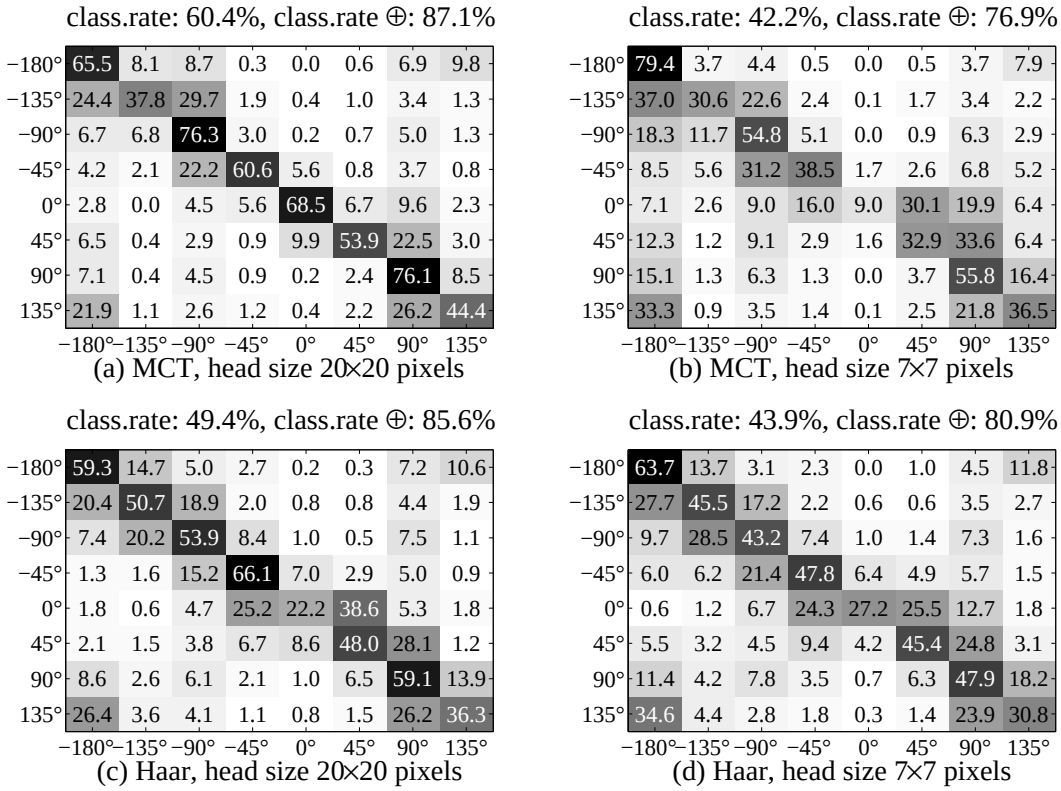


Figure 3.38: Confusion Matrices for MCT- ((a), (b)) and Haar-classifiers ((c), (d)) for head sizes of 20×20 pixels and 7×7 pixels.

classifier at 7×7pixels, a potential result from the fuzzy head pose labeling problem. It can be noticed that the MCT-classifiers score higher correct decision rates for higher resolutions when compared with the classifiers based on Haar-like features. The use of Haar-like features results in a slightly better performance only on the lowest considered image resolution. This can be explained, similar to the head localization results, by the nature of MCT-based features that rely on the pixel level structures, which may disappear in low resolution images. Haar-like features measure intensity differences between larger areas and are thus more robust against the low resolution problem. Figure 3.38 also shows, that most of the confusions of one particular head pose occur with the direct neighboring head poses. Therefore, similar to Section 3.4.2, in another evaluation round, a head pose estimation is also considered to be correct, if the predicted head pose class is identical to the true head pose or one of its direct neighboring head pose classes (classification rate $\oplus$). As a result, it is possible to get correct decision rates at a minimum of nearly 77% for 7×7 pixels head images and a maximum of 87% for head sizes of 20×20 pixels using MCT-based features. In the case of the actual eight-class problem, the classifiers using Haar-like features tend to confuse more between the neighboring poses than the classifiers based on the MCT. However, when dealing with modified eight-class problem, they seem to slightly outperform the MCT-classifiers, especially on the lowest resolution. I.e., MCT-based classifiers have a better class separation property compared to Haar-like-based classifiers.

(a) (b)

Figure 3.39: Correct results using MCT features for pedestrian heights of 140 pixels (a) and 50 pixels (b). White stripes indicate the pedestrians' head pose direction.

Figure 3.40: Samples of wrongly estimated head poses using MCT features at different image resolutions.

Figure 3.39 displays samples of correct head pose estimation results for heads of 20×20 and 7×7 pixels size with the MCT-based approach. To get an impression of the complexity of head pose estimation in very low resolution images samples of wrongly estimated head orientations are shown in Figure 3.40. The approach using Haar-like features results in a similar output. As the local structure features based on MCT slightly outperform Haar-like features in terms of class separation ability, in the following, further experiments will be presented using MCT-cascades only.

3.4.4 Head Pose Tracking Performance in Video-Sequences

In this section, the proposed method for head pose tracking in video-sequences (Sec. 3.2) is evaluated. Evaluation results are presented in details for every evaluation dataset presented in Section 3.3. Compared to the initial experiments for the single-frame based approach in Section 3.4.3, person image candidates are directly used for evaluation instead of up- or downscaling them to certain sizes.

Head Pose Tracking Evaluation on the CLEAR Dataset

The easiest dataset among all others presented in this work is assumed to be the CLEAR data (Sec. 3.3.1) as it contains images where person heads appear at sizes not smaller than 34×34 pixels. After setting up the trained head pose classifiers and system parameters (Sec. 3.4.1)

Person Dataset	1	2	3	4	5	Overall
Single-frame based approach	100.0	100.0	100.0	98.9	99.6	99.7
Tracking approach	100.0	100.0	100.0	100.0	100.0	100.0

Table 3.17: Head localization rates (%) for the CLEAR dataset including test persons 1 to 5

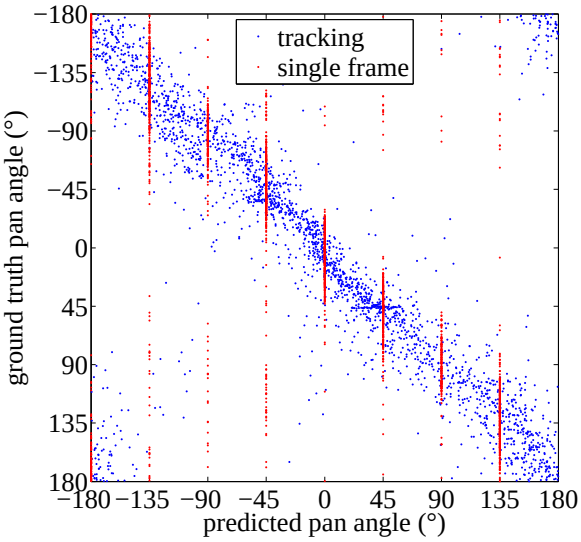
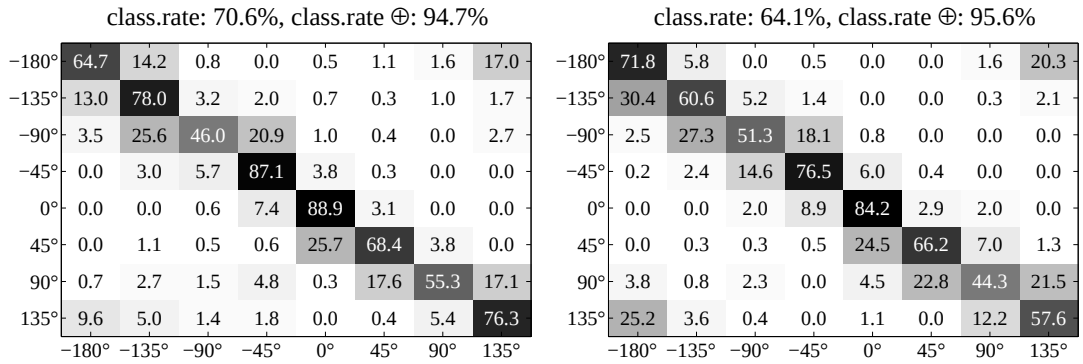


Figure 3.41: Distribution over predicted vs. ground truth head pan angles for the CLEAR dataset. Red data points: Estimates for single-frame based approach. Blue data points: Estimates for tracking approach.

for the CLEAR training data, evaluation for the overall system is performed on the CLEAR evaluation data. First, it is important to localize the head correctly in order to output a meaningful head pose estimation. For this, Table 3.17 shows the head localization rates on the test sets. Additionally, results are compared with the single frame based approach from Section 3.1. Both methods achieve high localization rates up to nearly 100%, which is not surprising as the head search area is already restricted by the person bounding box in addition to higher resolution images with homogeneous background. Through particle filtering, the method's output of the developed method is a continuous head pan angle estimate in comparison to only one of eight head pose classes given in Section 3.1 and [Orozco et al., 2009, Siriteerakul et al., 2010]. Figure 3.41 displays the distribution of discrete (single-frame based) and continuous (tracking approach) head pan angle estimates for the ground truth head pan angles given by the CLEAR evaluation datasets. Most of the estimates for both approaches are concentrated around the diagonal (ground truth angle equals predicted angle), which is the desired behavior. Having a look at the estimates for the single-frame based approach, one can get information about which continuous head pan angle range is mapped to a certain discrete head pose class. Qualitatively, the single-frame based approach is showing more outlier estimates compared to the tracking approach. For performance comparison using confusion matrices, the corresponding discrete head pose class is determined given the



(a) Single-frame based approach (b) Tracking approach

Figure 3.42: Confusion matrices for the CLEAR dataset for (a) single frame based approach (Sec. 3.1) and (b) using particle filtering (Sec. 3.2).

Person Dataset	1	2	3	4	5	Overall
Single-frame based approach	36.1	28.3	16.1	15.6	19.9	23.2
Tracking approach	29.0	20.5	16.7	11.7	14.1	**18.4**

Table 3.18: Mean pan angle error (deg.) for the CLEAR person datasets 1 to 5. Smallest head size: 34×34 pixels.

continuous angle estimate from the tracking approach. Depending on the later application within driver assistance system, this arrangement could already be sufficient. Figure 3.42 shows overall confusion matrices averaged over all person test sets for the tracking method compared to the single-frame based approach. Further, it is possible to compare both methods by means of head pose estimation accuracy, taking the head pan angle corresponding to the discrete output head pose class in case of the single frame based approach. A suitable performance measure is the mean absolute head pan angle error compared to the ground truth measurement. Table 3.18 shows the results. For both, single-frame-based and tracking approach, overall mean angular errors lie around 20 degrees. Tracking results in little less errors (18.4°) compared to evaluation on single images (23.2°). Nevertheless, the single-frame based method with discrete head pan angle estimates still results in a high accuracy. The highest sequence related mean angular error however is given by the single-frame based approach with about 36° for person 1. The main problems in correct head pose estimation occur, when persons are extremely moving their heads in vertical direction, i.e., having a high head tilt. As head pose detectors are trained to only capture head pan movements, the proposed method will obviously have problems here.

In addition, Figure 3.43 gives an overview about the distribution of angular errors when using the single-frame based and particle filtering approach, respectively. The particle filtering approach (blue) for head pose estimation results in a head pan angle error within the range of (-22.5°,22.5°) for a higher number of evaluated samples compared to the single-frame based approach (red).

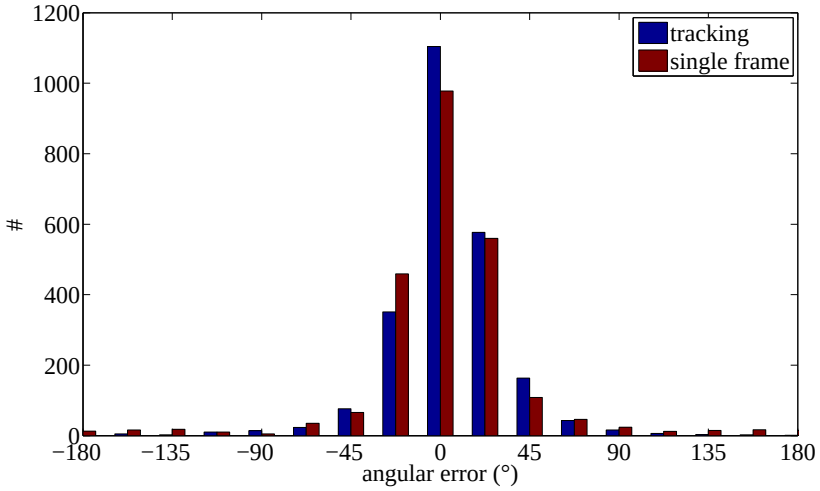


Figure 3.43: Distribution of angular errors for the CLEAR dataset

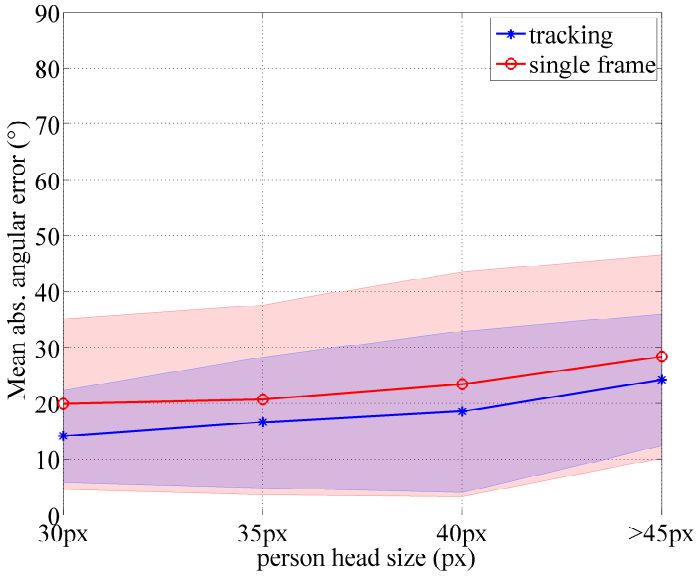


Figure 3.44: Mean absolute angular error (thick line) and its standard deviation (shaded area) over various head sizes for the CLEAR dataset.

For later application, it is worth to know, at which relative distances between pedestrian an oncoming vehicle, the head pose estimation system still results in sufficiently accurate estimates. This is directly dependent on the minimum head size the system can handle. Figure 3.44 therefore shows the mean angular errors and its standard deviation for ascending head sizes. In general, for the CLEAR dataset persons turn around a fixed point within a smart room. Hence, different head sizes in the images are mainly influenced by the persons hair structure. I.e., persons having long and thick hair result in larger head image areas than persons wearing short hair. Given this fact, the slightly decreasing accuracy in head pose estimation for increasing head sizes from 30×30 to 45×45 pixels can be explained. The tracking approach results in less angular errors of approximately 5 degrees on average compared to single-frame based evaluation. Considering the standard deviation, tracking yields

Figure 3.45: Exemplary head pose estimation results on the CLEAR dataset.

less dispersion in estimates and thus more stable results. Figure 3.45 shows some head pose estimation output images for the CLEAR dataset.
In the following paragraphs, experiments on more application related datasets will be shown, where heads appear on very low resolution.

Head Pose Tracking Evaluation on the CAVIAR Dataset

A public available dataset, which includes conditions similar those present in driver assistance scenarios, is given by the CAVIAR dataset (Sec. 3.3.2), as person heads appear on very low resolution. Evaluation is performed with the CAVIAR system setup explained in Section 3.4.1 for the single-frame based (Sec. 3.1) and tracking approach (Sec. 3.2) separately. For person distance- or head image size-dependent evaluation, the following critical areas are defined. The *far* range, capturing persons up to a height of 50 pixels and below, the *middle* range, including persons of heights between 50 and 100 pixels and the *near* range for pedestrians taller than 100 pixels. First, the head localization performance is investigated. Dependent on the introduced detection areas, head localization rates are displayed in Table 3.19.

Range	near	middle	far	overall
Single-frame based approach	98.1	93.8	90.1	94.0
Tracking approach	100.0	98.6	94.5	97.7

Table 3.19: Correct head localization rates (%) for different ranges and overall.

For tracking, little higher localization rates are achieved throughout the detection areas. Nevertheless, both single-frame based and tracking based approach achieve satisfying results. For quantitative evaluation of head pose estimation performance, the columns of Table 3.20 show combined results in head pose estimation by means of mean pan angle error and classification rate.
In general, performance decreases drastically compared to the results on the CLEAR data because of head appearances in very low resolution and the generalization problem which

	mean pan angle error (°)	classification rate (%)
Single-frame based approach	45.3	37.2
Tracking approach	38.7	41.6

Table 3.20: Mean head pan angle error and correct classification rates on the CAVIAR dataset. Smallest head size: 6×6 pixels.

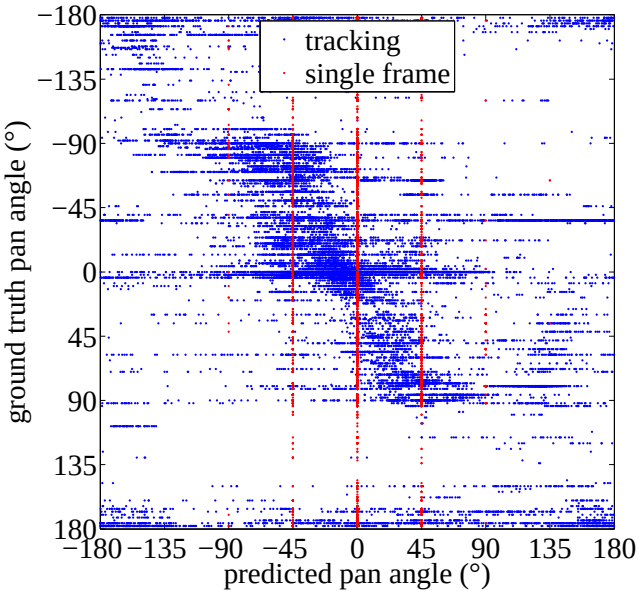
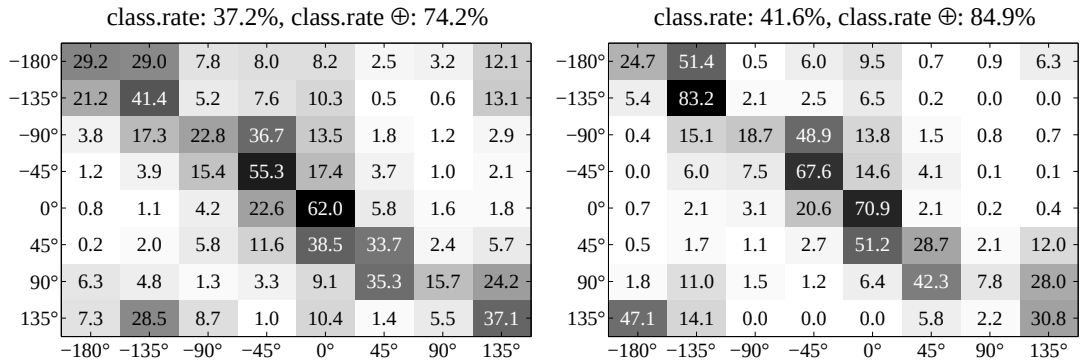


Figure 3.46: Distribution over predicted vs. ground truth head pan angles for the CAVIAR dataset. Red data points: Estimates for single-frame based approach. Blue data points: Continuous estimates for tracking approach.

originates from training the head pose classifiers on CLEAR data only. The benefit of the proposed tracking method is observable. Higher accuracy in head pose estimation is achieved having a lower mean angular error of approximately 39° compared to 45° for the single-frame based approach. Figure 3.46 displays the distribution of discrete (single-frame based) and continuous (tracking approach) head pan angle estimates for the ground truth head pan angles provided by the CAVIAR dataset. Still, there is a concentration of estimated head pan angles around the diagonal at least for the tracking approach. The single-frame based method results in unstable estimates for the range of -45° to 45°. To get conclusion, where most of the pan angle errors occur and how the different pose classes get mixed up by the tracking method, confusion matrices are used similar to the CLEAR data evaluation. Therefore, the discrete head pose class is determined from a given continuous pan angle estimate for the tracking approach. Again, rows correspond to the ground truth, while columns outline the system's predicted head pose class. Figure 3.47 shows the results. Most confusion occurs with direct neighboring head pose classes except of *Back*-views (-180°), that are sometimes mixed up with frontal views in 9.5% of the test samples in case of tracking. Here a prior probability distribution for the head pose based on the actual pedestrian body pose can avoid



(a) Single-frame based approach (b) Tracking approach

Figure 3.47: Head pose estimation performance for the CAVIAR dataset. Single frame approach (a) and head pose tracking method (b).

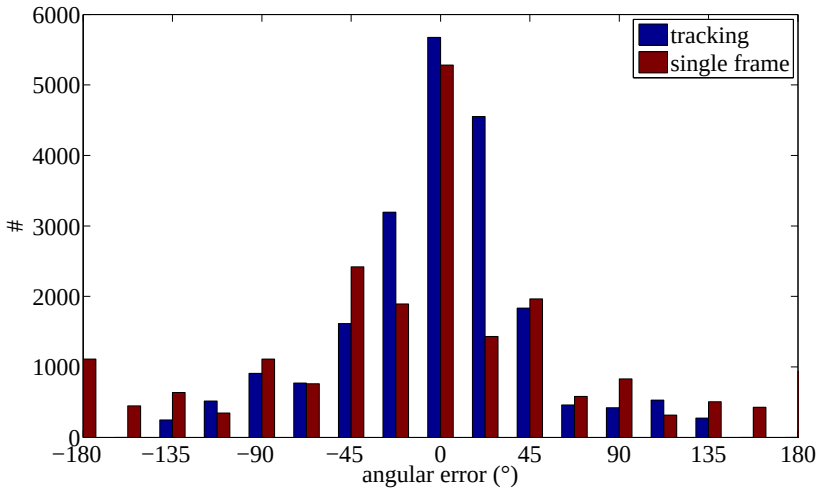


Figure 3.48: Distribution of angular errors in head pose estimation for the CAVIAR dataset

such confusions (cf. Sec. 4.2). To get the overall correct classification rate, the diagonal elements of each matrix are summed up and divided by the number of discrete head pose classes. These rates are included in the right column of Table 3.20. Again, the method using particle filtering achieves higher correct classification rates (41.6%) compared to the single frame based method (37.2%). When considering also estimates for neighboring classes to be correct for a given ground truth class (classification rate $\oplus$), the benefit of tracking is visible even clearer. It achieves overall 10% higher classification rates than the single-frame based method. Figure 3.48 displays the distribution of head pose estimation errors by means of mean angular error among all evaluated samples. Again, the benefit when using the tracking approach is clearly visible. More evaluated samples (blue) are concentrated around segments with lower angular errors compared to the method working on single frames only (red).
For later application, it is worth to know, at which relative distances between pedestrian an oncoming vehicle, the head pose estimation system still results in sufficiently accurate estimates. This is directly dependent of the minimum head size the system can handle. A head sample division in 5 pixels steps is performed for evaluation ranging from head sizes around

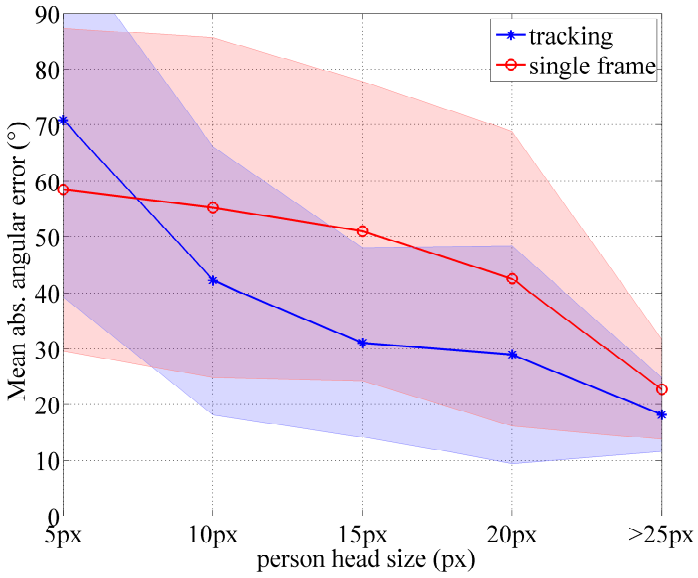


Figure 3.49: Mean absolute angular error (thick lines) and its standard deviation (shaded area) over ascending head sizes for the CAVIAR dataset.

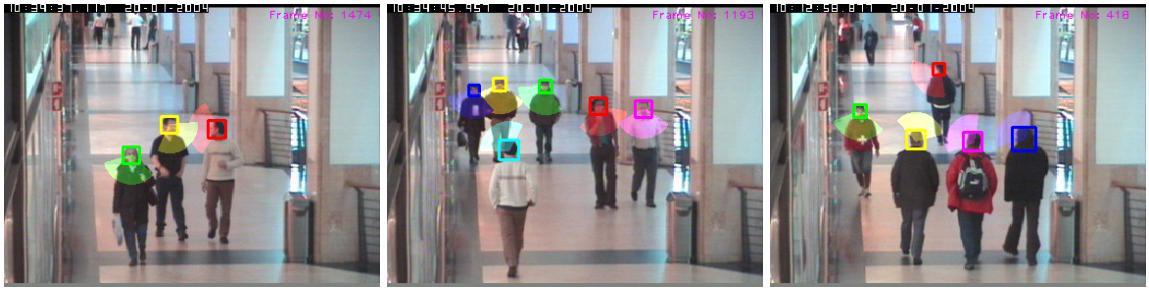


Figure 3.50: Exemplary head pose estimation results on the CAVIAR dataset.

5×5 pixels up to head sizes larger than 25×25 pixels. Figure 3.49 shows the mean absolute angular errors and its standard deviation for ascending head sizes. Head pose estimation performance decreases when evaluating smaller head samples, which is the expected behavior. Mean absolute angular errors up to 70° are observed for the tracking approach for head sizes around 5×5 pixels. Nevertheless, the averaged angular errors and its deviation significantly lies below the ones for the single frame based approach. Figure 3.50 shows some head pose estimation output images for the CAVIAR dataset.
The next section will focus on the evaluation of inner-city data from driver assistance scenarios.

Head Pose Tracking Evaluation on the Bosch Inner-City Dataset

As it is hard to get labeled data with accurate ground truth head pan angle measurements for inner-city scenarios, head pose estimation performance evaluation is restricted. Capturing and annotating at least eight discrete head pose classes results in the potential use of confusion matrices only. In later applications one is most interested in recognition of the two cases.

Figure 3.51: Pedestrian head pose estimation results in inner-city scenarios. Color gradient goes from green (frontal pose, i.e., less critical situation) to red (back poses, i.e., dangerous situation).

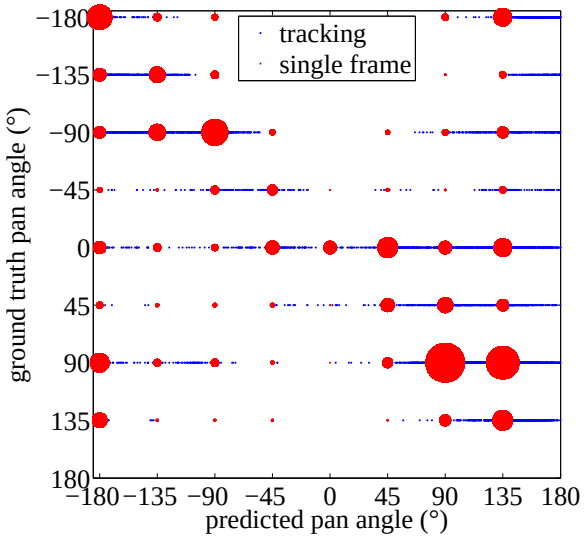
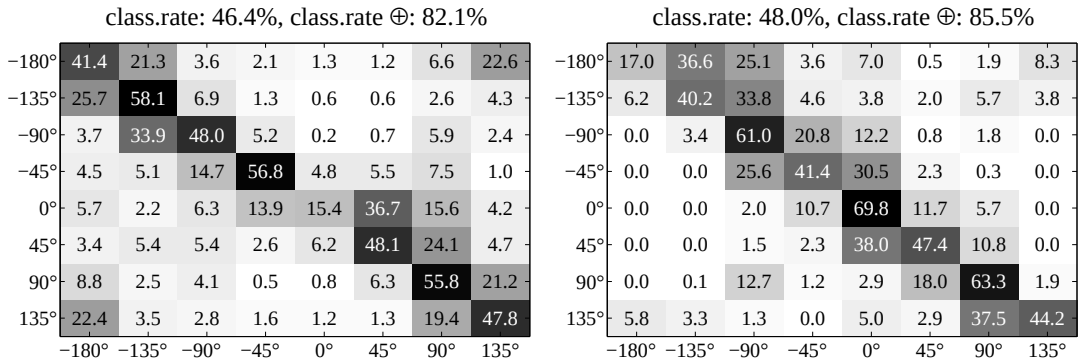


Figure 3.52: Distribution over predicted vs. ground truth head pan angles for the Bosch Inner-City dataset. Red data points: Discrete estimates for single-frame based approach. Blue data points: Continuous estimates for tracking approach

Is a pedestrian facing the approaching vehicle (the camera) and thus showing nearly a frontal head pose, or is he/she looking to another direction, i.e., having head poses from right over back to left. Hence, the use of eight head pose classes is sufficient here. Nevertheless, results are presented by means of calculated averaged angular errors keeping in mind the inaccurate ground truth head pan angle annotations. In Figure 3.51 some output images for head pose estimation are shown. Figure 3.52 displays the distribution of discrete (single-frame based) and continuous (tracking approach) head pan angle estimates for the ground truth head pan angles provided by the Bosch Inner-City dataset. As ground truth labels and single-frame based estimates are discrete (one of eight head pose classes), data points are displayed at different sizes reflecting the number of head pose estimates related to a certain head pose class given the ground truth class. There is a concentration of estimated head pan angles around



(a) Single-frame based approach (b) Tracking approach

Figure 3.53: Confusion matrices for the Bosch Inner-City dataset. Single frame based approach (a) and head pose tracking (b).

	head localization rate (%)	**classification rate (%)**	**classification rate ⊕(%)**
single-frame based approach	86.4	46.4	82.1
tracking approach	92.6	48.0	85.5

Table 3.21: Head localization rates and correct classification rate for the Bosch Inner-City dataset (left and middle column). Right column: Correct classification rate including direct neighboring pose predictions. Smallest head size: 6×6 pixels.

the diagonal at least for both, tracking and single-frame based approach. The corresponding quantitative head pose estimation results comparing the single-frame based- (Sec. 3.1) and tracking approach (Sec. 3.2) are given by the confusion matrices in Figure 3.53 and the performance measures in Table 3.21. Tracking improves head localization rates (92.6% compared to 86.4%) and outperforms the single frame based approach achieving an overall correct classification rate of 48.0% compared to 46.4%. One has to keep in mind, that more correct localized head samples are considered for classification rate calculation in the tracking case. Having a look at the values for classification rate ⊕ for both methods, the confusion with direct neighboring head pose classes compared to all other classes is little higher for the tracking method than for the single-frame based approach meaning a higher class stability. Figure 3.54 displays the distribution of head pose estimation errors among all evaluated samples. Again, the benefit when using the tracking approach is clearly visible. More evaluated samples (blue) are concentrated around segments with lower angular errors compared to the method working on single frames only (red). The maximum possible angular error of -180° is only occurring for the single-frame based method. Similar to the CAVIAR data, for head size dependent evaluation, a head sample division in 10 pixels steps is performed ranging from head sizes around 5×5 pixels up to head sizes larger than 55×55 pixels. Figure 3.55 shows the mean absolute angular errors and its standard deviation for ascending head sizes. Head pose estimation performance decreases when evaluating smaller head samples, which is the expected behavior. Mean absolute angular errors up to 76° and 67° are observed for

single frame based- and the tracking approach respectively, for head sizes around 5×5 pixels. For higher resolutions, errors lie at 30° and below for both approaches however, the tracking approach produces more stable estimates by means of a lower standard deviation for the angular error distribution. It has to be mentioned, that an angular error below 22.5° should be treated with caution as the labels are only given with a precision up to 45°. Given the camera constant f to be 1200 and assuming a average human head height of H=25cm, cf. [Young, 1993] the distance x of a pedestrian head with respect to the camera can be calculated using the head height h within the image as $x = fH/h$. A 15×15 pixel head appearance in the image then corresponds to a pedestrian distance of 20m. According to anatomy, the human eye field of view approximately is of 120°. Hence, an average angular error of less than 40° at a head size of 15×15 (Fig. 3.55) pixels is sufficient to realize a system for risk assessment where one needs to know at an early stage, if a pedestrian is recognizing an approaching vehicle or not.

3.5 Conclusion

In this chapter, a method was presented to estimate the head pose of pedestrians in video sequences. Given a pedestrian hypothesis as the outcome of a video-based pedestrian detection system, the method first initializes the position and the pose of the pedestrian head (Sec. 3.1) and further tracks the head pose over time using particle filtering (Sec. 3.2). In the experiments (Sec. 3.4) a detailed performance analysis on various datasets (CLEAR, CAVIAR and Bosch Inner-City) showed a good compromise between accuracy and runtime compared to state of the art approaches while having the advantage of a combined head localization and head pose estimation compared to the requirement of having a segmented head region resulting from a pre-processing step. It was discussed that an average angular error of less than 40° at small head sizes of 15×15 pixels shows sufficient performance in head pose estimation for a later application in risk assessment.

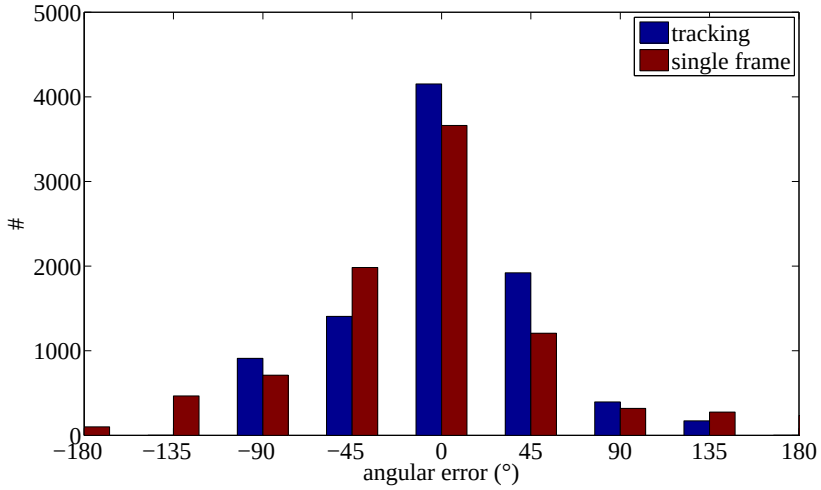


Figure 3.54: Distribution of angular errors for the Bosch Inner-City dataset.

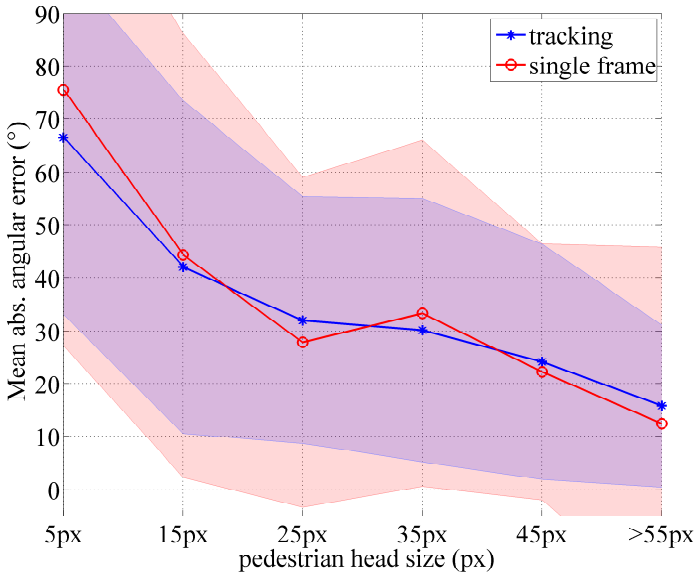


Figure 3.55: Mean absolute angular errors (thick lines) and its standard deviation (shaded area) over ascending head sizes for the Bosch Inner-City dataset.

The presented method can be used as an "on-top" component for a given overall pedestrian detection system, where pedestrians are detected and reliably tracked over time.

4 Robust 3D Head pose estimation using stereo vision

The overall developed system in this work is based on the input data from a stereo-video system for pedestrian detection and tracking (cf. Sec. 2.2). The image areas for detected pedestrian candidates are scanned with the method described in Chapter 3 in order to detect pedestrian heads and extract their corresponding head poses. The head detection or localization already achieves satisfying results. Nevertheless, there is still potential for improvement, especially at very low image resolutions. Here, the depth information, which can be gained from a stereo-vision sensor, shows high potential to excellently contribute for a more robust localization of pedestrian heads.
Within a following function for automatic emergency braking on pedestrians, all relevant object attributes such as relative position, velocity, etc. are specified with respect to a fixed world coordinate system, which has its origin at the middle of the vehicle's rear axle. This has the advantage that further independent sensor measurements and features can be taken into account for fusion in order to achieve a higher function robustification. At this point, estimating the pedestrian head pose relative to that coordinate system states a huge benefit.
Another important source of information resulting from a stereo-video sensor are given by the object candidate's derived velocity estimates. These can have direct influence on the following head pose estimation method in order to improve the same. The viewing and moving direction of a pedestrian, that is moving with a certain speed, will most probably coincide.
This chapter combines further improvements for the existing head pose estimation system presented in Chapter 3 using the potential benefits of a stereo-video sensor mentioned before. First, the transformation of estimated head poses with respect to 2D image coordinates into a 3D vehicle located coordinate system is explained in Section 4.1. Section 4.2 deals with the robustification for head pose estimation by incorporating the pedestrian's moving direction. A main component of this chapter is the improved head localization using stereo depth information presented in Section 4.3 followed by experiments in Section 4.5. Finally, this chapter concludes with Section 4.6.

4.1 Stereo Depth Information for 3D Head Pose Estimation

In the previous chapter a method for 2D head pose estimation on low resolution images was presented. In particular, a pedestrian's head pan angle with respect to the person's head 2D image appearance was determined using the output of a head pose detector array. For later application in risk assessment however, it is of a great value to transform the estimated 2D head pan angles into a global 3D coordinate system (e.g. defined by DIN70000 [DIN ISO 8855:2013-11, 2013]) to better cooperate with other system components. One way to achieve a 3D head pan angle representation is to use basic principles of projective geometry and stereo vision. As the final system will be based on stereo vision, this choice is intuitive. The presented algorithm builds upon pedestrian candidates detected by a stereo video system. Thus, accurate pedestrian 3D distance measurements can be obtained. In this work, the 3D head pose estimates are given with respect to a vehicle coordinate system with origin located at the left camera's center of the stereo vision system. The z-axis is pointing in longitudinal direction, the x-axis in right lateral direction and the y-axis vertically downwards as shown in Figure 4.1.

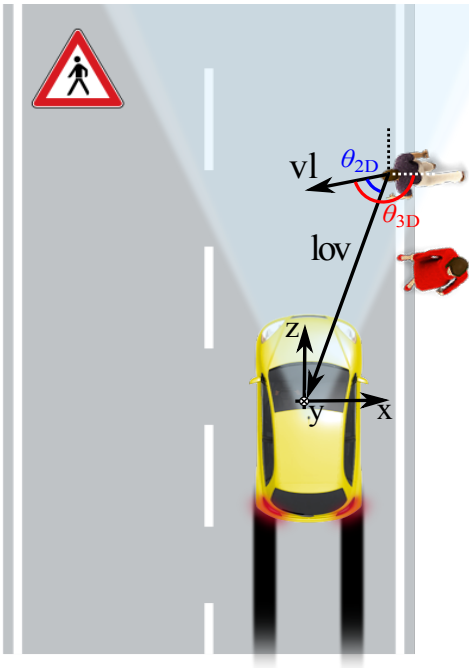


Figure 4.1: 3D head pan angle estimation with respect to vehicle located coordinate system. By calculating the line of view lov using stereo disparity estimates, the 3D head pan angle θ_{3D} can be obtained from the 2D head pose estimate θ_{2D}.

Using this setup, estimated 3D measurements can be transformed easily to any other 3D coordinate system (e.g. DIN70000) using simple mathematical rotations and translations. In the canonical stereo configuration, the following relation between a 3D point (x, y, z) and the

corresponding point in the image (u, v) holds, when considering homogeneous coordinates.

$$z \begin{pmatrix} u \\ v \\ d \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 0 & fB \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \tag{4.1}$$

with principal point (u_0, v_0), camera constant f, stereo base width B and estimated disparity d. Hence, a 3D point can be calculated from image coordinates and disparity by

$$x = (u - u_0)\frac{B}{d},\ y = (v - v_0)\frac{B}{d},\ z = f\frac{B}{d}. \tag{4.2}$$

First, a 3D line of view ($\mathbf{lov}$) pointing from the 3D head center H_c^{3D} to the left camera will be calculated given the median disparity over the pedestrian's head location in the image and the 2D head center H_c^{2D} using equation (4.2). The observed pedestrian's 2D head pan angle is then the difference angle between the vector pointing from the person's head center to the camera center ($\mathbf{lov}$) and the projected person's 3D view-line $\mathbf{vl}$. For a given 3D vector $\mathbf{v} = (v_x, v_y, v_z)'$ the corresponding pan angle θ can be calculated as

$$\theta_{\mathbf{v}} = \angle(\mathbf{v}) := \frac{180^\circ}{\pi} \arctan2(v_y, v_x). \tag{4.3}$$

The pan angle $\theta_{\mathbf{v},\mathbf{w}}$ between two 3D vectors $\mathbf{v}$, $\mathbf{w}$ restricted to the interval $[-180^\circ, 180^\circ)$ is determined by

$$\begin{aligned} \theta_{\mathbf{v},\mathbf{w}} &= \angle(\mathbf{v}; \mathbf{w}) := \theta_{\mathbf{v}} - \theta_{\mathbf{w}}, \\ \theta_{\mathbf{v},\mathbf{w}} &= \begin{cases} \theta_{\mathbf{v},\mathbf{w}} - 360^\circ, & \text{if } \theta_{\mathbf{v},\mathbf{w}} \geq 180^\circ \\ \theta_{\mathbf{v},\mathbf{w}} + 360^\circ, & \text{if } \theta_{\mathbf{v},\mathbf{w}} < -180^\circ \end{cases} \end{aligned} \tag{4.4}$$

with $\theta_{\mathbf{v}}$ and $\theta_{\mathbf{w}}$ calculated using equation (4.3). Now, the difference angle $\theta_{\mathbf{lov},x}$ between the camera's line of view ($\mathbf{lov}$) and the x-axis $(1, 0, 0)'$ is calculated using equation (4.4), i.e.,

$$\theta_{\mathbf{lov},x} = \angle(\mathbf{lov}; (1, 0, 0)') \tag{4.5}$$

Finally, the 3D head pan angle θ^{3D} is derived from the sum of $\theta_{\mathbf{lov},x}$ and the 2D head pan angle estimate θ^{2D}, that is

$$\theta^{3D} = \theta^{2D} + \theta_{\mathbf{lov},x}, \tag{4.6}$$

see Figure 4.1. In the following, the particle filter head pose estimates are given with respect to the defined fixed 3D coordinate system in Figuer 4.1. Internally, estimated 2D head pose classes are transformed to 3D for the particle weight update steps, see equation (3.40).

4.2 Combining Head Pose Estimation with Human Body Motion

[Benfold and Reid, 2009, Robertson and Reid, 2006] and [Chen et al., 2011] show, that incorporating the human moving direction or body pose as a prior for head pose estimation significantly improves the overall system performance. As pedestrian object hypotheses taken from a stereo-video system serve as input for the developed pedestrian head pose estimation approach, one directly obtains estimates for object movement direction and speed with respect to a fixed vehicle located world coordinated system. With the transformations presented in Section 4.1, the pedestrian motion estimates can directly be integrated into the state propagation model of the particle filter for a controlled head pose state prediction. In Chapter 3 a Bayes filter is used in order to achieve a smooth and stable head pose estimation over time. Now, the additional variable $\mathbf{v}_k$ is introduced representing the pedestrian's absolute velocity at time instance k. Then, a reformulation for the prediction and update steps of the filtering problem is given by

1. Prediction (prior):

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}, \mathbf{v}_{1:k}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{v}_k) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}, \mathbf{v}_{1:k-1}) d\mathbf{x}_{k-1} \text{ and} \tag{4.7}$$

2. Update (posterior):

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}, \mathbf{v}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:t-1}, \mathbf{v}_{1:k}). \tag{4.8}$$

Again, the state space vector $\mathbf{x}_k$ includes the filtered estimates for image position and scale of the pedestrian's head as well as the head pose, which is now given with respect to the vehicle coordinate system, i.e., $\mathbf{x_k} = (u_k, v_k, s_k, \theta_k^{\text{3D}})'$. In the following course of this work, head pose estimates always refer to the 3D vehicle coordinate system and the additional identifier is omitted. The first and second term under the integral in equation (4.7) again represent the dynamical model and the posterior probability for the previous time instance. The first term on the right hand side in equation (4.8) corresponds to the measurement model. Under the assumption of conditional independence, the dynamical model in equation (4.7) can be noted down as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{v}_k) &= p(\tilde{\mathbf{x}}_k|\tilde{\mathbf{x}}_{k-1}, \theta_{k-1}, \mathbf{v}_k) p(\theta_k|\tilde{\mathbf{x}}_{k-1}, \theta_{k-1}, \mathbf{v}_k) & (4.9) \\ &= p(\tilde{\mathbf{x}}_k|\tilde{\mathbf{x}}_{k-1}) p(\theta_k|\theta_{k-1}, \mathbf{v}_k), & (4.10) \end{aligned}$$

where $\tilde{\mathbf{x}}_k := (u_k, v_k, s_k)'$ includes the filtered head's image position and scale. Therefore, particle states are propagated for every time instance depending on the previous state estimate and the actual pedestrian's movement direction. The first term of the updated evolution

model can be described similar to equation (3.36) that is

$$p(\tilde{\mathbf{x}}_k|\tilde{\mathbf{x}}_{k-1}) \sim \mathcal{N}\left(\tilde{\mathbf{x}}_{k-1}, \begin{pmatrix} \sigma_u & 0 & 0 \\ 0 & \sigma_v & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}\right). \tag{4.11}$$

For the second term in equation (4.10) a formulation incorporating the von Mises distribution similar to [Chen et al., 2011] is chosen, i.e.,

$$p(\theta_k|\theta_{k-1}, \mathbf{v}_k) = \alpha_\theta \mathcal{M}(\theta_k; \theta_{k-1}, \kappa_\theta) + (1 - \alpha_\theta)\mathcal{M}(\theta_k; \angle(\mathbf{v}_k), \kappa_{\mathbf{v}_k}), \tag{4.12}$$

with the concentration parameters κ_θ and $\kappa_{\mathbf{v}_k}$ and the weighting factor $\alpha_\theta \in [0, 1]$. The function $\angle(\cdot)$ thereby outputs the angle of the person's velocity vector (moving direction) with respect to the vehicle coordinate system. The first term in equation (4.12) models the property, that the person's head pose will not change significantly with respect to the estimated head pose for the previous time instance. The second term models the anatomical limitation that pedestrians are normally looking in the direction of their movement. While α_θ and κ_θ get assigned constant values all over time, which are estimated based on a validation dataset, the parameter $\kappa_{\mathbf{v}_k}$ is dependent from the actual absolute value of the velocity signal, i.e.,

$$\kappa_{\mathbf{v}_k} = \begin{cases} \kappa_{\mathbf{v}}(\|\mathbf{v}_k\| - \xi)^2, & \text{if } \|\mathbf{v}_k\| > \xi \\ 0, & \text{otherwise} \end{cases}, \tag{4.13}$$

with the constant sharpening parameter $\kappa_{\mathbf{v}}$, see [Chen et al., 2011]. From equation (4.13) it follows, that only velocity measurements having an absolute value greater than ξ will have impact on the prediction step. The measurement model remains unchanged compared to Section 3.2.4 meaning that single particle weights are essentially determined by the confidence value outputs of head pose classifiers.

4.3 Head Localization using Stereo Depth Information

The presented system for pedestrian head pose estimation will later be embedded into an overall pedestrian protection system in order to predict future intentions of detected and tracked pedestrians. This system builds upon a stereo camera system for robust object localization and tracking in 3D. As there is already a stereo depth map available, in this section the possibilities to further increase the robustness for pedestrian head localization using the advantages of stereo vision is presented. For head pose estimation problems the use of human head depth profiles was investigated using a stereo camera system, cf. [Seemann et al., 2004], or Microsoft Kinnect, cf. [Bär et al., 2012, Fanelli et al., 2011, Martin et al., 2014]. Unfortunately, this requires head images on very high resolutions and is therefore not applicable for the task of pedestrian head pose estimation, presented here. However, for head

localization it is shown, how stereo vision can help in increasing the performance of the presented system. Considering previous experiments from Chapter 3, main problems in correct head localization occur, when there is a highly structured background, e.g. persons walking in front of buildings, trees etc. A reliable segmentation for the head search area into fore- and background regions could possibly avoid these miss-localizations.

4.3.1 Head Extraction based on u- and v-Disparity

For object detection tasks using a stereo camera system the use of line histogram representations of the calculated disparity map is proposed, see [Labayrade et al., 2002]. First, to reduce the dimensionality of data and second to easily extract raised obstacles. For the scenario of this work, this approach is applied restricted to the head search area in order to segment the pedestrian head from the background.

The u- and v-Disparity Images

It is assumed, that a disparity map D has been computed from a stereo image pair. The image V that is generated by accumulating disparity values within a histogram for each given image row v of the disparity map D is called v-*disparity* image. For a given image row v of V, the horizontal coordinate $u_{\mathbf{p}}$ of a point $\mathbf{p} := (u_{\mathbf{p}}, v_{\mathbf{p}})$ in V corresponds to the disparity $d_{\mathbf{p}}$ and its gray value $i_{\mathbf{p}}$ to the number of points with the same disparity $d_{\mathbf{p}}$ on the row v, that is

$$i_{\mathbf{p}} = \sum_{\mathbf{q} \in D} \delta_{v_{\mathbf{q}}, v_{\mathbf{p}}} \delta_{d_{\mathbf{q}}, d_{\mathbf{p}}}, \mathbf{p} \in V \tag{4.14}$$

where $\delta_{i,j}$ denotes the Kronecker delta. Once D has been computed, V is built by accumulating the pixels of same disparity in D along the v-axis. The same operations can be performed for accumulation of disparity values over the u-axis resulting in the so-called u-*disparity* image U. Here, the vertical coordinate $v_{\mathbf{p}}$ of a point $\mathbf{p} := (u_{\mathbf{p}}, v_{\mathbf{p}})$ in U corresponds to the disparity and its gray value $i_{\mathbf{p}}$ to the number of points with the same disparity $d_{\mathbf{p}}$ on the column u, that is

$$i_{\mathbf{p}} = \sum_{\mathbf{q} \in D} \delta_{u_{\mathbf{q}}, u_{\mathbf{p}}} \delta_{d_{\mathbf{q}}, d_{\mathbf{p}}}, \mathbf{p} \in U \tag{4.15}$$

An example of calculated u- and v-disparity images is given in Figure 4.2.

Head Extraction using Dominant Disparities

In the following, the calculated u- and v-disparity images build the basis for a further head segmentation in u- and v-direction. In a preprocessing step, the head search area is defined for a given pedestrian candidate identical to Section 3.1.5. The stereo-based head localization can be divided into three major parts each of which is described in abstract algorithms. First, a search for row- and column-wise foreground dominant disparities is performed using

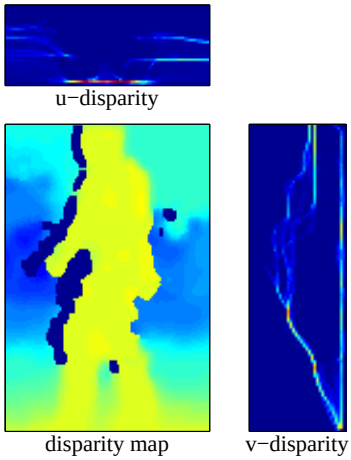


Figure 4.2: Calculated u- and v-disparity image for a given disparity map. Warmer colors correspond to larger disparities / smaller distances.

Algorithm 2 Dominant foreground disparities

Input: u- and v-disparity images (U,V) calculated by eq. (4.14) and (4.15)
Output: dominant foreground disparities in u- and v-direction Δ_u, Δ_v
Define n_{min}: minimum number of histogram bin count for valid disparity

Initialize Δ_u and Δ_v
for $v \leftarrow 1 : \text{rows}(V)$ **do**
    for $d \leftarrow \text{cols}(V) : -1 : 1$ **do**
        if $n_{\text{min}} \leq V(v, d)$ **then**
            $\Delta_v(v) \leftarrow d$, **break**
        end if
    end for
end for
for $u \leftarrow 1 : \text{cols}(U)$ **do**
    for $d \leftarrow \text{rows}(U) : -1 : 1$ **do**
        if $n_{\text{min}} \leq U(d, u)$ **then**
            $\Delta_u(u) \leftarrow d$, **break**
        end if
    end for
end for
apply averaging filter of window size 3 for gap removal
$\Delta_v \leftarrow \Delta_v \circ h_3$
$\Delta_u \leftarrow \Delta_u \circ h_3$

the v- and u-disparity images. For a given image row v or column u the dominant foreground disparity is defined as the largest disparity, that has sufficient support in terms of histogram entries in the u- or v- disparity image. Algorithm 2 shows a detailed scheme for both row- and column-wise dominant foreground disparity calculation. In the second step, the v-segmentation for the pedestrian head is determined. This is achieved by a row-wise search for concatenated regions in terms of similar dominant foreground disparities posed in algorithm 3. The result is a range of image rows $[v_{\text{min}}, v_{\text{max}}]$, which contains the pedestrian's

Algorithm 3 v-segmentation of pedestrian head

Input: dominant foreground disparities in v-direction Δ_v
Output: head v-segmentation $[v_{\text{min}}, v_{\text{max}}]$
Define h_{min}: minimum head height for given search region
Define ε_Δ: maximum disparity deviation of inner head regions

foundHead$_v \leftarrow$ *false*
$v_{\text{min}} \leftarrow \dim(\Delta_v), v_{\text{max}} \leftarrow \dim(\Delta_v)$
for $v \leftarrow \dim(\Delta_v) : -1 : 2$ **do**
    $d_v \leftarrow \Delta_v(v)$
    if $|d_v - \Delta_v(v-1)| \leq \varepsilon_\Delta$ **then**
        $v_{\text{min}} \leftarrow v - 1$
        if $1 = v_{\text{min}}$ **then**
            foundHead$_v \leftarrow$ *true*
        end if
    else
        dispGap $\leftarrow$ *false*
        for $v \leftarrow v - 1 : -1 : 2$ **do**
            if $|d_v - \Delta_v(v)| \leq \varepsilon_\Delta$ **then**
                $v_{\text{min}} \leftarrow v$, dispGap $\leftarrow$ *true*, **break**
            end if
        end for
        if $\neg$dispGap $\&(v_{\text{end}} - v_{\text{min}}) \geq h_{\text{min}}$ **then**
            foundHead$_v \leftarrow$ *true*, **break**
        end if
    end if
end for

head. Finally, a similar approach is applied for a column-wise search of head correspondences/clusters based on dominant foreground disparities calculated from the u-disparity image in Algorithm 4. At the end, the row- and column-wise segmentations form the location of the pedestrian head, see Algorithm 5. Due to the design of the head search area, the potential vertical search range is restricted to a certain limit meaning that valid disparity measurements corresponding to the pedestrian can already be expected from the bottom line of the v-disparity image. Therefore, the approach for vertical head segmentation is less complex than the search for head correspondences in the horizontal u-range. Here, the presence of structures from background like other pedestrian heads, street lamps and road signs for example requires the usage of a more intelligent clustering algorithm. An illustration for the introduced algorithms is given by Figure 4.3.

Algorithm 4 u-segmentation of pedestrian head

Input: dominant foreground disparities in u-direction Δ_u
Output: u-segmentation $\mathfrak{U}$ (set of u-clusters)
Define d_{min}: minimum considered disparity for head search
Define w_{min}: minimum head width for given search region
Define ε_Δ: maximum disparity deviation of inner head regions

$u_{\text{min}} \leftarrow 0, u_{\text{max}} \leftarrow 0, \mathfrak{U} \leftarrow \emptyset$, foundCluster $\leftarrow$ *false*
for $u \leftarrow 1 : \dim(\Delta_u)$ **do**
    $d_u \leftarrow \Delta_u(u)$;
    if ¬foundCluster **then**
        if $d_u \geq d_{\text{min}}$ **then**
            foundCluster $\leftarrow$ *true*, $u_{\text{min}} \leftarrow u, u_{\text{max}} \leftarrow u$;
        end if
    else
        if $(|\Delta_u(u_{\text{max}}) - d_u| \leq \varepsilon_\Delta) \& (d_u \geq d_{\text{min}})$ **then**
            $u_{\text{max}} \leftarrow u$;
            if $(u_{\text{max}} = \dim(\Delta_u)) \& (u_{\text{max}} - u_{\text{min}} + 1) \geq w_{\text{min}}$ **then**
                $\mathfrak{U} \leftarrow$ addCluster$(\{u_{\text{min}}, u_{\text{max}}, d_u\})$, **break**
            end if
        else
            if $(u_{\text{max}} - u_{\text{min}} + 1) \geq w_{\text{min}}$ **then**
                $\mathfrak{U} \leftarrow$ addCluster$(\{u_{\text{min}}, u_{\text{max}}, d_u\})$
            end if
            foundCluster $\leftarrow$ *false*;
        end if
    end if
end for

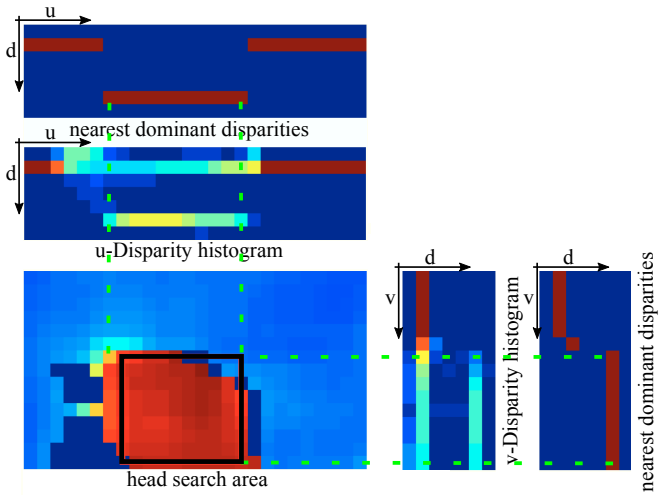


Figure 4.3: Stereo-based pedestrian head localization. Based on the u- and v-disparity images dominant foreground disparities are determined initially. Subsequently, concatenated areas or clusters are extracted, that have a common range of disparity values. The foreground clusters limit the head region in u- and v-direction and hereby described a rectangular region, which is then used as input for further pedestrian head pose estimation.

Algorithm 5 Determination of pedestrian head position

Input: v-segmentation $[v_{\text{min}}, v_{\text{min}}]$ and u-segmentation $\mathfrak{U}$
Output: head image location $H^{\text{2D}}_{\text{stereo}} = [u, v, w, h]$
Define $C \in \mathfrak{U}$: u-cluster element
Define A_{search}: defined head search area

foundHead$_u \leftarrow$ *false*
if $|\mathfrak{U}| = 1$ **then**
    foundHead$_u \leftarrow$ *true*, $[u_{\text{min}}, u_{\text{max}}] \leftarrow C \in \mathfrak{U}$
else if $|\mathfrak{U}| > 1$ **then**
    $[\hat{d}, \hat{C}] \leftarrow \arg\max_d \{d : d \in C\}_{C \in \mathfrak{U}}$
    $[u_{\text{min}}, u_{\text{max}}] \leftarrow \hat{C}$
    foundHead$_u \leftarrow$ *true*
end if
if foundHead$_v$&foundHead$_u$ **then**
    $H^{\text{2D}}_{\text{stereo}} \leftarrow [u_{\text{min}}, v_{\text{min}}, u_{\text{max}} - u_{\text{min}}, v_{\text{max}} - v_{\text{min}}]$
else
    $H^{\text{2D}}_{\text{stereo}} \leftarrow A_{\text{search}}$
end if

Depending on input data, the stereo-based head localization eventually results in a restricted area containing the pedestrian head, an area exactly surrounding the head or the overall defined head search area in cases, where a refined positioning is not possible.

4.3.2 Measurement Model Update

After extracting a potential head candidate, the additional knowledge is used for the particle filter measurement model update given in equation (3.39).

$$p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) = p_{\text{loc}}(\mathbf{z}_k|\mathbf{x}_k^{(i)}) \sum_{o \in \mathbb{O}} p(\mathbf{z}_k|\Theta = o, u_k^{(i)}, v_k^{(i)}, s_k^{(i)}) p(\Theta = o|\theta_k^{(i)}), \tag{4.16}$$

with

$$p_{\text{loc}}(\mathbf{z}_k|\mathbf{x}_k^{(i)}) = \exp(-d(\mathbf{z}_k, \mathbf{x}_k^{(i)})) = \exp(-d(H^{\text{2D}}_{\text{stereo}}, \mathbf{x}_k^{(i)}) \tag{4.17}$$

and $d(\cdot, \cdot)$ defining the Mahalanobis distance. The distance measure d represents the deviation of the stereo-based extracted head position from the actual considered particle's center and scale, that is

$$d(H^{\text{2D}}_{\text{stereo}}, \mathbf{x}_k^{(i)}) = \sqrt{\left\{ \begin{pmatrix} u^{\text{2D}}_{\text{stereo},c} \\ v^{\text{2D}}_{\text{stereo},c} \\ s^{\text{2D}}_{\text{stereo}} \end{pmatrix} - \begin{pmatrix} u^{(i)}_{k,c} \\ v^{(i)}_{k,c} \\ s^{(i)}_k \end{pmatrix} \right\}' \Sigma^{-1} \left\{ \begin{pmatrix} u^{\text{2D}}_{\text{stereo},c} \\ v^{\text{2D}}_{\text{stereo},c} \\ s^{\text{2D}}_{\text{stereo}} \end{pmatrix} - \begin{pmatrix} u^{(i)}_{k,c} \\ v^{(i)}_{k,c} \\ s^{(i)}_k \end{pmatrix} \right\}}, \tag{4.18}$$

where $u^{\text{2D}}_{\text{stereo},c}$, $v^{\text{2D}}_{\text{stereo},c}$, $u^{(i)}_{k,c}$ and $v^{(i)}_{k,c}$ define the center image coordinates of the stereo-based and the actual particle's associated head region, respectively.

The covariance matrix Σ consists of the following components

$$\Sigma = \begin{pmatrix} w^{(i)}/c_1 & 0 & 0 \\ 0 & h^{(i)}/c_2 & 0 \\ 0 & 0 & \tilde{\sigma}_s \end{pmatrix}. \tag{4.19}$$

Σ therefore depends on the particle associated head region's width $w^{(i)}$, height $h^{(i)}$ and a maximum allowed standard deviation in scale $\tilde{\sigma}_s$. The parameter values for c_1, c_2 and $\tilde{\sigma}_s$ have to be determined for later experiments. In other words, $p_{\text{loc}}(\mathbf{z}_k|\mathbf{x}_k^{(i)})$ gives a likelihood, how well the stereo-based extracted head location $H_{\text{stereo}}^{\text{2D}}$ fits to the particle positions. Hence, particles are guided through areas with dominant foreground disparity measurements. It should be noted that p_{loc} only defines a likelihood and not a probability measure having a range of values within the interval $[0,1]$. Therefore, empirically robust model fusion approaches like the *sum-rule* (additive combination) as proposed by [Kittler et al., 1998] cannot directly be used for the adapted measurement model. However, dedicated experiments for stereo-based head localization show stable estimates also for the *product-rule* as it is applied for the updated measurement model in equation (4.16).

4.4 Evaluation Datasets

The presented algorithms within the previous sections are evaluated on real world datasets. First, an internal dataset is introduced. For a another challenging dataset, a recently published dataset from Daimler by [Schneider and Gavrila, 2013] is chosen.

4.4.1 The Bosch Inner-City Stereo Dataset

For the purpose of stereo-based head localization and 3D head pose estimation, an internal inner-city dataset at Bosch was collected including labeled stereo image pairs. In total, the data consists of 23 sequences with 57 labeled pedestrian tracks. 9 of the sequences were recorded on test track with a small amount of background structures in order to have ideal scenarios for a proof of concept evaluation. For each labeled track, there is also a sequence of pedestrian detections available resulting from generic stereo-based object detection and tracking with a following pedestrian classification using a HoG SVM [Dalal and Triggs, 2005]. The stereo camera system has a base width of 12cm and the camera constant for the left virtual camera frame is around 1200. Labels are provided up to a measured pedestrian distance of 50m. An overview of the resulting samples for the Bosch stereo evaluation dataset can be found in Table 4.1.
Additionally, the pedestrian heads are annotated and the head pose is labeled again to be one of eight discrete head pose classes. The resulting sample distribution over all head pose classes is given in Table 4.2.

Number of sequences	23
Number of different persons	57
Minimum head size (px)	7×7
Total number of samples	5874 (labels and detections)

Table 4.1: Overview of relevant samples to be used for evaluation from the Bosch Inner-City Stereo dataset.

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
1153	523	564	347	1168	212	1343	564

Table 4.2: Total number of samples per head pose class for the Bosch Inner-City Stereo dataset.

4.4.2 The Daimler Inner-City Stereo Dataset

A real world inner-city dataset including sequences with pedestrians in critical situations was recently published by [Schneider and Gavrila, 2013]. This data contains pedestrian labels in addition to distance measurements using a stereo video sensor. The sequences include pedestrians in crossing, stopping, starting or bending in situations, which are later used as well in the chapters 5 and 6 for a combined pedestrian intention recognition and path prediction. In most of the sequences the ego vehicle is approaching a pedestrian. Hence, pedestrians appear at different resolutions, which holds the same for the appearances of human heads. The original dataset is divided into a training and test set, consisting of 35 and 32, respectively. For evaluation in this chapter, all sequences are put together into one overall set. Table 4.3 shows the number of sequences for each addressed scenario.

bending in	starting	stopping	crossing	total
23	9	17	18	67

Table 4.3: Number of sequences per scenario for the Daimler Inner-City Stereo dataset presented by [Schneider and Gavrila, 2013].

The underlying stereo camera system has a base width B of 22cm and a virtual camera constant f=1240. Note, that the stereo setup has a direct impact on the range for the proposed method in later experiments as the distance error $\epsilon_{\mathbf{x}}$ behaves proportionally with the reciprocal of the mentioned two factors, namely $\epsilon_{\mathbf{x}} \propto \frac{1}{fB}$. To evaluate the developed stereo-based head localization, pedestrian heads are labeled additionally for each frame, where a labeled pedestrian ground truth 3D position already is available. Together with the pedestrian labels, the dataset provides the measurement outputs resulting from a HoG-SVM pedestrian detector, see [Dalal and Triggs, 2005]. Table 4.4 gives an overview for all samples in the resulting evaluation set.

total number of samples	number of different persons	minimum head size (px)
9044(labels)/7844(detections)	4	6×6

Table 4.4: Overview of relevant samples to be used for evaluation from the Daimler Inner-City Stereo dataset presented by [Schneider and Gavrila, 2013].

Front	Front-Right	Right	Back-Right	Back	Back-Left	Left	Front-Left
544	4	153	7	2575	1285	3161	1315

Table 4.5: Total number of samples per head pose class for the Daimler Inner-City Stereo dataset presented by [Schneider and Gavrila, 2013].

For head pose estimation evaluation, only samples including pedestrians with a maximum distance of approximately 30m are considered. This results in the following distribution of head samples per head pose class displayed in Table 4.5. Due to the addressed scenarios within the Daimler dataset, the head pan angle range from 45° to 135° (front-right- to back-right views) is underrepresented.

4.5 Experiments

This section will present a performance evaluation for stereo-based head localization and 3D head pose tracking. Again, the amount of suitable datasets is quite low, which lead to the recording of a new dataset and additional labeling of existing data from related applications, see 4.4. First, the experimental setup for the chosen datasets is explained in 4.5.1. Then, the results for stereo-based head localization and 3D head pose tracking are presented in Section 4.5.2 and 4.5.3, respectively.

4.5.1 Experimental Setup

The presented methods for stereo-based head localization (Sec. 4.3) and 3D head pose estimation (Sec. 4.1 and Sec. 4.2), make the use of disparity maps for given stereo image pairs, which have to be calculated at first. The public available algorithm of [Geiger et al., 2010] therefore is able to provide accurate and dense disparity maps. For the updated particle state propagation model in Section. 4.2 additional model parameters have to determined. Here, heuristic assumptions are made resulting in the values given in Table 4.6. Pedestrian velocity

α_θ	κ_θ	κ_v	ξ
0.7	6	2	0.5m/s

Table 4.6: System parameters for head pose tracking with integrated human body motion (Sec. 4.2) determined for all evaluation datasets.

estimates are derived by implementing a simple Kalman filter with a linear constant velocity model, see [Bar-Shalom et al., 2002, Keller et al., 2011a]. Here, the filtered pedestrian state contains estimates for longitudinal and lateral velocity components, i.e., $X_t = (x_t, y_t, v_t^x, v_t^y)$ for each time instance t. The measurement vector $\mathbf{z} = (x, y)$ includes information about the actual pedestrian's relative longitudinal and lateral position, which are indirectly estimated by the stereo-video sensor using the pedestrian location in the image and an assigned median disparity calculated over the upper part of the corresponding image bounding box (eq. (4.2)).

Parameters for Stereo Head Localization

The proposed algorithms (2-5), realizing a stereo-based head localization, introduce a few parameters, that have to be determined in advance. Some parameters, like d_{min}, are directly set due to the stereo camera system setup and range limitation up to 50m. The remaining parameters values are set heuristically but turn out to guaranty an acceptable performance in later experiments. Table 4.7 displays the resulting parameters, their description and values.

Parameter	**Description**	**Bosch/Daimler**
d_{min}	minimum considered disparity for head search	$2.75/5.0$
n_{min}	number of histogram entries for valid head initialization	$3/6$
ε_{Δ}	maximum disparity deviation of inner head regions	1.75
h_{min}	minimum head height for given search region A_{search}	$0.5 \times h_{A_{\text{search}}}$
w_{min}	minimum head width for given search region A_{search}	h_{min}

Table 4.7: Chosen parameters values for stereo-based head localization depending on Bosch and Daimler dataset.

The parameters for the updated measurement model in Section 4.3.2 are optimized during experiments and given in Table 4.8.

c_1	c_2	$\tilde{\sigma}_s$
6	6	0.2

Table 4.8: Parameters for updated measurement model (Sec. 4.3.2) after integration of stereo head localization.

4.5.2 Results on Stereo Head Localization

First, the standalone stereo-based head localization is considered. Due to the stereo matching/aggregation window size, in general, high-frequency components like corners and edges in the image plane are smeared out in disparity space. This results in detected head regions not exactly fitting the head size in the image. However, for later comparison with the original 2D head pose estimation approach (Sec. 3.2.4) and the combined system presented in this chapter (Sec. 4.3.2), the conditions for associating a ground truth head annotation (wrt.

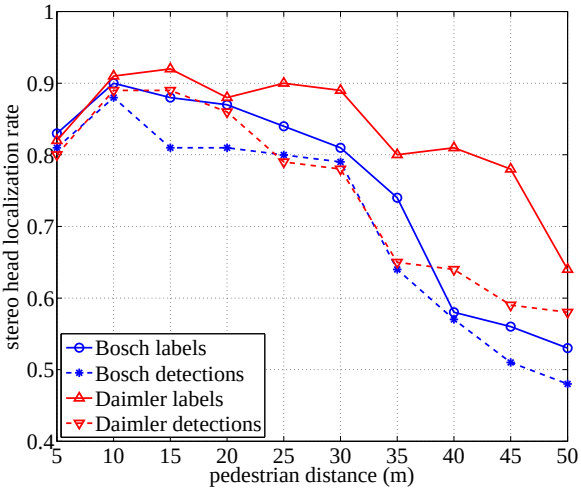


Figure 4.4: Stereo head localization rates over pedestrian distance (m) for the Bosch and Daimler dataset. Evaluation performed for pedestrian labels and system detections, separately.

	5*m*	10*m*	15*m*	20*m*	25*m*	30*m*	35*m*	40*m*	45*m*	50*m*
Daimler (labels)	0.82	0.91	0.92	0.88	0.90	0.89	0.81	0.81	0.78	0.64
Daimler (dets.)	0.80	0.89	0.89	0.86	0.79	0.78	0.65	0.64	0.59	0.58
Bosch (labels)	0.83	0.90	0.88	0.87	0.84	0.81	0.74	0.58	0.56	0.53
Bosch (dets.)	0.81	0.88	0.81	0.81	0.80	0.79	0.64	0.57	0.51	0.48

Table 4.9: Stereo head localization rates over pedestrian distance (m) for Bosch and Daimler dataset. Evaluation performed for pedestrian labels and system detections, separately.

image) with the stereo-based extracted head location have to be maintained. I.e., both cover and overlap have to lie above 0.7 as proposed in Section 3.4.3. This fact will directly have an impact on stereo-based head localization performance and has to be considered in later results. Localization rates are determined using the labeled head images from the Bosch and Daimler inner-city datasets. Figure 4.4 shows the calculated localization rates when using the ideal system input given by the origin pedestrian labels and for realistic system inputs using the pedestrian detection bounding boxes. The exact localization rate values can be found in Table 4.9. The general observation can be derived, that performance reduces when using a real detector input compared with the usage of ground truth annotations. The performance when using detections goes down more drastically for the Daimler dataset compared to the Bosch dataset. This can be explained by the usage of real HoG detections without further tracking in the Daimler case, whereas bounding boxes for the Bosch system are derived from tracked objects based on stereo-vision with a following classification which leads to more robust and smooth system inputs. At higher ranges up to 50m the head localization gives better results on the Daimler data compared with the Bosch dataset. The main reason for that is the higher base width for the used stereo video sensor. In the Bosch case, at 50m most of the measured raised obstacles share the same disparity with other background contents

	false loc.	**full A_{search}**	**surrounding areas**
Daimler (labels)	5%	4%	91%
Daimler (detections)	9%	17%	74%
Bosch (labels)	10%	14%	76%
Bosch (detections)	19%	24%	57%

Table 4.10: Percentage for three potential cases of an incorrect stereo-based head localization. Evaluation is performed on labels and pedestrian detections for the Bosch and Daimler dataset.

and therefore cannot be separated anymore. For the Daimler setup this is still possible to a limited extend.

According to Section 4.3, there are three possibilities for a head localization to be considered as incorrect, namely an area which has no sufficiently high overlap with the real head position (false location), an area, that completely surrounds the head or the full head search area. The percentage for all three cases among all false head localizations is given in Table 4.10. The real incorrect localized pedestrian heads only represent 5-19% of all miss-localized samples. The rest is either the complete head search area or a larger region including the head. The latter case does not state a critical issue for further processing, as particles get higher weights within this region therefore also at the real heads position. Hence, in this case there is no benefit loss compared to the head localization system only based on head pose detector confidences (Chap. 3). The percentage for a head localization resulting in the full head search area is higher for the Bosch data compared to the Daimler dataset. Most of these errors again appear at higher distances due to the base width limitation for the Bosch stereo camera system. In Figure 4.5 and Figure 4.6 samples for correct and incorrect localized pedestrian heads, respectively, are visualized. The method mostly fails, if there are other dominant objects in the foreground or at the same pedestrian's distance, like street lamps, bushes etc. Further failure root causes are pedestrians at higher distances, where background regions cannot be differentiated from the pedestrian appearance on disparity level, or general problems for stereo matching algorithms on periodic structures available on garage doors for example.

As a next step, the combined system is evaluated with a head localization that is mostly dependent on the confidence outputs of several head pose detectors, i.e., Section 3.2.3 and 3.2.4,

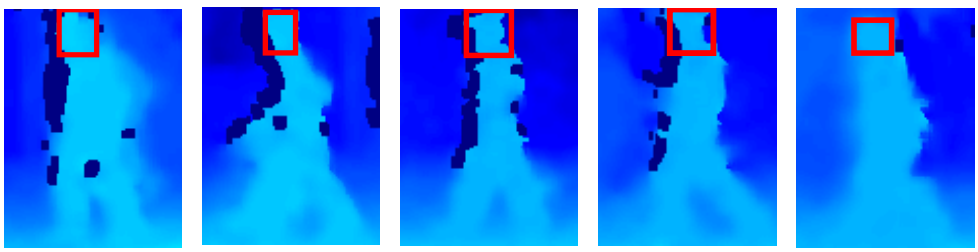


Figure 4.5: Collection of pedestrian samples with correct stereo-based head localization

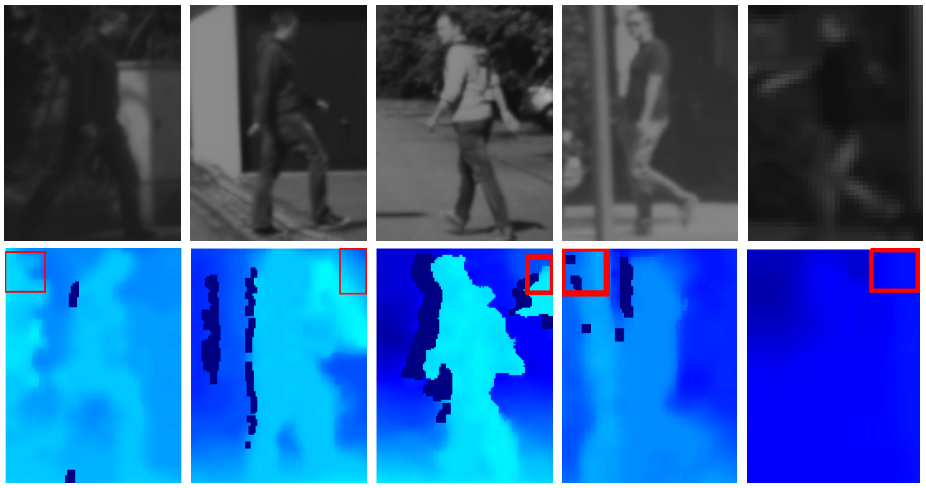


Figure 4.6: Collection of pedestrian samples with failing stereo-based head localization

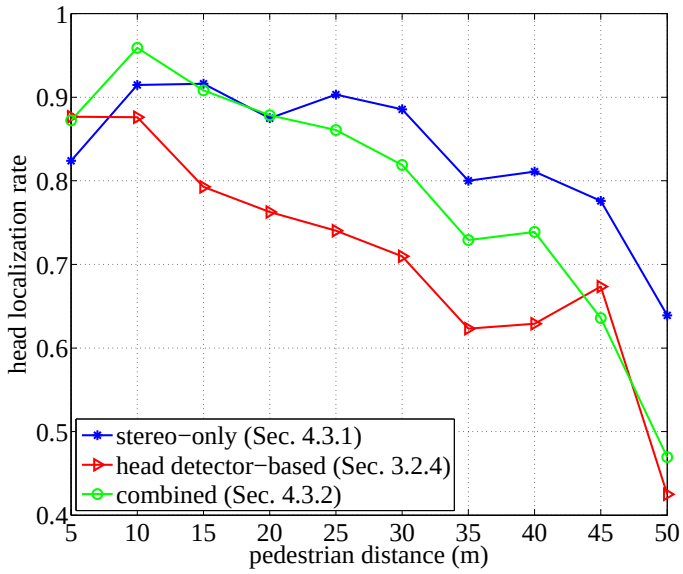


Figure 4.7: Head localization results over pedestrian distance on Daimler dataset. Blue curve: stereo-based only, see Section 4.3.1). Red curve: original 2D head pose detector-based version from Section 3.2.4. Green curve: combined version using updated measurement model from Section 4.3.2.

but now supported by stereo depth information as well, see Section. 4.3. Evaluation is performed on pedestrian label bounding boxes in order to derive the direct impact on benefit when adding the stereo feature. Figure 4.7 shows the resulting head localization rates over pedestrian distance for the single (Sec. 3.2) and combined system (Sec. 4.3.2) using the sequences from the Daimler dataset. When incorporating the stereo information into the combined system, a head localization rate increment of approximately 10% can be obtained nearly throughout the considered distance range. Performance reduces drastically at 45 to 50m (head size $\approx 6\times6$ pixel) for both methods resulting in head localization rates smaller than 50%, which is not surprising as the stereo-only approach suffers from the quadratically increasing disparity error. Stereo-only head localization performs better at higher distances than the 2D head pose detector-based approach. Here, incorporating stereo informa-

tion shows the highest benefit, while the head pose classifiers suffer from highly structured background. For evaluations in the following section, the combined system incorporating stereo information will be used.

4.5.3 Results on 3D Head Pose Tracking

For performance evaluation of the final developed system including all improvements discussed in the sections 4.1 to 4.3 confusion matrices are chosen in addition to the calculation of classification rates with respect to the eight discrete head pose classes. First, confusion matrices are presented in Figure 4.8 comparing the original method introduced in Section 3.2 with the system modifications developed in this chapter. Evaluation first is performed on the Daimler dataset.

Figure 4.8: Confusion matrices for Daimler dataset, when using (a) the original 2D head pose tracking module (Sec. 3.2.4) and (b) the combined version incorporating the human body movement direction (Sec. 4.2).

The integration of the pedestrian's velocity components (Fig. 4.8b) results in less confusion with more distant head pose classes than the initial implemented approach (Fig. 4.8a). Especially, the confusion of frontal poses with back views and vice versa can be reduced significantly from 23.8% to 2.7% and 18.9% to 9.3%, respectively. The 100% correct classification rate for back-right poses slightly distorts the overall performance as there are only a view samples available, see Table 4.5. With an overall correct classification rate of 54.8% the particle filtering system incorporating the pedestrian movement direction clearly outperforms the initially developed system with random head pose propagation, which only achieves 44.7%.

Evaluation on the internally recorded dataset shows similar observations when fusing head pose estimation results with human body movement direction as pictured in Figure 4.9.

Overall classification rate gets increased from 41.5% (Fig. 4.9a) to 51.0% (Fig. 4.9b) in case of combining head pose estimation with pedestrian movement direction. For both cases, normal tracking and combined version, left and right side views of a head can be classified with highest performance (52.2%/56.2%, 50.0%/59.9%), a result, which already was observed

Figure 4.9: Confusion matrices for Bosch dataset, when using (a) the origin head pose tracking module (Sec. 3.2.4) and (b) the combined version incorporating the human body movement direction (Sec. 4.2).

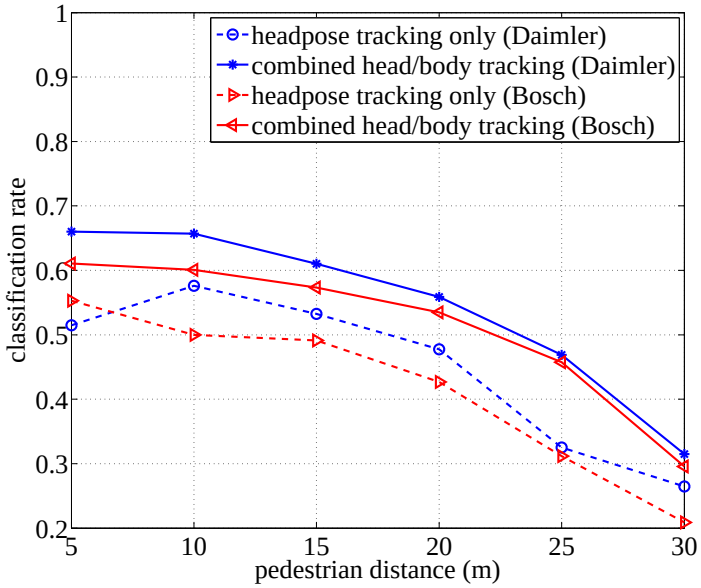


Figure 4.10: Head pose classification rates over pedestrian distance. Different curves show the results on different datasets (Bosch, Daimler), when using the origin head pose tracking module (Sec. 3.2.4) and the combined version incorporating the human body movement direction (Sec. 4.2).

within earlier experiments in Chapter 3. Again, the problem confusing head frontal and back views is greatly reduced with the combined system from 22.6% to 2.8% and 14.1% to 2.3%, respectively. In general, performance little decreases compared to the Daimler dataset. This can be explained by the presence of more dynamic and complex inner-city scenarios included within the internal Bosch dataset. Furthermore, the classification performance dependency on the pedestrian distance, which is directly related to the head size, is given in Figure 4.10. For distances higher than 30m, head pose classification performance breaks down. Hence, only the interval from 5 to 30m is considered, which is enough for later use in intention recognition and path prediction. Evaluation is performed using the images from the Bosch and Daimler datasets. Classification rates are calculated for the original 2D approach includ-

ing a random particle head pose state propagation (Sec. 3.2.4) and the method developed in this chapter integrating the pedestrian's movement direction into the dynamical model for the particle filter, see Section 4.2. In general, the system performs slightly worse on the Bosch real-world dataset. This can be explained by more complex scenarios included in the dataset compared to the simpler sequence setup for the Daimler dataset. Using the pedestrian movement direction derived from velocity estimates for particle state propagation significantly improves head pose classification rates about nearly 10% over the considered distance range for both Bosch and Daimler sequences.

4.6 Conclusion

The main focus of this chapter was the improvement and robustification of the earlier introduced head pose estimation system by making the use of stereo depth information. Improvement was achieved in two ways. First, velocity estimates from an accurate stereo-based object detection and tracking approach were used as prior knowledge for head pose tracking (Sec. 4.2). As a second measure, the stereo depth map was directly used for head extraction in order to support the intensity-based head localization in situations with highly structured background (Sec. 4.3). Experiments in Section 4.5 show an improvement in both head pose estimation and head localization of up to 10%.

5 Pedestrian Intention Recognition

This chapter presents a novel approach for a reliable pedestrian intention recognition in advanced video-based driver assistance systems using a Latent-Dynamic Conditional Random Field model (*LDCRF*). The model integrates pedestrian dynamics and situational awareness using the observations from a stereo-video system for pedestrian detection and human head pose estimation. Figure 5.1 presents a coarse system overview.

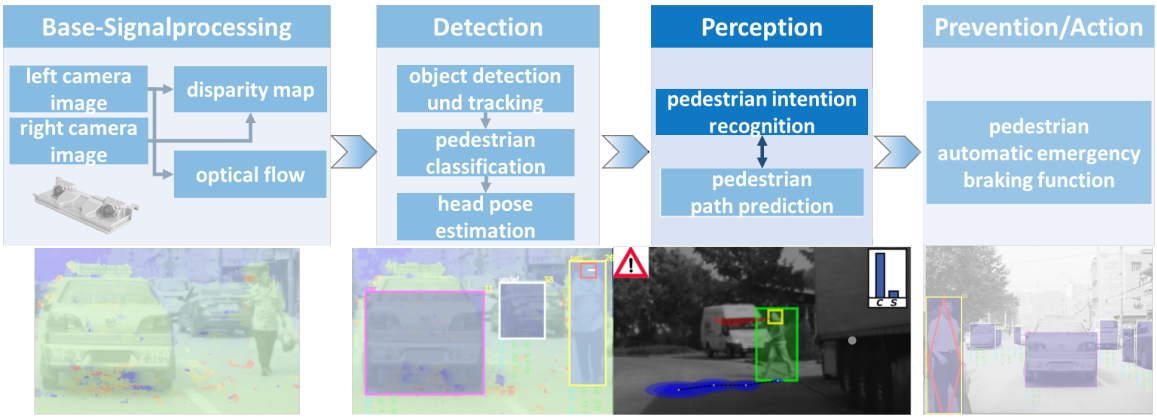


Figure 5.1: System Overview. Pedestrians are detected and tracked by a stereo generic obstacle detection approach and verified by a gray-value based classifier. For each pedestrian candidate the head orientation will be estimated. The previously extracted information is input for the intention recognition module.

The proposed latent-dynamic discriminative model applied on extracted time-series of features is then able to learn inner connections within a specific type of scenario and external correlations between different types of scenarios. Estimates for pedestrian intentions are provided in a probabilistic way based on pre-defined scenarios in daily traffic. The presented method is evaluated on a public available dataset addressing scenarios of lateral approaching pedestrians that might cross the road, turn into the road or stop at the curbside. Compared to recent literature, a wider range of scenarios is addressed also including pedestrians that initially are walking along the sidewalks but then bend in towards the road. Here, the importance in the knowledge of persons' head orientations will be demonstrated. During experiments, it can be shown, that the proposed approach leads to better stability and class separation compared to state-of-the-art pedestrian intention recognition approaches. The computational costs are comparatively low such that the approach can be easily integrated into an overall real-time system as an indicator for pedestrian path prediction helping to es-

timate potential future pedestrian states (cf. Chap. 6) within a later automatic emergency braking function. Similarly, the system output can be used for controlling the switching states of the Switching Linear Dynamical System (*SLDS*) presented in [Kooij et al., 2014a]. The remainder of this chapter is organized as follows. Section 5.1 presents the basic principles of Latent-Dynamic Conditional Random Fields together with the extraction of powerful features for pedestrian intention recognition. A public available dataset for later evaluation is introduced in Section 5.2. Furthermore, experimental results for the presented approach are presented for specified scenarios including crossing, stopping, bending and straight walking pedestrians in Section 5.3. This chapter ends up with a conclusion in Section 5.4.

5.1 Latent-Dynamic Conditional Random Fields for Pedestrian Intention Recognition

[Morency et al., 2007] first introduced LDCRF models as an extension of conventional Conditional Random Fields (*CRF*, [Lafferty et al., 2001]) by adding a layer of hidden latent states. These hidden state variables can model the intrinsic sub-structure of a specific class label and capture extrinsic dynamics between different classes. Furthermore, LDCRF models proved to outperform typical CRF models, the well-known Hidden Markov Models (*HMM*, [Rabiner, 1989]) and conventional machine learning algorithms like Support Vector Machines (*SVMs*) in the field of gesture recognition, cf. [Morency et al., 2007]. Figure 5.2 shows one potential configuration of a LDCRF model.

For this work, the task of action-/intention recognition is to find a sequence of labels $\boldsymbol{\eta} = \{\eta_1, \eta_2, \ldots, \eta_T\}$ that best predicts the pedestrian intention for the sequence of observations $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ collected within a total range of T time steps. LDCRF models originate from Conditional Random Field theory published by [Lafferty et al., 2001], which is one of famous activity recognition models that can capture extrinsic dynamics between class labels. As Figure 5.2 shows, a LDCRF model is an undirected graph consisting of sequential variable pairs of state variables $\mathbf{x}_t$ and class labels η_t plus an additional layer of non-observable, mostly probable hidden states $\mathbf{h} = \{h_1, h_2, \ldots, h_T\}$, for every time step. Using a larger set of observations as training data, the LDCRF is able to learn an intention recognition model and given a new observation sequence $X = \{\mathbf{x}_t\}_{t=1,\ldots,T}$ to infer the intention labels $\boldsymbol{\eta} = \{\eta_t\}_{t=1,\ldots,T}$. Each h_t is member of a finite set $\mathcal{H}_{\eta_t} = \{h_t^{(\eta_t,j)}\}_{j=1}^{H}$ of possible hidden states for the intention label $\eta_t \in \mathcal{Y} = \{1, 2, \ldots, L\}$, where L is the number of intention labels. For different labels, the model is restricted to have disjoint sets of hidden states $\mathcal{H}_{\eta_t}$, which later reduces complexity of the model. $\mathcal{H} := \bigcup_{\eta_t} \mathcal{H}_{\eta_t}$ is defined to be the union of all $\mathcal{H}_{\eta_t}$ sets, that is all possible hidden states are contained in $\mathcal{H}$. Following [Morency et al., 2007], the LDCRF defines a latent conditional model as

$$P(\boldsymbol{\eta}|X;\boldsymbol{\theta}) = \sum_{\mathbf{h}\in\mathcal{H}} P(\boldsymbol{\eta}|\mathbf{h}, X;\boldsymbol{\theta})P(\mathbf{h}|X;\boldsymbol{\theta}), \tag{5.1}$$

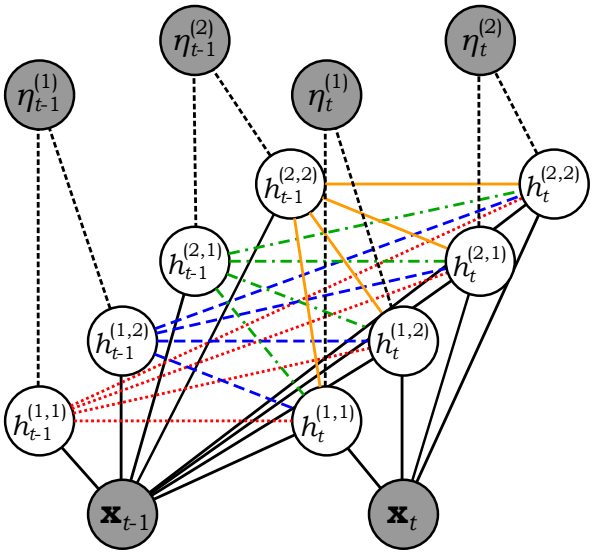


Figure 5.2: LDCRF for a 2-class problem with 2 hidden states per class label. Observables are displayed as shaded nodes. Black solid connections are modeled by the state functions s_l. Colored connections between hidden states are modeled by the transition functions t_m. The hidden states $h_t^{(i,1)}$ and $h_t^{(i,2)}$ model the intrinsic structure for class label $\eta_t^{(i)}$, $i \in \{1,2\}$, while the connections between $h_t^{(1,k)}$ and $h_t^{(2,k)}$, $k \in \{1,2\}$ model the extrinsic relations between the class labels $\eta_t^{(1)}$ and $\eta_t^{(2)}$.

with the model parameters $\boldsymbol{\theta}$. By definition, for sequences having any $h_t \notin \mathcal{H}_{\eta_t}$ it holds $P(\boldsymbol{\eta}|\mathbf{h}, X; \boldsymbol{\theta}) = 0$ and the model in equation (5.1) can be simplified to

$$P(\boldsymbol{\eta}|X;\boldsymbol{\theta}) = \sum_{\mathbf{h}:h_t \in \mathcal{H}_{\eta_t}, \forall t=1,\ldots,T} P(\mathbf{h}|X;\boldsymbol{\theta}). \tag{5.2}$$

Similar to the usual CRF formulation presented in [Lafferty et al., 2001], $P(\mathbf{h}|X;\boldsymbol{\theta})$ can be defined as

$$P(\mathbf{h}|X;\boldsymbol{\theta}) := \frac{1}{\mathcal{Z}(X,\boldsymbol{\theta})} \exp \underbrace{\left(\sum_k \theta_k \cdot F_k(\mathbf{h}, X) \right)}_{=:\Psi(\mathbf{h},X;\boldsymbol{\theta})}, \tag{5.3}$$

with the partition function

$$\mathcal{Z}(X;\boldsymbol{\theta}) = \sum_{\mathbf{h}:h_t \in \mathcal{H}} \exp \left(\sum_k \theta_k \cdot F_k(\mathbf{h}, X) \right). \tag{5.4}$$

The functionals $F_k(\mathbf{h}, X)$ can be written as sums of feature functions, namely state functions $s_l(h_t, X, t)$ and transition functions $t_m(h_t, h_{t-1}, X, t)$. For the potential function Ψ it then follows

$$\Psi(\mathbf{h}, X;\boldsymbol{\theta}) = \sum_{t=1}^{T} \left\{ \sum_l \lambda_l s_l(h_t, X, t) + \sum_m \mu_m t_m(h_t, h_{t-1}, X, t) \right\}, \tag{5.5}$$

with

$$\boldsymbol{\theta} = \{\theta_k\}_k = \{\lambda_l\}_l \cup \{\mu_m\}_m. \tag{5.6}$$

State functions s_l depend on a single hidden variable and observations in the model while transition functions t_m depend on pairs of hidden variables. In the original LDCRF model configuration, the number of state functions, s_l, will be equal to the dimension of the feature vector d times the number of possible hidden states. With the L different intention labels for the model and assuming H hidden states per label, the total number of state functions, s_l, and total number of associated weights λ_l will be $d \times L \times H$. In this work, a time window $w \geq 0$ can be defined additionally, connecting the observations from w previous time instances with the hidden states of the actual frame. The number of potential state functions then increases to $(1 + w) \times d \times L \times H$. For each hidden state pair (h', h''), the transition function t_m is defined as

$$t_m(h_{t-1}, h_t, X, t) = \begin{cases} 1, & \text{if } h_{t-1} = h' \text{ and } h_t = h'' \\ 0, & \text{otherwise} \end{cases}. \tag{5.7}$$

The weights μ_m associated with the transition functions model both the intrinsic and extrinsic dynamics. Weights associated with a transition function for hidden states that are in the same subset $\mathcal{H}_{\eta_t}$ will model the substructure patterns, while weights associated with the transition functions for hidden states from different subsets will model the external dynamic between intention labels. Each hidden state from the previous time instance can be connected by a transition function with all possible hidden states for the current frame. Therefore, the number of transition functions t_m in the model is $|\mathcal{H}| \times |\mathcal{H}|$. In comparison to LDCRFs, Hidden Markov Models (*HMM*) as presented in [Rabiner, 1989] are also able to model intrinsic structures with hidden states. However, for each class label η a separate model has to be trained. This results in unrelated model probabilities when evaluating a sequence with each trained HMM to determine the actual class label. Here, a benefit of using LDCRF models is, that they combine the single class marginals and output a meaningful probability for each $\eta_t^{(l)}$, i.e., $\sum_l P(\eta_t^{(l)}) = 1$.

5.1.1 Learning the Model Parameters

The training set consists of N labeled sequences $(X_i, \boldsymbol{\eta}_i)$, $i = 1 \ldots N$. Following [Lafferty et al., 2001] and [Morency et al., 2007], the objective function

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log P(\boldsymbol{\eta}_i | X_i; \boldsymbol{\theta}) - \frac{1}{2\sigma^2} \|\boldsymbol{\theta}\|^2 \tag{5.8}$$

is defined to learn the optimal parameter set $\boldsymbol{\theta}^*$. Equation (5.8) combines the conditional log-likelihood of the training data with the log of a Gaussian prior with variance σ^2, i.e., $P(\boldsymbol{\theta}) \sim \exp(\frac{1}{2\sigma^2} \|\boldsymbol{\theta}\|^2)$. The optimal parameter values under the criterion $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ can be found by gradient ascent using iterative solvers like *BFGS* [Broyden, 1970,Fletcher, 1970,

Goldfarb, 1970, Shanno, 1970]. Therefore, the gradient of $\log P(\boldsymbol{\eta}|X, \boldsymbol{\theta})$ for one particular training sequence $(X, \boldsymbol{\eta})$ with respect to the parameters λ_l associated with a state function s_l has to be calculated at first. Given Equations (5.2) and (5.3), it holds

$$\begin{aligned}
\frac{\partial}{\partial \lambda_l}\left(\log P(\boldsymbol{\eta}|X;\boldsymbol{\lambda})\right) &= \frac{\partial}{\partial \lambda_l}\left\{\log \sum_{\mathbf{h}:\forall h_t \in \mathcal{H}_{\eta_t}} \exp\left(\Psi(\mathbf{h}, X, \boldsymbol{\lambda})\right) - \log \mathcal{Z}(X;\boldsymbol{\lambda})\right\} \\
&= \frac{1}{\sum_{\mathbf{h}:\forall h_t \in \mathcal{H}_{\eta_t}} \exp(\Psi)} \sum_{\mathbf{h}:\forall h_t \in \mathcal{H}_{\eta_t}} \left\{\exp(\Psi)\frac{\partial \Psi}{\partial \lambda_l}\right\} - \frac{\frac{\partial}{\partial \lambda_l}\mathcal{Z}(X;\boldsymbol{\lambda})}{\mathcal{Z}(X;\boldsymbol{\lambda})} \\
&= \sum_{\mathbf{h}:\forall h_t \in \mathcal{H}_{\eta_t}} \left\{\frac{\exp(\Psi)}{\sum_{\mathbf{h}:\forall h_t \in \mathcal{H}_{\eta_t}} \exp(\Psi)} \sum_{t=1}^{T} s_l(h_t, X, t)\right\} - \frac{\frac{\partial}{\partial \lambda_l}\mathcal{Z}(X;\boldsymbol{\lambda})}{\mathcal{Z}(X;\boldsymbol{\lambda})} \\
&= \sum_{t,h'} \left\{\frac{\sum_{\mathbf{h}:h_t = h' \wedge \forall h_t \in \mathcal{H}_{\eta_t}} P(\mathbf{h}|X;\boldsymbol{\lambda})}{\sum_{\mathbf{h}:\forall h_t \in \mathcal{H}_{\eta_t}} P(\mathbf{h}|X;\boldsymbol{\lambda})} s_l(h', X, t)\right\} - \frac{\frac{\partial}{\partial \lambda_l}\mathcal{Z}(X;\boldsymbol{\lambda})}{\mathcal{Z}(X;\boldsymbol{\lambda})} \\
&= \sum_{t,h'} P(h_t = h'|\boldsymbol{\eta}, X;\boldsymbol{\lambda}) s_l(h', X, t) - \frac{\frac{\partial}{\partial \lambda_l}\mathcal{Z}(X;\boldsymbol{\lambda})}{\mathcal{Z}(X;\boldsymbol{\lambda})} \\
&= \sum_{t,h'} P(h_t = h'|\boldsymbol{\eta}, X;\boldsymbol{\lambda}) s_l(h', X, t) \\
&\quad - \sum_{\boldsymbol{\eta}',t,h'} P(h_t = h', \boldsymbol{\eta}'|X;\boldsymbol{\lambda}) s_l(h', X, t). \qquad (5.9)
\end{aligned}$$

According to [Morency et al., 2007], the single derivation steps of the gradient in equation (5.9) show, that the summations for $P(h_j = h'|\boldsymbol{\eta}, X;\boldsymbol{\lambda})$ are simply constrained versions of the partition function $\mathcal{Z}$ over the conditional random field for $\mathbf{h}$. Hence, it can be shown, that the gradient is computable with $O(T)$ using belief propagation [Pearl, 1988], for a sequence of length T. The objective function's gradient with respect to the parameters μ_m associated with a transition function t_m can be derived the same way, that is

$$\begin{aligned}
\frac{\partial}{\partial \mu_m}\left(\log P(\boldsymbol{\eta}|X;\boldsymbol{\mu})\right) &= \sum_{t,h',h''} P(h_t = h'', h_{t-1} = h'|\boldsymbol{\eta}, X;\boldsymbol{\mu}) t_m(h', h'', X, t) \\
&\quad - \sum_{\boldsymbol{\eta}',t,h',h''} P(h_t = h', h_{t-1} = h''|X;\boldsymbol{\mu}) t_m(h', h'', X, t). \qquad (5.10)
\end{aligned}$$

The marginal probabilities on edges, $P(h_j = h', h_k = h''|\boldsymbol{\eta}, X, \boldsymbol{\mu})$, necessary for this gradient can also be computed efficiently using belief propagation. For gradient ascent a more efficient version in speed and robustness of the BFGS technique is used, cf. [Liu and Nocedal, 1989].

5.1.2 Inference

To test a previously unseen sequence X, the most probable label sequence $\boldsymbol{\eta}^*$ will be estimated that maximizes the trained model using the optimal parameter values $\boldsymbol{\theta}^*$:

$$\boldsymbol{\eta}^* = \arg\max_{\boldsymbol{\eta}} P(\boldsymbol{\eta}|X, \boldsymbol{\theta}^*), \tag{5.11}$$

Applying equation (5.2) once again results in

$$\boldsymbol{\eta}^* = \arg\max_{\boldsymbol{\eta}} \sum_{\mathbf{h}:h_t \in \mathcal{H}_{\eta_t}, \forall t=1,\ldots,T} P(\mathbf{h}|X, \boldsymbol{\theta}^*). \tag{5.12}$$

To predict the label η_t^* of frame t, the marginal probabilities $P(h_t = h'|X, \boldsymbol{\theta}^*)$ are calculated for all possible hidden states $h' \in \mathcal{H}$. Single marginal probabilities are summed according to the disjoint sets of hidden states $\mathcal{H}_{\eta_t}$. Finally, the label associated with the optimal set is chosen. Similar to the training process, the marginal probabilities can efficiently be computed using belief propagation.

5.1.3 Feature Computation

In this section the set of features is presented, that are used for pedestrian intention recognition. [Hamaoka et al., 2013], [Oxley et al., 1995] and [Schmidt and Färber, 2009] describe dominant features to recognize persons' intention during road crossing events. The measurement outputs from the presented system for stereo-based object detection and tracking combined with a pedestrian head pose estimation provides most of these features. The defined features are extracted for each frame and concatenated into a time-series of T frames as an input for the LDCRF model introduced in earlier sections. The individual used features are explained in the following.

Pedestrian Dynamics

Detected pedestrians will be tracked in lateral and longitudinal direction using a Kalman filter, cf. [Bar-Shalom et al., 2002]. A simple constant velocity model (CV) for pedestrians is assumed. See [Schneider and Gavrila, 2013, Keller et al., 2011a] for details. As measurements the center foot point of the pedestrian image bounding box plus an additional median disparity value calculated over the upper pedestrian body are extracted. Using this information, the measured pedestrian lateral and longitudinal position in 3D is calculated with the help of camera calibration parameters, see eq. (4.1). In addition, vehicle dynamics are incorporated into the dynamical model for ego-motion compensation, cf. [Schneider and Gavrila, 2013]. As a result, for each frame, one obtains the filtered pedestrian's positions on the ground (x_t, z_t) relative to the ego-vehicle and absolute velocities $(\dot{x}_t, \dot{z}_t)$ in lateral and longitudinal direction.

Pedestrian Head Pose

To capture the pedestrians' awareness of an oncoming vehicle the human head pose gives a dominant cue, see [Hamaoka et al., 2013, Oxley et al., 1995] and [Schmidt and Färber, 2009]. The presented method for pedestrian intention recognition builds upon the system presented in the previous chapters 3 to 4, that tries to estimate pedestrian head poses in monocular gray value images supported by stereo depth information. The basic idea is to train multiple classifiers for different head pose classes related to a specified pan angle range. Furthermore, the single head pose estimation results are filtered for usage in video sequences by implementing a particle filter (Sec. 3.2). To more robustly guide the particles around future head regions, depth information within a detected pedestrian bounding box as well as estimated human movement directions are incorporated, see Chapter 4. Hence, for an existing pedestrian track the head pan angle θ_t^H from the continuous range $[-180, 180)$ will be extracted for each frame.

Formation of a Time-series

For a video sequence the above mentioned features will be integrated into a time-series of observations. The time-series X over T frames then includes the following data

$$X = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\} = \{(x_1, z_1, \dot{x}_1, \dot{z}_1, \theta_1^H)', \ldots, (x_T, z_T, \dot{x}_T, \dot{z}_T, \theta_T^H)'\}. \tag{5.13}$$

5.2 Evaluation Dataset

[Schneider and Gavrila, 2013] presented a new dataset containing labeled stereo-video images. The image data includes different situations of persons approaching the curb. Ground truth is provided by means of labeled pedestrian bounding boxes, distance measurements estimated from calculated disparity maps and ego-motion data from on-board inertial sensors. Additionally, the authors provide the output from a HOG/linSVM pedestrian detector ([Dalal and Triggs, 2005]) evaluated on regions of interest supplied by a stereo-video based obstacle detection component. For pedestrian action recognition the so-called "time-to-event" (*TTE*) in frames is labeled, identifying, when a person is crossing, starting to cross, bending in or stopping at the curb. In total, 35 training and 32 test scenarios are recorded.

The goal of this work is to address a large number of possible situations pedestrians can interact in daily traffic, to get an overall scene understanding at the end. Therefore, the additional scenario of a walking pedestrian on the sidewalk is also addressed, which is named as "straight". "straight"-samples are extracted out of "bending-in"-scenarios with an adequate time distance to the turning event. Later, the fact states a very high value to be able to differentiate, if a person will continue his/her way on the sidewalk or suddenly bend into the road. When taking all data together into an overall set, one obtains the distribution over addressed scenarios displayed in Table 5.1. The contained samples for "starting"-scenarios

bending in	starting	stopping	crossing	straight	total
23	9	17	18	20	87

Table 5.1: Number of sequences per scenario for Daimler dataset [Schneider and Gavrila, 2013] including straight scenarios.

Figure 5.3: Crossing (left) and stopping scenario (right) from Daimler dataset [Schneider and Gavrila, 2013].

Figure 5.4: Bending in (left) and straight walking scenario (right) from Daimler dataset [Schneider and Gavrila, 2013].

will not be part for further evaluation. Figure 5.3 and 5.4 show some of the evaluated images for each addressed scenario.

5.3 Experiments

5.3.1 Experimental Setup

All sequences are reprocessed in order to calculate the features mentioned in Section 5.1.3. Similar to [Keller and Gavrila, 2014], a uniform noise of up to 10% of the original label bounding box height is added to the height and center to simulate real-world behavior given by a pedestrian tracker. To capture the pedestrians awareness of the underlying scene, continuous head pose angels are calculated using the algorithms of Chapter 3 and Chapter 4. Here, for each of 8 discrete head pose classes boosting cascades including MCT-Features [Fröba and Ernst, 2004] are trained on a large set of manually labeled pedestrian head images with

approximately 2500 samples per class, see the BOSCH data setup in Section 3.4.1. For a comparison, pedestrian dynamics are captured using a similar approach to PHTM [Keller and Gavrila, 2014], where motion histogram features are extracted from dense optical flow. In this work, a public available version of the *TV-L1* flow [Zach et al., 2007, Sanchez et al., 2012] is used. Vehicle ego-motion compensation is achieved by applying an efficient and highly accurate algorithm for visual odometry, cf. [Geiger et al., 2011]. In [Keller and Gavrila, 2014] only flow vectors inside a pedestrian depth mask contribute to the histogram calculation. This mask is derived by considering only disparity measurements within a given pedestrian patch that have a small deviation from the patch-wise calculated median disparity. The required disparity maps over whole frames are calculated using the method of [Geiger et al., 2010].

Without loss of generality, the system is restricted to lateral approaching pedestrians from the right side. There is one sample in the training set and testing set respectively, where a pedestrian is crossing from the left side. For evaluation, both samples are converted into right-to-left-crossings by inverting the extracted features for lateral position, lateral velocity and head pose. Facing the problem of a very low number of samples the use of leave-one-out cross-validation (*LOO*) is proposed. Here, in multiple rounds, models are learned on a complete set including original training- and test data, leaving one particular sample out for testing. Two types of one-vs.-one classifiers are trained to differentiate between "stopping"- and "crossing"- (*SC*) or "bending-in"- and "straight"- (*BS*) scenarios. The corresponding class label sets $\mathcal{Y}$ consist of the events $\mathcal{Y}_{SC} = \{\textit{"crossing"}, \textit{"stopping"}\}$ and $\mathcal{Y}_{BS} = \{\textit{"bending in"}, \textit{"straight"}\}$, respectively. The idea for LDCRF is now to replace one abstract class label by a specified number of latent variables in order to model the extrinsic and intrinsic class dependencies. Additionally, the number of feature functions and model parameters can be controlled by setting a time window for temporal feature dependencies to be learned. If, for instance, the number of hidden states for the trained LDCRF model is chosen to be 3, the following subsets $\mathcal{H} = \{\mathcal{H}_i\}_{i=1,2}$ for a crossing vs. stopping classifier can be defined as $\mathcal{H}_1 = \{\textit{"crossing}_1\textit{"}, \textit{"crossing}_2\textit{"}, \textit{"crossing}_3\textit{"}\}$ and $\mathcal{H}_2 = \{\textit{"stopping}_1\textit{"}, \textit{"stopping}_2\textit{"}, \textit{"stopping}_3\textit{"}\}$. During experiments, LDCRF models are trained with various time window sizes w from 0 to 5 and number of hidden states H per class label from 1 to 10 (cf. Sec. 5.1) plus additional parameters to be set for weight initialization, system regularization and the gradient ascent method. Only the best performing models are analyzed in the following. The original LDCRF training algorithm is adapted to not take data for future frames into account when using a positive window size parameter. This will avoid a system delay during evaluation. Observations taken from a stopping scenario related to TTE values larger than eight frames are members of the crossing class. Indeed, modifying this threshold has strong impact on the course of the performance plots displayed later. The value of eight frames is motivated by the time period assumed for a pedestrian to change intention and action, in this case $8 * 1/16\text{fps} = 0.5s$. The same assumption is made for bending-in scenarios shortly before the turning event.

For evaluation, a sliding window is shifted over a whole pedestrian trajectory to collect frame-based system responses. The goal is to analyze the system's capability to early predict critical situations with a high reliability. Therefore, the system's output is shown, which is the event probability for a pedestrian to stop at the curbside or turning in. In addition averaged classification rates for both scenarios are presented. Interesting and relevant is the system behavior within a short time range around the actual event for both types of scenarios. As proposed by [Keller and Gavrila, 2014] and [Kooij et al., 2014a], the focus is set on the TTE interval [-5,20], where positive TTE values reflect the time period before event occurrence. Plotted are the mean and standard deviation for event probabilities over all tested sequences. For a system realizing pedestrian action/intention-recognition to be used for collision warning or even avoidance, the desired event-probabilities should have the following properties. In an early stage, e.g. 20 to 15 frames prior to the event, the event probabilities should have considerable low values. Towards the occurring event, the probabilities should increase rapidly showing a strong gradient. This behavior will lead to systems less sensitive to false activations. This fact is later considered during interpretation of results.

5.3.2 Results on Pedestrian Intention Recognition

Event Probabilities

Evaluation is done on single features only and on their combined versions. Figure 5.5 and Figure 5.6 show the evaluation results for the best trained SC-models and BS-models respectively. Compared to the single-feature-based models, the combined SC-model (Pos+Vel+Hp) is able to reliably recognize crossing situations that is the stopping probability is approximately zero within the considered time period for most of the test sequences. For stopping scenarios the probability increases continuously towards the stopping event (TTE=0). While at an early stage (TTE=20) for most of the stopping sequences the system still tends to predict a crossing scenario (average stopping probability smaller than 0.025 with a standard deviation near to 0), this behavior rapidly changes for getting closer the actual stopping event (TTE$\in$[0,5]). The combined BS-model shows similar behavior. The velocity components do not seem to have the high impact on an accurate intention recognition for BS-scenarios compared to CS-scenarios. This can be explained by the fact that absolute velocity values do not change that significant for a bending in rather than for a stopping scenario, where the values tend to zero. Another interesting observation is related to the power of the head pose feature. While for the CS-model, the head pose cannot contribute significantly to a performance gain, for the BS-model the opposite is the case. Here, the model only trained on the head pose feature outperforms the one trained on velocity inputs. Having a deeper look at the sequences, this fact can be explained by the presence of a dominant pedestrian head turning towards the oncoming vehicle in most of the bending in situations. This also reflects pedestrian behavior in real-world scenarios. According to Figure 5.7, for the stopping scenarios, the head turn towards the approaching vehicle and focusing of the same mainly occurs less

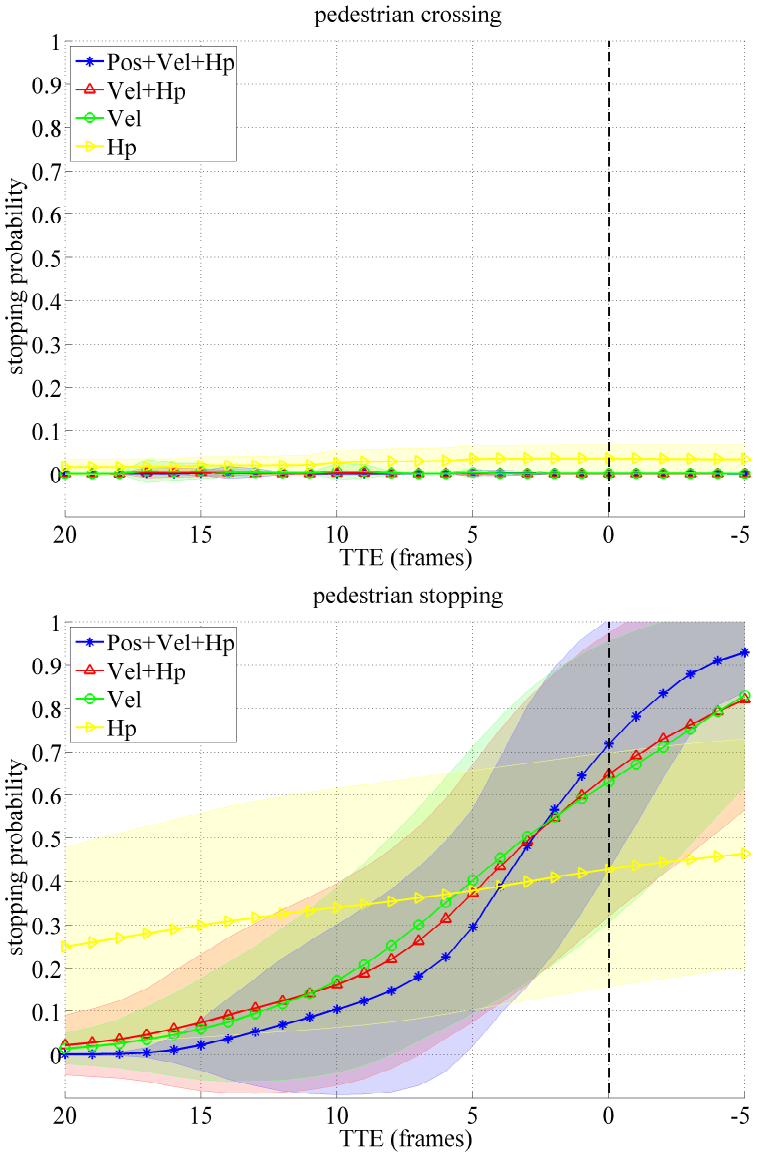


Figure 5.5: Stopping probabilities over TTE (frames) for the SC LDCRF model. Road-crossing scenarios (upper graph) and stopping scenarios (lower graph). Visualized are mean and standard deviation (shaded area) over all tested sequences.

than five frames ($<0.3s$!) before or even significantly after the actual event, which is contradictory to normal behavior demonstrated by the works of [Hamaoka et al., 2013] and [Oxley et al., 1995]. Hence, the expected performance gain of the head pose feature with respect to intention recognition in stopping situations cannot be proven exclusively when evaluating on the Daimler dataset. The motion histogram features introduced by [Keller and Gavrila, 2014] are used in two ways. First, by applying a PHTM-like version (Fig. 2.4) and second by integrating the original features into a LDCRF model, see Figure 5.8.
Compared to the best trained LDCRF model in this work, the PHTM approach (green) results in more unstable intention estimates over the whole dataset especially for crossing situations. Nevertheless, the motion histogram features show high potential for a robust intention recognition in combination with a LDCRF model (red).

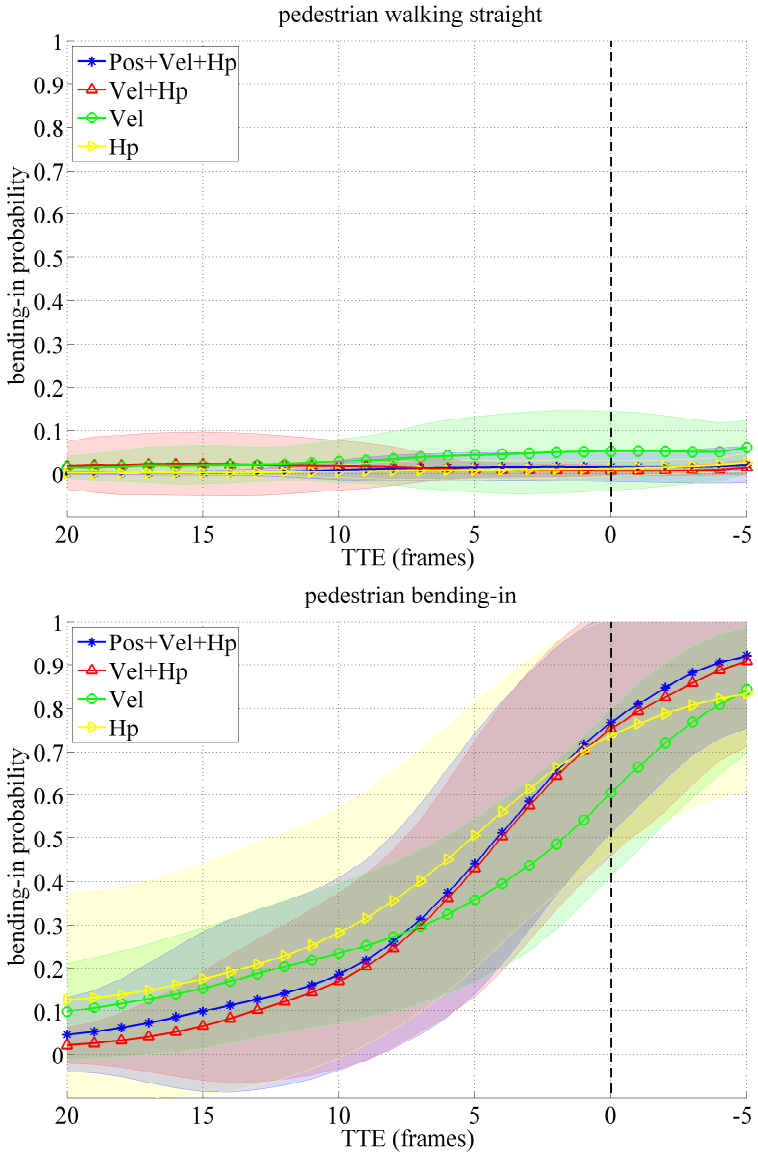


Figure 5.6: Bending-in probabilities over TTE (frames) for the BS LDCRF model. Straight-walking scenarios (upper graph) and bending-in scenarios (lower graph).

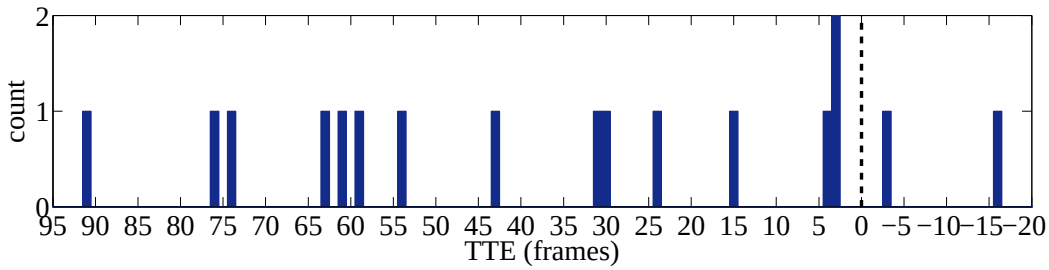


Figure 5.7: First occurrence of frontal head pose over TTE in stopping situations given by the Daimler dataset, [Schneider and Gavrila, 2013]. For some scenarios the expected pedestrian head turn towards the approaching vehicle occurs only after the actual stopping action.

For another reference, standard machine learning approaches were trained – here, SVMs and Random Forests (RF) – to test their suitability compared to LDCRF. For this, single observa-

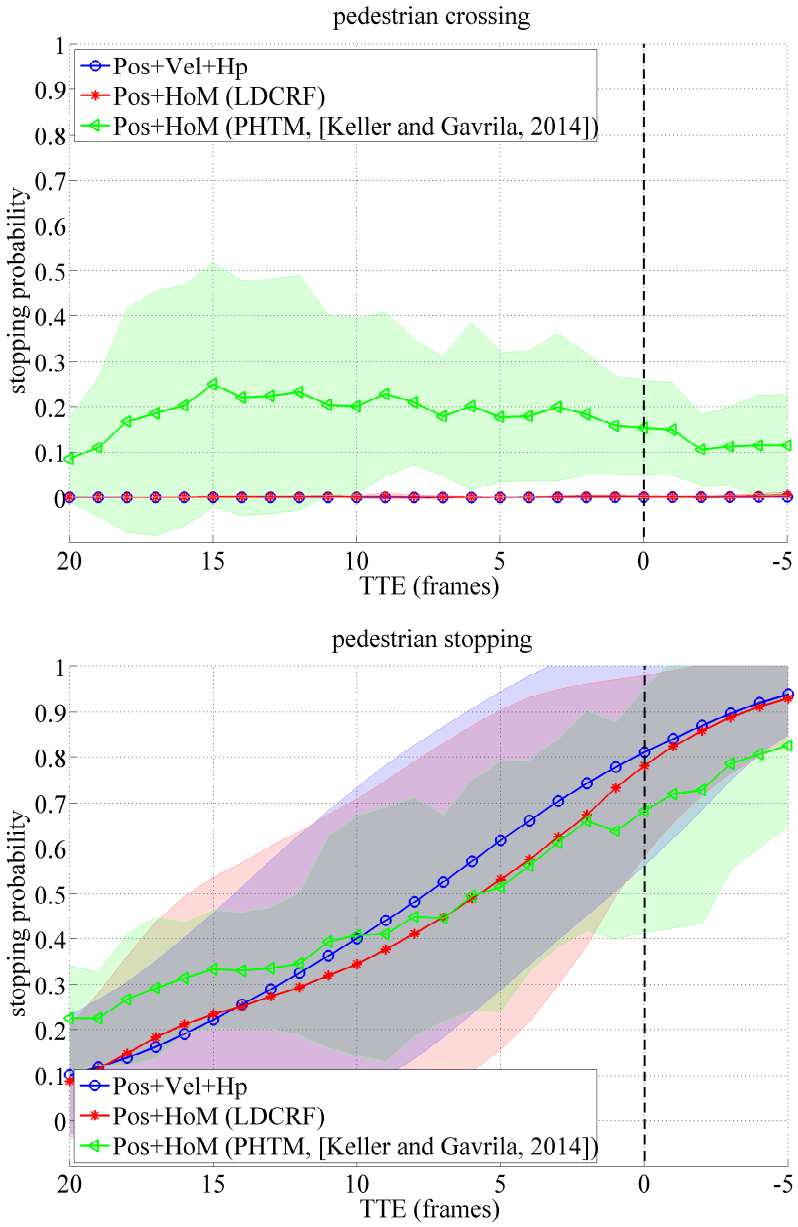


Figure 5.8: Stopping probabilities over TTE (frames) for the best LDCRF (blue), a LDCRF model learned on pedestrian position and motion histograms (red) and the PHTM-like approach [Keller and Gavrila, 2014] (green).

tions are integrated over a time window of 20 frames. Results for the stopping vs. crossing classifiers are given in Figure 5.9. Compared to LDCRF, SVM (green) and RF models (red) result in unstable estimates for the underlying scenarios. At an earlier stage there is still a comparably high confusion between stopping- and crossing-scenarios for both SVM and RF models, whereas the LDCRF results in more stable estimates. A similar behavior is observed for bending-in and straight walking scenarios in Figure 5.10. When considering the probability estimates resulting from the SVM-based classifier, it is hardly possible to differentiate a bending-in- from a straight walking scenario due to weakly increasing probability estimates towards bending-in events. For the Random Forest classifier probability estimates result in very high ranges (high standard deviation) for bending-in situations leading to po-

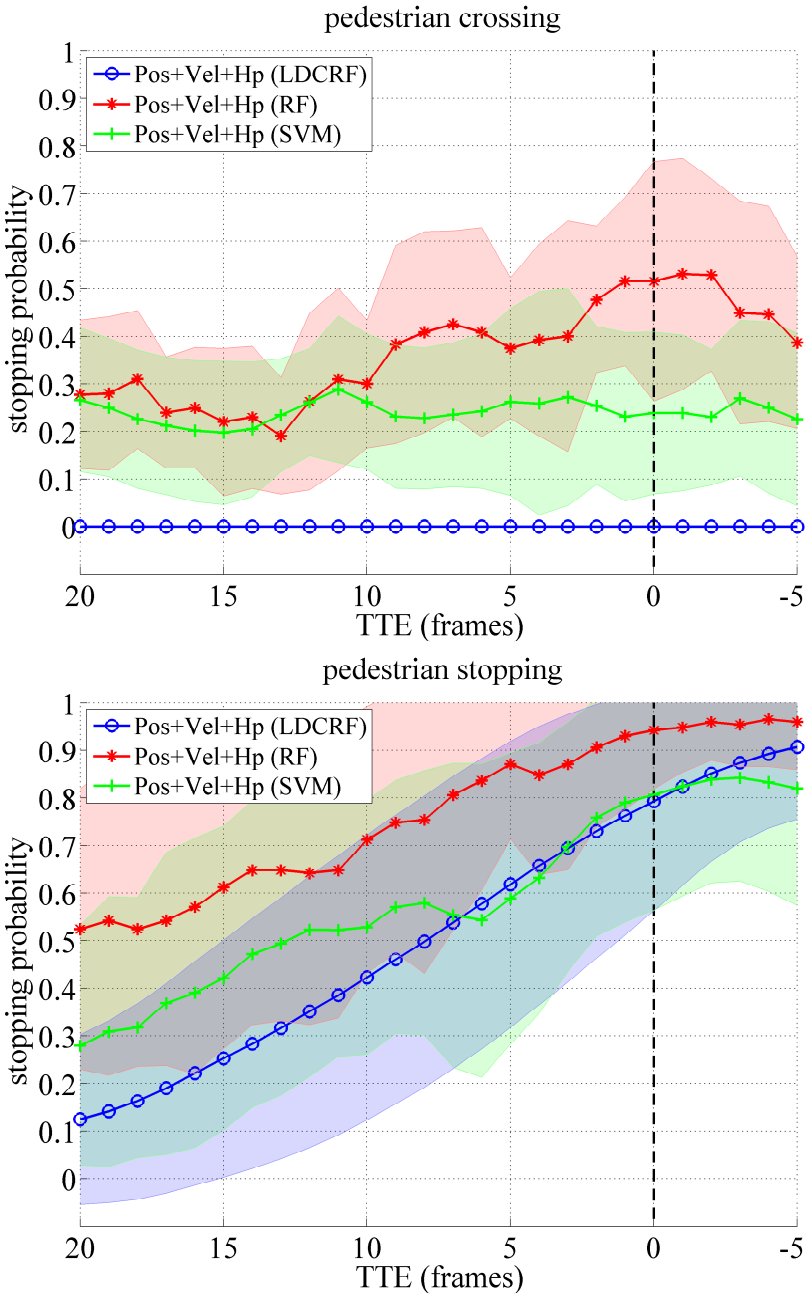


Figure 5.9: Stopping probabilities over TTE (frames) for the best SC LDCRF model compared to other machine learning approaches (SVM/green and Random Forest/red). Crossing scenarios (upper graph) and stopping scenarios (lower graph).

tential confusions till shortly before event occurrence (TTE>5). In comparison, the LDCRF based classifier again results in smoother and more stable estimates. The benefit of using a LDCRF based classifier for intention recognition points out more clearly when analyzing classification rates in the following.

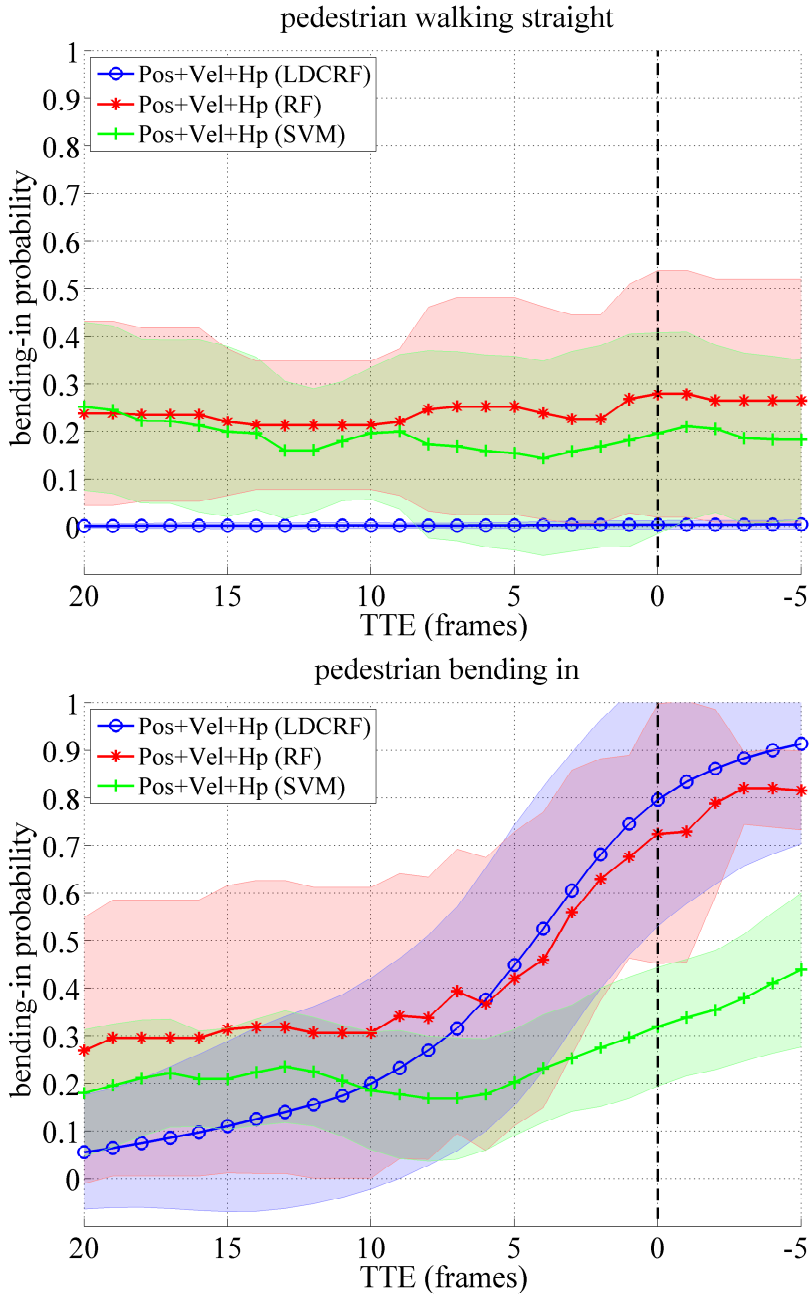


Figure 5.10: Bending-in probabilities over TTE (frames) for the best BS LDCRF model (blue) compared to other machine learning approaches (SVM/green and Random Forest/red). Straight walking scenarios (upper graph) and bending-in scenarios (lower graph).

Classification Rates

For further comparison, classification rates are provided for different feature combinations and classifiers. For this, the stopping- or bending-in-probability values are decisive with regard to the prediction of the correct intention. Similar to *Receiver Operating Characteristic* curves (ROC), used for general detection algorithms, optimal thresholds for the stopping- and bending in probabilities are determined, that lead to optimal performance in terms of a maximized classification rate. The classification rate for a two-class problem can be ex-

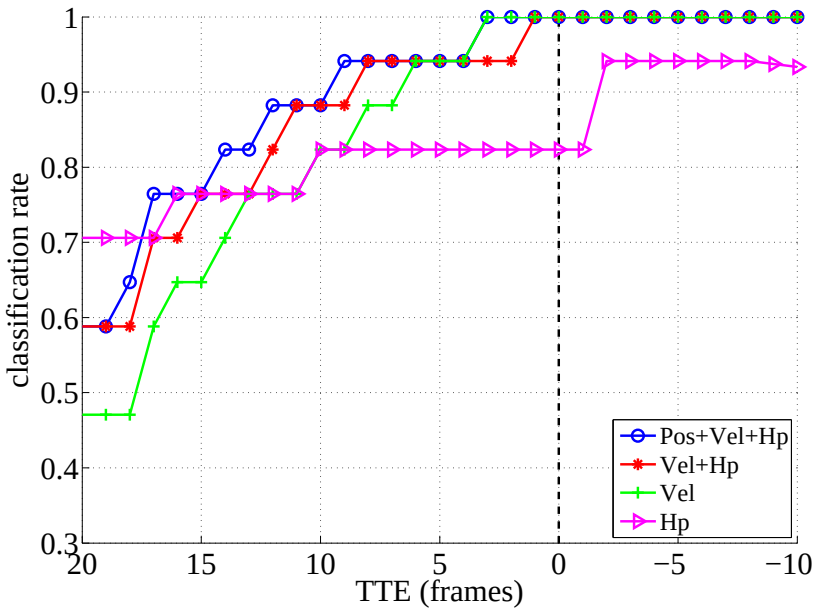


Figure 5.11: LDCRF classification rates in stopping scenarios for different feature combinations.

pressed as

$$\text{classRate} = \frac{TP + TN}{P + N}, \tag{5.14}$$

with TP (true positive) and TN (true negative) the number of correct decisions and P and N the number of positive and negative samples, respectively. Later application of the developed intention recognition approach is a system for autonomous braking. Within this system, the predicted intention will serve as an additional indicator, that a person tends to cross the street or not. In case of a successful predicted stopping intention, the function will avoid the trigger of an emergency braking and therefore has an additional measure to reduce a potential false activation while maintaining a certain degree of value, i.e., safe more pedestrian lives. Concerning the system design, the false prediction of a crossing intention in case of a real stopping pedestrian is the most crucial mistake the system can do. In the context of this work, this would be related to a false negative (FN) stopping intention recognition. Therefore, for determining the optimum thresholds for the stopping probability and deriving a classification rate that particular value is taken that minimizes the false negative rate and hence the miss-classification rate (1-classification rate). The optimum criteria to determine the optimal threshold for the stopping- probability is the minimum sum of single averaged miss-classification rates over the considered time interval TTE $\in$ [-10,20]. The corresponding classification rate then results from equation (5.14). Figure 5.11 shows the classification rates in stopping situations for the trained LDCRF models using different feature combinations. A general observation is that classification rates increase towards the event occurrence (TTE=0), which is the desired behavior. Except of the LDCRF with the head pose as a stand-alone feature (magenta curve) all combined versions achieve 100% classification rate few frames before the stopping event. The benefit of integrating the human head pose can

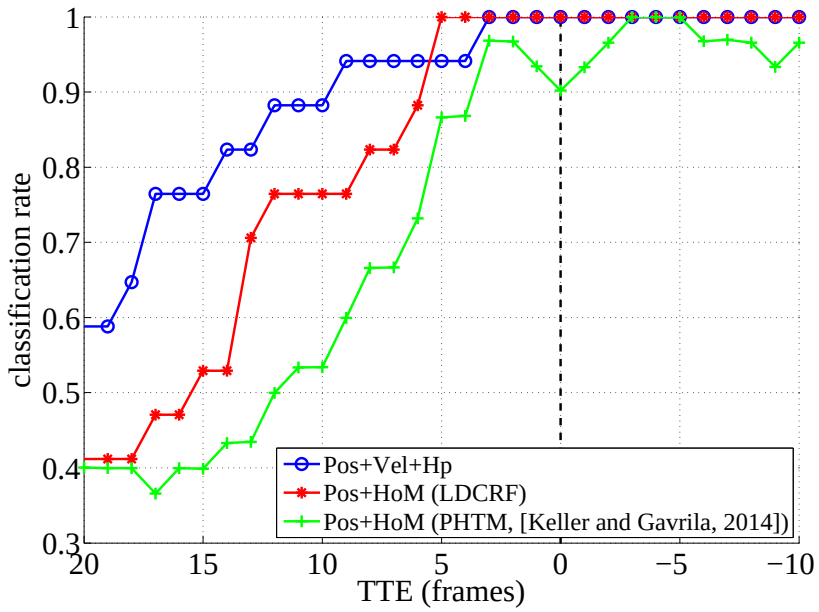


Figure 5.12: Classification rates in stopping scenarios for LDCRF model compared to PHTM-like approach from [Keller and Gavrila, 2014].

clearly be seen, when comparing the classification rates for velocity only (green curve) and combined version (red curve) all over the considered time interval. For instance, at 10 frames prior to the event (TTE=10), combining velocity with head pose achieves 6% higher classification rates than the velocity-only based LDCRF model. Best performance is given by combining relative position, velocity and head pose (blue curve). Here, almost 95% correct intention estimation can be achieved 8 frames before pedestrians are actually stopping.

Figure 5.12 shows a comparison of classification rates with the PHTM-like approach of [Keller and Gavrila, 2014]. The trained LDCRF model (blue curve) outperforms the PHTM-like approach (green curve) in terms of the ability to earlier increasing classification rates and hence earlier successful prediction of a stopping intention. For eight frames before the stopping event, the trajectory matching approach achieves only 77% accuracy in predicting the correct intention, whereas the LDCRF model already achieves 95%. As also observed previously in analyzing the event probabilities, the motion features presented in [Keller and Gavrila, 2014] are well performing for the task of intention recognition in stopping situations. A LDCRF model trained on these features (red curve) outperforms the conventional approach and achieves higher classification rates towards the stopping event. Hence, the probabilistic binary search tree used for the PHTM approach seems to be the limiting factor.

A comparison of classification rates for the trained LDCRF model with conventional machine learning approaches, SVM and Random Forest, is displayed in Figure 5.13.

Both, SVM and RF behave similar towards the stopping event, however do not reach the performance of the LDCRF model. For instance at TTE=5 both, SVM and RF, only score 77% accuracy while the LDCRF model already achieves 95% in correctly predicting the stopping intention. Even after event occurrence for it is not possible to reach a stable 100% classification rate for both conventional machine learning approaches.

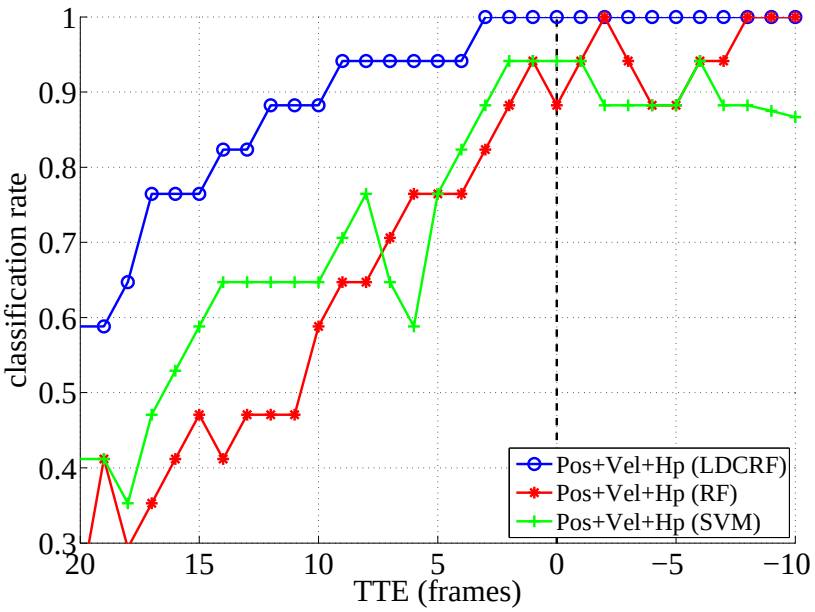


Figure 5.13: Classification rates in stopping scenarios for LDCRF model compared with other machine learning approaches (SVM and Random Forest).

In the bending-in vs. straight case, considerations for finding the optimal threshold for the estimated bending-in probability are slightly different. To increase the benefit for an existing system it is important to have a high classification rate for bending-in intentions while on the other hand achieving a very low miss-classification rate in situations where pedestrians are just walking along the sidewalk. In other words, the situation of a predicted bending-in intention in case of a real straight walking pedestrian would be the crucial case for a potential false activation. For the trained bending-in vs. straight classifier this situation is related to a false positive (FP) predicted bending-in intention. Hence, to determine optimal thresholds for the bending-in vs. straight classifier, the FP-rate is minimized for evaluation. With the resulting operating point, classification rates are calculated correspondingly given by equation (5.14). Figure 5.14 shows the resulting classification rates for the trained LDCRF models using different feature combinations in bending-in situations. Assuming that a bending-in scenario based on velocity as stand-alone feature can only be detected when the event is actually occurring, integrating the human head pose for intention recognition should help significantly. Sequence analysis shows that a dominant head turn occurs within 5 frames before event occurrence. Analyzing the head pose feature only (magenta curve), this head turn is recognized by the LDCRF model and directly assigned to the corresponding bending in intention yielding a strong increase in classification rates 5 frames before the pedestrian turns in. Combining head pose and velocity components (red curve) further improves results especially in situations, where the pedestrian is oncoming and thus not showing the dominant backward head turn. Best performance again is achieved with integration of all derived features including also the pedestrians relative position with respect to the approaching vehicle (blue curve). Due to the challenging scenario of a bending-in pedestrian, a 100% accuracy can only be achieved after the turning action for all trained LDCRF models.

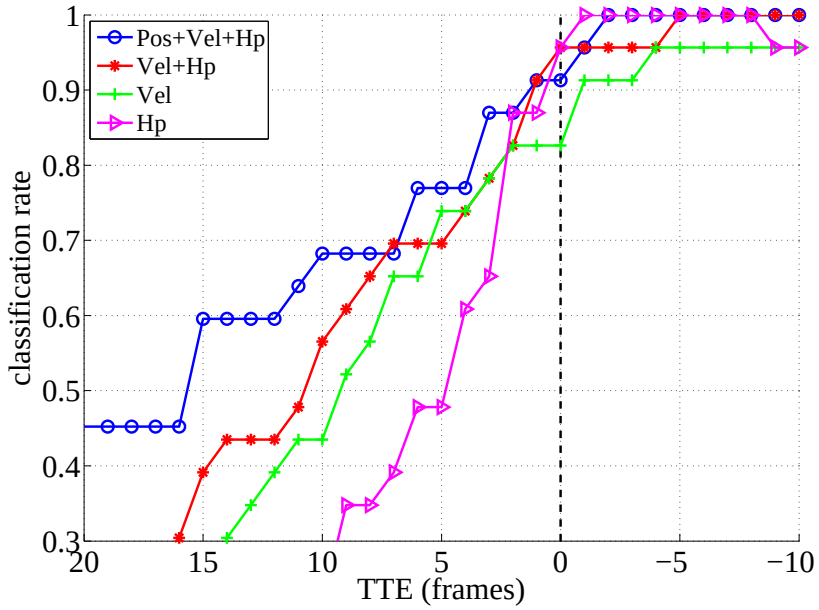


Figure 5.14: LDCRF classification rates in bending-in scenarios for different feature combinations.

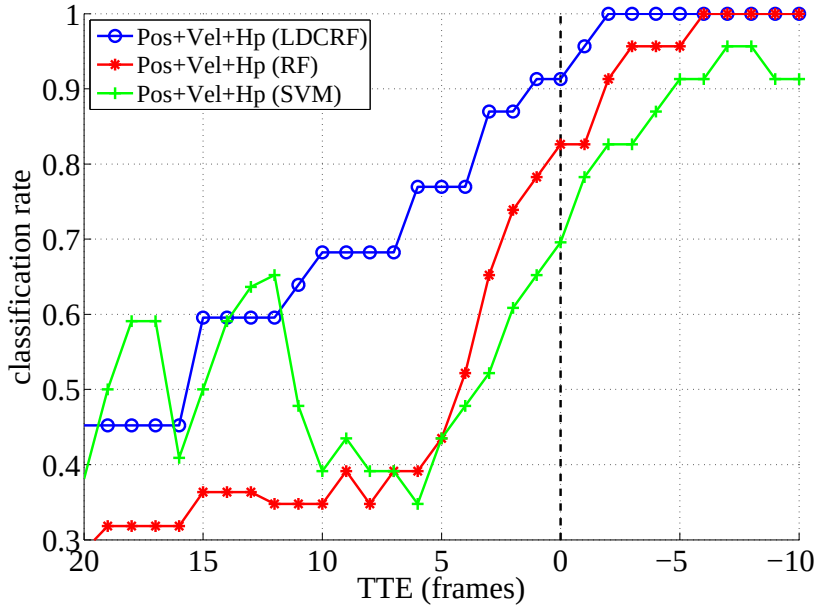


Figure 5.15: Classification rates in bending scenarios for LDCRF model compared with other machine learning approaches (SVM and Random Forest).

A comparison for classification rates with SVMs and Random Forests is given in Figure 5.15. Similar to the stopping vs. crossing models, the LDCRF model outperforms conventional machine learning approaches by means of general higher and more stable classification rates over the considered TTE interval. While the SVM and RF classifier are only able to reliable detect the bending-in action shortly after event occurrence (classification rate $>$90% at TTE$\leq$-2 for RF), the LDCRF model predicts the correct intention shortly before the pedestrian is turning in.

5.4 Conclusion

In this chapter a method to estimate the intention of lateral approaching pedestrians in the domain of intelligent vehicles was presented. Multiple features capturing the pedestrian dynamics and the awareness of the nearby traffic situation were used to learn a LDCRF model. The proposed model has the advantage to automatically learn intrinsic structure and feature dependencies as well as temporal dynamics between different actions. Evaluation of the model in Section 5.3 showed more stable intention estimates for different scenarios compared to other machine learning approaches and a state of the art approach of [Keller and Gavrila, 2014] in terms of smoother evolution of the event probabilities and higher classification rates. The model provides evidence for potential risky situations and therefore can serve for better pedestrian path prediction, cf. Chapter 6, or be directly integrated into a system implementing a pedestrian warning or automatic emergency braking function to reduce false alarms.

6 Pedestrian Path Prediction

This chapter focuses on the integration of the pedestrian intention recognition approach presented in Chapter 5 for an improved path prediction to be used in an AEB function. The use of a controlled Interacting Multiple Model filter is investigated in combination with the presented Latent-Dynamic Conditional Random Field model for the task of intention recognition and path prediction in different scenarios. Again, situations with pedestrians walking along or towards the road curbside on their way to cross, stop or just keep on going in the same direction will be addressed. Pedestrians are detected be the previously introduced stereo-video based object detection and tracking system with additional human head pose estimation. Using extracted features, a potential intention for a tracked pedestrian can be derived, cf. Chap. 5. Now, for each time instance the goal is to predict the pedestrian's future states which is crucial to know in order to react properly in the later AEB function. The path prediction uses an iterative approach, where prediction results for a specified time window are controlled by updated pedestrian intention estimates. Figure. 6.1 displays the adapted system overview.

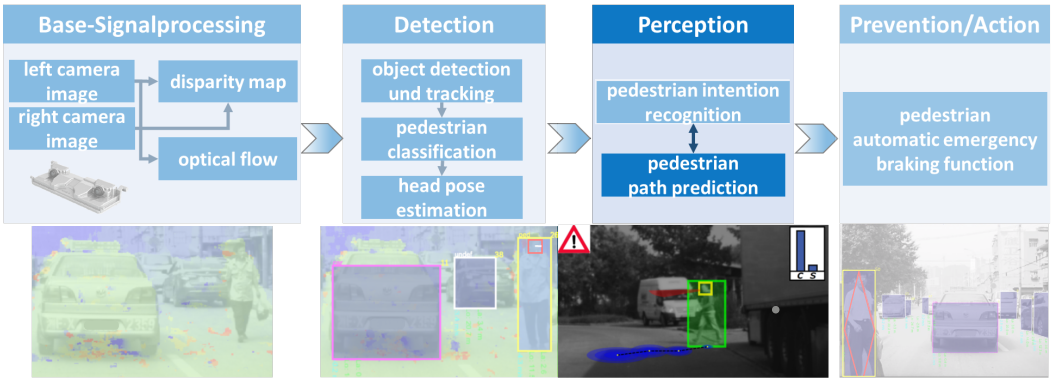


Figure 6.1: System overview. Pedestrians are detected and tracked by a stereo-video based generic obstacle detection and tracking approach and verified by an image intensity-value based classifier. For each pedestrian candidate the head orientation will be estimated from the image data. The previous extracted information serves as input for an iterative intention recognition and path prediction.

Estimated pedestrian intentions will serve as control input for a more reliable and robust path prediction. Similar to [Kooij et al., 2014a], a dynamic system for prediction is effected by latent information, i.e., the LDCRF output has a direct impact on the model probabilities controlling the behavior of the IMM filter. The benefit in choosing the LDCRF model is the

fact that it can work with time series of arbitrary lengths, where additional confidence can be retrieved over temporal integration. Additionally, the IMM filter can handle multiple motion models and hence is not restricted to a simpler linear motion model.
This chapter is structured as follows. Section 6.1 presents the theory about the presented approach for path prediction. Starting with the introduction of the general IMM filter the method for a controlled state prediction by incorporating intention estimates is explained. A detailed analysis of system performance is later performed within the experiments in Section 6.2. Section 6.3 finally concludes this chapter.

6.1 A Controlled Interacting Multiple Model Filter for Pedestrian Path Prediction

In the addressed subject area, it is reasonable to assume that a pedestrian dynamical model can change over time. For example, a pedestrian is able to de-/accelerate or turn within a very short time period. These varying system characteristics are hard to describe with only one single model. Hence, improvements in object tracking can be achieved taking into account the potential dynamical change in motion behavior. With regard to this work, a discrete set of n models is considered, which is denoted by $M = \{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(n)}\}$. Each model $\mathbf{m}^{(j)}$ will have some prior probability $\mu_0^{(j)} = P(\mathbf{m}_0^{(j)})$. Additionally, the probabilities of switching from model i to model j in the next time step are assumed to be known and denoted by $p_{ij} = P(\mathbf{m}_t^{(j)}|\mathbf{m}_{t-1}^{(i)})$. This defines a transition probability matrix of a first order Markov chain characterizing the mode transitions, and hence systems of this type are commonly referred as Markovian switching systems. The next section presents a detailed description of the IMM filter, which can cope with such systems and is used throughout this work.

6.1.1 Interacting Multiple Model Filter

For general tracking tasks with an underlying Markovian switching system, where a tracked object can be subject to multiple motion patterns, the IMM-filter, presented in [Bar-Shalom et al., 2002, Blackman and Popoli, 1999, Li and Jilkov, 2005], turned out to be a computationally efficient and in many cases well performing suboptimal estimation algorithm. The IMM filter can be described by three major steps per time step, namely interaction (mixing), filtering and combination. First, for each time instance the initial filter state condition for each of the considered models is obtained by mixing the state estimates of all model filters from the previous time step under the assumption, that the actual considered model is the best matching one. In the second step regular Kalman filtering steps are applied for each underlying model in order to obtain updated filter states after incorporating actual system measurements. Finally, the combination of all updated filter states in form of weighted sum yields the final state estimate and covariance for the current time step. Here, the weights

related to the model probabilities are calculated in the earlier filtering step and depend on the fitness of actual observations with respect to the model assumptions. A detailed notation of the IMM algorithm for one cycle is given as follows.

Interaction

In a first step, the mixing probabilities $\mu_t^{(i|j)}$ for each model $\mathbf{m}^{(i)}$ and $\mathbf{m}^{(j)}$ are calculated as

$$\mu_t^{(i|j)} = \frac{1}{\bar{c}_j} p_{ij} \mu_{t-1}^{(i)}, \tag{6.1}$$

with a normalization factor

$$\bar{c}_j = \sum_{i=1}^{n} p_{ij} \mu_{t-1}^{(i)},\ j = 1, \dots n. \tag{6.2}$$

$\mu_{t-1}^{(i)}$ is the probability of model $\mathbf{m}^{(i)}$ in the time step $t-1$. Now the mixed inputs for each model filter $j = 1, \dots, n$ can be computed as

$$\hat{\mathbf{x}}_{t-1}^{(0j)} = \sum_{i=1}^{n} \mu_t^{(i|j)} \hat{\mathbf{x}}_{t-1}^{(i)}, \tag{6.3}$$

$$\mathbf{P}_{t-1}^{(0j)} = \sum_{i=1}^{n} \mu_t^{(i|j)} \times \left\{ \mathbf{P}_{t-1}^{(i)} + (\hat{\mathbf{x}}_{t-1}^{(i)} - \hat{\mathbf{x}}_{t-1}^{(0j)}) \times (\hat{\mathbf{x}}_{t-1}^{(i)} - \hat{\mathbf{x}}_{t-1}^{(0j)})' \right\}, \tag{6.4}$$

where $\hat{\mathbf{x}}_{t-1}^{(i)}$ and $\mathbf{P}_{t-1}^{(i)}$ are the updated mean and covariance for model $\mathbf{m}^{(i)}$ at time step $t-1$.

Filtering

After interaction, the actual filtering for each model $\mathbf{m}^{(j)}$ is done as

$$\left[\hat{\tilde{\mathbf{x}}}_t^{(j)}, \tilde{\mathbf{P}}_t^{(j)} \right] = \text{EKF}_p(\hat{\mathbf{x}}_{t-1}^{(0j)}, \mathbf{P}_{t-1}^{(0j)}, \mathbf{A}_{t-1}^{(j)}, \mathbf{Q}_{t-1}^{(j)}), \tag{6.5}$$

$$\left[\hat{\mathbf{x}}_t^{(j)}, \mathbf{P}_t^{(j)} \right] = \text{EKF}_u(\hat{\tilde{\mathbf{x}}}_t^{(j)}, \tilde{\mathbf{P}}_t^{(j)}, \mathbf{z}_t, \mathbf{h}_t^{(j)}, \mathbf{H}_t^{(j)}, \mathbf{R}_t^{(j)}), \tag{6.6}$$

where Eq. (6.5) and (6.6) denote the prediction and update steps of the standard extended Kalman filter for non-linear systems with $\text{EKF}_p(\cdot)$ and $\text{EKF}_u(\cdot)$, correspondingly, cf. [Bar-Shalom et al., 2002]. The matrices $\mathbf{A}^{(j)}$, $\mathbf{Q}^{(j)}$ and $\mathbf{R}^{(j)}$ define transition matrix, process noise and measurement noise for each model $\mathbf{m}^{(j)}$. The vector $\mathbf{z}_t$ includes the actual system measurements at the t-th time step, while $\mathbf{h}^{(j)}$ and $\mathbf{H}^{(j)}$ denote model $\mathbf{m}^{(j)}$'s in general non-linear measurement model and its corresponding derivative. In addition to mean and covariance, the measurement likelihood for each filter is determined by

$$\Lambda_t^{(j)} = \mathcal{N}(\mathbf{r}_t^{(j)}; 0, \mathbf{S}_t^{(j)}), \tag{6.7}$$

where $\mathbf{r}_t^{(j)} = \mathbf{z}_t - \mathbf{h}_t^{(j)}(\hat{\tilde{\mathbf{x}}}_t^{(j)})$ is the measurement residual and $\mathbf{S}_t^{(j)} = \mathbf{H}_t^{(j)}\tilde{\mathbf{P}}_t^{(j)}(\mathbf{H}_t^{(j)})' + \mathbf{R}_t^{(j)}$ it's covariance for model $\mathbf{m}^{(j)}$ in the Kalman Filter update step. Depending on the fact, how well the measurements fit into the model assumptions, the probabilities of each model $\mathbf{m}^{(j)}$ at time step t are calculated as

$$\mu_t^{(j)} = \frac{1}{c}\Lambda_t^{(j)}\bar{c}_j, \tag{6.8}$$

with the normalization factor

$$c = \sum_{i=1}^{n} \Lambda_t^{(i)}\bar{c}_i. \tag{6.9}$$

Combination

In the final stage of the IMM filter the combined estimate for the state mean and covariance are derived from the single model estimates as

$$\hat{\mathbf{x}}_t = \sum_{j=1}^{n} \mu_t^{(j)}\hat{\mathbf{x}}_t^{(j)}, \tag{6.10}$$

$$\mathbf{P}_t = \sum_{j=1}^{n} \mu_t^{(j)} \times \left\{\mathbf{P}_t^{(j)} + (\hat{\mathbf{x}}_t^{(j)} - \hat{\mathbf{x}}_t) \times (\hat{\mathbf{x}}_t^{(j)} - \hat{\mathbf{x}}_t)'\right\}. \tag{6.11}$$

6.1.2 Dynamical Models

In this work three different dynamic models $\mathbf{m}^{(i)}, i = 1, \ldots 3$, are integrated into an IMM filter, motivated by the scenarios involving a crossing, stopping or turning pedestrian. These models are well known as constant velocity (CV), constant position (CP) and coordinated turn (CT), see [Bar-Shalom et al., 2002, Schneider and Gavrila, 2013]. For each model the pedestrian state $\mathbf{x}_t$ at the actual time step t is defined to include lateral and longitudinal relative positions (x_t, z_t), lateral and longitudinal absolute velocities $(\dot{x}_t, \dot{z}_t)$ and a turn rate ω_t, i.e., $\mathbf{x}_t = (x_t, z_t, \dot{x}_t, \dot{z}_t, \omega_t)'$. The different dynamical models will be explained in detail in the following.

Constant Velocity Model (CV)

The state vector for the CV model takes the following form $\mathbf{x}_t = (x_t, z_t, \dot{x}_t, \dot{z}_t, \omega_t = 0)'$. Using Newton mechanics, the dynamical model for a constant velocity assumption can be written as

$$\mathbf{x}_{t+1} = \underbrace{\begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{F}^{\mathrm{CV}}} \mathbf{x}_t + \mathbf{q}^{\mathrm{CV}}, \tag{6.12}$$

with the Gaussian process noise parameters for velocity components and turn rate $\mathbf{q}_t^{\text{CV}} \sim \mathcal{N}(\mathbf{0}, I_{5\times5}(0, 0, \sigma_{\dot{x}}^2, \sigma_{\dot{z}}^2, 0)')$.

Constant Position (CP)

For the CP model, the state vector has the following form $\mathbf{x}_t = (x_t, z_t, \dot{x}_t = 0, \dot{z}_t = 0, \omega_t = 0)'$. The discrete form for the dynamic model of a stationary pedestrian can be written as

$$\mathbf{x}_{t+1} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{F}^{\text{CP}}} \mathbf{x}_t + \mathbf{q}^{\text{CP}}, \tag{6.13}$$

with the Gaussian process noise parameters $\mathbf{q}_t^{\text{CP}} \sim \mathcal{N}(\mathbf{0}, I_{5\times5}(\sigma_x^2, \sigma_z^2, 0, 0, 0)')$.

Coordinated Turn Model (CT)

The discretization for the coordinated turn model can be written as

$$\mathbf{x}_{t+1} = \underbrace{\begin{pmatrix} 1 & 0 & \frac{\sin(\omega_t\Delta t)}{\omega_t} & \frac{\cos(\omega_t\Delta t)-1}{\omega_t} & 0 \\ 0 & 1 & \frac{1-\cos(\omega_t\Delta t)}{\omega_t} & \frac{\sin(\omega_t\Delta t)}{\omega_t} & 0 \\ 0 & 0 & \cos(\omega_t\Delta t) & -\sin(\omega_t\Delta t) & 0 \\ 0 & 0 & \sin(\omega_t\Delta t) & \cos(\omega_t\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{F}_t^{\text{CT}}} \mathbf{x}_t + \mathbf{q}^{\text{CT}}, \tag{6.14}$$

where $\mathbf{q}_t^{\text{CT}} \sim \mathcal{N}(\mathbf{0}, I_{5\times5}(0, 0, \sigma_{\dot{x}}^2, \sigma_{\dot{z}}^2, \sigma_{\omega}^2)')$ encapsulates the process noise for the turn rate parameter. This model is, despite the matrix form, non-linear with respect to the turn rate state parameter ω. Therefore before usage within the IMM filter the Jacobian $\partial\mathbf{F}_t^{\text{CT}}/\partial\mathbf{x}_t$ at each time step t has to be computed for linearization, cf. [Bar-Shalom et al., 2002] and [Blackman and Popoli, 1999]. With $a := \omega_t\Delta t$ the Jacobian can be written as

$$\left.\frac{\partial\mathbf{F}_t^{\text{CT}}}{\partial\mathbf{x}}\right|_{\mathbf{x}_t} = \begin{pmatrix} 1 & 0 & \frac{\sin(a)}{\omega_t}\Delta t & \frac{\cos(a)-1}{\omega_t}\Delta t & [\frac{\cos(a)}{\omega_t}\Delta t - \frac{\sin(a)}{\omega_t^2}]\dot{x}_t - [\frac{\sin(a)}{\omega_t}\Delta t + \frac{\cos(a)-1}{\omega_t^2}]\dot{z}_t \\ 0 & 1 & \frac{1-\cos(a)}{\omega_t}\Delta t & \frac{\sin(a)}{\omega_t}\Delta t & [\frac{\sin(a)}{\omega_t}\Delta t - \frac{1-\cos(a)}{\omega_t^2}]\dot{x}_t + [\frac{\cos(a)}{\omega_t}\Delta t - \frac{\sin(a)}{\omega_t^2}]\dot{z}_t \\ 0 & 0 & \cos(a) & -\sin(a) & -\Delta t\sin(a)\dot{x}_t - \Delta t\cos(a)\dot{z}_t \\ 0 & 0 & \sin(a) & \cos(a) & \Delta t\cos(a)\dot{x}_t - \Delta t\sin(a)\dot{z}_t \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.15}$$

6.1.3 Measurement Model

Measurements result from a pedestrian detector applied on image sequences recorded with a stereo camera system. A measurement vector $\mathbf{z}_t = (u_t, d_t)$ is derived from the center foot-point $\mathbf{p}_t^f = (u_t, v_t)$ and the median disparity d_t of a pedestrian bounding box in the image. Following [Schneider and Gavrila, 2013] for a 3D point in camera coordinates on the ground-plane $(x_c, 0, z_c)$, the measurement model $\mathbf{h} = (h_1, h_2)'$ can be written as

$$\mathbf{z} = \begin{pmatrix} u \\ d \end{pmatrix} = \begin{pmatrix} h_1(x_c, z_c) \\ h_2(x_c, z_c) \end{pmatrix} = \begin{pmatrix} u_0 + fx_c/z_c \\ fB/z_c \end{pmatrix}, \tag{6.16}$$

with principal point u_0, camera constant f and stereo baseline B. To use the nonlinear measurement model $\mathbf{h}$ within the IMM filter framework the Jacobian $\mathbf{H}_t = \partial \mathbf{h}_t / \partial \mathbf{x}$ has to be provided as well, that is

$$\left. \frac{\partial \mathbf{h}_t}{\partial \mathbf{x}} \right|_{\mathbf{x}_t} = \begin{pmatrix} f/z_t & -fx_t/z_t^2 & \mathbf{0}_{1\times 3} \\ 0 & -fB/z_t^2 & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 3} \end{pmatrix} \tag{6.17}$$

6.1.4 Incorporating Pedestrian Intention Recognition using Latent-Dynamic Conditional Random Fields

To control the model transitions in the presented IMM filter results from the intention recognition system using Latent-Dynamic Conditional Random Fields from Chapter 5 are incorporated. The task of intention recognition is to assign intention labels $\boldsymbol{\eta} = \{\eta_1, \eta_2, \ldots, \eta_T\}$ that best explain a sequence of observations $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ for a range of T time steps. The LDCRF model is defined as

$$P(\boldsymbol{\eta}|X; \boldsymbol{\theta}) = \sum_{\mathbf{h}:\forall h_t \in \mathcal{H}_{\eta_t}} \frac{1}{\mathcal{Z}(X, \boldsymbol{\theta})} \exp\left(\sum_k \theta_k \cdot F_k(\mathbf{h}, X) \right), \tag{6.18}$$

with the partition function

$$\mathcal{Z}(X, \boldsymbol{\theta}) = \sum_{\mathbf{h}} \exp\left(\sum_k \theta_k \cdot F_k(\mathbf{h}, X) \right) \tag{6.19}$$

and functionals $F_k(\mathbf{h}, X)$ being a linear combination of feature functions modeling the internal structure of a particular class label and the external dependencies between different classes. Using a larger set of observations from labeled sequences as training data, the LDCRF is able to learn the optimal parameters $\boldsymbol{\theta}^*$ for an intention recognition model and given a new observation sequence X to infer the intention labels $\boldsymbol{\eta}$. Here, the trained models from Chapter 5 to recognize stopping, crossing, bending in and straight moving intentions are

used. For each time step t the probabilities $\bar{P}_t^{(K_i)}, i = 1, \ldots, 3$ related to each dynamical model integrated within the IMM filter from Section 6.1.2 are derived from intention probability estimates by summing up the LDCRF's marginal probabilities according to the set of hidden states $\mathcal{H}_{\eta_t^{(k)}}$ related to class label $\eta_t^{(k)}$, that is

$$\bar{P}_t^{(K_i)} := \frac{1}{|K_i|} \sum_{k \in K_i} P(\eta_t^{(k)} | X_{1:t}, \boldsymbol{\theta}^*). \tag{6.20}$$

The set of intention label indices K_i includes the relation between intention and corresponding motion model. I.e., a "bending in"-intention relates to the CT model, the stopping intention to the CP model and the straight or crossing intention to the CV model, respectively.

6.1.5 Pedestrian Path Prediction

Given the results from the pedestrian IMM tracking and the LDCRF, path prediction will be performed as follows. In general, future pedestrian states can be easily derived by applying the prediction functions given the process models from the IMM filter. Depending on future pedestrian behavior, one of the different models within the filter can be better suited for a more accurate path prediction than others. Unfortunately, the IMM filter is not able to manage that in the absence of additional state measurements. To overcome this problem in order to obtain a more precise prediction, estimates for pedestrian intention from the LDCRF model will be incorporated. For this, the IMM interaction, prediction, and combination steps are applied over a fixed time horizon of $\mathcal{T}$ future time steps. In between the temporary single model probabilities $\bar{\mu}^{(i)}$, $i = 1, \ldots, 3$ will be updated with the probabilities resulting from intention recognition, that is

$$\bar{\mu}_{t,0}^{(i)} := \mu_t^{(i)}, \tag{6.21}$$

$$\bar{\mu}_{t,\tau}^{(i)} := \bar{\mu}_{t,\tau-1}^{(i)} \bar{P}_{\tau-1}^{(K_i)} \left(\sum_{j=1}^{3} \bar{\mu}_{t,\tau-1}^{(j)} \bar{P}_{\tau-1}^{(K_j)} \right)^{-1}, \tag{6.22}$$

for $\tau = 1, \ldots, \mathcal{T}$ with $\mu_t^{(i)}$ from equation (6.8) and $\bar{P}_\tau^{(K_i)}$ from equation (6.20). One has to mention that, during prediction, the iteratively updated estimates on position and velocity are directly looped back for a more precise intention recognition, while the head pose estimate remains constant on the latest obtained measurement. This holds under the assumption that the essential pedestrian head movement occurs within a sufficiently long time interval (>1 Second) before one of the addressed actions actually occurs.

6.1.6 Features for Pedestrian Intention Recognition

The trained LDCRF models presented in Chapter 5 require feature inputs extracted from pedestrian dynamics as well as the pedestrian's situational awareness in terms of the human head pose. Similar to the procedure there, these features are extracted using the measurements from an on-board stereo-video based pedestrian tracking and head pose estimation system for each frame and concatenated into a time-series of T frames. The pedestrian's relative position (x_t, z_t) and absolute velocity components $(\dot{x}_t, \dot{z}_t)$ at time instance t will be directly taken from the related system state of the IMM filter presented in Section 6.1.1. For human head pose estimation, the system presented in the chapters 3 and 4 provides frame-wise stable continuous angle measurements $\theta_t^H \in [-180, 180)$ resulting from a tracking module integrating multiple head pose classifiers related to a specified pan angle range. Therefore, for intention recognition, the time-series X of features over T frames is then given by

$$X = \{(x_t, z_t, \dot{x}_t, \dot{z}_t, \theta_t^H)'\}_{t=1,\ldots,T}. \tag{6.23}$$

6.2 Experiments

This section presents results for pedestrian path prediction. Similar to the experiments in Chapter 5, evaluation will be performed on the public available dataset presented by [Schneider and Gavrila, 2013], see Section 5.2. First the setup of the system parameters is explained. Then, evaluation over time-to-event is performed in comparison to a state of the art method presented by [Keller and Gavrila, 2014]. In addition, also the scenarios for straight walking and bending pedestrians are addressed.

6.2.1 Experimental Setup

All sequences are reprocessed to extract the features mentioned in Section 6.1.6. Similar to [Keller and Gavrila, 2014] and the previous chapter, a uniform noise of up to 10% of the original height of the labeled bounding boxes is added to their height and center to simulate real-world performance. To capture the pedestrians awareness of the underlying scene, continuous head pose angels are calculated using the algorithms of Chapter 3 and 4 with the identical training setup as presented in Section 5.3. As mentioned before, the best performing LDCRF models from Chapter 5 are integrated for intention recognition. These consist of the two models for *stopping* against *crossing* (SC) and *bending-in* against *straight* (BS) scenarios, respectively. In the presented controlled IMM filter for path prediction, the single model probabilities for the CV model $\bar{\mu}_{t,\tau}^{(\mathrm{CV})}$ are calculated by averaging the intention probabilities for *crossing* and *straight* from the SC- and the BS-LDCRF models, respectively, whereas the model probabilities for CP ($\bar{\mu}_{t,\tau}^{(\mathrm{CP})}$) and CT ($\bar{\mu}_{t,\tau}^{(\mathrm{CT})}$) are directly updated by the intention probabilities for *stopping* and *bending in*. Following [Schneider and Gavrila, 2013], the relevant

measurement noise		process noise						
		CV		CP		CT		
σ_u	σ_d	$\sigma_{\dot{x}}$	$\sigma_{\dot{z}}$	σ_x	σ_z	$\sigma_{\dot{x}}$	$\sigma_{\dot{z}}$	σ_ω
6.15	0.32	0.9	0.9	0.1	0.1	0.4	0.4	0.9

Table 6.1: Optimized IMM model parameters presented in [Schneider and Gavrila, 2013].

IMM model parameters from Table 6.1 are used. There, the measurement noise parameters for σ_u and σ_d are derived statistically by calculating the displacement error of a pedestrian bounding box with respect to the corresponding ground truth label in image and disparity space. Process noise parameters are optimized for each filter model with a parameter search in discrete space. In this work, the same principle is applied on the training data subset for optimizing the CP model parameters. For this, the objective function

$$\arg\min_{\sigma_{x/z}} \frac{1}{T(\mathcal{T}+1)} \sum_{t=1}^{T} \sum_{\tau=0}^{\mathcal{T}} \text{MSE}(\mathbf{x}_t(\tau)) \tag{6.24}$$

is minimized, where MSE$(\cdot)$ calculates the mean-squared error for an estimated or predicted position compared to ground truth. For τ=0, $\mathbf{x}_t(\tau)$ is the estimated filter state while for a positive $\tau = 1, \ldots, \mathcal{T}$, $\mathbf{x}_t(\tau)$ is the τ-th predicted state at time instance t. Additionally, the CP model parameters are optimized using only trajectory snippets around the stopping event (TTE $\in$ [10,-50]). In addition, the model transition probabilities are determined as described by [Blackman and Popoli, 1999]. The diagonal elements p_{ii} of the transition matrix stating the probability, that the tracked object stays in model $\mathbf{m}_i$, $i = 1, \ldots n$, are related to the mean sojourn times τ_i and the cycle time T_c, that is

$$p_{ii} = 1 - \frac{T_c}{\tau_i} \tag{6.25}$$

[Schneider and Gavrila, 2013] present mean sojourn times based on different pedestrian movement patterns for the used dataset as given in Table 6.2.
Transition probabilities p_{ij} between different models can be estimated by

$$p_{ij} = \frac{n_{ij}}{n_i}(1 - p_{ii}), \tag{6.26}$$

Motion type	mean sojourn time τ_i (s)
straight walking	6.66
maneuver (starting/stopping)	1.67
turning	2.50

Table 6.2: Mean estimated sojourn times for Daimler dataset presented in [Schneider and Gavrila, 2013].

where n_{ij} denotes the number of transitions from model $\mathbf{m}_i$ to $\mathbf{m}_j$ and n_i the number of all transitions from model $\mathbf{m}_i$. In general the values $\{p_{ij}\}_{i,j}$ have to be determined on a larger validation set. However in this work, the transition probabilities between different models are estimated under heuristic assumptions but prove to achieve sufficiently accurate results in later experiments. Here, the fact is considered, that the transition from stopping to walking is more probable than from stopping to bending. This results in the overall transition probability matrix

$$\{p_{ij}\}_{i,j=1,\ldots,3} = \begin{pmatrix} 0.99 & 0.005 & 0.005 \\ 0.03 & 0.96 & 0.01 \\ 0.01 & 0.01 & 0.98 \end{pmatrix}. \tag{6.27}$$

For comparison, pedestrian path prediction is evaluated using an approach similar to PHTM [Keller and Gavrila, 2014], where motion histogram features are extracted from dense optical flow within a pedestrian depth mask. To calculate the optical flow fields a public available version of the *TV-L1* flow [Zach et al., 2007] is used. Stereo disparity maps are computed with the method of [Geiger et al., 2010]. The efficient and highly accurate algorithm for visual odometry from [Geiger et al., 2011] serves for vehicle ego-motion compensation. The system's output behavior is evaluated within a short time range around the occurring event of a crossing, stopping or turning in pedestrian. As proposed in literature [Keller and Gavrila, 2014, Kooij et al., 2014a], focus for evaluation is set on the TTE interval [-10,20], which relates to 20 frames prior to the event and 10 frames after event occurrence. Following [Kooij et al., 2014a] and depending on later application within an AEB function, for path prediction evaluation the Euclidean distance between lateral predicted expected position $\hat{x}_{t+\mathcal{T}}$ and lateral ground truth position $G_{t+\mathcal{T}}$ is calculated, that is

$$\text{err}_{t,\mathcal{T}} := |\hat{x}_{t+\mathcal{T}} - G_{t+\mathcal{T}}| \tag{6.28}$$

defines one performance metric. In addition the *predictive log-likelihood*

$$\text{predll}_{t,\mathcal{T}} := \log\left[p_{t,\mathcal{T}}(G_{t+\mathcal{T}})\right], \tag{6.29}$$

with

$$p_{t,\mathcal{T}}(x_{t+\mathcal{T}}) := p(x_{t+\mathcal{T}}|\mathbf{z}_{1:t}) \tag{6.30}$$

and

$$x_{t+\mathcal{T}} \sim p(x_{t+\mathcal{T}}|\mathbf{z}_{1:t}), \tag{6.31}$$

is investigated, which provides information about the uncertainty of state prediction. The prediction time window $\mathcal{T}$ is set to 16 frames which corresponds to forecasting the pedestrian's position 1 second ahead.

		crossing		stopping	
		TTE∈[-10, 20]	TTE=0	TTE∈[-10, 20]	TTE=0
IMM	Mean	0.20	0.25	0.33	0.31
	± Std	0.23	0.22	0.22	0.20
PHTM-like [Keller and Gavrila, 2014]	Mean	0.25	**0.21**	0.34	0.31
	± Std	0.29	0.21	0.29	0.27
IMM-LDCRF	Mean	**0.19**	0.23	**0.29**	**0.14**
	± Std	0.22	0.21	0.21	0.18
IMM-IntGT	Mean	0.18	0.23	0.14	0.07
	± Std	0.22	0.21	0.13	0.09

Table 6.3: Mean lateral prediction error (m) for crossing and stopping scenarios using a prediction horizon of 16 frames

6.2.2 Results on Pedestrian Path Prediction

Crossing vs. Stopping

First, situations with a stopping or crossing pedestrian are analyzed. Figure 6.2 shows the prediction errors in lateral position for these two kind of scenarios. Plotted are the mean error and its standard deviation over all scenario-related sequences. For detailed numbers, the lateral prediction errors averaged over the considered time interval and at the time of event occurrence (TTE=0) are displayed in Table 6.3. The proposed LDCRF-controlled IMM filter (IMM-LDCRF, eq. (6.22)) is compared with the conventional IMM filter prediction (IMM, eq. (6.5)) and the forecasted positions using a controlled IMM filter, where the ground truth intention serves as an input (IMM-IntGT). The latter approach represents an optimum version of a controlled IMM filter. Additionally, a reimplementation of the PHTM method [Keller et al., 2011c,Keller and Gavrila, 2014] serves as a baseline algorithm. As one would expect, for crossing pedestrians (Figure 6.2a), no significant benefit can be observed compared to other approaches, when applying the presented controlled IMM filter. This is due to the fact, that the pedestrian motion state remains homogeneous and therefore is not rapidly changing such that the conventional IMM filter still is able to provide accurate predictions for future pedestrian positions. The PTHM-like approach performs slightly worse on average. There are two peaks in the plots, where path prediction results in higher errors for all methods. In these two scenarios the pedestrian is appearing out of occlusion and hence causing inaccuracies during filter initialization and the following path prediction. However, for all different approaches the mean lateral path prediction error lies within a small range around 20cm. Shortly after event occurrence (TTE<0), conventional IMM filter predictions and its controlled versions coincide with the optimal predictions resulting from a ground truth control input.

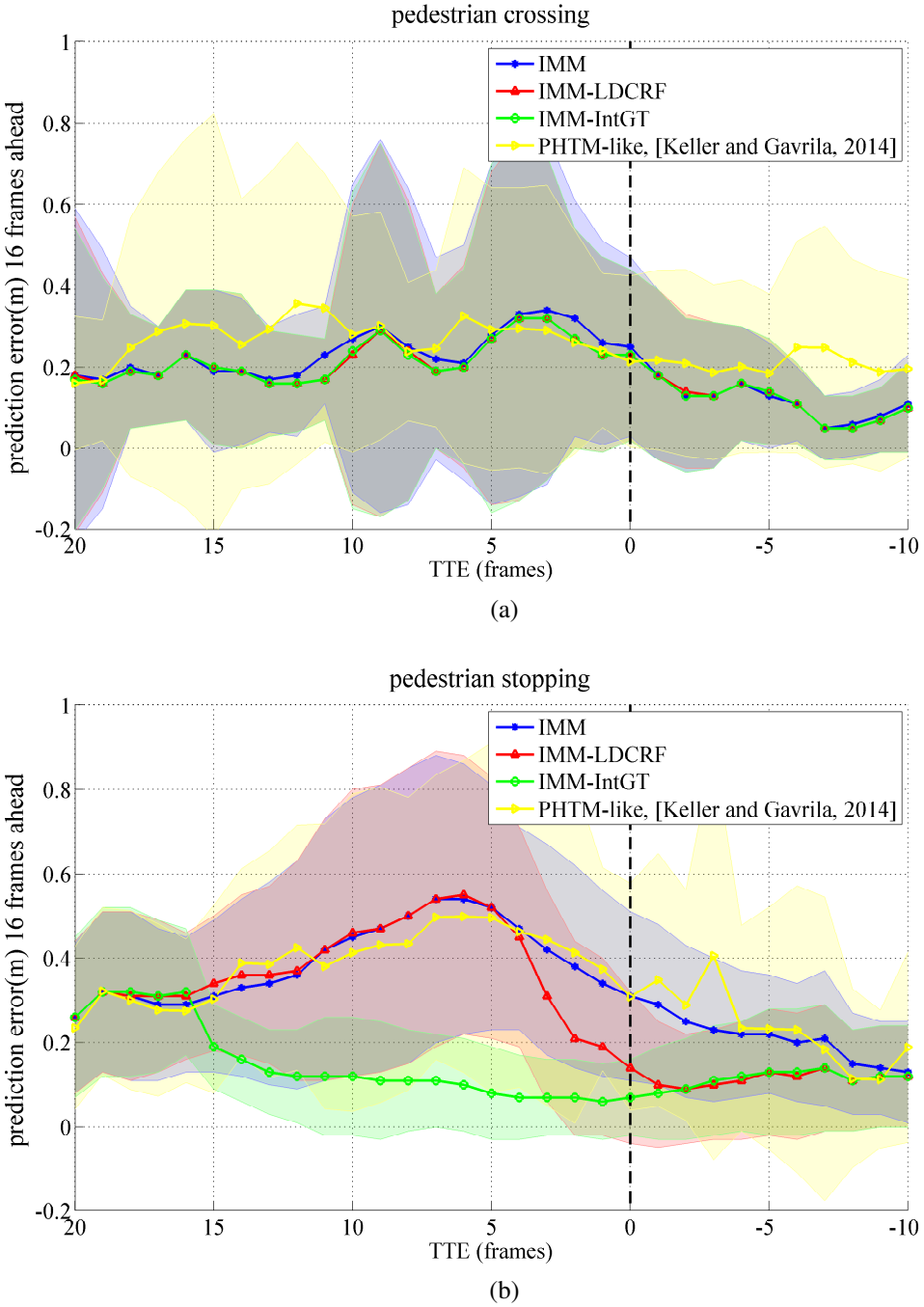


Figure 6.2: Lateral prediction error (m) when predicting 16 frames ahead around event occurrence for (a) crossing and (b) stopping pedestrians. Visualized are the mean error and its standard deviation (shaded area) over all crossing and stopping scenarios from the evaluation dataset. Comparison between a standard IMM filter prediction (blue), the presented LDCRF-controlled IMM filter prediction (red), a controlled IMM filter, when using the underlying GT intention for prediction (green) and a PTHM-like approach [Keller and Gavrila, 2014] (yellow).

In comparison, for a stopping pedestrian the motion state changes abruptly from a homogenous movement to stationary which results in an increased prediction error towards the occurring stopping event (Figure 6.2b). In this case, the presented LDCRF-controlled IMM filter (red curve) profits from the intention recognition and therefore is able to forecast the future

pedestrian lateral position more accurately compared to the conventional IMM filter (blue curve) and the PTHM-like method (yellow curve) in reducing the average path prediction error by $0.17m$ to a value of $0.14m$ at TTE=0. Shortly after the stopping event (TTE<-2), the presented model completely fits the error curve of a controlled IMM filter with ideal ground truth intention inputs (green curve). A few frames prior to the stopping event (TTE$\in[0, 5]$) the LDCRF-controlled IMM filter is able to adapt faster to the rapid change in the pedestrian motion state. The PTHM-like approach does not result in a significant performance gain compared to the conventional IMM filter prediction.

The next step evaluates the fact how well the actual pedestrian ground truth position is represented by the estimated predicted state probability distribution. For this, the earlier introduced predictive log-likelihood is used, see equation (6.29). Ideally, the predicted state probability density function is represented by a Dirac-Delta Distribution with expectation value equal to the ground truth position. In this case the predictive log-likelihood outputs positive values. For a ground truth position, not fitting into the given predicted state probability estimate the predictive log-likelihood will take strong negative values. Table 6.4 shows a comparison for the averaged predictive log-likelihood values on a time interval shortly before event occurrence (TTE $\in$ [0, 15]) in stopping and crossing situations for different IMM filter configurations.

	crossing	**stopping**
IMM	-0.75	-1.29
IMM-LDCRF	**-0.69**	**-0.57**
IMM-IntGT	-0.67	0.01

Table 6.4: Averaged predictive log-likelihood for TTE $\in$ [0, 15] of the pedestrian ground truth position in crossing and stopping scenarios using a prediction horizon of 16 frames ($\approx 1s$).

Whereas for crossing situations values for all filters lie within a similar range (predll $\approx$ 0.7), the benefit of incorporating the pedestrian intention within a controlled IMM filter is clearly visible. The conventional IMM filter is only able to adapt slowly to the change in motion state after a sufficiently high amount of observations that contribute to the static model assumption. This results in an averaged predictive log-likelihood of -1.29. In comparison, for the controlled IMM filter a stopping intention is earlier detected leading to more accurate estimates and a higher averaged predictive log-likelihood of -0.57, approaching the optimal value of 0.01 when using ideal ground truth intentions for a controlled IMM filter prediction.

Bending in vs. Straight

Due to the nature of the PHTM algorithm presented by [Keller and Gavrila, 2014] crossing scenarios cannot be easily differentiated from a bending in or straight walking scenario as trajectories are transferred into each other using translation and rotation operations. There-

		bending in	
		TTE∈ [-10, 20]	TTE=0
IMM	Mean	0.26	0.47
	± Std	0.18	0.24
IMM-LDCRF	Mean	**0.22**	**0.44**
	± Std	0.17	0.25
IMM-IntGT	Mean	0.21	0.40
	± Std	0.18	0.23

Table 6.5: Mean lateral prediction error (m) for bending scenarios using a prediction horizon of 16 frames

fore, the PHTM approach will not be considered in the following. Figure 6.3 shows the prediction errors in lateral position for bending-in scenarios, where pedestrians initially are walking straight along the road. Table 6.5 again shows detailed numbers for lateral prediction errors averaged over the considered time interval and at the time of event occurrence (TTE=0). In scenarios, where a pedestrian is turning into the road, the controlled IMM filter only performs slightly better before event occurrence compared to the standard IMM filter. A significant higher accuracy can only be observed shortly after the turning event (TTE≤0). Even the IMM-IntGT filter predictions result in lateral prediction errors up to 0.40m right before the pedestrian turns into the road (TTE≥0). This can be explained by the

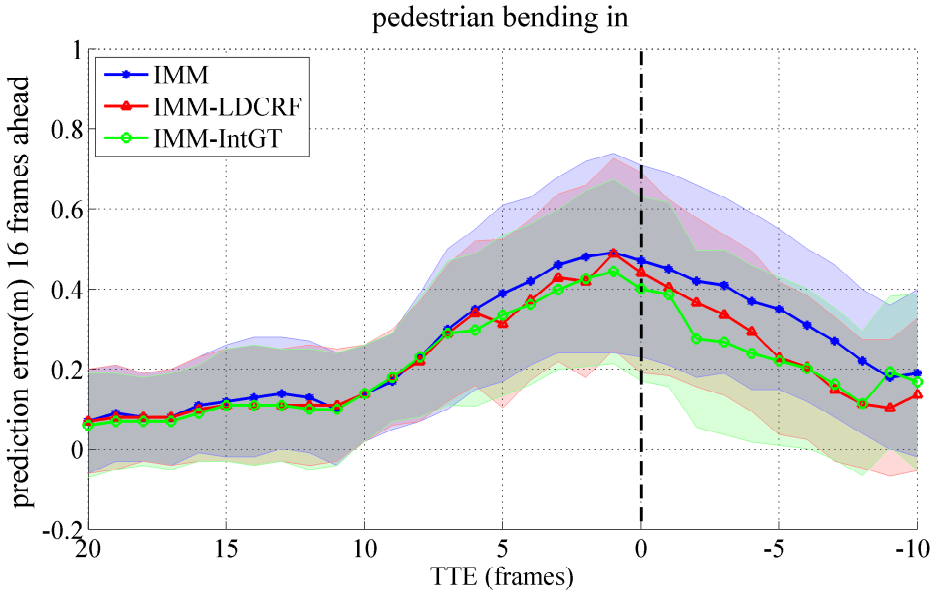


Figure 6.3: Lateral prediction error (m) when predicting 16 frames ahead around event occurrence for bending-in pedestrians. Visualized are the mean error and its standard deviation (shaded area) over all bending in scenarios from the evaluation dataset. Comparison between standard IMM filter prediction (blue), presented LDCRF-controlled IMM filter prediction (red) and a controlled IMM filter, when using the underlying GT intention for prediction (green).

	predll
IMM	-0.67
IMM-LDCRF	**-0.60**
IMM-IntGT	-0.59

Table 6.6: Averaged predictive log-likelihood for TTE $\in$ [0, 15] of the pedestrian ground truth position in bending-in scenarios using a prediction horizon of 16 frames ($\approx 1s$).

fact, that no direct turn rate control is implemented which could guide the predicted path in the correct direction with a small amount of uncertainty. In contrast, for the stopping case, the pedestrian velocity components are directly set to zero, when applying the CP model. The effect is also observed when analyzing the calculated predictive log-likelihood values averaged over the time window TTE $\in$ [0,15] in Table 6.6. Although the controlled IMM performs slightly better than the conventional IMM filter, absolute values for the predictive log-likelihood are comparatively high. Even when incorporating the ground truth intention, no performance gain is observed resulting from a missing direct turn rate control in cases where the CT model is chosen as most appropriate.

6.3 Conclusion

In this chapter a method was presented that deals with the combined intention recognition and path prediction of lateral approaching pedestrians in the domain of intelligent vehicles. The core algorithm consisted of a IMM filter for pedestrian tracking and path prediction controlled by the intention estimates from a LDCRF model learned for different scenarios. Multiple features capturing the pedestrian dynamics and the awareness of the nearby traffic situation served for both intention recognition and path prediction. For a prediction horizon of 1 second, evaluation showed a reduction of the average path prediction error for stopping scenarios (up to $0.17m$) compared to a conventional IMM filter and a PHTM-like approach. In contrast to the PHTM approach the proposed method can handle a wider range of scenarios including straight walking and bending in pedestrians. However, for situations, where pedestrians are turning into the road, there is still a high potential for improvement in stronger guiding the predicted path along the turning direction. The model provides evidence for potential risky situations and therefore can serve for better pedestrian path prediction or be directly integrated into a system implementing a pedestrian forward collision warning or automatic emergency braking function in order to reduce system false alarms.

7 Conclusion

This last chapter concludes the presented work on pedestrian intention recognition and path prediction for video-based driver assistance systems realizing a forward collision warning or automatic emergency braking in critical situations involving an approaching vehicle and pedestrians. Through the earlier chapters single components and contributions have been presented starting with the video-based head pose estimation for pedestrians and ending up with an improved path prediction with integrated estimates for pedestrian intention.
This work essentially consisted of three major parts. The first part was developed in Chapter 3 and dealt with the integrated problem of pedestrian head localization and full pan angle head pose estimation under the harsh conditions of low resolution gray-value images including complex and highly dynamic scenarios taken from a moving camera. The presented system built upon an existing system for video-based pedestrian detection and tracking. A detector-array based approach was proposed, where two state-of-the-art face detection techniques built the basic block. Head pose classifiers were trained in a modified one-vs.-all framework, that guaranteed a good localization rate over the continuous head pose pan angle and prevented instabilities during the training process. For initialization, a sliding window at different scales was used to collect classification confidence values all over the head search area. Based on the normalized confidence values, the head could be localized and the head pose be estimated in a probabilistic framework. The single-frame based approach was extended for pedestrian head pose estimation in video sequences using tacking based on particle filtering in order to provide more accurate and stable results (sec. 3.2). Furthermore, this yields an output that is not restricted to one of eight discrete head poses but provides continuous head pan angle estimates. The developed system was evaluated on large and diverse datasets including publicly available datasets, namely CLEAR and CAVIAR, and internally recorded real world inner-city data. Results were presented in form of confusion matrices and showed over 90% correct head localization and 50% correct head pose estimation using eight discrete head pose classes even when dealing with very low resolution images including heads down to a size of 6×6 pixels. The probabilistic form of the head pose estimation results provided a suitable type of information for further integration within the overall system. In Chapter 4 additional improvement for pedestrian head pose estimation was achieved by incorporating stereo video depth information in different ways. First, the estimated image-based 2D head pan angle was transformed to 3D for direct usage within

the overall system referring to a vehicle located coordinate system (sec. 4.1). Secondly, the usage of head depth profiles for a more robust head localization especially in situations with a heavily structured background (sec. 4.3), and finally, the usage of pedestrian velocity estimates extracted from a stereo-based generic obstacle detection and tracking approach as prior knowledge for a more robust head pose estimation (sec. 4.2).
In the second major building block of this thesis, a method was presented to estimate the intention of lateral approaching pedestrians in the domain of intelligent vehicles in Chapter 5. Multiple features capturing the pedestrian dynamics and the awareness of the nearby traffic situation were used to learn a highly performant Latent-dynamic Conditional Random Field model. The proposed model has the advantage to automatically learn intrinsic structure and feature dependencies as well as temporal dynamics between different intention classes. Evaluation of the trained models showed stable intention estimates for different scenarios compared to other machine learning approaches and state-of-the-art methods. The model provides evidence for potential risky situations and therefore can serve for better pedestrian path prediction or be directly integrated into a system implementing a pedestrian forward collision warning or emergency braking function in order to reduce system false alarms.
As the last major contribution, a method for combined intention recognition and path prediction of lateral approaching pedestrians was investigated in Chapter 6. The core algorithm consisted of an Interacting Multiple Model filter for pedestrian tracking and path prediction controlled by the intention estimates from a LDCRF model learned for different scenarios. For a prediction horizon of 1 second, evaluation showed a reduction of average path prediction error for stopping scenarios (up to $0.17m$) compared to a conventional IMM filter and a PHTM-like approach of [Keller and Gavrila, 2014]. In contrast to recent literature, cf. [Keller and Gavrila, 2014, Kooij et al., 2014a], a wider range of scenarios was addressed including pedestrians that initially are walking along the sidewalks but then suddenly bend in towards the road in addition to lateral crossing or stopping pedestrians.

Future Work

With the current implementation, head pose estimation relies on an accurate and highly available pedestrian detection and tracking system, which can only be guaranteed to a limited extend. Thus, future steps should integrate the head pose estimation process into the pedestrian detection system, first to support or reject given pedestrian hypotheses, similar to part-based detectors, and second to improve person tracking through better path prediction.
In order to further improve intention recognition, additional shape and texture features from intensity and depth and optical flow (inner pedestrian movement) could be integrated into the LDCRF model especially when dealing with short tracks, where pedestrians are appearing out of occlusion. Following the works of [Bonnin et al., 2014, Kooij et al., 2014a] further contextual features like distance to curb stone, detected zebra crossings or the presence of other traffic participants will extend the existing method to cope with a wider range of sit-

uations. Also group behavior could be analyzed and integrated to improve single intention estimates.

For path prediction in turning situations, the main problem is the control of the dynamical models especially the coordinated turn model with adapted turn rate. In comparison to a stopping person, where the model parameters are clear, i.e., the person velocity estimate has to be set to zero, a bending in person can behave with more variability. Especially, approaching or moving away pedestrians would show opposite turn rates. Here, an additional consideration of the underlying situation will help to adapt the model parameters accordingly for path prediction. Using multiple turn rate models with different parameters could additionally help to handle the variability of movement patterns. The single model parameters could then be adapted by anatomical restrictions and additional user studies.

Publications

[Schulz et al., 2011] Schulz, A., Damer, N., Fischer, M., and Stiefelhagen, R. (2011). Combined head localization and head pose estimation for video–based advanced driver assistance systems. In *Proceedings of the German Conference on Pattern Recognition*, volume 6835 of *Lecture Notes in Computer Science*, pages 51–60. Springer Berlin, Heidelberg.

[Schulz and Stiefelhagen, 2012] Schulz, A. and Stiefelhagen, R. (2012). Video-based pedestrian head pose estimation for risk assessment. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC), 2012*, pages 1771–1776.

[Schulz and Stiefelhagen, 2015a] Schulz, A. and Stiefelhagen, R. (2015). Pedestrian intention recognition using latent-dynamic conditional random fields. In *Proceedings of the Intelligent Vehicles Symposium (IV), 2015*, pages 622–627.

[Schulz and Stiefelhagen, 2015b] Schulz, A. and Stiefelhagen, R. (2015). A controlled interactive multiple model filter for combined pedestrian intention recognition and path prediction. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC), 2015*, pages 173–178.

Bibliography

[Antonini et al., 2006] Antonini, G., Martinez, S., Bierlaire, M., and Thiran, J. (2006). Behavioral priors for detection and tracking of pedestrians in video sequences. In *International Journal of Computer Vision (IJCV)*, volume 69(2).

[Arulampalam et al., 2002] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. In *Transactions on Signal Processing*, volume 50(2), pages 174–188.

[Avineri et al., 2012] Avineri, E., Shinar, D., and Susilo, Y. O. (2012). Pedestrians' behaviour in cross walks: The effects of fear of falling and age. In *Accident Analysis & Prevention*, volume 44(1), pages 30–34. Safety and Mobility of Vulnerable Road Usears: Pedestrians, Bicyclists, and Motorcyclists.

[Ba and Odobez, 2004] Ba, S. O. and Odobez, J.-M. (2004). A Probabilistic Framework for Joint Head Tracking and Pose Estimation. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 4, pages 264–267, Los Alamitos, CA, USA. IEEE Computer Society.

[Badino et al., 2009] Badino, H., Franke, U., and Pfeiffer, D. (2009). The stixel world - a compact medium level representation of the 3D-World. In *Proceedings of the German Conference on Pattern Recognition (DAGM)*, volume 31, pages 51–60. Springer Berlin, Heidelberg.

[Bajracharya et al., 2009] Bajracharya, M., Moghaddam, B., Howard, A., Brennan, S., and Matthies, L. H. (2009). Results from a Real-time Stereo-based Pedestrian Detection System on a Moving Vehicle. In *Proceedings of the International Conference on Robotics and Automation (ICRA), Workshop on People Detection and Tracking*.

[Bandyopadhyay et al., 2013] Bandyopadhyay, T., Won, K. S., Frazzoli, E., Hsu, D., Lee, W. S., and Rus, D. (2013). Intention-Aware Motion Planning. In *Proceedings of the Algorithmic Foundations of Robotics Workshop*, volume 10, pages 475–491. Springer Berlin, Heidelberg.

[Bär et al., 2012] Bär, T., Reuter, J. F., and Zöllner, J. M. (2012). Driver head pose and gaze

estimation based on multi-template ICP 3-D point cloud alignment. In *Proceeding of the Intelligent Transportation Systems Conference (ITSC)*, pages 1797–1802.

[Bar-Shalom et al., 2002] Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R. (2002). *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, USA.

[Bäuml et al., 2010] Bäuml, M., Bernardin, K., Fischer, M., and Ekenel, H. K. (2010). Multi-pose face recognition for person retrieval in camera networks. In *Proceedings of the Advanced Video and Signal-based Surveillance Conference (AVSS)*, volume 7.

[Benenson et al., 2012a] Benenson, R., Mathias, M., Timofte, R., and Van Gool, L. (2012a). Fast Stixel Computation for Fast Pedestrian Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 11–20. Springer Berlin, Heidelberg.

[Benenson et al., 2012b] Benenson, R., Mathias, M., Timofte, R., and Van Gool, L. (2012b). Pedestrian detection at 100 frames per second. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Benenson et al., 2014] Benenson, R., Omran, M., Hosang, J., and Schiele, B. (2014). Ten years of pedestrian detection, what have we learned? In *Proceedings of the European Conference on Computer Vision (ECCV), Workshop on computer vision for road scene understanding and autonomous driving*.

[Benenson et al., 2011] Benenson, R., Timofte, R., and Gool, L. V. (2011). Stixels estimation without depth map computation. In *Proceedings of the International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2010–2017.

[Benfold and Reid, 2008] Benfold, B. and Reid, I. (2008). Colour Invariant Head Pose Classification in Low Resolution Video. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[Benfold and Reid, 2009] Benfold, B. and Reid, I. (2009). Guiding visual surveillance by tracking human attention. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 20.

[Bernardin and Stiefelhagen, 2008] Bernardin, K. and Stiefelhagen, R. (2008). Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. In *EURASIP Journal on Image and Video Processing, Article ID 246309*, pages 1–10.

[Bernhoft and Carstensen, 2008] Bernhoft, I. M. and Carstensen, G. (2008). Preferences and behaviour of pedestrians and cyclists by age and gender. In *Transportation Research Part F: Traffic Psychology and Behaviour*, volume 11(2), pages 83–95.

[Bertozzi et al., 2005] Bertozzi, M., Binelli, E., Broggi, A., and Rose, M. D. (2005). Stereo Vision-based approaches for Pedestrian Detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23–28.

[Blackman and Popoli, 1999] Blackman, S. and Popoli, R. (1999). *Design and Analysis of Modern Tracking Systems*. Artech House radar library. Artech House.

[Bogin and Varela-Silva, 2010] Bogin, B. and Varela-Silva, M. I. (2010). Leg Length, Body Proportion, and Health: A Review with a Note on Beauty. In *International Journal of Environmental Research and Public Health*, volume 7(3), pages 1047–1075.

[Bonnin et al., 2014] Bonnin, S., Weisswange, T. H., Kummert, F., and Schmuedderich, J. (2014). Pedestrian crossing prediction using multiple context-based models. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC)*, pages 378–385.

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis.

[Broyden, 1970] Broyden, C. G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. In *Journal of Applied Mathematics*, volume 6(1), pages 76–90.

[Bulut et al., 2011] Bulut, Y., Vines-Cavanaugh, D., and Bernal, D. (2011). Process and Measurement Noise Estimation for Kalman Filtering. In *Proceedings of the IMAC, A Conference on Structural Dynamics*, volume 28(3), pages 375–386, New York, NY. Springer New York.

[Canton-Ferrer et al., 2007] Canton-Ferrer, C., Casas, J. R., and Pardàs, M. (2007). Head pose detection based on fusion of multiple viewpoint information. In *Multimodal Technologies for Perception of Humans: International Evaluation Workshop on Classification of Events, Activities and Relationships (CLEAR)*, pages 305–310. Springer Berlin, Heidelberg.

[Canton-Ferrer et al., 2008] Canton-Ferrer, C., Casas, J. R., and Pardàs, M. (2008). Head orientation estimation using particle filtering in multiview scenarios. In *Multimodal Technologies for Perception of Humans: International Evaluation Workshop on Classification of Events, Activities and Relationships (CLEAR)*, pages 317–327. Springer Berlin, Heidelberg.

[CAVIAR, 2004] CAVIAR (2004). http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1.

[Chen et al., 2011] Chen, C., Heili, A., and Odobez, J. (2011). A joint estimation of head and body orientation cues in surveillance video. In *Proceedings of the International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 860–867.

[Chen and Odobez, 2012] Chen, C. and Odobez, J. (2012). We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video. In

Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), pages 1544–1551.

[Chen and Medioni, 1992] Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. In *Journal on Image and Vision Computing*, volume 10(3), pages 145–155. Elsevier.

[Chen et al., 2008] Chen, Z., Ngai, D. C. K., and Yung, N. H. C. (2008). Pedestrian Behavior Prediction based on Motion Patterns for Vehicle-to-Pedestrian Collision Avoidance. In *Proceedings of the 11th Intelligent Transportation Systems Conference (ITSC)*, pages 316–321.

[Crammer and Singer, 2002] Crammer, K. and Singer, Y. (2002). On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. In *Journal of Machine Learning Research*, volume 2, pages 265–292. JMLR.org.

[Crow, 1984] Crow, F. C. (1984). Summed-area Tables for Texture Mapping. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, volume 11, pages 207–212, New York, NY, USA. ACM.

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

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

[Debnath et al., 2004] Debnath, R., Takahide, N., and Takahashi, H. (2004). A decision based one-against-one method for multi-class support vector machine. In *Pattern Analysis and Applications*, volume 7(2), pages 164–175.

[Dimitrijevic et al., 2006] Dimitrijevic, M., Lepetit, V., and Fua, P. (2006). Human body pose detection using Bayesian spatio-temporal templates. In *Proceedings of Computer Vision and Image Understanding*, volume 104, pages 127–139.

[DIN ISO 8855:2013-11, 2013] DIN ISO 8855:2013-11 (2013). Road vehicles – Vehicle dynamics and road-holding ability – Vocabulary.

[Ding and Xiao, 2012] Ding, Y. and Xiao, J. (2012). Contextual boost for pedestrian detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2895–2902.

[Dollár et al., 2012a] Dollár, P., Appel, R., and Kienzle, W. (2012a). Crosstalk Cascades for Frame-Rate Pedestrian Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 645–659. Springer Berlin, Heidelberg.

[Dollár et al., 2010] Dollár, P., Belongie, S., and Perona, P. (2010). The Fastest Pedestrian Detector in the West. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 68.1–68.11. BMVA Press.

[Dollár et al., 2009a] Dollár, P., Tu, Z., Perona, P., and Belongie, S. (2009a). Integral Channel Features. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[Dollár et al., 2009b] Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2009b). Pedestrian detection: A benchmark. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Dollár et al., 2012b] Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012b). Pedestrian detection: An evaluation of the state of the art. In *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 34(4).

[Doucet et al., 1998] Doucet, A., Godsill, S., and Andrieu, C. (1998). On sequential Monte Carlo sampling methods for Bayesian filtering. In *Statistics and Computing*, volume 10(3), pages 197–208.

[Enzweiler et al., 2010] Enzweiler, M., Eigenstetter, A., Schiele, B., and Gavrila, D. (2010). Multi-cue pedestrian classification with partial occlusion handling. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Enzweiler and Gavrila, 2010] Enzweiler, M. and Gavrila, D. (2010). Integrated pedestrian classification and orientation estimation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 982–989.

[Enzweiler and Gavrila, 2009] Enzweiler, M. and Gavrila, D. M. (2009). Monocular pedestrian detection: Survey and experiments. In *Transactions on Pattern Analysis and Machine Intelligence*, volume 31(12), pages 2179–2195.

[Enzweiler and Gavrila, 2011] Enzweiler, M. and Gavrila, D. M. (2011). A Multilevel Mixture-of-Experts Framework for Pedestrian Classification. In *Transactions on Image Processing*, volume 20(10), pages 2967–2979.

[Enzweiler et al., 2012] Enzweiler, M., Hummel, M., Pfeiffer, D., and Franke, U. (2012). Efficient Stixel-based object recognition. In *Proceedings of the Intelligent Vehicles Symposium (IV)*, pages 1066–1071.

[Ess et al., 2009] Ess, A., Leibe, B., Schindler, K., and Van Gool, L. (2009). Moving obstacle detection in highly dynamic scenes. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 56–63.

[Ess et al., 2007] Ess, A., Leibe, B., and Van Gool, L. (2007). Depth and Appearance for Mobile Scene Analysis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1–8.

[Euro NCAP, 2016] Euro NCAP (2016). http://www.euroncap.com/en.

[Evans and Norman, 2003] Evans, D. and Norman, P. (2003). Predicting adolescent pedestrians' road-crossing intentions: an application and extension of the Theory of Planned Behaviour. In *Health Education Research*, volume 18(3), pages 267–277.

[Fanelli et al., 2011] Fanelli, G., Weise, T., Gall, J., and Gool, L. V. (2011). Real Time Head Pose Estimation from Consumer Depth Cameras. In *Proceedings of the Annual Symposium of the German Association for Pattern Recognition (DAGM)*, volume 33.

[Fisher, 1996] Fisher, N. I. (1996). *Statistical Analysis of Circular Data*. Cambridge University Press.

[Fletcher, 1970] Fletcher, R. (1970). A New Approach to Variable Metric Algorithms. In *The Computer Journal*, volume 13(3), pages 317–322.

[Flohr et al., 2014] Flohr, F., Dumitru-Guzu, M., Kooij, J., and Gavrila, D. (2014). Joint probabilistic pedestrian head and body orientation estimation. In *Proceedings of the Intelligent Vehicles Symposium (IV)*, pages 617–622.

[Freund and Schapire, 1997] Freund, Y. and Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. In *Journal of Computer and System Sciences*, volume 55(1), pages 119–139.

[Freund and Schapire, 1999] Freund, Y. and Schapire, R. E. (1999). A Short Introduction to Boosting. In *Journal of Japanese Society for Artificial Intelligence*, volume 14(5), pages 771–780.

[Friedman et al., 2000] Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). In *The Annals of Statistics*, volume 28(2), pages 337–407. The Institute of Mathematical Statistics.

[Fröba and Ernst, 2004] Fröba, B. and Ernst, A. (2004). Face detection with the modified census transform. In *Proceedings of the International Conference on Face and Gesture Recognition (FG)*, volume 6.

[Fugger et al., 2001] Fugger, T. F., Randles, B. C., Wobrock, J. L., Stein, A. C., and Whiting, W. C. (2001). Pedestrian Behavior at Signal-Controlled Crosswalks. In *SAE Technical Paper*.

[Furuhashi and Yamada, 2011] Furuhashi, R. and Yamada, K. (2011). Estimation of street crossing intention from a pedestrian's posture on a sidewalk using multiple image frames. In *Proceedings of the Asian Conference on Pattern Recognition (ACPR)*, pages 17–21.

[Gandhi and Trivedi, 2007] Gandhi, T. and Trivedi, M. (2007). Pedestrian Protection Systems: Issues, Survey, and Challenges. In *Transactions on Intelligent Transportation Systems*, volume 8(3), pages 413–430.

[Gandhi and Trivedi, 2008] Gandhi, T. and Trivedi, M. (2008). Image based estimation of pedestrian orientation for improving path prediction. In *Proceedings of the Intelligent Vehicles Symposium (IV)*, pages 506–511.

[Gavrila and Munder, 2007] Gavrila, D. M. and Munder, S. (2007). Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle. In *International Journal of Computer Vision*, volume 73(1), pages 41–59. Springer Berlin, Heidelberg.

[Gee and Cipolla, 1994] Gee, A. H. and Cipolla, R. (1994). Determining The Gaze Of Faces In Images. In *Journal of Image and Vision Computing*, volume 12, pages 639–647.

[Geiger et al., 2012] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361.

[Geiger et al., 2010] Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient Large-Scale Stereo Matching. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

[Geiger et al., 2011] Geiger, A., Ziegler, J., and Stiller, C. (2011). StereoScan: Dense 3D Reconstruction in Real-time. In *Proceedings of the Intelligent Vehicles Symposium (IV)*.

[Goldfarb, 1970] Goldfarb, D. (1970). A Family of Variable-Metric Methods Derived by Variational Means. In *Mathematics of Computation*, volume 24(109), pages 23–26. American Mathematical Society.

[Goldhammer et al., 2014] Goldhammer, M., Doll, K., Brunsmann, U., Gensler, A., and Sick, B. (2014). Pedestrian's Trajectory Forecast in Public Traffic with Artificial Neural Networks. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 4110–4115.

[Haar, 1911] Haar, A. (1911). Zur Theorie der orthogonalen Funktionensysteme. In *Mathematische Annalen*, volume 71(1), pages 38–53. Springer-Verlag.

[Hamaoka et al., 2013] Hamaoka, H., Hagiwara, T., Tada, M., and Munehiro, K. (2013). A study on the behavior of pedestrians when confirming approach of right/left-turning vehicle while crossing a crosswalk. In *Proceedings of the Intelligent Vehicles Symposium (IV)*.

[Helbing and Molnár, 1995] Helbing, D. and Molnár, P. (1995). Social force model for pedestrian dynamics. In *Physical Review E*, volume 51, pages 4282–4286. American Physical Society.

[Hirata et al., 2008] Hirata, J., Morimoto, M., and Fujii, K. (2008). Estimating face direction from low resolution images. In *Proceedings of the Automation Congress*, pages 1–6.

[Hoiem et al., 2006] Hoiem, D., Efros, A., and Hebert, M. (2006). Putting Objects in Perspective. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2137–2144.

[Horn and Schunck, 1981] Horn, K. P. and Schunck, G. (1981). Determining Optical Flow. In *Proceedings of the Conference on Artificial Intelligence (AI)*, volume 17, pages 185–203.

[Horprasert et al., 1996] Horprasert, T., Yacoob, Y., and Davis, L. S. (1996). Computing 3-D head orientation from a monocular image sequence. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG)*, pages 242–247.

[Horton et al., 2007] Horton, M., Cameron-Jones, M., and Williams, R. (2007). Multiple Classifier Object Detection with Confidence Measures. In *Proceedings of the 20th Australian Joint Conference on Advances in Artificial Intelligence (AI)*, pages 559–568. Springer Berlin, Heidelberg.

[Isard and Blake, 1998] Isard, M. and Blake, A. (1998). CONDENSATION–Conditional Density Propagation for Visual Tracking. In *International Journal of Computer Vision (IJCV)*, volume 29(1), pages 5–28.

[ISO 26262-1:2011, 2011] ISO 26262-1:2011 (2011). Road vehicles – Functional safety – Part 1: Vocabulary.

[Jia and Zhang, 2007] Jia, H.-X. and Zhang, Y.-J. (2007). Fast Human Detection by Boosting Histograms of Oriented Gradients. In *Proceedings of the International Conference on Image and Graphics (ICIG)*, pages 683–688.

[Jin et al., 2004] Jin, H., Liu, Q., Lu, H., and Tong, X. (2004). Face detection using improved LBP under Bayesian framework. In *Proceedings of the International Conference on Image and Graphics (ICIG)*, pages 306–309.

[Jones and Viola, 2003] Jones, M. and Viola, P. (2003). Fast multi-view face detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Kadari and Vedagiri, 2013] Kadari, B. R. and Vedagiri, P. (2013). Modelling pedestrian road crossing behaviour under mixed traffic condition. In *European Transport*, volume 55.

[Kearns, 1988] Kearns, M. (1988). Thoughts on hypothesis boosting. (Unpublished).

[Keller et al., 2011a] Keller, C., Enzweiler, M., and Gavrila, D. M. (2011a). A New Benchmark for Stereo-based Pedestrian Detection. In *Proceedings of the Intelligent Vehicles Symposium (IV)*.

[Keller et al., 2011b] Keller, C., Enzweiler, M., Rohrbach, M., Fernandez Llorca, D., Schnorr, C., and Gavrila, D. (2011b). The Benefits of Dense Stereo for Pedestrian Detection. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC)*, volume 12(4), pages 1096–1106.

[Keller et al., 2011c] Keller, C., Hermes, C., and Gavrila, D. (2011c). Will the pedestrian cross? Probabilistic Path Prediction based on Learned Motion Features. In *Proceedings of the German Conference on Pattern Recognition*, volume 6835, pages 386–395.

[Keller and Gavrila, 2014] Keller, C. G. and Gavrila, D. M. (2014). Will the pedestrian cross? A study on pedestrian path prediction. In *Intelligent Transportation Systems (ITS)*, volume 15(2).

[Kelly et al., 2008] Kelly, P., O'Connor, N. E., and Smeaton, A. F. (2008). A framework for evaluating stereo-based pedestrian detection techniques. In *Transactions on Circuits and Systems for Video Technology*, volume 18(8), pages 1163–1167.

[Kitani et al., 2012] Kitani, K. M., Ziebart, B. D., Bagnell, J. A., and Hebert, M. (2012). Activity Forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12, pages 201–214. Springer Berlin, Heidelberg.

[Kittler et al., 1998] Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998). On combining classifiers. In *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 20(3), pages 226–239.

[Köhler et al., 2012] Köhler, S., Goldhammer, M., Bauer, S., Doll, K., Brunsmann, U., and Dietmayer, K. (2012). Early detection of the Pedestrian's intention to cross the street. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC)*, pages 1759–1764.

[Köhler et al., 2013] Köhler, S., Schreiner, B., Ronalter, S., Doll, K., Brunsmann, U., and Zindler, K. (2013). Autonomous evasive maneuvers triggered by infrastructure-based detection of pedestrian intentions. In *Proceedings of the Intelligent Vehicles Symposium (IV)*, pages 519–526.

[Kooij et al., 2014a] Kooij, J., Schneider, N., Flohr, F., and Gavrila, D. (2014a). Context-Based Pedestrian Path Prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 8694 of *Lecture Notes in Computer Science*, pages 618–633. Springer Berlin, Heidelberg.

[Kooij et al., 2014b] Kooij, J. F. P., Schneider, N., and Gavrila, D. M. (2014b). Analysis of pedestrian dynamics from a vehicle perspective. In *Proceedings of the Intelligent Vehicles Symposium (IV)*.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. In *The Annals of Mathematical Statistics*, volume 22, pages 79–86. The Institute of Mathematical Statistics.

[Labayrade et al., 2002] Labayrade, R., Aubert, D., and Tarel, J.-P. (2002). Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In *Proceedings of the Intelligent Vehicle Symposium (IV)*, volume 2, pages 646–651.

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 18, pages 282–289. Morgan Kaufmann Publishers Inc.

[Laptev et al., 2008] Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Lawrence et al., 2004] Lawrence, G. J. L., Hardy, B. J., Carroll, J. A., Donaldson, W. M. S., Visviskis, C., and Peel, D. (2004). A study on the feasibility of measures relating to the protection of pedestrians and other vulnerable road users. In *Transportation Research Library*.

[Leibe et al., 2007] Leibe, B., Cornelis, N., Cornelis, K., and Van Gool, L. (2007). Dynamic 3D Scene Analysis from a Moving Vehicle. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Leibe et al., 2008] Leibe, B., Schindler, K., Cornelis, N., and Van Gool, L. (2008). Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles. In *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 30(10), pages 1683–1698.

[Leibe et al., 2005] Leibe, B., Seemann, E., and Schiele, B. (2005). Pedestrian detection in crowded scenes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 878–885.

[Li and Jilkov, 2001] Li, X. R. and Jilkov, V. P. (2001). Survey of Maneuvering Target Tracking. Part III: Measurement models. In *Proceedings of the Conference on Signal and Data Processing of Small Targets (SPIE)*.

[Li and Jilkov, 2002] Li, X. R. and Jilkov, V. P. (2002). Survey of Maneuvering Target Tracking. Part IV: Decision-Based Methods. In *Proceedings of the Conference on Signal and Data Processing of Small Targets (SPIE)*.

[Li and Jilkov, 2003] Li, X. R. and Jilkov, V. P. (2003). Survey of maneuvering target tracking. Part I. Dynamic models. In *Transactions on Aerospace and Electronic Systems*, volume 39(4), pages 1333–1364.

[Li and Jilkov, 2005] Li, X. R. and Jilkov, V. P. (2005). Survey of maneuvering target tracking. Part V. Multiple-model methods. In *Transactions on Aerospace and Electronic Systems*, volume 41(4), pages 1255–1321.

[Li and Jilkov, 2010] Li, X. R. and Jilkov, V. P. (2010). Survey of Maneuvering Target Tracking. Part II: Motion Models of Ballistic and Space Targets. In *Transactions on Aerospace and Electronic Systems*, volume 46(1), pages 96–119.

[Lienhart and Maydt, 2002] Lienhart, R. and Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. In *Proceedings of the International Conference on Image Processing (ICIP)*, volume 1, pages 900–903.

[Littlestone, 1988] Littlestone, N. (1988). Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. In *Machine Learning*, volume 2(4), pages 285–318. Kluwer Academic Publishers-Plenum Publishers.

[Liu and Nocedal, 1989] Liu, D. C. and Nocedal, J. (1989). On the Limited Memory BFGS Method for Large Scale Optimization. In *Mathematical Programming*, volume 45(3), pages 503–528, Secaucus, NJ, USA. Springer-Verlag New York, Inc.

[Llorca et al., 2012] Llorca, D., Sotelo, M., Hellín, A., Orellana, A., Gavilán, M., Daza, I., and Lorente, A. (2012). Stereo regions-of-interest selection for pedestrian protection: A survey. In *Transportation Research Part C: Emerging Technologies*, volume 25, pages 226–237.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision*, volume 60, pages 91–110.

[Maji et al., 2008] Maji, S., Berg, A., and Malik, J. (2008). Classification using intersection kernel support vector machines is efficient. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Marchal et al., 2003] Marchal, P., Gavrila, D. M., Letellier, L., Meinecke, M.-M., Morris, R., and Töns, M. (2003). SAVE-U: An innovative sensor platform for Vulnerable Road User protection. In *Intelligent Transportation Systems (ITS)*.

[Martin et al., 2014] Martin, M., v. d. Camp, F., and Stiefelhagen, R. (2014). Real Time Head Model Creation and Head Pose Estimation on Consumer Depth Cameras. In *Proceedings of the 2nd International Conference on 3D Vision*, volume 1, pages 641–648.

[Matikainen et al., 2009] Matikainen, P., Hebert, M., and Sukthankar, R. (2009). Trajectons: Action recognition through the motion analysis of tracked features. In *Proceedings of the International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 514–521.

[Mehta and Patel, 1995] Mehta, C. R. and Patel, N. R. (1995). Exact logistic regression: Theory and examples. In *Statistics in Medicine*, volume 14(19), pages 2143–2160. Wiley Subscription Services, Inc.

[Morency et al., 2007] Morency, L., Quattoni, A., and Darrell, T. (2007). Latent-dynamic discriminative models for continuous gesture recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Mu et al., 2008] Mu, Y., Yan, S., Liu, Y., Huang, T., and Zhou, B. (2008). Discriminative local binary patterns for human detection in personal album. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Murphy-Chutorian et al., 2007] Murphy-Chutorian, E., Doshi, A., and Trivedi, M. M. (2007). Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC)*, pages 709–714.

[Murphy-Chutorian and Trivedi, 2009] Murphy-Chutorian, E. and Trivedi, M. (2009). Head pose estimation in computer vision: A survey. In *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 31, pages 607–626.

[Nedevschi et al., 2009] Nedevschi, S., Bota, S., and Tomiuc, C. (2009). Stereo-Based Pedestrian Detection for Collision-Avoidance Applications. In *Transactions on Intelligent Transportation Systems (ITS)*, volume 10(3), pages 380–391.

[NHTSA, 2016] NHTSA (2016). National Center for Statistics and Analysis. Pedestrians: 2014 Data. In *Washington, DC: National Highway Traffic Safety Administration, Traffic Safety Facts*.

[Niese et al., 2006] Niese, R., Al-Hamadi, A., and Michaelis, B. (2006). A Stereo and Color-based Method for Face Pose Estimation and Facial Feature Extraction. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 1, pages 299–302.

[Orozco et al., 2009] Orozco, J., Gong, S., and Xiang, T. (2009). Head pose classification in crowded scenes. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[Oxley et al., 1995] Oxley, J., Fildes, B., Ihsen, E., Day, R., and Charlton, J. (1995). An Investigation of Road Crossing Behaviour of older pedestrians. In *Monach University-Accident Research Centre*, volume 81.

[Papageorgiou and Poggio, 1999] Papageorgiou, C. and Poggio, T. (1999). Trainable Pedestrian Detection. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 35–39. IEEE.

[Pavlovic et al., 2000] Pavlovic, V., Rehg, J. M., and Maccormick, J. (2000). Learning Switching Linear Models of Human Motion. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 981–987.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Pellegrini et al., 2009] Pellegrini, S., Ess, A., Schindler, K., and van Gool, L. (2009). You'll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking. In *Proceedings of the International International Conference on Computer Vision (ICCV)*.

[Pfeiffer and Franke, 2011] Pfeiffer, D. and Franke, U. (2011). Modeling Dynamic 3D Environments by Means of The Stixel World. In *Transactions on Intelligent Transportation Systems (ITS)*, volume 3(3), pages 24–36.

[Proesmans et al., 1994] Proesmans, M., Van Gool, L., Pauwels, E., and Oosterlinck, A. (1994). Determination of Optical Flow and its Discontinuities using Non-Linear Diffusion. In *Proceedings of the 3rd European Conference on Computer Vision (ECCV)*, volume 2, pages 295–304, London, UK. Springer-Verlag.

[Quintero et al., 2014] Quintero, R., Almeida, J., Llorca, D. F., and Sotelo, M. A. (2014). Pedestrian path prediction using body language traits. In *Proceedings of the Intelligent Vehicles Symposium (IV)*, pages 317–323.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. In *Readings in Speech Recognition*, volume 77(2), pages 257–286.

[Rehder et al., 2014] Rehder, E., Kloeden, H., and Stiller, C. (2014). Head detection and orientation estimation for pedestrian safety. In *Proceedings of the Intelligent Transportation Systems Conference (ITSC)*, pages 2292–2297.

[Rifkin and Klautau, 2004] Rifkin, R. and Klautau, A. (2004). In Defense of One-Vs-All Classification. In *Journal of Machine Learning Research*, volume 5, pages 101–141. JMLR.org.

[Robertson and Reid, 2006] Robertson, N. and Reid, I. (2006). Estimating gaze direction from low-resolution faces in video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 9.

[Rohrbach et al., 2009] Rohrbach, M., Enzweiler, M., and Gavrila, D. M. (2009). High-Level Fusion of Depth and Intensity for Pedestrian Classification. In *Proceedings of the 31st German Conference on Pattern Recognition*, pages 101–110. Springer Berlin, Heidelberg.

[Rowley et al., 1998] Rowley, H., Baluja, S., and Kanade, T. (1998). Rotation invariant neural network-based face detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38–44.

[Roy and Marcel, 2009] Roy, A. and Marcel, S. (2009). Haar Local Binary Pattern Feature for Fast Illumination Invariant Face Detection. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[Runkler, 2012] Runkler, T. A. (2012). *Data Analytics, Models and Algorithms for Intelligent Data Analysis*. Vieweg+Teubner Verlag, 1 edition.

[Sanchez et al., 2012] Sanchez, J., Meinhardt-Llopis, E., and Facciolo, G. (2012). TV-L1 Optical Flow Estimation. In *Image Processing Online*.

[Schindler and Van Gool, 2008] Schindler, K. and Van Gool, L. (2008). Action snippets: How many frames does human action recognition require? In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[Schmidt and Färber, 2009] Schmidt, S. and Färber, B. (2009). Pedestrians at the kerb-Recognizing the action intentions of humans. In *Proceedings of Transportation Research Part F: Traffic Psychology and Behaviour*, volume 12(4).

[Schneider and Gavrila, 2013] Schneider, N. and Gavrila, D. (2013). Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, volume 8142 of *Lecture Notes in Computer Science*, pages 174–183. Springer Berlin, Heidelberg.

[Seemann, 2007] Seemann, E. (2007). Dissertation, Pedestrian Detection in Crowded Street Scenes. In *Selected Readings in Vision and Graphics*, volume 46. Hartung-Gorre, Konstanz.

[Seemann et al., 2004] Seemann, E., Nickel, K., and Stiefelhagen, R. (2004). Head pose estimation using stereo vision for human-robot interaction. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG)*, pages 626–631.

[Shanno, 1970] Shanno, D. F. (1970). Conditioning of quasi-Newton methods for function minimization. In *Mathematics of Computation*, volume 24(111), pages 647–656.

[Shimizu and Poggio, 2004] Shimizu, H. and Poggio, T. (2004). Direction estimation of pedestrian from multiple still images. In *Proceedings of the Intelligent Vehicles Symposium (IV)*, pages 596–600.

[Siriteerakul et al., 2010] Siriteerakul, T., Sugimura, D., and Sato, Y. (2010). Head Pose Classification from Low Resolution Images Using Pairwise Non-Local Intensity and Color Differences. In *Proceedings of the 4th Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pages 362–369.

[Stein, 2004] Stein, F. (2004). Efficient Computation of Optical Flow Using the Census Transform. In *Proceedings of the 26th DAGM Symposium*, volume 3175 of *Lecture Notes in Computer Science*, pages 79–86. Springer Berlin, Heidelberg.

[Stiefelhagen et al., 2008] Stiefelhagen, R., Bowers, R., and Fiscus, J. (2008). Multimodal Technologies for Perception of Humans. In *Proceedings of the International Evaluation Workshops CLEAR 2007 and RT 2007*.

[Stiefelhagen et al., 1996] Stiefelhagen, R., Yang, J., and Waibel, A. (1996). A Model-Based Gaze Tracking System. In *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems*, pages 304–310.

[Tamura et al., 2012] Tamura, Y., Le, P. D., Hitomi, K., Chandrasiri, N. P., Bando, T., Yamashita, A., and Asama, H. (2012). Development of pedestrian behavior model taking account of intention. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 382–387.

[Torralba, 2003] Torralba, A. (2003). Contextual priming for object detection. In *International Journal of Computer Vision*, volume 53(2), pages 169–191.

[Turaga et al., 2008] Turaga, P., Chellappa, R., Subrahmanian, V. S., and Udrea, O. (2008). Machine Recognition of Human Activities: A Survey. In *Transactions on Circuits and Systems for Video Technology*, volume 18(11), pages 1473–1488.

[Tuzel et al., 2008] Tuzel, O., Porikli, F., and Meer, P. (2008). Pedestrian Detection via Classification on Riemannian Manifolds. In *Transactions on Pattern Analysis and Machine Intelligence*, volume 30(10), pages 1713–1727, Washington, DC, USA. IEEE Computer Society.

[Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 2 edition.

[Vatahska et al., 2007] Vatahska, T., Bennewitz, M., and Behnke, S. (2007). Feature-based head pose estimation from images. In *Proceedings of the International Conference on Humanoid Robots*, pages 330–335.

[Viola and Jones, 2001a] Viola, P. and Jones, M. (2001a). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518.

[Viola and Jones, 2001b] Viola, P. and Jones, M. (2001b). Robust Real-time Object Detection. In *International Journal of Computer Vision*.

[Viola and Jones, 2004] Viola, P. and Jones, M. (2004). Robust Real-time Face Detection. In *International Journal of Computer Vision*, volume 57, pages 137–154.

[Viola et al., 2003] Viola, P., Jones, M., and Snow, D. (2003). Detecting Pedestrians Using Patterns of Motion and Appearance. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 734–741.

[Voit et al., 2008] Voit, M., Nickel, K., and Stiefelhagen, R. (2008). Head Pose Estimation in Single- and Multi-view Environments – Results on the CLEAR'07 Benchmarks. In *Proceedings of the International Evaluation Workshops CLEAR and RT*, pages 307–316.

[Voit and Stiefelhagen, 2009] Voit, M. and Stiefelhagen, R. (2009). A System for Probabilistic Joint 3D Head Tracking and Pose Estimation in Low-Resolution, Multi-view Environments. In *Computer Vision Systems*, volume 5815 of *Lecture Notes in Computer Science*, pages 415–424. Springer Berlin, Heidelberg.

[Wakim et al., 2004] Wakim, C. F., Capperon, S., and Oksman, J. (2004). A Markovian model of pedestrian behavior. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, volume 4, pages 4028–4033.

[Walk et al., 2010] Walk, S., Schindler, K., and Schiele, B. (2010). Disparity Statistics for Pedestrian Detection: Combining Appearance, Motion and Stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11, pages 182–195. Springer Berlin, Heidelberg.

[Wang et al., 2009] Wang, X., Han, T., and Yan, S. (2009). An HOG-LBP human detector with partial occlusion handling. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 32–39.

[Weston et al., 1998] Weston, J., Watkins, C., et al. (1998). Multi-class support vector machines.

[WHO, 2013] WHO (2013). Global status report on road safety 2013. In *World Health Organization*.

[Withopf, 2007] Withopf, D. (2007). *Reliable Real-Time Vehicle Detection and Tracking*. Dissertation, IWR, Fakultät für Mathematik und Informatik, Univ. Heidelberg.

[Wojek et al., 2009] Wojek, C., Walk, S., and Schiele, B. (2009). Multi-cue onboard pedestrian detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 794–801.

[Yannis et al., 2015] Yannis, G. D., Evgenikos, P., Broughton, J., Pace, J.-F., and Sanmartín, J. (2015). Traffic Safety Basic Facts on Pedestrians. In *European Commission, Directorate General for Transport*.

[Young, 1993] Young, J. W. (1993). Head and Face Anthropometry of Adult U.S. Civilians. In *U.S. Department of Administration*.

[Zabih and Woodfill, 1994] Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–158. Springer Berlin, Heidelberg.

[Zach et al., 2007] Zach, C., Pock, T., and Bischof, H. (2007). A Duality Based Approach for Realtime TV-L1 Optical Flow. In *Proceedings of German Conference on Pattern Recognition (DAGM)*.

[Zhang et al., 2007] Zhang, Z., Hu, Y., Liu, M., and Huang, T. (2007). Head pose estimation in seminar room using multi view face detectors. In *Proceedings of the International Evaluation Conference on Classification of Events, Activities and Relationships*, volume 1, pages 299–304.

[Zheng et al., 2010] Zheng, J., Ramírez, G., and Fuentes, O. (2010). Face detection in low-resolution color images. In *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR)*, volume 6111, pages 454–463.

[Zhu et al., 2006] Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. (2006). Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1491–1498.