Inaugural-Dissertation

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich–Mathematischen Gesamtfakultät
der
Ruprecht–Karls–Universität
Heidelberg

vorgelegt von
Dipl. Inf. Christoph Gustav Keller
aus Heidelberg

Tag der mündlichen Prüfung: 06.11.2014

Stereo-based Pedestrian Detection and Path Prediction

Gutachter:	**Prof. Dr. Christoph Schnörr** Universität Heidelberg
Zweitgutachter:	**Prof. Dr. Dariu M. Gavrila** Universität von Amsterdam

Zusammenfassung

In den letzten Jahren gab es eine rasante Entwicklung von Fahrerassistenzsystemen (Englisch: Advanced Driver Assistance Systems oder kurz ADAS). Diese Systeme unterstützen nicht nur den Fahrer, sondern erhöhen durch das automatische Einleiten von Sicherheitreaktionen des Fahrzeuges selber auch die Sicherheit aller anderen Verkehrsteilnehmer. Zukünftige aktive Fußgängerschutzsystem in Intelligentem Fahrzeugen müssen nun noch einen Schritt weiter gehen und lernen, ein genaues Bild ihrer Umgebung und der darin während der Fahrt zu erwartenden Änderungen zu entwickeln.

Diese Arbeit widmet sich der Verbesserung bildgestützter Fußgängerschutzsysteme. Es werden darin neue Methoden der Bildhypothesengenerierung (englisch: region of interest (ROI) generation), Fußgängerklassifikation, Pfadvorhersage und Absichtserkennung entwickelt. Die Leistung der Fußgängererkennung in realen, dynamischen Umgebungen mittels einer bewegten Kamera wird durch die Verwendung von dichtem Stereo in den unterschiedlichen Modulen verbessert.

In einer Experimentalstudie wurde die Effizienz eines Systems zur monokularen Fußgängererkennung mit einem System verglichen, dass erweitert wurde um dichtes Stereo für die Hypothesengenerierung und der Fußgängerverfolgung (englisch: tracking) zu nutzen. Das neue System erwies sich hierin als deutlich effizienter als das monokulare System. Diese Leistungssteigerung gab Anlass für eine erweiterte Nutzung von dichtem Stereo bei der Fußgängererkennung. Die Hypothesengenerierung wurde durch die dynamische Schätzung der Kameraorientierung und des Straßenprofils weiter verbessert. Insbesondere bei hügeligen Straßen steigerte sich die Erkennungsleistung durch die Optimierung des Suchbereichs. Zusätzlich konnte die Klassifikationsleistung durch die Fusion von unterschiedlichen Merkmalen aus Bild und Tiefeninformation verbessert werden.

Aufbauend auf den Erfolgen bei der Fußgängererkennung wird in der Arbeit ein System für den Aktiven Fußgängerschutz vorgestellt, welches die Funktionen Fußgängererkennung, Situationsanalyse und Fahrzeugsteuerung kombiniert. Für die Fußgängerkennung wurden Ergebnisse eines Verfahrens zur bewegungsbasierten Objekterkennung mit Ergebnissen eines Fußgängerklassifikators fusioniert. Das System wurde in einen Versuchsträger eingebaut und half dabei, Unfälle durch einen aktiven Lenkeingriff oder ein Notbremsemanöver zu vermeiden.

Der letzte Teil der Arbeit befasst sich mit dem Problem der Pfadvorhersage und dem Erkennen der Fußgängerabsicht in Situationen, in denen sich der Fußgänger nicht mit einer konstanten Geschwindigkeit bewegt. Zwei neue, lernbasierte

Ansätze werden vorgestellt und mit aktuellen Verfahren verglichen. Durch die Verwendung von Merkmalen, die aus dichtem optischem Fluss generiert werden, ist es möglich den Pfad und die Absicht einer Fußgängers vorherzusagen. Das erste Verfahren lernt eine niedrigdimensionale Mannigfaltigkeit der Merkmale, die eine Vorhersage von Merkmale, Pfad und Absicht erlaubt. Das zweite Verfahren verwendet einen Suchbaum in dem Trajektorien abgelegt sind die mit Bewegungsmerkmalen erweitert wurden. Ein probabilistischer Suchalgorithmus ermöglicht die Vorhersage des Fußgängerpfads und Absicht. Die Leistungsfähigkeit der Systeme wurde zusätzlich mit der Leistung von menschlichen Probanden verglichen.

In dieser Arbeit wurde großer Wert auf die ausführliche Analyse der vorgestellten Verfahren und die Verwendung von realistischen Testdatensätzen gelegt. Die Experimente zeigen das die Leistungsfähigkeit eines Systems zur Fußgängererkennung durch die Verwendung von dichtem Stereo verbessert werden kann. Die Vorgestellten Verfahren zur Pfadvorhersage und Absichtserkennung ermöglichen ein frühzeitiges erkenne der Fußgängerabsicht. Die Zuverlässigkeit zukünftiger System für den Aktiven Fußgängerschutz, die durch Aktiven Lenkeingriff oder Notbremsemanöver Unfälle vermeiden, kann mit den vorgestellten Verfahren verbessert werden. Dadurch können Unfälle vollständig verhindert oder die Schwere einer Kollision reduziert werden.

Abstract

Over the last years there has been a rapid evolution of advanced driver assistance systems (ADAS). Such systems not only support drivers in safely steering their vehicle, but also increase the safety of all traffic participants by adding systems able to trigger autonomous safety reactions of the vehicle itself. Future active pedestrian protection systems for intelligent vehicles will now have to go a step further and learn how to acquire a thorough understanding of a car's environment and how this environment will change in the course of driving.

This thesis focuses on improving the performance of vision based pedestrian protection systems. In this course new methods for region of interest generation, classification, pedestrian path prediction and action classification are introduced. All methods address the problem of pedestrian detection from a moving camera in a real-world cluttered environment by additionally using dense stereo data for region of interest generation, classification, tracking and path prediction.

In an experimental study a state-of-the-art monocular based pedestrian recognition system is compared with a system using dense stereo data for the region of interest (ROI) generation and for tracking the pedestrian position. Performance gains of the stereo-based system motivate the further use of dense stereo for different modules of a pedestrian recognition system. Dense stereo data is further exploited by dynamically estimating camera parameters and road profile information for a refined ROI generation in a complex environment. Constraining on possible pedestrian locations is especially beneficial in scenes with undulating, hilly roads. Additionally, the different characteristics of depth and intensity features are fused to improve the performance of a pedestrian classifier.

An integrated active pedestrian safety system is presented as a combination of sensing, situation analysis, decision making and vehicle control. For the task of pedestrian detection this system fuses generic motion-based object segmentation and pedestrian recognition. The system has been implemented in a demonstrator vehicle and can reliably prevent collisions by automatically initiating braking or evasive steering maneuvers.

Finally, the problem of pedestrian path prediction and action classification in challenging situations where a constant velocity assumption fails is addressed. Two novel learning-based approaches are introduced and compared to state-of-the-art methods. Using features extracted from optical flow, the newly proposed methods allow the prediction of a pedestrian‘s path and its intended action at short, sub-second time intervals. The first approach learns a low dimensional

representation of motion features that allow feature, path and action prediction. The second approach uses a probabilistic search tree containing trajectories extended with motion features to predict the trajectory and action of a pedestrian. A further comparison puts the action classification performance of the newly proposed systems up against human performance.

For all problems addressed in this work, emphasis has been placed on the use of realistic, real-world datasets and an in-depth performance evaluation of the proposed methods. The experiments show that the use of dense stereo for ROI generation and classification significantly improves the performance of a pedestrian detection system. The proposed path prediction and action classification methods further allow an early detection of pedestrian actions. Integrating the developed methods in the next-generation of active pedestrian safety systems will lead to a faster, improved system decision whether or not to initiate emergency vehicle maneuvers (braking, steering) in critical situations. This increases their potential of preventing accidents with pedestrians or reducing the severity of a collisions.

Acknowledgements

Thanks to all the people who made this thesis possible.

First, I am grateful to Prof. Dr. Christoph Schnörr and Prof. Dr. Dariu M. Gavrila for their help and guidance. In particular I want to thank Dariu for his constant support, advice and the painstaking proof-reading of this thesis and all the papers. I learned a lot from your tenacity to push things forward.

Also I would like to thank all my friends and colleges from the Department of Environment Perception. Most of all thank you Dr. Markus Enzweiler, my office mate, for always listening to my ideas, having fruitful discussions and making work fun.

Although we were not sitting in one room, the same applies for Dr. Sebastian Zuther and Dr. Marc Muntziger, so I owe you thanks, too.

If not already mentioned, I would also like to thank the co-authors of my papers for valuable feedback and meaningful discussions: Dr. Christoph Hermes, Dr. Clemens Rabe, Dr. Thao Dang, Dr. Hans Fritz, Markus Rohrbach and Armin Joos. In this context many people supported my work by providing algorithms, professional and personal assistance. Many thanks: Prof. Dr. Christian Wöhler, Dr. Tilo Schwarz, Dr. Andreas Wedel, Dr. Martin Fritzsche, Dr. Ulrich Kressel and Dr. Uwe Franke.

My family has been a rock for me throughout the years. To my family, particularly my parents, my brother and my sister, thank you for your love, support, and unwavering belief in me. Above all I would like to thank Salome for her love, constant support and for keeping me sane over the past years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the triumphal procession of the car in the 19th century an interesting question arose: "What goes on when a man drives an automobile" [77]. Whilst driving a car the driver constantly selects and transforms mainly visual information in his environment. In order to make correct driving decisions not only has the current situation to be comprehended, but also possible future changes of the situation have to be anticipated.

This is illustrated in the *situation awareness model* [45] which is subdivided into three levels (see Figure 1.1): The first level handles the perception of elements in a constantly changing environment, such as the course of the road, its surface conditions, pedestrians, traffic, etc.. Once the elements have been identified, the second level assigns a meaning to these elements. For example, seeing a pedestrian at a crosswalk will not result in the desired behavior of stopping, if the driver has no knowledge of this rule. With an understanding of the current situation, foreseeing the near future is modeled in the third level. By using the information obtained in the previous levels and the experience of the dynamics and possible behavior of identified objects around us, changes in the environment can be predicted. For example, given our current driving speed and direction we can

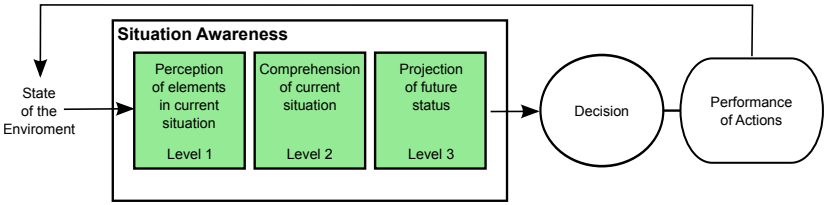


Figure 1.1: Model of situation awareness in dynamic decision making, adapted from [45].

predict a possible hazardous situation that requires braking, when a pedestrian walks onto the road. From this prediction, a decision is made that results in the next action that modifies the state of the environment. At this point the cycle of perception, comprehension and projection of the environment starts again.

But a driver's information processing capabilities are limited. Depending on the situation and the drivers constitution, only a limited amount of sensory information can be processed and erroneous actions might be taken. Studies show that more than 90% of traffic accidents are due to human error [55]. With more than 90% due to visual information acquisition problems [86, 133, 154]. The most common explanation for car accidents is, "I looked, but I didn't see" [27]. Advanced driver assistance systems (ADAS) support the driver in complicated situations and can help to prevent accidents where humans are inattentive. This requires perceiving the environment using sensors. Besides artificial sensors that are similar to the human sensory system, e.g. cameras, additional sensors that go beyond the human senses, e.g., radar, can be used to perceive the environment. But even with a complex sensory setup, current systems are outperformed by humans when it comes to the complex tasks of understanding and forecasting the current situation. It will be the paramount objective of this thesis to develop and improve the methods for pedestrian protection.

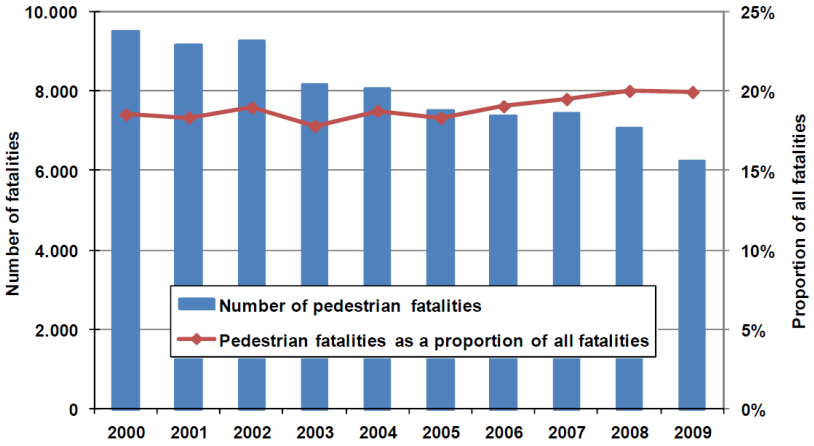


Figure 1.2: Number of pedestrian fatalities and proportion of total fatalities in Europe [24].

Motivation and Challenges

Pedestrians are without doubt the most vulnerable traffic participants. In Europe (EU-24, 2009), 20% of all the fatalities in road traffic accidents have been pedestrians [24]. In the last 10 years the number of deadly accidents involving pedestrians have been reduced by 34% from 9476 deaths in 2000 to 6233 in 2009 (Figure 1.2). Nations with emerging economies have an even higher pedestrian fatality incidence [93]. This reduction can partially be explained by the increasing awareness of the plight of vulnerable road users at the EU level. In 2003, the EU passed Phase 1 of Directive *2003/102/EC* on pedestrian protection, focusing on passive safety, i.e. meaning to reduce injury levels upon impact, by specifying various maximum impact criteria (e.g. head, leg). Data collected during the time of the first stage, showed that the more restricted criteria set for the second stage was not feasible. Different and newer technologies can contribute to the reduction of pedestrian and other vulnerable road user casualties. Studies showed that pedestrian protection can be significantly improved by *Brake Assist Systems (BAS)* [110]. These systems are designed to detect a emergency brake situation and help the driver to archive the maximum deceleration to prevent an accident. In 2009 the EU Parliament approved Regulation *2009/78/EC* on pedestrian protection which requires car manufactures to integrate active safety systems into cars. Pedestrian protection is meanwhile also a major theme for consumer rating groups like Euro NCAP.

(a) **(b)**

Figure 1.3: (a) Active bonnet to reduce pedestrian injuries on an impact by enlarge the deformation zone. (b) Collision prevention by automated evasive steering maneuver.

Figure 1.4: Typical dangerous situation: pedestrian stepping unexpectedly onto the street.

Systems for pedestrian protection can be grouped into active and passive safety systems. Passive pedestrian safety measures involve vehicle structures (e.g. bonnet, bumper) that expand during collision in order to minimize the impact of the pedestrian leg or head hitting the vehicle. For example, Mercedes-Benz introduced the active bonnet (Figure 1.3a) as standard for the new E-Class 2009. The system includes three impact sensors in the front section as well as special bonnet hinges pre-tensioned by powerful springs. Upon impact with a pedestrian, the rear section of the bonnet is pushed upwards by $50mm$ in a fraction of a second, thus enlarging the deformation zone. The system is reversible and can be reset manually by the driver.

Although important, passive pedestrian safety measures are constrained by the laws of physics in terms of ability to reduce collision energy and thus injury level. Moreover, passive measures cannot account for injuries sustained in the secondary impact of the pedestrian hitting the road. Much effort is therefore spent towards the development of active safety systems, which detect dangerous situations involving pedestrians ahead of time, allowing the possibility to warn

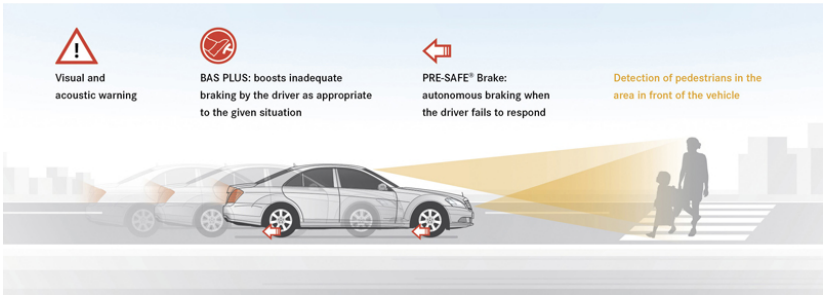


Figure 1.5: Pedestrian protection system introduced in the Mercedes-Benz E-Class and S-Class in 2013. Dangerous situations can be detected by a combination of a stereo camera system and radar sensors. An autonomous emergency braking maneuver is initiated if a collision is imminent.

the driver or to automatically control the vehicle. A system that can prevent accidents by automatic braking or evasion (Figure 1.3b) is presented in Chapter 6. Such systems are particularly valuable when the driver is distracted or visibility is poor (Figure 1.4). The first night vision systems that detect and highlight pedestrians have reached the market (e.g. Mercedes-Benz E-Class 2009, BMW 7 series 2008 and Audi A8 2010). In 2010 Volvo introduced a collision mitigation braking system (CMS) for pedestrians with the S60 limousine, which is based on monocular vision and radar. A new safety system that can help to prevent collisions with pedestrians and vehicles has been introduced in the Mercedes-Benz E-Class and S-Class in 2013. For the task of pedestrian protection (Figure 1.5), the system uses a Stereo Multi-Purpose Camera (SMPC) mounted behind the windshield and two short-range radar sensor. Visual pedestrian detection is realized with a pedestrian recognition system using dense stereo. Possible collision with a pedestrian in the driving corridor can be detected and the driver is warned using visual and acoustic signals. The necessary braking power is automatically adjusted to prevent the collision, if the driver tips the brake pedal. If the driver does not react an autonomous emergency braking maneuver is initiated. The system operates at speeds up to $72\ km/h$ and can autonomously prevent accidents up to $50\ km/h$.

Robustness of these systems is very important, they have to be able to detect a variety of possible dangerous situations in order to prevent an accident. At

the same time, false system actions have to be at a minimum to fulfill necessary automotive safety integrity levels (ASIL). Additional constraints apply for vision based systems. Operation should be possible at day, night, different weather conditions and in an always changing, complex environment. So computer vision algorithms, developed for pedestrian protection, need to be robust with respect to the input image data while still having a reliable detection performance at a low false positive rate. At the same time algorithms have to operate in real-time, handling large amounts of image data.

Chapter 2

Related Work

An extensive amount of literature has been written on the subject of pedestrian safety. See [69] for a broad survey on passive and active pedestrian protection methods, discussing multiple sensor types (e.g. cameras in visible/NIR/FIR spectrum, radars, laser range finders) and methods for collision risk assessment. In this thesis, we focus on vision-based pedestrian detection which is a key problem in the domain of intelligent vehicles (IV). Large variations in human pose and

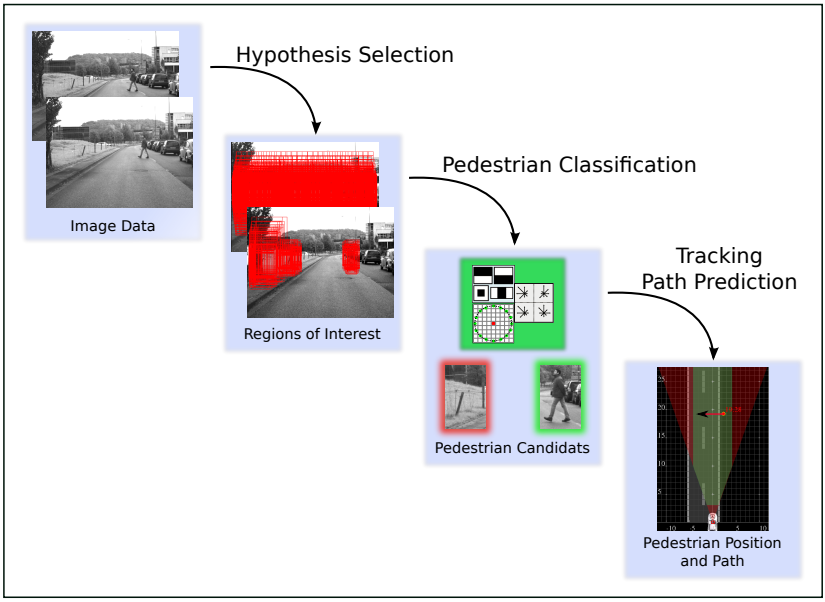
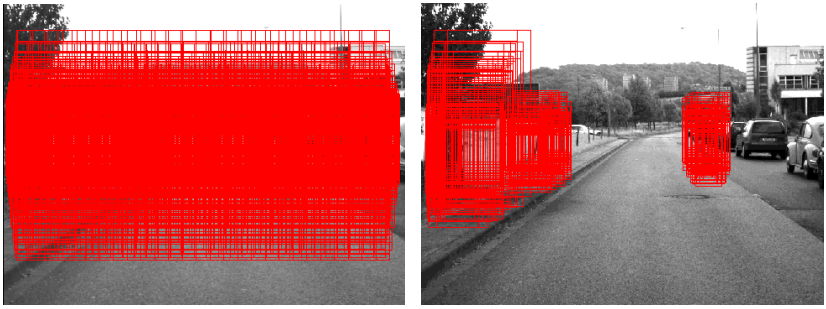


Figure 2.1: Different modules of a pedestrian recognitions system.



(a) (b)

Figure 2.2: (a) Regions-of-Interest (ROI) generated in a sliding window fashion at various scales. (b) ROIs with sufficient support from dense stereo data.

clothing, as well as varying backgrounds and environmental conditions make this problem particularly challenging. Many interesting approaches for vision-based pedestrian detection have been proposed. Most approaches follow a module-based strategy (Figure 2.1) comprising generation of possible pedestrian location hypotheses (regions-of-interest, ROI) using some computationally efficient method, followed by a more expensive pattern classification step utilizing features from intensity images (gray-scale or color) ([23, 72, 130]). Most benchmark studies on pedestrian detection have dealt with monocular based systems, see [41, 49, 69, 74, 91, 126] for relevant surveys and benchmark studies. Detected pedestrian candidates are then tracked over time to allow predicting the pedestrians path, which is important for possible actions of an integrated pedestrian safety system.

2.1 Hypotheses Selection

Various modalities (e.g. intensity, motion, depth) are used in ROI generation to extend the sliding window technique, where detector windows at various scales and locations are shifted over the image to obtain object hypotheses for classification (Figure 2.2). This pre-processing step is applied to reduce the number of hypotheses which are processed by a more powerful but computationally expen-

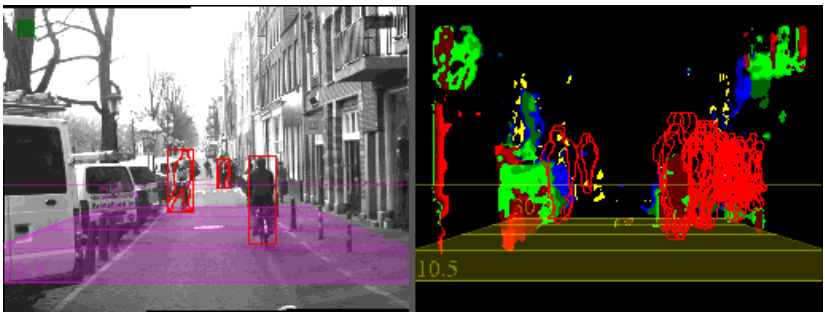


Figure 2.3: Regions-of-Interest (ROI) generated from dense stereo combined with shape based pedestrian detection.

sive classifier. In [72], the locations where the number of depth features from sparse stereo exceeds a percentage of the search window area are added to the ROI list for the subsequent shape detection module (Figure 2.3). This approach has been extended in [103] using dense stereo in ROI the preprocessing step. In [185], a foreground region is obtained by clustering in disparity space. [23, 79] propose to select ROIs by considering the x- and y-projections of the disparity space following the v-disparity representation [108]. In [2], object hypotheses are obtained by using a subtractive clustering in the 3D space in world coordinates. The "stixel world" [7, 15, 141] describes obstacles in the three dimensional world using a set of thin, vertically oriented rectangles. This representation is used in in [14] for candidate region selection. Motion information is utilized in [52] as a pre-processing step for ROI generation.

Most approaches for ROI generation involve the assumption of a planar road, as well as constant camera height and pitch. Violations of these constraints are typically handled by relaxing the scene constraints, e.g. allowing a certain amount of deviation from the ground-plane assumption. Recently, some approaches for estimating road shape and camera parameters have been presented. To estimate camera height and pitch, linear fitting in the v-disparity space [131], in world coordinates [57, 76] and in the so-called virtual-disparity image [157] has been proposed. In [108], the road surface is modeled by fitting piecewise linear functions in the v-disparity space. Other approaches involve fits of quadratic polynomials [134] or clothoid functions [131] in the v-disparity space.

2.2 Pedestrian Classification

Given a set of initial object hypotheses the task of the pedestrian classifier is the decision whether the given image region contains a pedestrian (pedestrian class) or not (non-pedestrian class). Pedestrian classifiers can be roughly characterized as either generative or discriminative models [164].

2.2.1 Generative Models

Generative approaches model pedestrian appearance in terms of its class-conditional density function. In combination with the class priors, the posterior probability for the pedestrian class can be inferred using a Bayesian approach. Most generative approaches use a 2D pedestrian shape model that is learned from example shapes. Using shape data for pedestrian detection has the advantage of robustness to illumination changes and different clothing. The shape space can be described using discrete or continues models.

Discrete shape models represent the space using a set of example shapes [70, 72]. Instance-based learning approaches have the advantage that new example shapes can easily be added to the model but they require a large amount of samples to cover the possible shape space. In [70, 72] several thousand shapes have been collected to represent possible pedestrian appearances. To allow real-time operation an efficient shape hierarchy is used in combination with matching technique based on distance-transforms to compute shape similarities. In [84] a Field Programmable Gate Array (FPGA) implementation is presented to allow real-time vehicle on-board operation.

Continues shape models learn a parametric representation from the set of example shapes [32, 33]. Suitable landmark points, e.g. feet of a pedestrian, are extracted either manually [33] or automatically [13, 128] to allow aligning the training set. Reducing the dimensionality of the landmarks space and deriving the main modes of variation can be realized using linear methods , e.g. Principal Component Analysis (PCA) [33] or non-linear methods, e.g. Local linear embedding (LLE) [44]. Integrating shapes that represent different pedestrian poses (e.g. feet apart or feet closed) into a common subspace can result in degenerated models that lead to implausible shapes. This can be prevented by separating dissimilar shapes into different clusters and generating cluster specific subspaces [47, 128]. The advantage of continues shape models is the possibility to generate synthetic shape examples that can e.g. be used for tracking [44, 78].

However, matching a test sample to the model requires finding optimal model parameters. Iterative parameter estimation [33] can be rohibitively expensive for real-time operation. Using Monte Carlo techniques [78] can be used to speed up computation.

Besides shape information texture is an important cue for pedestrian recognition. Different generative models have been proposed that combine shape and texture information [48, 31, 54].

2.2.2 Discriminative Models

Regarding pedestrian classification, most approaches use discriminative models comprising a combination of intensity-based feature extraction and classification. Such features can be categorized into texture-based and gradient-based.

Local binary pattern (LBP) descriptors [125] and derivatives [20, 26, 25, 26, 83] have successfully been used for pedestrian classification. The basic idea is to encode the neighborhood of each pixel as a binary number by thresholding. A ROI is divided in to several cells. The LBP feature vector is formed by concatenating normalized histograms over cells. Popularity of LBP can be explained due to their robustness to illumination changes and fast feature computation.

Non-adaptive Haar wavelet features have been popularized by [140] and adapted by many others [123, 170]. Feature evaluation involves computing the sum of pixels within rectangular image areas. Fast feature evaluation can be realized using integral images [170]. In [123, 140] significant wavelet feature located on the exterior boundary of the pedestrian body have manually been selected. Adaptive Boosting (AdaBoost) [65] in combination with integral image feature evaluation is used in [170], to overcome the manual feature selection from an over-complete set of Haar wavelets. Automatically combining a set of non-adaptive features using boosting allows a certain adaptation to the training data.

Motivated by the function of the visual cortex, local receptive fields (LRF) [178] or convolutional neural network (CNN) [113] can be used to learn adaptive features. These methods learn the underlying spacial structure of the data and show superior results [127, 158] compared to non-adaptive Haar wavelet features.

Gradient-based features, e.g. Histograms of oriented gradients (HOG) [34, 46, 132, 179, 184, 186] have found wide use for pedestrian classification. The idea is that pedestrian appearance and shape can be described by the distribution of edge gradients. A ROI is divided into several overlapping blocks and each block contains a grid of cells. Gradients of each cell are accumulated in

a orientation-based histogram. The feature vector is formed by concatenating locally normalized blocks. The HOG feature is robust to illumination changes and outperforms non-adaptive Haar wavelet features for the task of pedestrian recognition [34, 49]. Computational costs of HOG features can be reduced by using integral images [186] to allow fast feature evaluation. Field Programmable Gate Array (FPGA) [11], Graphics Processing Unit (GPU) [143, 156] and Digital Signal Processor (DSP) [28] implementations allow real-time HOG feature extraction for on-board vehicle use.

Discriminative models approximate the Bayesian maximum-a-posteriori decision by learning the parameters of a discriminant function (decision boundary) between the pedestrian and non-pedestrian classes from training examples.

Support vector machines (SVM) are widely used in both linear [35, 46, 179, 184, 186] and non-linear variants [123, 140]. Other popular classifiers include neural networks [72, 178] and AdaBoost cascades [121, 163, 170, 179, 181, 184, 186]. Some approaches additionally apply a component-based representation of pedestrians as an ensemble of body parts [46, 56, 121, 123, 136, 181]. Context information from neighboring regions can additionally be used [40, 148] for classification.

Cascaded architectures for pedestrian detection, involving modules using different cues to narrow down the image search space, are prevalent (e.g. [72, 76, 128, 130]). A recent trend involves the integration of multiple features (Haar wavelets, HOG, LRF, LBP, etc.) and/or modalities (intensity, depth, motion, etc.) into a single pattern classification module [46, 132, 147, 152, 171, 175, 179, 182]. One fusion approach involves integration of all cues into a single joint feature space [152, 175, 179]. Here, the enlarged dimensionality of the joint space can cause over-fitting problems or is practically intractable, cf. [152]. Boosting approaches have also been proposed to automatically select the "best" features from a pool of different features and modalities [179, 182]. In contrast, [46, 132, 147] utilize fusion on classifier-level by training a specialized classifier for each feature or modality. Classifier fusion is done using fuzzy integration [132], simple classifier combination rules [147] or a mixture-of-experts framework [46].

2.3 Tracking / Path Prediction

Tracking describes the problem of estimating the trajectory of a pedestrian and deriving properties that are not directly observable, e.g. velocity of a pedestrian. For that purpose a tracker has to assign consistent labels to pedestrians detected in each image frame. Recursive Bayesian estimation is commonly used

to estimate the probability density function describing the pedestrian state over time. Integrated in a pedestrian recognition system, the tracker declares whether the estimated pedestrian state is valid. This is a challenging task, especially in situations involving multiple pedestrians and complicated occlusion situations.

Most approaches follow a Track-by-Detection approach, considering this as frame-by-frame association of detections based on geometry and dynamics without particular pedestrian appearance models [2, 71, 72, 120]. Other approaches utilize pedestrian appearance models coupled with geometry and dynamics [21, 53]. Detection-by-Tracking approaches (e.g. [4, 128]) furthermore integrate detection and tracking in a Bayesian framework, combining appearance models with an observation density, dynamics, and probabilistic inference of the posterior state density. In [128] a multi-cue approach using a generative shape model, a discriminative texture classifier and a temporal transition model is combined in a particle filter framework. System detection rate and tracking performance could be improved significantly by allowing the tracker to decide about the object class and state.

Active pedestrian safety system require not only an accurate estimate of the pedestrian position but additionally the possibility to predict the pedestrian path [36, 98]. One way to perform path prediction relies on closed-form solutions for Bayesian filtering. In the Kalman Filter (KF) [10], the current state of a dynamic system can be propagated to the future by means of the underlying dynamical model, without the incorporation of new measurements. The same idea can be applied to KF filter extensions to either multiple linear dynamical models, e.g. the Interacting Multiple Model (IMM) KF [10], or to non-linear models, e.g. the Extended KF or the Unscented KF (see [120, 159] for applications to pedestrian tracking). Model based prediction accuracy mainly depends on the correctness of the latest state estimate in combination with a valid dynamical model, see [151] for a comparison and combination of different dynamical model. An alternative approach for path prediction involves non-parametric stochastic models. Possible trajectories are generated by Monte Carlo simulations, taking into account the respective dynamical models. In [1] a constant motion model is combined with particle filtering to perform impact prediction. Nicolao *et al.* [39] distinguish lateral and longitudinal pedestrian velocity and model these independently by a random walk. Wakim *et al.* [172] model pedestrian motion by means of four states of a Markov chain, corresponding to standing still, walking, jogging and running. Each state is associated with probability distributions of magnitude and direction of pedestrian velocity; the state changes are controlled by vari-

ous transition probabilities. Recently, more complex pedestrian motion models also account for group behavior and spatial lay-out (e.g. entry/exit points), e.g. see [5] for a discussion of methods for surveillance applications. These latter approaches, although interesting, are less relevant to the traffic safety domain considered in this work.

The limited amount of available training data precludes the use of modeling approaches which compute joint probability distributions over time intervals explicitly. Indeed, most pedestrian motion models consist of states that correspond to single time steps and are first-order Markovian. This potentially limits their expressiveness and precision. In contrast, Black and Jepson [17] describe an extension of particle filtering to incrementally match trajectory models to input data. It is used for motion classification of 2D gestures and expression. Sidenbladh *et al.* [153] add an efficient tree search in the context of articulated 3D human pose recovery. Käfer *et al.* [97] apply this technique to vehicle motion prediction, utilizing the quaternion-based rotationally invariant longest common subsequence (QRLCS) metric for trajectory matching. In our work [102] we combine positional and optical flow features in the QRLCS matching to perform pedestrian path prediction and action classification (continue-walking vs. stopping at the curbside) from a vehicle. Following up on the analysis of pedestrian intention at the curbside, Köhler *et al.* [106] address the continue-standing vs. starting-to-walk classification task, from a stationary, monocular camera. They combine a motion contour image based HOG-like descriptor with a linear SVM. Chen *et al.* [29] propose a multi-level prediction model, in which the higher levels are long-term predictions based on trajectory clustering matching, whereas the low level uses an Auto-Regressive model to predict the next time step.

A common assumption when dealing with human motion is that measurements in a high dimensional space can be represented in a low dimensional, non-linear manifold. Non-linear dimensionality reduction methods allow learning the internal model of the data (see [168] for an overview of techniques). It often depends on the data and the task at hand (e.g. visualization, classification) to determine which of the techniques is best suited. Because measurements from human motions are time-dependent it is desirable to consider the dependency of the data over time. The Gaussian process latent variable model (GPLVM) [111], which is a generalization of the Probabilistic Principal Component Analysis (PPCA) [161], can be extended to model the dynamics of the data. This Gaussian Process Dynamical Model (GPDM) [174] allows for a non-linear mapping from the latent space to the observation space as well as a smooth prediction of latent

points. Especially the mapping (or prediction) of data-points on the latent space makes this technique interesting for tracking application. Urtasun *et al.* [167] use GPDM to track a small number of 3D body points that have been derived using an image-based tracker. The system is trained using one gait cycle from six subjects and is able to handle several frames of occlusions. A Detection-by-Tracking approach using a dynamic part based limb detector in combination with a GPDM is presented in [4]. Especially in scenarios with many persons and long term occlusions the system has a robust detection and tracking performance. Raskin *et al.* [145] use a GPDM with an articulated model of the human body in combination with an Annealed Particle Filter (APF) for tracking and action classification. Action classification is realized by comparing observed sequences with template sequences in latent space.

2.4 Integrated Systems

A number of pedestrian systems were installed on-board vehicles [9, 22, 68, 72, 115, 116, 119, 130]. Some of these not only implement a perception component but also collision risk estimation in combination with acoustical driver warning and/or automatic vehicle braking, see systems by Daimler [116], Ibeo [68], VW [116, 119], and the Universities of Alcala [115] and Parma [22]. While these systems sole rely on on-board sensors for pedestrian protection, Car-2-X communication systems have been proposed [3, 124] which use wireless communication modules to allow an interaction between a pedestrian, vehicles and infrastructure. Especially in situations where pedestrian can not be detected by sensors, e.g. due to complete occlusion by a parked car, drivers can be warned and/or safety measures can be prepared (e.g. pressure buildup for an emergency braking). Other work dealt with pedestrian perception, collision risk estimation and vehicle actuation by means of simulation [95].

Systems for collision avoidance and mitigation by braking are already in the market for passenger cars and commercial vehicles. Suitable methods for criticality assessment have already been proposed (e.g. [85]). However, collision avoidance by steering has not been covered in depth in the literature. Most work on trajectory generation for collision avoidance has been done in the field of robotics. Powerful methods to solve non-holonomic motion planning problems with dynamic obstacles have been proposed (e.g. [58, 109]), yet the computational complexity of many of the proposed algorithms prohibits the application on current automotive hardware. To overcome this limitation, efficient plan-

ning algorithms to evaluate possible collision avoidance maneuvers by human drivers in highly structured scenarios have been introduced [149]. Optimal vehicle trajectory control for obstacle avoidance within shortest distance is presented in [82]. The Proreta Project [94] evaluated driver assistance systems that initiate automatic braking when an object vehicle cuts into the ego vehicle's lane, and automatic steering when an object vehicle is standing in front of the ego vehicle and the driver does not react.

In comparison to collision avoidance systems, autonomous vehicles have different requirements for sensors and algorithms. An autonomous vehicle has to be able to generate an understanding of the environment(e.g. drivable corridor, intention of traffic participants) in order to be able to derive the necessary next actions. The majority of autonomous vehicles demonstrated at the Defense Advanced Research Projects Agency (DARPA) Urban Challenge [37] and the Google self-driving car [81] used high-end laser scanners coupled with radars for long range sensing. Using high-end laser scanners (e.g. Velodyne) allows generating a detailed 3D map of the surrounding. Mode of operation and costs prohibit the use of these sensors for mass market products.

Although computer vision played a minor role in most of the vehicles participating in the DARPA Challenge [165], early approaches to autonomous driving [60] evaluate the requirements and importance of a vision-based system. Especial in an inner-city environment pedestrian recognition is of great importance. In August 2013, a Mercedes-Benz S-Class vehicle with close-to-production sensors drove completely autonomously for about $100km$ from Mannheim to Pforzheim, Germany, following the well-known historic Bertha Benz Memorial Route [63]. The autonomous vehicle relied solely on vision and radar sensors in combination with accurate digital maps to obtain a comprehensive understanding of complex traffic situations. For the task of pedestrian detection the systems used a dense stereo-base ROI generation [100] and a multi-cue pedestrian classifier [51] using disparity and intensity image data. Detecting and understanding the behavior of pedestrians has shown to be crucial to enable reliable autonomous driving in an inner-city environment.

Chapter 3

Outline and Contributions

The main focus of this thesis is to develop methods for vision based active pedestrian safety systems. Overall system performance is improved by focusing on the use of dense stereo in all modules of the pedestrian safety system. For all methods developed in this thesis, the practical application for an online safety system is analyzed and assessed in detail. Performance evaluation is done using real world data recorded from a moving vehicle or system-in-the-loop tests using a demonstrator vehicle on a dedicated test track. Assets and drawbacks of the proposed methods are shown by comparing the performance to state-of-the-art systems.

3.1 An Experimental Study on Stereo-based Pedestrian Detection (Chapter 4)

The area of pedestrian detection has rapidly evolved in the intelligent vehicles domain. Different modalities for hypothesis generation have been proposed, see Chapter 2. Stereo vision is an attractive sensor for this purpose. In Chapter 4 we focus on the use of stereo vision for candidate selection and pedestrian localization in comparison to a purely monocular based system. But unlike for monocular vision [41, 49, 69, 74, 91, 126], there are no realistic, large scale benchmarks available for stereo-based pedestrian detection, to provide a common point of reference for evaluation. We present a thorough evaluation methodology for the evaluation of integrated multi-module pedestrian recognition systems and make our dataset publicly available for benchmarking purposes. Furthermore the benefits of stereo vision for ROI generation and localization are quantified. The monocular system and the stereo bases system use a ROI generation with a flat world assumption. The stereo-based systems additionally uses dense stereo data for hypothesis selection and 3D position estimation of the pedestrian. False

positives are reduced by a factor of $4-5$ with stereo over mono, using the same classification component.

3.2 The Benefit of Dense Stereo (Chapter 5)

Starting with Chapter 4, we focus on the use of dense stereo for ROI generation to detect candidate locations. In Chapter 5 the benefits of dense stereo are further exploited. Dense stereo allows to dynamically estimate camera parameters and road profile which in turn provides strong scene constraints on possible pedestrian locations. For classification we extract spatial features (gradient orientation histograms) directly from dense depth and intensity images. Both modalities are represented in terms of individual feature spaces, in which discriminative classifiers are learned. Our experiments involve challenging image data captured in complex urban environments (i.e. undulating roads, speed bumps, etc.). Our results show a performance improvement by up to a factor of 7.5 at classification-level and up to a factor of 5 at tracking-level (reduction in false alarms at constant detection rates) over a system with static scene constraints and intensity-only classification.

3.3 Fusion of Generic Obstacle Detection and Pedestrian Recognition (Chapter 6)

Chapter 6 presents a novel active pedestrian safety system, which combines sensing, situation analysis, decision making and vehicle control. Active safety systems which use sensors to survey surroundings hold great potential to reduce the accident frequency and severity, by warning the driver and/or exerting automatic vehicle control ahead of crashes. The sensing component is based on stereo vision; it fuses two complementary approaches for added robustness: motion-based object detection and pedestrian recognition. Based on the techniques developed in Chapter 4 and Chapter 5, the pedestrian recognition module uses dense stereo for candidate selection and pedestrian localization. The highlight of the system is the ability to decide within a split second whether to perform automatic braking or evasive steering, and to execute this maneuver reliably, at relatively high vehicle speeds (up to $50\ km/h$). Extensive pre-crash experiments with the system on the test track have been performed (22 scenarios with real pedestrians and a dummy). We obtained a significant benefit in detection performance and

improved lateral speed estimation by the fusion of motion-based object detection and pedestrian recognition. On a reproducible scenario subset, involving the dummy entering laterally into the vehicle path from behind an occlusion, the system executed in over 40 trials the intended vehicle action: automatic braking (if a full stop is still possible) or else, automatic evasive steering.

3.4 Path Prediction and Action Classification (Chapter 7)

Chapter 7 presents a study on pedestrian path prediction and action classification at short, sub-second time intervals. Future vehicle systems for active pedestrian safety will not only require a high recognition performance, but also an accurate analysis of the developing traffic situation. State-of-the-art collision avoidance systems, such as described in Chapter 6, rely on an accurate velocity estimation for predicting the pedestrian path. One major challenge for pedestrian path prediction is the highly dynamic behavior of pedestrians, which can change their walking direction in an instance, or start/stop walking abruptly. To address this challenge, we introduce two novel learning based approaches using augmented features derived from dense optical flow. The first approach uses a probabilistic search tree containing trajectories extended with motion features to predict the path and action of a pedestrian. The second approach learns low dimensional representations, using Gaussian Process Dynamical Models (GPDM), describing the temporal dynamics of the pedestrians optical flow field. Path prediction and action classification performance of the proposed methods is compared to two baseline systems that use positional information only (Kalman Filter and its extension to Interacting Multiple Models). In experiments using stereo vision data obtained from a vehicle, the accuracy of path prediction at various time horizons is investigated, as well as the effect of various errors (image localization, vehicle ego-motion estimation). During stopping events the newly proposed methods using non-linear and/or higher-order models achieved a more accurate position prediction compared to the baseline systems. To put the action classification performance of the different methods in context, we additionally evaluated the performance humans archive.

3.5 Publications

This thesis has led to a number of publications that are listed in Appendix A. Corresponding publications have partially been included in the discussion of related work in Chapter 2.

Chapter 4

An Experimental Study on Stereo-based Pedestrian Detection

4.1 Overview

The main contribution of this chapter is to carefully quantify the benefit of stereo-vision over an otherwise identical monocular system for pedestrian detection, see Figure 4.1. We do not present entirely new systems, but evaluate a variant of the well-known HOG-based pedestrian detector, e.g. [34], in both monocular and stereo vision set-ups. We assume our results to generalize to other established pedestrian detectors, e.g. [41, 52, 69, 74, 91, 126].

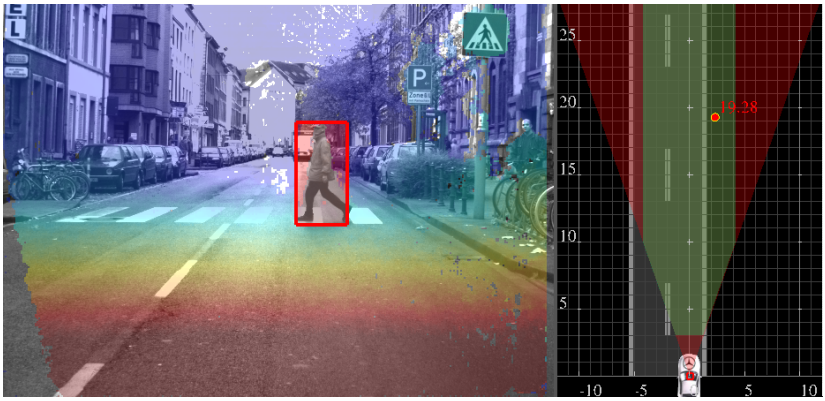


Figure 4.1: Pedestrian detection using the stereo-based system.

A second contribution involves a new large real-world stereo dataset for pedestrian detection which is used in our experiments. We make this dataset publicly

available for non-commercial purposes to encourage research and benchmarking[1]. The data is based on the established monocular *Daimler Pedestrian Detection Benchmark* [52], which is extended in several ways. First, the new benchmark includes the corresponding (left and right) stereo image pairs for the same 27-minute urban test sequence as used in [52], where previously only the left image was published. We further present a new stereo-vision sequence not containing pedestrians for bootstrapping. Instead of generating 3D ground-truth by back-projecting manually acquired pedestrian labels from the image into the world using the ground-plane constraint, we now derive more exact 3D ground-truth using shape information and stereo-vision. Finally, we enrich our test sequence by releasing vehicle data (velocity, yaw rate) estimated by on-board sensors to develop and evaluate more robust tracking algorithms.

Evaluation, comparison and ranking of pedestrian detection systems requires publicly available datasets which can be used as a common reference ground to benchmark many different systems. As a result of various systems having different requirements in terms of data used (e.g. gray-level appearance, optical flow, stereo, color or vehicle data), a multitude of datasets are available. Data acquisition further varies with the actual application area of the system, e.g. surveillance, IV or action recognition. Roughly, pedestrian datasets can be categorized into classification and detection datasets.

Classification datasets, e.g. [34, 46, 51, 75, 126, 135, 137], are mainly used to evaluate a combination of a feature set and a pattern classifier using a given set of pedestrian (positive) and non-pedestrian (negative) cut-out samples. For pedestrians, such samples are typically extracted from manually labeled image data resulting in accurately aligned pedestrian cut-outs. Non-pedestrian cut-outs can be extracted randomly or by some pre-processing method from images not containing pedestrians. In this context, pre-processing is used to focus on application-relevant "difficult" samples. A fixed set of positive and negative training and test samples is supplied for benchmarking. To allow for classifier bootstrapping, additional negative images are often provided.

Detection datasets, e.g. [4, 41, 52, 53, 73, 179, 180], containing cut-outs for training and full images for test data are used to benchmark integrated pedestrian detection systems. Although the pedestrian classifier is the most important module of most systems, differences in relative performance can also arise from varying hypotheses generation or tracking modules. Further, the extended scanning of an image skews the relation of pedestrian and non-pedestrian windows

[1] See http://www.gavrila.net/Datasets/datasets.html or contact the author.

	Training				Testing									
	# pedestrians	# pos. image	#neg. samples	# neg. images	# labels	# images	Traj. Labels	Focal Length (mm)	Stereo	Baseline (cm)	Vehicle Data	Platform	City / Setup	Year
ETH [53]	1578	490	-	-	10k	2293	-	8	✓	40	-	stroller	Zurich / city	2007
CALTECH [42]	192k	67k	-	61k	155k	65k	$\approx 1k$	7.5	-	-	-	vehicle	Los Angeles / urb.	2009
TUD-Brussels [179]	1776	1092	-	192+26	1326	508	-	8	-	-	-	vehicle	Brussels / city	2009
Daimler Mono [52]	3915	-	-	6744	56k	22k	259	12	-	-	-	vehicle	Aachen / urb.	2009
CVC-02 [73]	1016	-	7650	153	7983	4634	-	6	✓	12	-	vehicle	Barcelona / urb.	2010
Daimler Stereo [99]	3915	-	-	7129	56k	22k	259	12	✓	30	✓	vehicle	Aachen / urb.	2011

Table 4.1: Summary of the available pedestrian datasets recorded from a moving platform in an urban environment $(1k = 10^3)$.

used for testing - typically, the test images only contain a few pedestrians, whereas many thousands of regions not corresponding to pedestrians may be scanned per image.

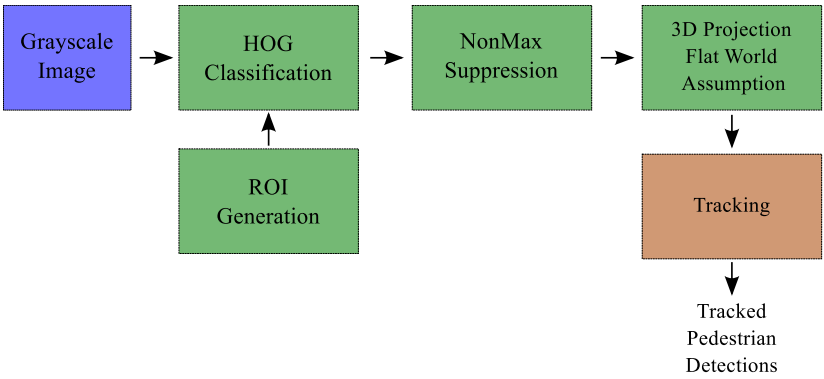
Although a classification dataset allows the isolated performance analysis of a classification module, results do not necessarily generalize to the performance of a fully integrated pedestrian detection system, as noted above. On the other hand, evaluating the classification module of an integrated system in an isolated brute-force (monocular) sliding window detection setting, e.g. [41], does not necessarily correspond to the actual application context either. Both evaluation methodologies have their justification and the choice strongly depends on the application and evaluation context.

In the context of advanced driver assistance systems (ADAS) in the intelligent vehicles domain, video sequences acquired in a realistic urban traffic environment are crucial for an adequate evaluation of state-of-the-art systems. Depending on the design of the systems under consideration, different image cues may be required. Systems utilizing optical flow require a sufficiently large frame rate while stereo based systems need additional image data to derive depth information. Table 4.1 shows an overview of available pedestrian detection datasets recorded from a moving platform, as well as their main properties. Manually annotating video data is a time-consuming and tedious work. In [41], an interactive procedure where the system generated intermediate labels by interpolation between manually assigned labels is proposed. Especially for sequences recorded with a large frame rate this approach can reduce the costs for labeling at the expense of accuracy [73].

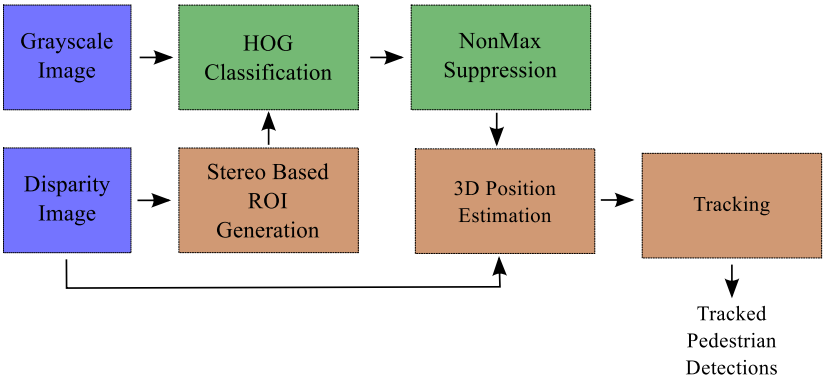
In the remainder of this chapter, we introduce the systems used for benchmarking and present our new stereo-based benchmark dataset and our experimental evaluation.

4.2 Selected Pedestrian Detection Systems

In our experiments, we compare the performance of two state-of-the-art baseline systems. The first system solely depends on a monocular camera setup for detection and tracking, see [52]. In contrast, the second system utilizes stereo data for hypotheses generation and refined pedestrian localization, i.e. an adapted version of [72]. Stereo data is computed using the "Semi-Global Matching" (SGM) algorithm [89] algorithm which provides dense disparity maps. Figure 4.2 illustrates the processing steps of the selected systems.



(a) Mono System



(b) Stereo System

Figure 4.2: Comparison of the processing steps for the (a) mono and (b) stereo system.

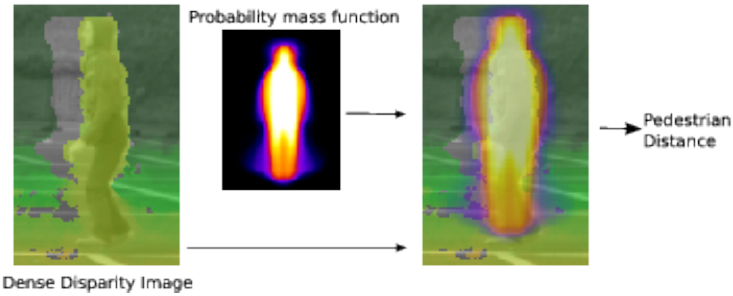


Figure 4.3: Pedestrian distance estimation using weighted disparity values.

Both systems utilize an initial set of ROIs generated for various detector scales and image locations using a flat-world assumption and ground-plane constraints. For the stereo-based system, ROIs at a certain distance are only generated if the number of depth features for the distance exceeds a percentage of the ROI area. ROIs are then passed to the classification module which uses histograms of oriented gradients (HOG) features [34] on gray-scale image data. Extracted features are classified by a linear support vector machine (linSVM). To speed-up the feature computation, we implemented the integral histograms of oriented gradients approach e.g. [186], which does not allow for the inclusion of tri-linear interpolation steps, as described in [34]. The resulting computational speed-up comes at the cost of a lower detection performance [186].

Multiple detector responses at near-identical locations and scales are addressed by applying confidence-based non-maximum suppression to the detected bounding boxes using pairwise box coverage. Two system detections a_i and a_j are subject to non-maximum suppression if their coverage

$$\Gamma(a_i, a_j) = \frac{A(a_i \cap a_j)}{A(a_i \cup a_j)},$$

the ratio of intersection area and union area, is above θ_n. For the following experiments $\theta_n = 0.5$ has been selected.

To allow possible collision mitigation maneuvers, the pedestrian position with respect to the vehicle is required. From the available stereo data, the pedestrian position is estimated by averaging the weighted disparity values in the detected box in the image and back-projecting the foot-point into 3D world coordinates

onto the ground-plane using known camera geometry, see [72]. With manually labeled pedestrian shapes, a mask has been derived for importance weighting of disparity values depending on their location, as shown in Figure 4.3. Pedestrian positions for the monocular system are computed with the assumption that pedestrians are standing on the (flat) ground-plane (ground-plane constraint).

Lateral (x) and longitudinal (z) pedestrian positions are tracked using a Kalman filter [10] with measurement vector $\mathbf{z} = \left(x, z\right)^T$ and the state vector $\mathbf{x_k} = \left(x, z, v_x, v_z\right)^T$, with v_x and v_z denoting the pedestrian velocity. We assume no abrupt velocity changes of the pedestrian and consequently use a constant velocity (CV) model. With vehicle velocity v^e and yaw-rate $\dot{\psi}^e$, estimated from on-board sensors, the vehicle ego-motion is compensated. As a possible extension, visual measurements could additionally be incorporated at this point. Figure 4.4 illustrates the simplified motion of the vehicle using the one-track vehicle model [139].

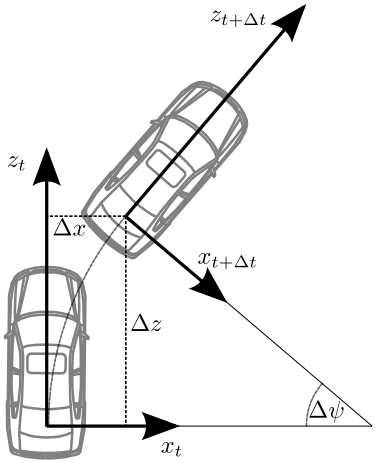


Figure 4.4: Single-Track model used for ego-motion compensation.

Ego-motion compensation is integrated into the prediction step of the Kalman filter. Between time-step t and $t + \Delta t$ the vehicle travels the distance $\left(\Delta x, \Delta z\right)$ with orientation change $\Delta\psi^e$. Moving on the curve radius $r = v^e \cdot \dot{\psi}^e$ following translation and rotation parameters apply:

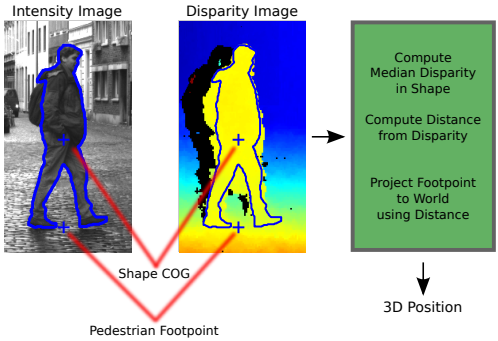


Figure 4.5: Pedestrian 3D world position derived from manual labeled pedestrian shaped and dense stereo data

$$\begin{aligned} \Delta\psi^e &= \dot{\psi}^e \Delta t \\ \Delta x &= v^e (\dot{\psi}^e)^{-1} [1 - cos(\Delta\psi)] \\ \Delta z &= v^e (\dot{\psi}^e)^{-1} sin(\Delta\psi) \end{aligned}$$

So the predicted pedestrian state $\hat{x}_{k|k-1}$ in the vehicle coordinate system for $t + \Delta t$ is computed using

$$\hat{\mathbf{x}}_{\mathbf{k|k-1}} = F[\hat{\mathbf{x}}_{\mathbf{k-1}} - \mathbf{x}_{\mathrm{cog}}] + [\mathbf{x}_{\mathrm{cog}} - \begin{pmatrix} \Delta x \\ \Delta z \\ 0 \\ 0 \end{pmatrix}]$$

with $\mathbf{x}_{\mathrm{cog}}$ describing the translation to the vehicle center-of-gravity and F describing the state transition matrix respecting the vehicle ego-orientation change.

$$F = \begin{pmatrix} cos(\Delta\psi) & sin(\Delta\psi) & cos(\Delta\psi)\Delta t & sin(\Delta\psi)\Delta t \\ -sin(\Delta\psi) & cos(\Delta\psi) & -sin(\Delta\psi)\Delta t & cos(\Delta\psi)\Delta t \\ 0 & 0 & cos(\Delta\psi) & sin(\Delta\psi) \\ 0 & 0 & -sin(\Delta\psi) & cos(\Delta\psi) \end{pmatrix}$$

Measurement to track associations in the track management are handled using the global nearest neighbor algorithm [10] with prior rectangular gating on the

Training	
# unique pedestrians	3915
# pedestrian samples	15660
# neg. frames (stereo pairs)	7129
Testing	
# frames (stereo pairs)	21790
# labels	56484
# pedestrian traj.	259

Table 4.2: Daimler Stereo-Vision Pedestrian Benchmark dataset statistics.

predicted pedestrian position. New tracks result from measurements that can not be assigned to an existing track. Starting in the state *hidden*, new tracks enter the state *confirmed* after n measurement to track associations. After m missed associations *confirmed* tracks are terminated. Here we use $n = 2$ and $m = 2$ for the track management. Only confirmed tracks are regarded as valid system outputs.

4.3 Dataset Overview

We extend the benchmarking dataset of [52] to contain stereo image pairs to allow the computation of distance data using different stereo algorithms. Stereo video data not containing pedestrians is additionally supplied to allow training and bootstrapping of different classification algorithms.

Test data has been recorded with 15 frames per second (fps) enabling the computation of optical flow data. Vehicle velocity and yaw-rate measurements from on-board sensors are provided for each frame to enable integration into a tracking and decision making system. All sequences are recorded in an urban environment representing a realistic challenge for today's pedestrian detection systems. Example images from training and testing data are given in Figure 4.6.

A summary of the dataset statistics is given in Table 4.2. By shifting and mirroring, 15660 pedestrian training samples are created from 3915 unique pedestrian samples. A training sample resolution of 48×96 pixels with a border of 12 pixels around the pedestrians is used. Negative training samples (≈ 15600) are randomly cropped from the bootstrapping image sequence using ground-plane constraints.

In [52], 3D ground truth from camera geometry in addition to bounding box

Figure 4.6: Overview of the detection benchmark dataset: (a) pedestrian training samples. (b) non-pedestrian training images. (c) annotated test images.

labels has been provided. The 3D ground truth data has been revised. We use 3D ground truth from stereo data because of its robustness to vehicle pitch variations and violations of the flat-world assumption. Figure 4.5 illustrates the ground truth generation. To increase precision of estimated 3D positions, unoccluded pedestrians in the required detection area (see Section 4.4) have manually been shape labeled. Pedestrian distance is derived from the median of disparity values located on the pedestrian body. In combination with the pedestrian footpoint determined from the shape center-of-gravity (COG) and known camera parameters the 3D position is computed.

4.4 Experiments

In the following the performance for the classifier modules and complete system configurations of the two selected baseline systems is compared. System setup and evaluation parameters are described in detail to allow reproducibility of the results.

4.4.1 System Configuration

Parameters for the ROI generation have been chosen to correspond to pedestrians at a longitudinal distance of $10\ m$ to $25\ m$ in front of the vehicle and $\pm 4\ m$ in lateral direction. Pedestrians with a height of $1.6\ m$ up to $2.0\ m$ standing on the ground are searched in the detection area. To cover the detection area, ROIs ranging from $h_{min} = 72\ px$ to $h_{max} = 206\ px$ are required. ROIs with an aspect ratio of $2{:}1$ are generated in a multi-scale sliding window fashion on the ground-plane using a flat world assumption with a pitch tolerance of $\pm 1^{\circ}$. Given the pitch tolerance, ROIs are located at most $11\ px$ above or below the ground plane. With a scale step factor $\Delta_s = 1.1$ a total of 12 scales are generated. ROI locations are shifted at fractions $\Delta_x = 0.1$ of their height and $\Delta_y = 0.25$ of their width resulting in a total of 5920 generated ROIs, see [52].

The HOG/linSVM classifiers are trained and iteratively bootstrapped, as in [52, 126]. Gradients for the HOG features are computed with $\left(-1, 0, 1\right)$ masks. Orientation histograms with 8 bins are generated from cells with a size of 8×8 pixels. Overlapping descriptor blocks (2×2) are normalized using the L_2-norm. An initial classifier (iter0) has been trained with the positive and negative training samples described in Section 4.3. For both systems, this initial classifier is iteratively applied to the set of non-pedestrian images to collect additional

false positives for the next round of classifier training. This process is repeated until (test) performance saturates.

4.4.2 Evaluation

For evaluation, we follow the well-established methodology of [52, 72]. To compare system output with ground-truth, we need to specify the localization tolerance, i.e. the maximum positional deviation that still allows to count the system detection as a match. This localization tolerance is the sum of an application-specific component (how precise does the object localization have to be for the application?) and a component related to measurement error (how exact can we determine true object location?). Object localization tolerance is defined (see [52, 72]) as percentage of distance, for longitudinal and lateral direction (Z and X), with respect to the vehicle. For our evaluation of the video sensing component, we use $Z = 30\%$ and $X = 10\%$, which means that, for example at $10\ m$ distance, we tolerate a localization error (including ground truth measurement error) of $\pm 3\ m$ and $\pm 1\ m$ in the position of the pedestrian, longitudinal and lateral to the vehicle driving direction, respectively. Partial visible pedestrians are matched in 2D with a box coverage of $\theta_n = 0.25$. Pedestrians outside the detection area or partial visible are regarded as optional and are neither credited nor penalized. For this application we allow many-to-many correspondences, i.e. a ground truth object is considered matched if there is at least one system detection matching it.

Classification Performance

Figure 4.7 and 4.8 illustrates the performance of the two systems after each bootstrapping iteration. Both classifiers improve with additional bootstrapping iterations. For the monocular system (Figure 4.7) performance saturates after three iterations. By augmenting the set of negative training samples with "difficult" examples performance is pushed by a factor of 14 at similar detection rates (60%). Because the stereo system generates ROIs only at highly structured locations the benefit of bootstrapping is less evident. After the first bootstrapping iteration performance does no longer improve.

A direct comparison of the monocular system with the stereo system (Figure 4.9) shows the benefit of the stereo-based ROI generation and improved localization. For a detection rate of 60% the number of false positives is reduced by a factor of 4. We attribute this to the reduced number of generated ROIs

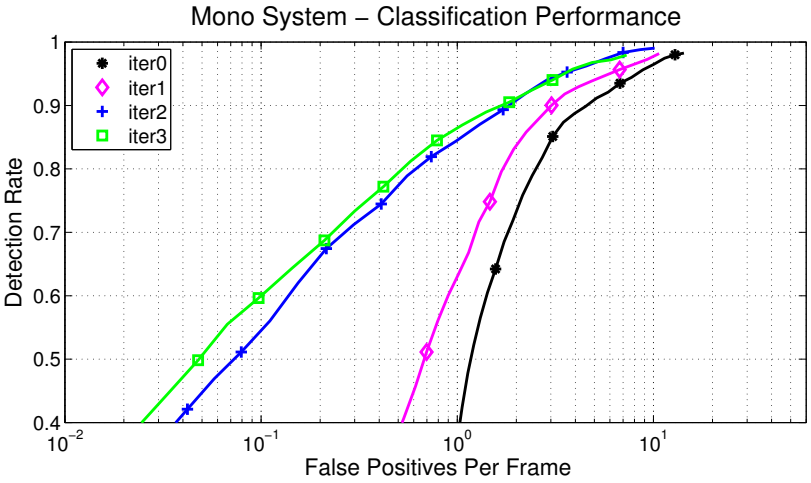


Figure 4.7: Classification performance of the mono system for different bootstrapping iterations.

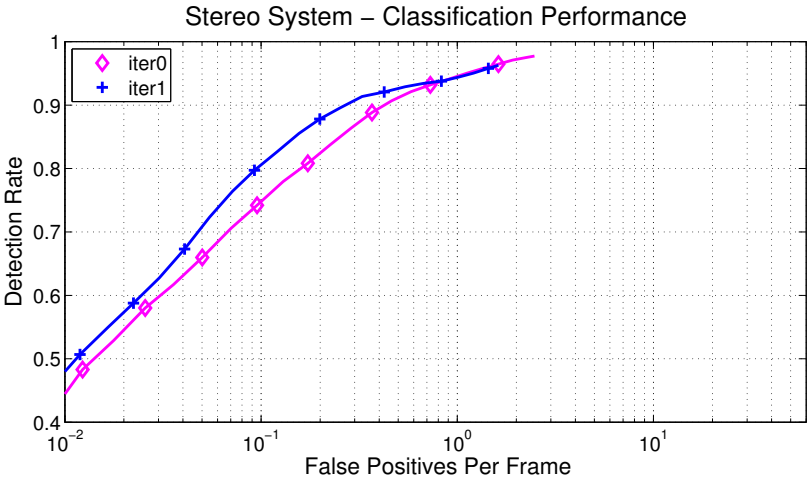


Figure 4.8: Classification performance of the stereo system for different bootstrapping iterations.

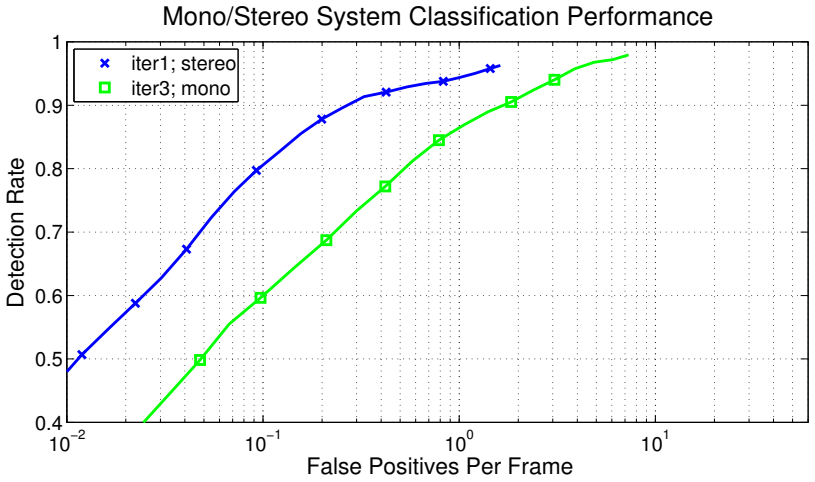


Figure 4.9: Performance comparison of the mono and the stereo system.

containing random structures. Figure 4.10b and 4.10c illustrate some typical false positive examples of the detectors.

System Performance

Overall detection performance of the systems including the tracking module is given in Table 4.3. Classifier thresholds are selected from Figure 4.9 using a common reference point of 60% detection rate. For additional insight, we consider detection rate and precision (percentage of system detections that are correct) on both the frame- and trajectory-level. For the latter, we distinguish two types of trajectories: "class-A" and "class-B" which have 50% and 1 frame entries matched. Thus, all "class-A" trajectories are also "class-B" trajectories; the different classes of trajectories represent different quality levels that might be relevant for particular applications. At comparable detection rate levels, the stereo system has a significant higher precision (approximately 20%). False alarms are reduced by a factor of $4-5$ over the mono system, similar to the previous evaluation of the classification modules (see Figure 4.9).

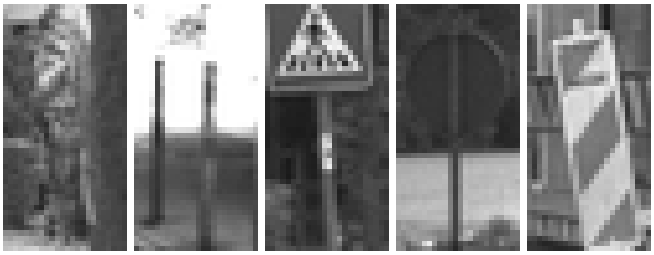
		F	A	B
Mono System	Detection Rate (all)	66.58%	70.21%	78.72%
	Precision (all)	39.45%	32.50%	39.19%
	FA min	0.11	13.12	11.82
Stereo System	Detection Rate (all)	58.75%	53.19%	72.34%
	Precision (all)	62.14%	50.0%	56.10%
	FA min	0.02	3.05	2.68

Table 4.3: System performance of the mono system vs. the stereo system after tracking.

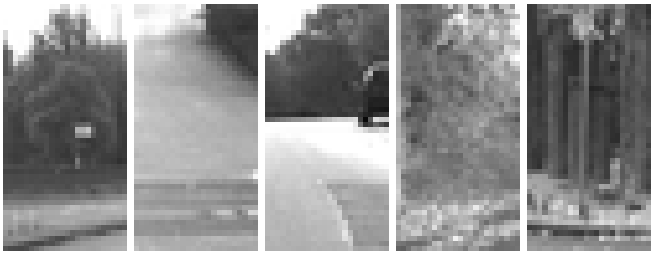
4.5 Conclusion

This chapter presented an experimental study comparing monocular and stereo-base pedestrian detection. Furthermore the benefit of stereo vision for ROI generation and localization has been quantified. At equal detection rates, false positives are reduced by a factor of 4-5 with stereo over mono, using the same HOG/linSVM classification component. To allow reproducibility of the results, system configurations and evaluation parameters were described in detail.

(a) Examples of correct detections of the mono and stereo system



(b) Examples of false detections of the stereo system.



(c) Examples of false detections of the mono system.

Chapter 5

The Benefits of Dense Stereo

5.1 Overview

In this chapter, we propose the use of dense stereo information in two modules of our pedestrian detection system: First, we estimate the varying road profile and camera orientation from dense stereo, to refine regions-of-interest with respect to possible pedestrian locations, see Section 5.3. Second, we enrich an intensity-based feature space with features operating on dense depth images to improve pedestrian classification performance, see Section 5.4.

Previous IV applications have typically used sparse, feature-based stereo approaches (e.g. [2, 72, 128]) because of lower processing cost. However, with recent hardware advances, real-time dense stereo has become feasible [169] (here a hardware implementation of the semi-global matching (SGM) algorithm [61, 89]) is used.

Both sparse and dense stereo approaches have proved suitable to dynamically estimate camera height and pitch angle, in order to deal with road imperfections, speed bumps, car accelerations, etc. But dense stereo also holds the potential to reliably estimate the vertical road profile. The more accurate estimation of ground location of pedestrians can be expected to improve system performance, especially when considering undulating, hilly roads.

Dense stereo can furthermore provide additional cues for pedestrian recognition. Up to now, the use of stereo information has been mainly limited to recovering 3D scene structure [53, 114], partial occlusion [46] and providing a focus-of-attention mechanism (e.g. [72, 76, 128, 185]).

The main contribution in this chapter is the use of dense stereo information in two modules of our pedestrian detection system: ROI generation and pedestrian classification. For ROI generation, we recover scene geometry in terms of camera height, camera pitch and road profile from dense stereo information on a frame-by-frame basis. Constraints on possible pedestrian locations are dynamically

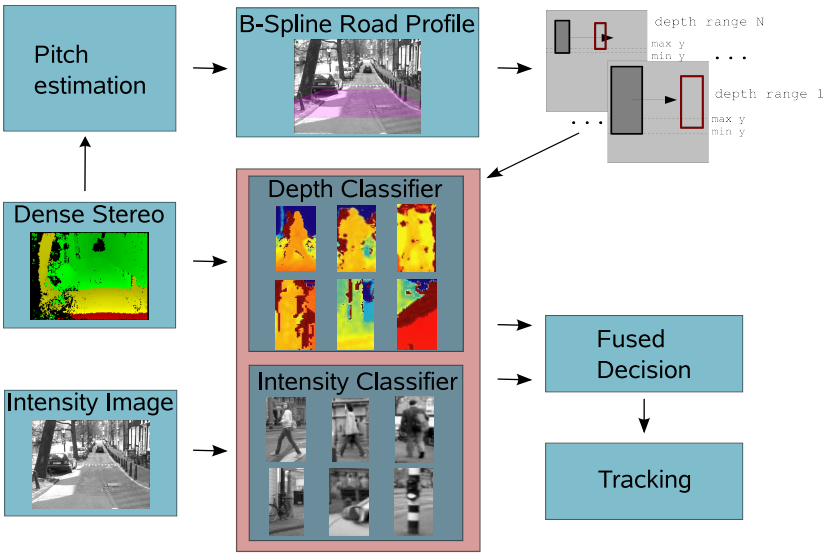


Figure 5.1: Overview of the dense stereo-based ROI generation and high-level fusion of intensity and depth classifiers. For depth images, warmer colors represent closer distances to the camera. Dense stereo is used for pitch estimation, B-Spline road profile modeling, obstacle detection and depth-based classification.

derived from the recovered models of camera and road geometry. With regard to pedestrian classification, we extract spatial features from dense depth images at medium resolution (pedestrian heights up to 80 pixels) and fuse them with an intensity-based feature set on classifier-level.

See Figure 5.1 for a system overview: First, the camera pitch angle is estimated by determining the slope with highest probability in the v-disparity map, for a reduced distance range. Second, a corridor of predefined width is computed using the vehicle velocity and the yaw rate. Only points that belong to that corridor will be used for subsequent road surface modeling. The ground surface is represented as a parametric B-Spline surface and tracked using a Kalman filter [176]. Reliability on the road profile estimation is an important issue which has to be considered for real implementations. Regions-of-interest (ROIs) are finally obtained by analyzing the multiplexed depth maps as in [72]. The re-

maining ROIs are classified using linear support vector machine (SVM) classifiers operating on histograms of oriented gradient (HOG) features, extracted from both intensity and dense depth data. We follow a classifier-level fusion strategy which bases the final decision on a combined vote of the individual classifiers. As opposed to fusion approaches using a joint feature space, e.g. [152, 175, 179], this strategy does not suffer from the increased dimensionality of the joint space, see [147, 152]. We assume our approach to generalize to other state-of-the-art features and classifiers, which are complex enough to capture the appearance of the pedestrian class, see [52]. Finally the detected pedestrians are tracked over time.

5.2 Dense Stereo

With two (or more) cameras, 3D information of the environment can be derived by finding the corresponding points across multiple cameras. A known stereo camera configuration constraints the location of corresponding image points to be on a single epipolar line. To simplify the matching process camera images are often rectified, resulting in epipolar lines that are parallel to image lines. For a point $l(u, v)$ in the left image and the corresponding point $r(u, v)$ in the right image, the disparity $d(u, v)$ can be computed using:

$$d(u, v) = l(u, v) - r(u, v) \tag{5.1}$$

Feature-based stereo vision systems typically provide depth measurements at points with sufficient image structure, whereas dense stereo algorithms estimate disparities at every pixel, including untextured regions. Only for regions which are visible in only one image no disparity values can be computed causing a "stereo shadow". Here we use a hardware implementation of the "Semi-Global Matching" (SGM) [89] algorithm which provides dense disparity maps in real-time, see Figure 5.7b.

Given the camera geometry with focal length f and the distance between the two cameras B, dense depth maps containing distance information can be computed using:

$$Z(u, v) = \frac{fB}{d(u, v)} \text{at pixel}(u, v). \tag{5.2}$$

These dense disparity/depth maps are used for the following ROI generation, road-profile estimation, obstacle detection and pedestrian classification.

5.3 Dense Stereo-Based ROI Generation

5.3.1 Modeling of Non-Planar Road Surface

Before computing the road profile, the camera pitch angle α is estimated using the v-disparity space. We assume that the camera is installed in a way that the roll angle is insignificant. A planar road surface in the camera coordinate system can be described using

$$Y(Z) \quad = \quad e \cdot Z - H, \tag{5.3}$$

with $e = \tan(\alpha)$ and camera height H. In v-disparity space this road is described using

$$v(d) \quad = \quad ad + c \tag{5.4}$$

where v is the image row and a, c are the slope and the offset which depend on camera height and tilt angle respectively. With the assumption of a fixed camera height H only the offset c of the line needs to be estimated in v-disparity space. Integrating the camera projection formula allows the computation of the slope

$$e(u, v) \quad = \quad \frac{v_0 - v}{f} + \frac{H}{Bf} d(u, v) \tag{5.5}$$

with the camera principal point v_0. Results are accumulated into a slopes histogram and the slope with the highest probability is selected to obtaining a first estimation of the camera pitch angle. Outliers are suppressed by computing a maximum disparity deviation for each image row depending on the tolerance of the camera height and tilt angle.

The next step consists in computing the predicted driving corridor in front of the vehicle. This is particularly important when the vehicle is taking a curve, since most of the points in front of the vehicle do not correspond to the road. Using a single track model with yaw-rate measurements $\dot{\psi}$ and velocity v from on-board sensors the vehicle path can be predicted. Moving on the curve radius $r = v \cdot \dot{\psi}$ the lateral (X) and longitudinal (Z) positions in the future t are calculated as

$$X(t) \quad = \quad v \cdot \dot{\psi}^{-1}[1 - cos(\dot{\psi}t)] \tag{5.6}$$

$$Z(t) \quad = \quad v \cdot \dot{\psi}^{-1} sin(\dot{\psi}t) \ . \tag{5.7}$$

The region of interest for selecting disparity values is computed by projecting

the corridor into image space using the estimated camera pitch. Here we use a corridor of width $\pm 1.5\ m$ and distance range $3 - 40\ m$ in the camera coordinate system.

The road profile is represented as a parametric B-Spline surface as in [176]. B-Splines are a basis for the vector space of piecewise polynomials with degree d. The basis-functions are defined on a knot vector c using equidistant knots within the observed distance interval. A simple B-Spline least square fit tries to approximate the 3D measurements optimally. However, a more robust estimation over time is achieved by integrating the B-Spline parameter vector c, the camera pitch angle α and the camera height H into a Kalman filter. Finally, the filter state vector is converted into a grid of distances Z_i and their corresponding road height values h_i as depicted in Figure 5.2. The number of bins of the grid will be as accurate as the B-Spline sampling.

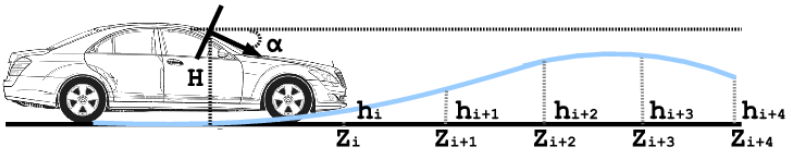


Figure 5.2: Road surface modeling. Distances grid and their corresponding height values along with camera height and tilt angle.

5.3.2 Outlier Removal

In general, the method of [176] works well if the measurements provided to the Kalman filter correspond to actual road points. The computation of the corridor removes a considerable amount of object points. However, there are a few cases in which the B-Spline road modeling still leads to bad results. These cases are mainly caused by vertical objects (cars, motorbikes, pedestrians, cyclists, etc.) in the vicinity of the vehicle. Reflections in the windshield can cause additional correlation errors in the stereo image. If we include these points, the B-spline fitting achieves a solution which *climbs* or *wraps* over the vertical objects.

In order to avoid this problem, the variance of the road profile for each bin σ_i^2 is computed. Thus, if the measurements for a specific bin are out of the bounds defined by the predicted height and the cumulative variance, they are not

added to the filter. Although this alternative can deal with spurious errors, if the situation remains for a consecutive number of iterations (e.g., when there is a vehicle stopped in front of the host vehicle), the variance increases due to the in-availability of measurements, and the points pertaining to the vertical object are eventually passed to the filter as measurements. This situation is depicted in Figure 5.3.

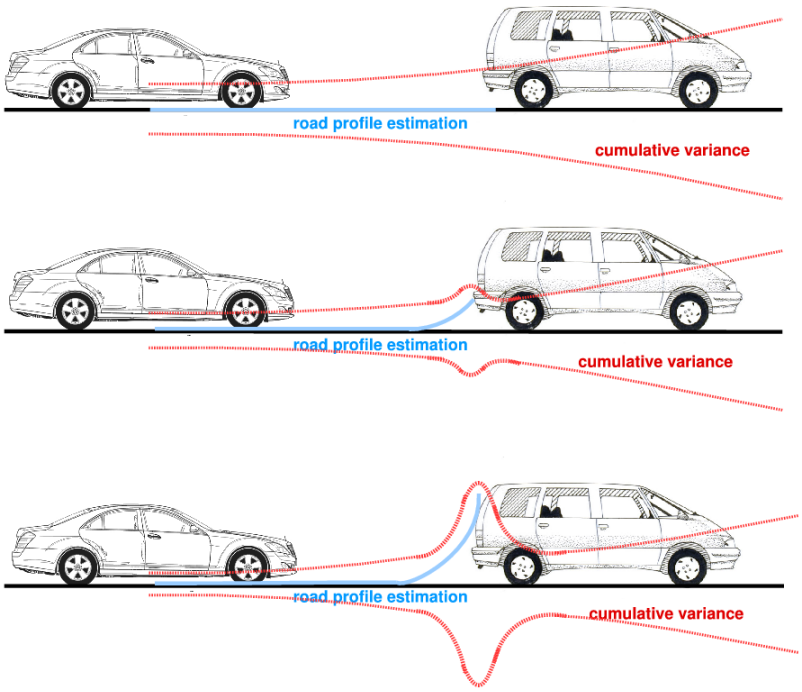


Figure 5.3: Wrong road profile estimation when a vertical object appears in the corridor for a consecutive number of frames. The cumulative variance for the bin in which the vertical object is located increases and the object points are eventually passed to the Kalman filter.

Accordingly, a mechanism is needed in order to ensure that points corresponding to vertical objects are never passed to the filter. We compute the variance of all measurements for a specific bin and compare it with the expected variance in

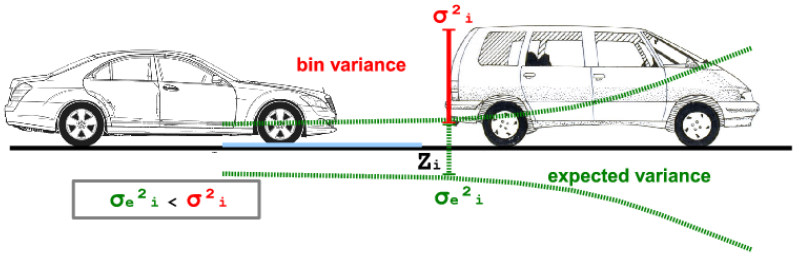


Figure 5.4: Rejected measurements for bin i at distance Z_i since measurements variance σ_i^2 is greater than the expected variance σ_{ei}^2 in that bin.

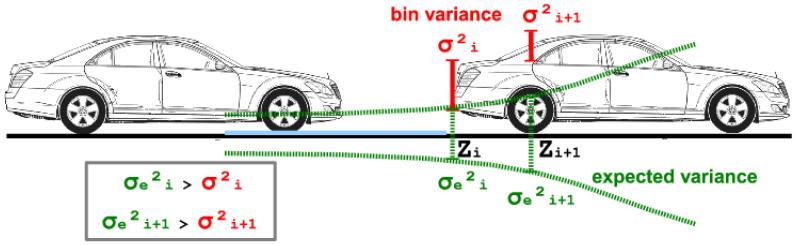


Figure 5.5: Accepted measurements for bins i and $i+1$ at distances Z_i and Z_{i+1} since measurements variances σ_i^2 and σ_{i+1}^2 are lower than the expected variances σ_{ei}^2 and σ_{ei+1}^2 in these bins.

the given distance. The latter can be computed by using the associated standard deviations σ_m via error propagation from stereo triangulation [134, 176]. If the computed variance σ_i^2 is greater than the expected one σ_{ei}^2, we do not rely on the measurements but on the prediction for that bin. This is useful for cases in which there is a vertical object like the one depicted in Figure 5.4.

However, in cases in which the rear part of the vertical object produces 3D information for two consecutive bins, this approach may fail depending on the distance to the vertical object. For example, in Figure 5.5 the rear part of the vehicle yields 3D measurements in two consecutive bins Z_i and Z_{i+1} whose variance is lower than the expected one for those bins. In this case, measurements will be added to the filter which will yield unpredictable results.

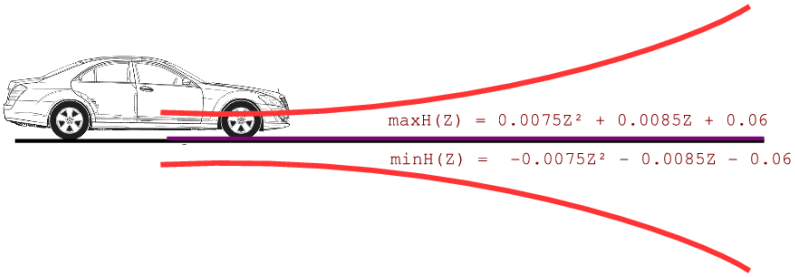


Figure 5.6: Second order polynomial function used to accept/reject measurements at all distances.

We therefore define a fixed region of interest, in which we restrict measurements to lie. To that effect, we quantify the maximum road height changes at different distances and fit a second order polynomial, see Figure 5.6. The fixed region can be seen as a compromise between filter stability and response to sharp road profile changes (undulating roads). Apart from this region of interest, we maintain the before-mentioned test on the variance, to see if measurements corresponding to a particular grid are added or not to the filter.

5.3.3 System Integration

Initial ROIs R_i are generated using a sliding window technique where detector windows at various scales and locations are shifted over the depth map. In previous work [72], the flat-world assumption along with known camera geometry restricted the search space. Pitch variations were handled by relaxing the scene constraints [72], e.g. camera pitch and camera height tolerances. In our approach, the use of dense stereo allows a reliable estimation of the vertical road profile, camera pitch and tilt angle.

In order to adapt the subsequent detection modules, we compute new camera heights H_i' and pitch angles α_i' for all bins of the road profile grid. After that, standard equations for projecting 3D points into the image plane are used.

First of all, dense depth maps are filtered as follows: points $P_r = (X_r, Y_r, Z_r)$ under the actual road profile, i.e., $Z_i < Z_r < Z_{i+1}$ and $Y_r < h_i$ and over the actual road profile plus the maximum pedestrian size, i.e. $Z_i < Z_r < Z_{i+1}$ and $Y_r > h_i + H_{max}$, are removed since they do not correspond to obstacles

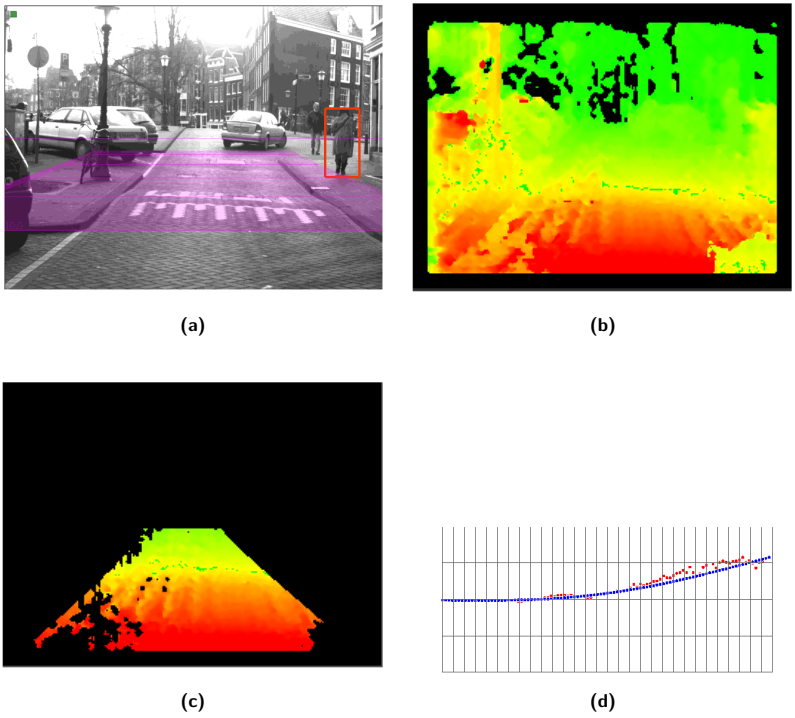


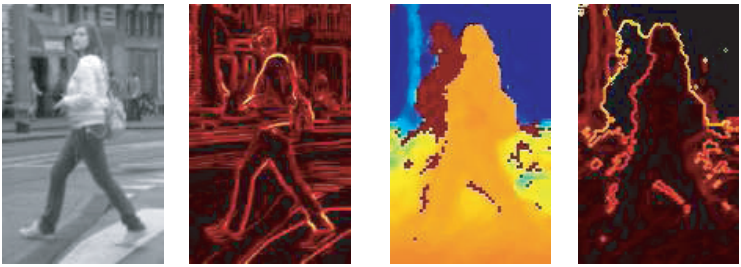
Figure 5.7: System example with estimated road profile and pedestrian detection. (a) Final output with detected pedestrian marked red. The magenta area illustrates the system detection area. (b) Dense stereo image. (c) Corridor used for spline computation after outlier removal. (d) Spline (blue) fitted to the measurements (red) in system profile view.

(possible pedestrians). The resulting filtered depth map is multiplexed into N discrete depth ranges, which are subsequently scanned with windows related to minimum and maximum extent of pedestrians. Possible window locations (ROIs) are defined according to the road profile grid (we assume the pedestrian stands on the ground). Each pedestrian candidate region R_i is represented in terms of the

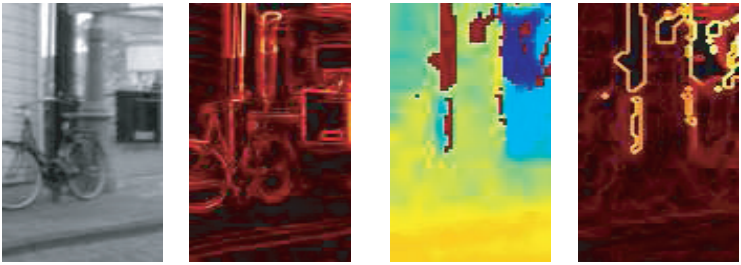
number of depth features DF_i. A threshold θ_R governs the amount of ROIs which are committed to the subsequent module. Only ROIs with $DF_i > \theta_R$ trigger the evaluation of the next cascade module. Others are rejected immediately.

5.4 Multi-Modality Classification

5.4.1 Spatial Depth and Intensity Features



(a) Pedestrian



(b) Non-Pedestrian

Figure 5.8: Intensity and depth images for pedestrian (a) and non-pedestrian samples (b). From left to right: intensity image, gradient magnitude of intensity, depth image, gradient magnitude of depth.

Dense stereo provides disparity and depth information for most image areas, apart from regions which are visible only by one camera (stereo shadow). See the dark red areas to the left of the pedestrian torso in Figure 5.8(a). Spatial features can be based on either depth Z (in meters) or disparity d (in pixels). As

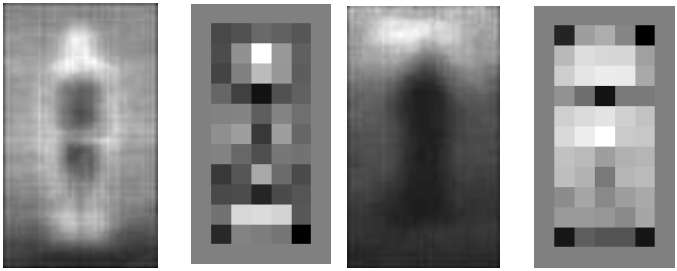
shown in Section 5.2, both are inverse proportional given the camera geometry with focal length f and the distance between the two cameras B.

Objects in the scene have similar foreground/background gradients in depth space, irrespective of their location relative to the camera. In disparity space however, such gradients are larger, the closer the object is to the camera. To remove this variability, we base our spatial features on depth instead of disparity.

A visual inspection of the depth images vs. the intensity images in Figure 5.8 reveals distinct properties which are unique to each modality. In intensity images, lower body features (shape and appearance of legs) are the most significant features of a pedestrian (see results of part-based approaches, e.g. [123]). The texture of the pedestrian exhibits lots of gradients and characteristic structure resulting from clothing. In contrast, the upper body area has dominant foreground/background gradients and is particularly characteristic for a pedestrian in the depth image. There are no significant depth gradients on areas corresponding to the pedestrian body (we assume pedestrians in an upright position). Additionally, the stereo shadow is clearly visible in the upper-body area (to the left of the pedestrian torso) and represents a significant local depth discontinuity. This might not be a disadvantage but rather a distinctive feature. The various salient regions in depth and intensity images motivate our use of fusion approaches between both modalities to benefit from the individual strengths, see Section 5.4.2.

To instantiate feature spaces involving depth and intensity, we utilize well-known state-of-the-art features, which focus on local discontinuities: Histogram of oriented gradient features with a linear support vector machine classifier (HOG/linSVM), see [35]. We assume our approach to generalize to other state-of-the-art features and classifiers, see [52]. To get an insight into the resulting HOG features, Figure 5.9 depicts the average gradient magnitude of all pedestrian training samples for both intensity and depth. We observe that gradient magnitude is particularly high around the upper body contour for the depth image, while being more evenly distributed for the intensity image. Further, almost no depth gradients are present on areas corresponding to the pedestrian body. Figure 5.9 further shows the weights of the linear SVM classifier after training on the corresponding feature sets. In this visualization, each "pixel" results from averaging the SVM weights over the underlying block of HOG features. In the intensity domain, HOG blocks corresponding to head/shoulder and leg regions have the highest weight. In case of the depth features, the upper body (coarse depth contrast between foreground and background) and torso areas (uniform

texture) are most indicative of a pedestrian.



(a) Intensity features **(b)** Depth features

Figure 5.9: Average gradient magnitude and SVM weights averaged over HOG blocks for intensity (a) and depth images (b) in the training set.

5.4.2 Classifier-Level Fusion Approach

A popular strategy to improve classification is to split-up a classification problem into more manageable sub-parts on data-level, e.g. using mixture-of-experts or component-based approaches [51]. A similar strategy can be pursued on classifier-level. Here, multiple classifiers are learned on the full dataset and their outputs combined to a single decision. Particularly, when the classifiers involve uncorrelated features, benefits can be expected. We follow a *Parallel Combination* strategy [43], where multiple feature sets (i.e. based on depth and intensity, see Section 5.4.1) are extracted from the same underlying data. Each feature set is then used as input to a single classifier and their outputs are combined. As opposed to creating a joint feature-space, classifier-level fusion does not suffer from effects related to the increased dimensionality of the joint space, see [147, 152].

For classifier fusion, we utilize a set of fusion rules which are explained below. An important prerequisite is that the individual classifier outputs are normalized, so that they can be combined homogeneously. The outputs of many state-of-the-art classifiers can be converted to an estimate of posterior probabilities [142]. We use this in our experiments.

Let $\mathbf{x}_k, k = 1, \ldots, n$, denote a (vectorized) sample. The posterior for the k-th sample with respect to the j-th object class (e.g. pedestrian, non-pedestrian),

estimated by the i-th classifier, $i = 1, \ldots, m$, is given by: $p_{ij}(\mathbf{x}_k)$. Posterior probabilities are normalized across object classes for each sample, so that:

$$\sum_j \left(p_{ij}\left(\mathbf{x}_k\right)\right) = 1 \tag{5.8}$$

Classifier-level fusion involves the derivation of a new set of class-specific confidence values for each data point, $q_j(\mathbf{x}_k)$, out of the posteriors of the individual classifiers, $p_{ij}(\mathbf{x}_k)$. The final classification decision $\omega(\mathbf{x}_k)$ results from selecting the object class with the highest confidence:

$$\omega(\mathbf{x}_k) = \arg\max_j \left(q_j\left(\mathbf{x}_k\right)\right) \tag{5.9}$$

We consider the following fusion rules to determine the confidence $q_j\left(\mathbf{x}_k\right)$ of the k-th sample with respect to the j-th object class:

Product Rule Individual posterior probabilities are multiplied to derive the combined confidence:

$$q_j\left(\mathbf{x}_k\right) = \prod_i \left(p_{ij}\left(\mathbf{x}_k\right)\right) \tag{5.10}$$

Linear SVM Rule A linear support vector machine is trained as a fusion classifier to discriminate between object classes in the space of posterior probabilities of the individual classifiers:

Let $\mathbf{p}_{jk} = \left(p_{1j}\left(\mathbf{x}_k\right), \ldots, p_{mj}\left(\mathbf{x}_k\right)\right)^T$ denote the m-dimensional vector of individual posteriors for sample $\mathbf{x}_k$ with respect to the j-th object class. The corresponding hyperplane is defined by:

$$f_j\left(\mathbf{p}_{jk}\right) = \mathbf{w}_j \cdot \mathbf{p}_{jk} + b_j \tag{5.11}$$

Here, $\mathbf{w}_j$ denotes the linear SVM weight vector, b_j a bias term and $\cdot$ the dot product. This linear SVM fusion rule equals a weighted sum of the individual classifier outputs, with weights and an additional bias term learned from the training set. The SVM decision value $f_j\left(\mathbf{p}_{jk}\right)$ (distance to the hyperplane) is used as confidence value:

$$q_j\left(\mathbf{x}_k\right) = f_j\left(\mathbf{p}_{jk}\right) \tag{5.12}$$

5.5 Experiments

We tested our integrated pedestrian detection system on a 6:40 min (5919 images) sequence recorded from a vehicle driving through the canal area of the city of Amsterdam during daytime. Because of the many bridges and speed bumps, the sequence is quite challenging for the road profiling module. Additionally, due to the complexity of the scenery this sequence is very demanding for a pedestrian classifier.

Our training samples comprise non-occluded pedestrian (in an upright position) and non-pedestrian cut-outs from both intensity and corresponding depth images, captured from a moving vehicle in an urban environment. See Table 5.1 and Figure 5.10 for an overview. All samples are scaled to 48×96 pixels with an eight-pixel border to retain contour information. For each manually labelled pedestrian cut-out, we randomly created 18 samples by horizontal mirroring and geometric jittering. Non-pedestrian samples were the result of a pedestrian shape detection pre-processing step with a relaxed threshold setting, i.e. containing bias towards more difficult patterns. We further applied an incremental bootstrapping technique, e.g. [52], by collecting additional false positives of the corresponding classifiers on an independent sequence and re-training the classifiers on the increased data set.

HOG features are extracted from those samples using 8×8 pixel cells, accumulated to 16×16 pixel blocks with 8 gradient orientation bins. Identical feature / classifier parameters were used for intensity and depth modalities.

In our test sequence, pedestrian bounding boxes were manually labelled. Their 3D position is obtained by triangulation in the two camera views. Only pedestrians with a distance of 12-27 in longitudinal and ± 4m in lateral direction were considered required. Pedestrians beyond this detection area were regarded as optional, i.e. the systems are not rewarded / penalized for correct / missing detections. This results in 1684 required pedestrian single-frame instances in 66 distinct trajectories which are required to be detected by our pedestrian detection system.

The match of a ground truth bounding box g_i to a system alarm a_j we use the bounding box coverage $\Gamma(g_i, a_j)$ as described in Chapter 4.2. In the following experiments we chose $\theta_n = 0.25$.

We evaluate the benefit of dense stereo on ROI generation and pedestrian classification both in isolation (Section 5.5.1 and 5.5.2) and in an integrated system variant (Section 5.5.3). Our baseline system involves static scene ge-

	Pedestrians (labelled)	Pedestrians (jittered)	Non-Pedestrians (bootstrapped)
Training Set (intensity)	16497	296946	183501
Training Set (depth)	16497	296946	188301

Table 5.1: Training set statistics. The number of pedestrian samples is identical for depth and intensity images. Non-Pedestrians samples for intensity and depth slightly vary due to the bootstrapping process.

ometry (flat-world assumption with fixed camera height and pitch) combined with intensity-only HOG/linSVM classification (we use the original code provided by [34]).

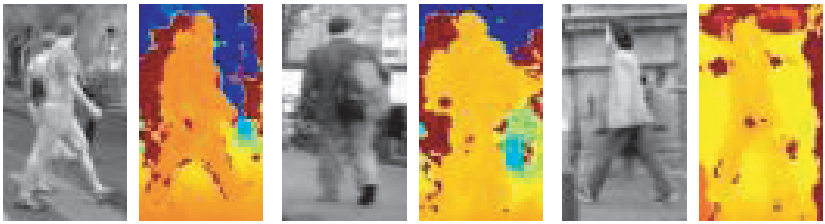
5.5.1 ROI Generation

The performance of the ROI generation module is evaluated in combination with the HOG/linSVM pedestrian classifier on intensity features only. Figure 5.11 compares the performance of the baseline system (flat-world assumption, fixed camera height and pitch) to the proposed ROI generation technique using a) pitch estimation with a flat-world assumption and b) pitch estimation with road profiling. It is observed, that pitch estimation (magenta $\times$) already improves the performance over the baseline (blue $+$), by distributing ROIs on a more adequate ground. An additional improvement is obtained by disregarding the flat-world assumption and estimating the actual road profile in front of the vehicle (green $\square$). For a detection rate of, say, 60%, the number of false positives is reduced by a factor of 2.3 using integrated pitch estimation and road-profiling compared to the baseline.

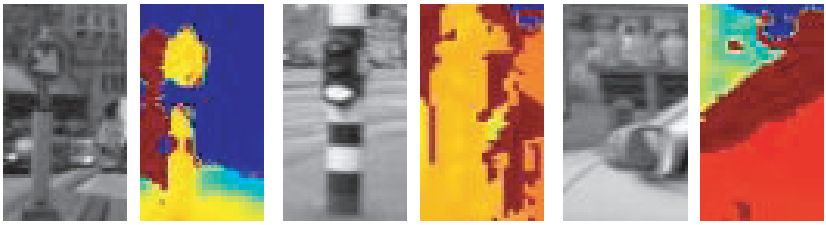
5.5.2 Multi-Modality Classification

Figure 5.12 compares the performance of classifiers in different modalities (depth and intensity), as well as fusion strategies. All classifiers are used with the base assumption of flat world and fixed camera height and pitch, i.e. the proposed dense stereo based dynamic scene constraints are not (yet) in place. Our results show, that a HOG/linSVM classifier on intensity features (blue $+$) outperforms the corresponding classifier on depth features (red $\times$).

The application of any proposed multi-modality fusion strategies, see Section 5.4.2 results in a significant performance boost (magenta $\diamond$ and green $\square$).



(a) Pedestrian samples



(b) Non-Pedestrian samples

Figure 5.10: Overview of (a) pedestrian and (b) non-pedestrian samples (intensity and corresponding depth images).

The performance difference between both fusion strategies is only minor. At a detection rate of 60% for example, the combined intensity-depth classifier reduces false positives by a factor of 3.3 over the intensity-only classifier. This clearly shows, that the different characteristics of depth and intensity can indeed be exploited, see Section 5.4.1.

5.5.3 Combined System Performance

In our next experiment, we combine the two best performing variants for ROI generation and pedestrian classification from our previous experiments: ROI generation using dense stereo based dynamic scene geometry and intensity-depth classification. Results are given in Figure 5.13. The integrated system (green $\square$) significantly boosts performance over the baseline system (blue $+$). At a detection rate of 60% for example, the number of false positives is reduced by a factor of 7.5, which almost equals the product (a factor of 7.6) of the individual benefits shown (factors of 2.3 for ROI generation and 3.3 for classification, respectively).

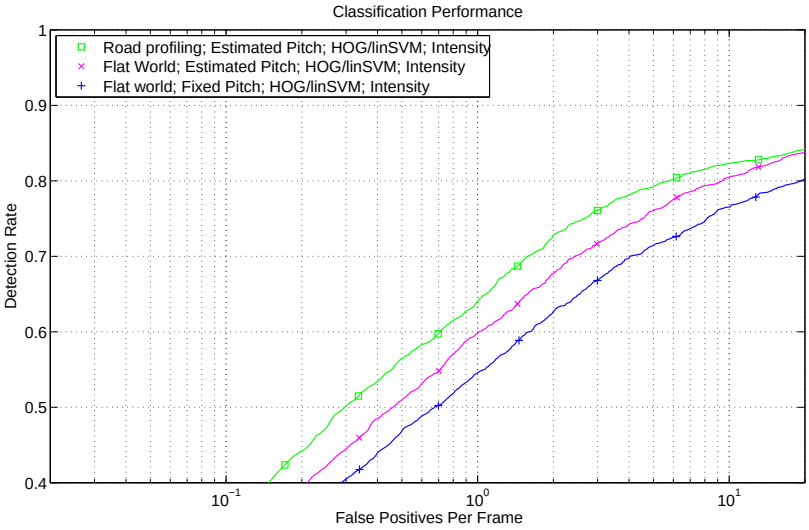


Figure 5.11: ROC performance of different variants of stereo-based ROI generation combined with an intensity-only HOG/linSVM pedestrian classifier.

This shows that the obtained performance boosts in the two different system modules are highly orthogonal to each other.

In our final experiment, we add a (rather simple) tracker to the system, to obtain results on trajectory-level. As described in Chapter 4.4, we distinguish between two types of trajectories:"class-A" and "class-B" trajectories. We compare the performance of the integrated system (dynamic scene geometry and intensity-depth classification) versus the baseline system (static scene geometry and intensity-only classification). Input to the tracker are pedestrian detections which were obtained from both systems by setting the classifier thresholds to correspond to a detection rate of 50% at frame-level.

Non-maximum suppression using the classifier outputs is applied to overlapping detections with a bounding box coverage of 50%. Remaining detections are tracked using a 2.5D $\alpha - \beta$ tracker, see [72]. New tracks are started after 3 continuous detections and closed after 2 successional missed detections. Table 5.2 summarizes the performance of the two systems. Frame-level sensitivity of the

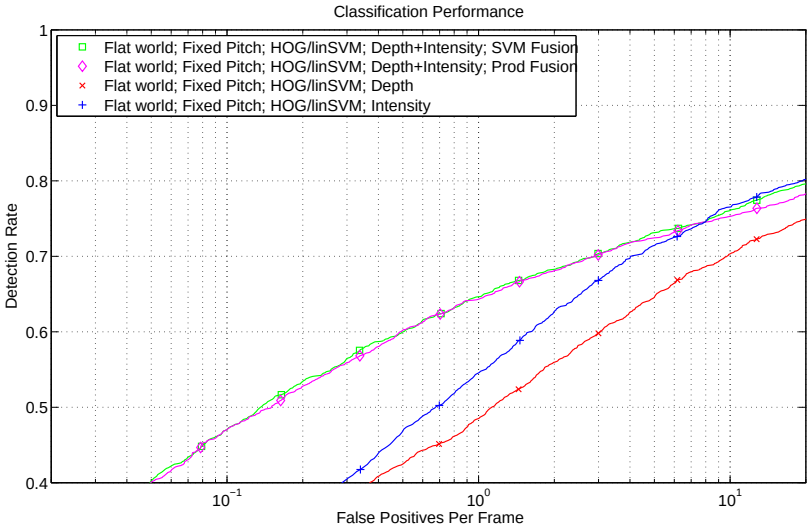


Figure 5.12: ROC performance of stereo-based ROI generation combined with intensity-depth HOG/linSVM pedestrian classification.

system using stereo information is slightly increased compared to the baseline system. But the main benefit lies in the reduction of false positives by a factor of approximately 5. The use of dense stereo information for both road profiling and classification reduces the number of false positives per frame from 0.336 to 0.066. A comparison of the observed benefit (factor of 5) to the system performance without tracking (benefit of factor 7.5) shows, that tracking reduces the absolute performance differences of the systems. Similar effects have been observed in [52]. Figure 5.14 illustrates system performance, including a typical false positives in a cluttered image region, and a missed pedestrian in not fully upright pose.

5.5.4 Processing Time

The hardware implementation of our SGM stereo requires $17\ ms$ per frame. Other system components run in (unoptimized) C/C++ code on a single core $2.66\ GHz$ Intel CPU. Camera pitch estimation requires $3.5\ ms$ per frame on

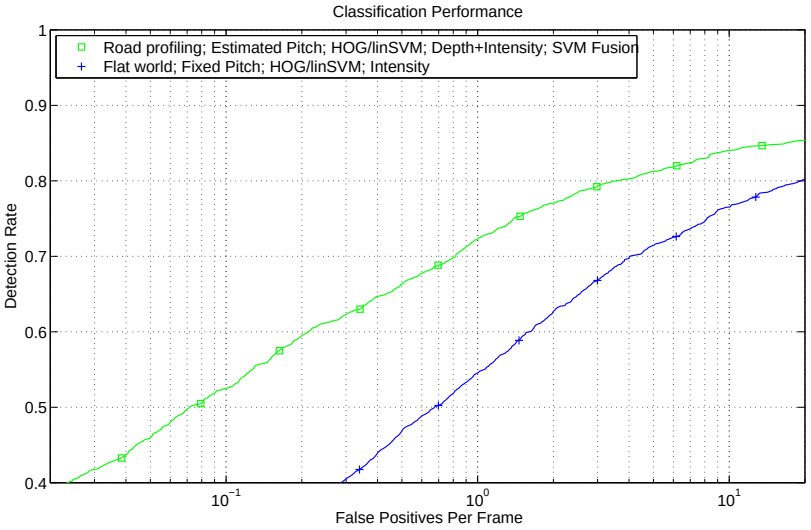


Figure 5.13: ROC performance comparing the baseline system using HOG/linSVM classifier on intensity images with the proposed system using road-profiling, pitch estimation and HOG/linSVM classifiers on depth and intensity images with SVM fusion.

average with the additional road profiling taking $26\ ms$. With a static pitch and flat world assumption the ROI grid is generated only once and reused in every frame. Incorporating pitch or road profile information requires an adjustment of the grid which takes $4\ ms$ per frame. Depending on the configuration of earlier modules, the number of ROIs passed to the classifier vary. For the system using static pitch and flat world, about 700 ROIs per frame need to be classified, on average. Using pitch and road profile estimation this number is reduced to about 600 ROIs per frame. HOG features need to be extracted and classified from the depth and intensity data which doubles the costs for classification. On a multiprocessor architecture, feature extraction and classification for each modality could be processed in parallel. Processing time for any of the described rules to fuse the classifier decision values are minor and hence neglected. In our setup, feature extraction, classification, fusion and tracking requires approx. $500\ ms$ per frame, on average. Note that processing costs do scale sub-linearly

		F	A	B
Base System	Sensitivity	55.58%	60.53%	77.63%
	Precision	64.07%	52.36%	56.74%
	FA frame, min	0.336	40.80	37.05
Prop. System	Sensitivity	57.54%	63.16%	78.95%
	Precision	90.38%	81.71%	84.09%
	FA frame, min	0.066	9.30	8.10

Table 5.2: System performance of the integrated system vs. the baseline system after tracking.

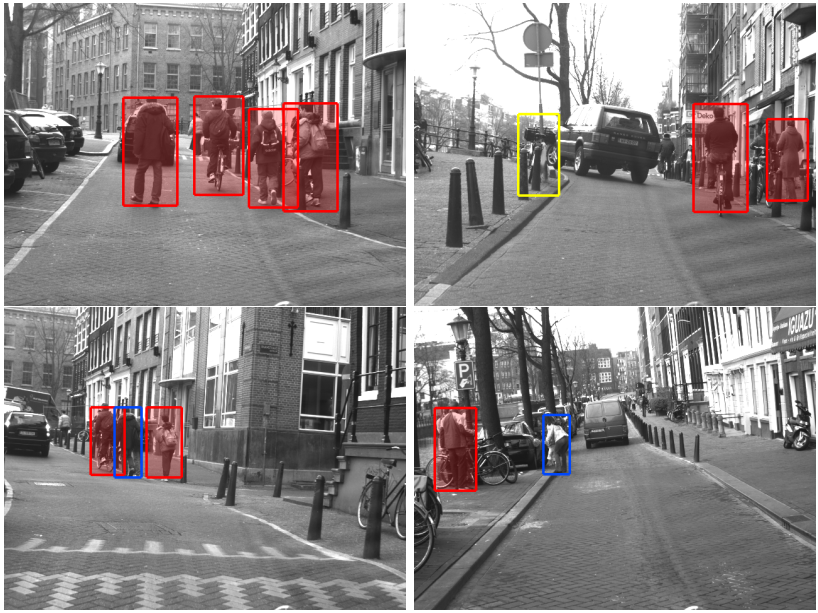


Figure 5.14: Examples of system detections (red), false positives (yellow) and missed pedestrians (blue).

with the number of ROIs, since feature computation can be shared among several overlapping ROIs (in the same modality), e.g. using integral histograms [186].

5.6 Discussion

Our performance evaluation focused on demonstrating the relative improvements arising from the use of dense stereo, i.e. the reduction of false positives at constant sensitivity levels by a factor of 7.5 after the classification module and by a factor of 5 after the tracker, respectively. On absolute terms, the (class-B) trajectory-level system performance of approximately 80% sensitivity and 8 false detections per minute (cf. Table 5.2) seems far from performance levels necessary in a realistic application. However, this perceived performance gap for the most part stems from the exceeding difficulty of our test sequence (undulating roads, bridges, speed bumps, very complex urban scenery), which was specifically chosen as a challenging test bed for the proposed road profiling module, see Section 5.5. Other studies have demonstrated differences of orders of magnitude in the performance of otherwise identical systems resulting from the use of different datasets, e.g. [41, 170].

In this work, we did not heavily optimize the feature sets with regard to the different modalities. Instead, we transferred general knowledge and experience from the behavior of features and classifiers from the intensity domain to the depth domain. At this point, it is not clear, if (and how) additional modification and adaptation of the feature sets could further improve performance.

We did not particularly focus on processing time constraints in this chapter. However, we do expect that software optimization and hardware implementation (e.g. DSP, FPGA) can result in real-time applicability of the proposed algorithms, cf. [12, 87, 122].

5.7 Conclusion

We investigated the benefits of dense stereo for a pedestrian detection system on challenging real-world data (i.e. undulated roads, bridges and speed bumps). The improved ROI generation utilizes dense stereo data for pitch estimation, road profiling and obstacle detection. Compared to our base system with flat world assumption and fixed pitch a reduction of false positives by a factor of 2.3 at similar detection rates was demonstrated. By fusing classifier responses from different modalities (intensity and depth), we additionally obtained a reduction of false positives by a factor of 3.3. Combining the proposed ROI generation and high-level fusion resulted in a reduction of false positives by a factor of 7.5 at classification-level and by a factor of 5 at tracking-level, respectively.

Chapter 6

Fusion of Generic Obstacle Detection and Pedestrian Recognition

6.1 Introduction

The contributions in this chapter are as follows. The main contribution is the description of an integrated active pedestrian safety system, which combines sensing, situation analysis, decision making and vehicle control. The secondary contribution concerns the sensing component; it is based on stereo vision and fuses two complementary approaches for added robustness: motion-based object segmentation and pedestrian recognition. The highlight of the system is the ability to decide within a split second whether to perform automatic braking or evasive steering, and to execute this maneuver reliably, at relatively high vehicle speed (up to $50\ km/h$).

6.2 Video-based Pedestrian Sensing

6.2.1 Single-Frame Pedestrian Recognition (PedRec)

Initial regions of interest (ROIs) are generated using the sliding window technique described in [72]. The depth image, obtained by stereo vision, is scanned with windows related to the maximum extents of pedestrians, assuming the latter are standing on the ground plane, while taking into account appropriate positional tolerances (e.g. vehicle pitch, slightly curved roads vertically). The locations where the number of (depth) features exceeds a percentage of the window area are added to the ROI list for the subsequent pedestrian classification. Candidates are classified following the HOG/linSVM approach of Dalal and Triggs [34]. Multiple detector responses at near identical locations and scales are addressed by applying confidence-based non-maximum suppression to the detected bounding boxes using pairwise box coverage: two system detections a_i and a_j are subject

to non-maximum suppression if their coverage $\Gamma(a_i, a_j) = \frac{A(a_i \cap a_j)}{A(a_i \cup a_j)}$, the ratio of intersection area and union area, is above θ_n.

The distance of a detected pedestrian in the image is estimated using the computed dense stereo image. Because the exact contour of the pedestrian is unknown all possible pedestrian shapes are considered in the depth estimation process using a probability mass function, as described in Chapter 4.2. Figure 4.3 illustrates the depth estimation procedure. Distance values in the depth image for a given bounding box are weighted and averaged using the probability mass function. The 3D position of the pedestrian is given by projecting the vertical line going through the bounding box center and the computed box distance. Detected 3D pedestrian locations are passed untracked to the fusion module.

6.2.2 Detection of Moving Objects (6D-Vision)

Using a stereo camera set-up, the 3D structure of the observed scene can be immediately obtained by a stereo algorithm (e.g. [62, 88]). Usually, to identify individual objects, this information is accumulated in an evidence-grid-like structure, followed by a connected-component analysis [117]. To obtain the motion of the identified objects, the objects are then tracked over time and their velocity is estimated by means of filtering. The disadvantage of this standard approach is that the performance of the detection depends highly on the correctness of the segmentation. Especially moving objects close to stationary ones – e.g. the moving pedestrian behind the standing vehicle are often merged and therefore not detected.

To overcome this problem, we proposed in [64, 144] to base the detection not only on the stereo information, but also on the 3D motion field. The reconstruction of the 3D motion field is performed by the so called 6D-Vision algorithm. The basic idea is to track points with depth known from stereo vision over two and more consecutive frames and to fuse the spatial and temporal information using Kalman filters. The result is an improved accuracy of the 3D-position and an estimation of the 3D-motion of the considered point at the same time. This fusion implies the knowledge of the motion of the observer, also called the ego-motion. It is estimated from the image points found to be stationary, using a Kalman filter based approach. However, other methods, like for example [6] or [105] can be easily integrated.

In the current setup, the image points are tracked by a KLT tracker [162], which provides sub-pixel accuracy and tracks the image points robustly for a

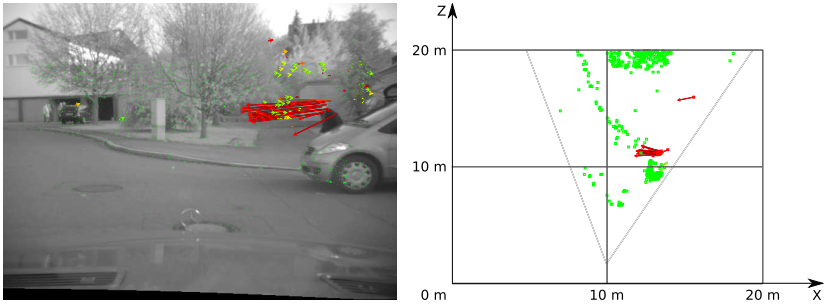


Figure 6.1: Estimation result of the 6D-Vision algorithm. The arrows point to the estimated 3D position in 0.5 s, projected back onto the image. The color encodes the absolute velocity: Static points are encoded green, points moving at a speed of 4.0 m/s or above are encoded red.

long sequence of images. It was optimized with respect to speed, allowing the complete motion-based object detection module to analyze up to 5000 points in real-time (25 fps). The stereo computation is performed by a hardware implementation of the semi-global matching algorithm [88]. However, any comparable optical flow and stereo algorithm can be used.

The estimation result of the 6D-Vision algorithm is shown in Figure 6.1. Here, the arrows point from the current 3D-position to the predicted 3D-position in 0.5 s. Looking at the bird's-eye view in the right image, the moving pedestrian is now easily distinguished from the standing vehicle.

Objects are identified as groups of contiguous coherent motion vectors. Since the 6D-Vision algorithm provides not only the state estimates, but also their uncertainty, the Mahalanobis distance is used as a similarity measure in the cluster analysis.

6.2.3 Fusion of Motion-based Object Detection (6D-Vision) and Pedestrian Recognition (PedRec)

For an accurate prediction of pedestrian movement, both positional and velocity information is important. Input from 6D-Vision and PedRec modules are fused

using a Kalman filter. The state **S** of the filter is modeled as

$$\mathbf{S} = [\ x\ y\ v_x\ v_y\ ]^T$$

with x/y being the longitudinal/lateral position of the pedestrian to the vehicle and v_x/v_y being its absolute longitudinal/lateral velocity in the world. The measurement vectors associated with the 6D-Vision and PedRec modules are

$$\mathbf{z}_{6d} = [\ x\ y\ v_x\ v_y\ ]^T, \quad \mathbf{z}_{ped} = [\ x\ y\ ]^T,$$

where x/y and v_x/v_y are various measurements of the state variables defined above (the mapping from state to measurements is thus trivial). Current measurements from both modules are integrated into the filter using successive update steps.

We assume a constant velocity pedestrian motion model (acceleration is modeled in the process noise covariance). The transition matrix F is given by

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & \mathsf{T} & 0 \\ 0 & 1 & 0 & \mathsf{T} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.1}$$

with T being the cycle time of the camera (40 ms).

Ego-motion of the vehicle is compensated in the prediction step of the Kalman Filter. Object translation with respect to the vehicle can be computed assuming a "bicycle" model [104] for the vehicle motion with constant steering angle and velocity between two measurement points. The required velocity and yaw rate data for the ego-motion-compensation is given by on-board sensor data and is accessible in the camera cycle time.

Measurement to track association is done using a global nearest neighbor (GNN) approach with prior rectangular gating on object positions. The Mahalanobis distance between predicted state and measurement is used for the data association. For pedestrian detections this means the position is used for measurement to track association, while for 6D-Vision detections the velocity is used additionally.

Track initialization and termination is handled depending on the number of associations to a track. New tracks are initialized using measurements that could not be assigned to an existing track. In order to suppress spurious detections, tracks start in the state *hidden*. A track enters the state *confirmed* after a certain

number n of measurements have been assigned to the track. Here we use $n = 2$, which means a detection from both modules at the same time directly results in a confirmed pedestrian track. Only tracks where a pedestrian detection has been assigned to are marked as valid pedestrian track. For all tracks a history of their state over time, including measurement to track associations is kept. Tracks are terminated after a user defined number of missed associations m.

Both modules operate independently at different cycle times. The 6D-Vision module operates in the fixed camera cycle time (25 fps). Processing time of the PedRec module varies depending on the scene complexity with a lower limit of 15 fps. Measurements have a common time-stamp defined by the frame-stamp of the image they have been generated on. In situations where measurements arrive out of sequence and can not be integrated in the common filter state, the track history is used to check measurements to track associations in the past. Possible assignments lead to an update of the association information. Although the filter state is not updated using the out of sequence measurements the updated association information effects the track management, allowing a track to enter the state *confirmed*. Additionally PedRec associations lead to a validated pedestrian track.

The initial state of the Kalman filter is derived from the first measurement. If a track is initialized by a pedestrian detection the velocities of the system state are set to zero. A track started by a 6D-Vision detection uses the measured velocities as initial value.

Finally, position, velocity and extent of the tracked pedestrians are passed to the situation analysis module.

6.3 Situation Analysis, Decision, Intervention, and Vehicle Control

Situation analysis and vehicle control are the components of a driver assistance system which generate a machine level understanding of the current situation (based on the previously described sensor information) and take appropriate actions. Figure 6.2 depicts the relationships between *trajectory generation*, *situation analysis*, *decision & intervention*, and *vehicle control*.

Situation analysis predicts how the current driving situation will evolve and automatically evaluates its criticality using measures as e.g. time-to-collision, time-to-steer, and time-to-brake. This criticality assessment serves as the basis for a decision module which triggers appropriate maneuvers for collision avoid-

ance and collision mitigation. Such maneuvers are realized by specialized vehicle controllers. Naturally, vehicle control and situation analysis are closely coupled, since both rely on accurate, realistic models of evasive maneuvers. These models are provided by a trajectory generation module. The following sections will briefly describe the aforementioned modules (see [98] for a detailed description).

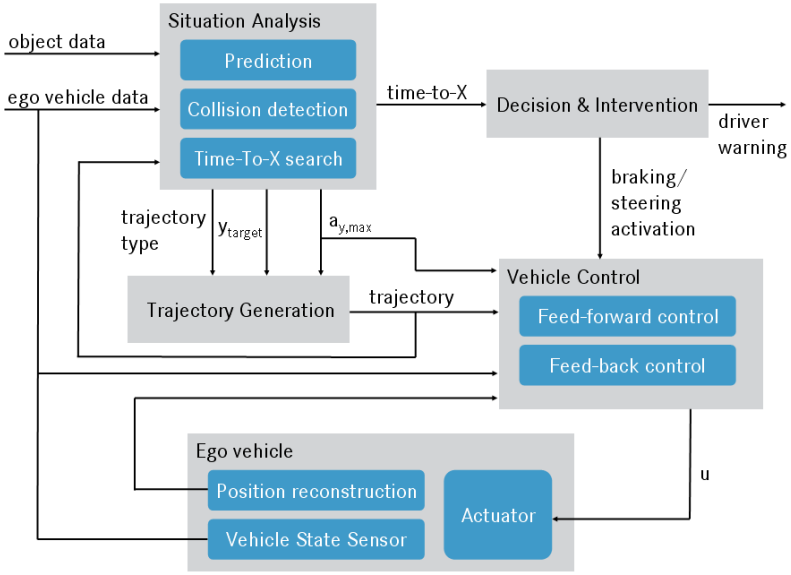


Figure 6.2: System structure of situation analysis and vehicle control.

6.3.1 Trajectory Generation

The objective of trajectory generation is twofold. First, trajectory generation has to provide accurate models of evasive steering maneuvers that fulfill several requirements: The generated trajectory for evasion should be as comfortable as possible, feasible (i.e. drivable by the ego vehicle), and should also lead to a safe transition with minimal side-slipping of the vehicle during the automatic evasive maneuver. Snatch of steering wheel can be dangerous and must therefore be avoided.

Second, trajectory generation should also provide the reference input variables for lateral control such as yaw angle, yaw rate, etc. Different trajectory types have been investigated and a sigmoidal blending function based on a polynomial approach as proposed in [66] is used to model the evasive maneuver path.

A polynomial model of seventh degree for the evasive path

$$y_{trj} = f(x) = \sum_{i=0}^{7} b_i \cdot x^i \,, \tag{6.2}$$

where y_{trj} is the desired lateral and x the longitudinal offset from the starting point of the evasion maneuver, allows to fulfill the requirements regarding comfort and feasibility. To meet these specifications, the determination of the polynomial coefficients b_i is based on several constraint equations which limit the maximum lateral acceleration $a_{y,max}$, the derivatives of the lateral offset and of the curvature, respectively. For the derivation of the polynomial coefficients b_i in Eq. (6.2) we refer to [98].

Based on the polynomial function and on the measured vehicle velocity v, the important input variables for lateral control (lateral offset y_{trj}, curvature c_{trj}, heading angle χ_{trj}) are determined at every sample time step.

6.3.2 Situation Analysis

A commonly employed approach for collision risk assessment involves criticality measures such as *Time-To-Brake* (TTB), *Time-To-Steer* (TTS), etc.. TTB, for example, denotes the remaining time span in which the driver can still avoid a collision by braking with maximum deceleration. Detailed descriptions of Time-To-X criticality measures and their application in driver assistance systems for collision avoidance and mitigation can be found in [85].

In this chapter, TTB and TTS are used to trigger automatic collision avoidance by either braking or steering maneuvers. The algorithm not only needs to find the latest steering maneuver which avoids a collision with the pedestrian in our driving path, but also has to ensure that the emergency maneuver does not result in a collision with any other detected object in the scene (e.g. cars, pedestrians; the integration of such free-space sensing component is left for future work, see Section 6.5). To fulfill these requirements we employ a numerical simulation method, which allows efficient, real time computation of Time-To-X criticality measures even for complex maneuvers. In addition, this numerical method can

verify if an evasive steering maneuver can be performed without collision.

As depicted in Figure 6.2, the numerical simulation methods consist of three main components: *prediction*, *collision detection*, and *Time-To-X search*. In the prediction step, a sequence of potential future ego and other object states

$$\left\{t_k, \mathbf{z}_{ego,k}, \mathbf{z}^1_{obj,k}, \ldots, \mathbf{z}^M_{obj,k}\right\}, \quad k = 1 \ldots K, \tag{6.3}$$

is computed, where t_k is the k-th time stamp of the prediction, K the prediction horizon, $\mathbf{z}_{ego,k}$ a vector describing the ego vehicle's pose and motion at time t_k, and $\mathbf{z}^1_{obj,k}, \ldots, \mathbf{z}^M_{obj,k}$ the pose and motion of all M objects provided by the sensor data fusion (Section 6.2.3). To obtain these predictions, we rely on appropriate motion models for all objects and the ego vehicle, thus making assumptions on their future behaviors.

Given the predicted states, we can identify potential collisions between the system vehicle and all objects in the scene by intersecting corresponding positions resulting from $\mathbf{z}_{ego,k}$ and $\mathbf{z}^1_{obj,k}, \ldots, \mathbf{z}^M_{obj,k}$, respectively. If a collision is detected, we start the search for the latest possible collision avoidance maneuver.

To accomplish this task, we have defined two emergency maneuvers representing braking with maximum deceleration of $-10\mathrm{m/s}^2$ and steering as modeled in Section 6.3.1, respectively. Each pairing $(t_k, \mathbf{z}_{ego,k})$ of Eq. (6.3) constitutes a potential starting point for an automatic emergency maneuver. Using a binary search algorithm, we can efficiently find the latest time steps at which braking or steering maneuvers have to be triggered that do not lead to a collision with any object in the scene. These time steps are discrete estimates of TTB and TTS.

6.3.3 Decision & Intervention

The "decision & intervention" module is the core of the assistance system, since it associates the function with the driver's behavior. Due to the high injury risk of a pedestrian in an accident, collision avoidance is the primary objective of the function. In order to identify the best way to support the driver, it is necessary to know the driver's current driving intention. The driver monitoring algorithm is using signals from the vehicle, e.g. accelerator and brake pedal position, speed, lateral and longitudinal acceleration, steering angle and steering rate to determine the current driving maneuver of the driver. If the driver is not reacting appropriately to the dangerous situation, an optical and acoustic warning will be given, so he can avoid the collision himself. In the case a function intervention is necessary to avoid the collision, full braking takes priority over the evasive

maneuver. The full braking will be triggered when $TTB = 0$ and the driver is neither doing an accelerating nor an evasive maneuver. If the collision cannot be prevented with full braking any more ($TTB < 0$), the evasive steering maneuver will be activated at $TTS = 0$, provided the situation analysis has computed that this can be executed without collision; the evasive maneuver using the vehicle control to compute the necessary steering torque. The function ramps down the steering torque, when the evasive maneuver has finished. Afterwards the function is available immediately, when needed. Automatic evasion results in a fixed lateral offset of the vehicle in the range of 80–100 cm. In case collision free evasive steering would not be possible because of, say, detected oncoming traffic, the decision would be to brake (collision mitigation).

The design of the prototype function allows the driver to overrule the steering intervention at any time. If the driver holds the steering wheel, he will weaken or suppress the steering of the system. A distinct activity of the accelerator or brake pedal cancels the evasive maneuver immediately. Similar exit conditions exist for the full braking intervention.

In order to minimize dynamic misalignment of the passengers during the system intervention additional protective measures are triggered. The function controls the electromotive reversible seat-belt pretensioners and the side-gated air cushions of the seating and backrests will be inflated. When the system has finished the intervention, the tension of the reversible seat belt pretensioners is released automatically and the air cushions of the seats are vented to the previous position.

6.3.4 Vehicle Control

Vehicle control consists of two parts: longitudinal control for automatic braking and lateral control for evasion. Automatic braking is triggered when TTB=0s (i.e. at the latest point in time when the ego vehicle can avoid the collision by full emergency braking), thus the longitudinal vehicle controller will set a maximum deceleration of $-10\ m/s^2$. Steering maneuvers (or lateral control) for automatic collision avoidance entail highly dynamic lateral movements of the ego vehicle (here, lateral motion refers to motion perpendicular to our driving lane). The dynamics of such maneuvers with high lateral acceleration are nonlinear. In general, the lateral offset y_{target} as defined in Section 6.3.1 may vary from only a few centimeters to a full lane change depending on the size of the obstacle, its velocity, and the free space available for the evasive maneuver. Here, however, for pedestrian evasion a fixed lateral offset is used.

Figure 6.3: (top) Test track set-up with the pedestrian dummy sliding along a traverse in front of the vehicle (bottom) Close-up of pedestrian dummy.

Collision avoidance by steering requires precise lateral control of the ego vehicle. The controller permanently compares the reference position along the evasive maneuver trajectory as specified in Eq. (6.2) to the actual vehicle position and thus requires accurate and reliable knowledge of the ego vehicle's pose. The position of the vehicle is reconstructed from odometers and inertial sensors readily available in today's vehicles. To account for the nonlinear lateral dynamics of the evasive maneuver, a control strategy combining feed forward and feed back control is used, see [67, 98] for a detailed description.

6.4 Experiments

6.4.1 Set-up

Our vehicle prototype is a Mercedes-Benz S-Class, with a stereo camera mounted behind the windshield. Figure 6.4 depicts the main components of the prototype. The stereo base line is $30cm$ and each camera has a resolution of 640×480 pixels and a focal length of $12mm$. Two computers are mounted in the trunk; a $4GHz$ Quad Core Pentium with the image processing and fusion algorithms and a $2.2GHz$ dual core Pentium with the function specific algorithms. They are connected to the CAN network of the vehicle, which provides the required vehicle signals, such as speed and steering angle.

The vehicle prototype works with a conventional power steering, as it is used

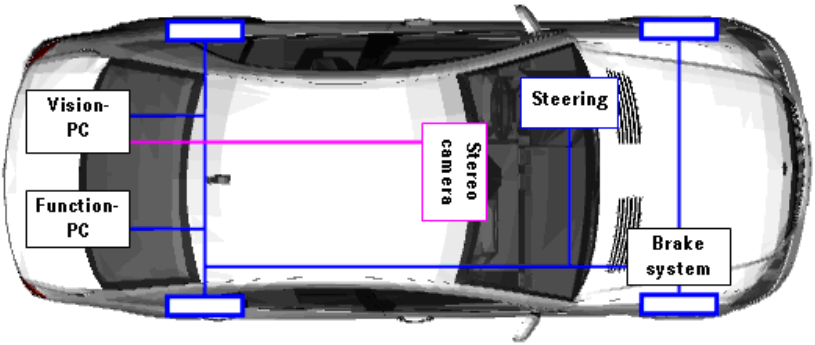


Figure 6.4: Main components of the prototype system.

in the production vehicles. In addition, the steering possesses an electric steering torque actuator. It allows inducing an additional steering torque up to $5Nm$ at the steering wheel to realize the automatic evasive maneuver. Braking and driver warning (display instrument panel, loudspeakers) were implemented using the Mercedes-Benz series control units. In addition, seat air cushions were inflated and seat belts were pre-tensioned in the event of a near-crash.

In order to test the prototype functionality, a traverse construction was installed on a proving ground, under which a pedestrian dummy, hung by a set of wires, can be moved across the road. See Figure 6.3. An electronic device allowed reproducible movement of the pedestrian dummy. The synchronisation of the pedestrian dummy and the vehicle was achieved by a light barrier.

6.4.2 Test of Video Sensing Component

We first discuss the evaluation of the video sensing component only. A total of 22 scenarios were staged, covering real world situations of varying complexity, see Figure 6.11. The scenarios involve different numbers of pedestrians, geometrical lay-outs, walking speeds and visibility conditions. For safety reasons, lateral pedestrian movement resulting in near-collisions was solely performed with the dummy. Furthermore, vehicle speed was reduced to $30\ km/h$ in those scenarios (S11, S13, S17, S21, S22) where a real pedestrian was nearing the vehicle side up to $1.5\ m$.

3D ground truth positions of pedestrians with respect to the vehicle were obtained by manual labeling the corresponding bounding boxes in the camera images and by triangulation. Partially occluded pedestrians were labeled by a bounding box containing the visible body parts. We defined a sensor coverage range of $7 - 27m$ in front and up to $6m$ to each side of the vehicle medial axis, which was covered by both the 6D-Vision and PedRec modules. In this area all pedestrians were 'required', i.e. were needed to be detected by the system (even if only partially visible). Outside this area, pedestrians were 'optional', we did not credit or penalize for system detections. In all, this resulted in 48 required pedestrian trajectories and 1700 pedestrian single-frame instances. We now consider four performance metrics in turn: detection performance, position- and speed-accuracy and time-to-detect.

Detection performance is related to the number of matches between ground truth and system-detected object locations. There are two aspects: sensitivity and precision. Sensitivity relates to the percentage of true solutions that were found by the system (i.e. detection percentage), whereas precision relates to the percentage of system solutions that were correct. A sensitivity and precision of 100% is ideal: the system finds all real solutions and produces no false positives. Performance is evaluated using the frame- and trajectory-level criteria as described in Chapter 4.4. We distinguish three types of trajectories: "class-A+", "class-A", "class-B", which have 75%, 50% and 1 frame entries matched.

In comparing system output with ground truth, we need to specify the localization tolerance, i.e. the maximum positional deviation that still allows us to count the system detection as a match. This localization tolerance is the sum of an application-specific component (how precise does the object localization have to be for the application) and a component related to measurement error (how exact can we determine true object location). We define object localization tolerance as percentage of distance, for longitudinal and lateral direction (X and Y), with respect to the vehicle. For our evaluation of the video sensing component, we took $X = 15\%$ and $Y = 4\%$, which means that, for example at $10m$ distance, we tolerate a localization error (including ground truth measurement error) of ± 1.5 and ± 0.4 m in the position of the pedestrian, longitudinal and lateral to the vehicle driving direction, respectively.

For this application we allow many-to-many correspondences. A ground truth location is considered matched if there is at least one system detection matching it. In practice, this means that in the case a group of pedestrians walking sufficiently close together in front of the vehicle, the system would not necessarily

have to detect all of them in isolation, it suffices if each true pedestrian is within the localization tolerance of a detected pedestrian.

Table 6.1 summarizes the pedestrian detection performance. First two columns relate to 6D-Vision and (single-frame) PedRec output, which form the components of the fused system, shown in the last column. The third column represents the baseline case (termed 'PedRecTrack'): PedRec in combination with the previously described Kalman filter, without integrating the 6D-Vision detections. Two consecutive detections are required for a track to be initialized. After three missed detections tracks are closed.

From Table 6.1 one observes an improved performance of the fusion system (fourth column) vs. the baseline PedRec tracking system (third column). This is mainly due by the successful detections of 6D-Vision of the partially occluded pedestrians (i.e. upper body visible above parked car), which are not captured by the current PedRec, see Figure 6.7. By relying on motion, 6D-Vision cannot always be of help, however. Pedestrian standing or walking slowly (especially in longitudinal direction) are not well detected, which accounts for the somewhat lower detection rate (first column). As 6D-Vision is a generic moving object detection system, false pedestrian positives do not apply (see N/A entries).

Table 6.2 summarizes the obtained **positional accuracy** for the required pedestrians which were detected (i.e. within before-mentioned localization tolerance). Lateral localization is quite accurate for all the 6D-Vision and PedRec components and fusion. Not surprisingly, longitudinal accuracy is lower for all variants. Here, PedRec has an edge, partly because its measurements are restricted to fully visible pedestrians.

For a reliable automatic vehicle maneuver, **speed accuracy** is important in addition to position accuracy. Figure 6.5 illustrates the estimated speed of the various configurations on scenario S01, from the time the pedestrian is partially visible coming behind the parked car. The speed of the pedestrian dummy (2 m/s) is exactly known from the test setup.

Although the dummy is detected early by PedRecTrack system, the initial estimated position is not exact enough to allow a correct two-point filter initialization. This is because of small errors in depth estimation, caused by including disparity values belonging to the parked car that is occluding the dummy. Therefore, PedRecTrack is initialized with a speed of zero (same applies for the fused system). As Figure 6.5 shows, it takes about one second for the PedRecTrack system to converge to the correct speed of 2 m/s. The 6D-Vision module, however, tracks the correct feature points on the moving target, and is able to converge fast to

	6D-Vision	PedRec (single-frame)	**PedRecTrack**	**Fusion**
Sensitivity (frame level)	66.2% (70.4%)	75%	76.2%	88.9%
Precision (frame level)	N/A	N/A	100%	100%
# False detected objects (frame level)	N/A	N/A	0	0
Sensitivity (class-A+ trajectory)	56.3% (62.8%)	54.2%	60.4%	89.6%
Sensitivity (class-A trajectory)	75.0% (83.7%)	81.25%	81.3%	95.8%
Sensitivity (class-B trajectory)	91.7% (100%)	100%	100%	100%
# False Trajectories	N/A	N/A	0	0
Precision (class-A+ trajectory)	N/A	N/A	100%	100%
Precision (class-A trajectory)	N/A	N/A	100%	100%
Precision (class-B trajectory)	N/A	N/A	100%	100%

Table 6.1: Pedestrian detection performance of baseline system (PedRecTrack, third column) and of proposed fusion approach (Fusion, last column) on full dataset, 22 scenarios. Between brackets, results on data subset containing moving pedestrians only.

	6D-Vision	PedRec (not tracked)	Fusion
lateral	0.06 (0.06)	0.05 (0.05)	0.06 (0.05)
longitudinal	0.40 (0.16)	0.17 (0.17)	0.32 (0.31)

Table 6.2: Localization accuracy over defined sensor coverage area (longitudinal 7-27 m, lateral up to 6 m): root mean squared error and (between brackets) standard deviation in meters

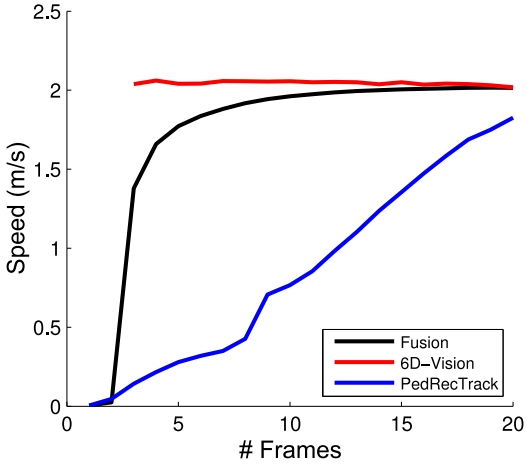


Figure 6.5: Estimated pedestrian speed using the baseline PedRecTrack, 6D-Vision and the proposed fusion system. The ground truth speed is $2\ m/s$.

the correct speed. For the fused system, integrating the speed information from the 6D-Vision module helps the filter to converge faster to the correct speed than the baseline PedRecTrack system.

Finally, we analyze the performance regarding **time-to-detect**, here defined as the number of frames it requires to detect a ground-truth trajectory, from first instance of full pedestrian visibility (a system trajectory that is started beyond the required sensor coverage range will result in a "time-to-detect" of one frame). Trajectories that can not be detected by all configurations are excluded. A total of 42 trajectories remain, the results are shown in Figure 6.6. In analyzing the results of the individual sequences, we observe that lateral moving pedestrians (2 m/s), for which the lower body part is occluded by the parked cars, are detected early by the 6D-Vision module, see Figure 6.7. Table 6.3 summarizes the results for this scenario subset. On the other hand, longitudinal moving pedestrians close to parked cars are more difficult to segment but pose no problem for the PedRec module. By fusing detections of both modules, the time to detect a pedestrian is reduced on average.

6.4.3 Test of Integrated System

We tested the integrated system (sensing, situation analysis, decision making and vehicle control) on two scenarios S01 and S02. In both scenarios, the vehicle drives close to $50\ km/h$ and the pedestrian dummy moves from the right side onto the vehicle's lane with a lateral speed of $2\ m/s$. In scenario S01 the pedestrian dummy is only partially occluded by a parking passenger vehicle. In scenario S02, the dummy appears behind a parking van and thus can only be detected by our system significantly later than in scenario S01. The desired vehicle action is to brake if still possible, otherwise to evade. See Figure 6.12 and 6.13 for snapshots of the integrated system choosing the correct vehicle action in scenarios S01 and S02.

PedRecTrack (baseline)	6D-Vision	Fusion
2.4 (2.8)	1.4 (1.3)	1.5 (1.6)

Table 6.3: Number of frames until the pedestrian dummy is detected, from moment of full visibility: mean and standard deviation (in brackets). Data computed over 10 trajectories, with initial partial occlusion.

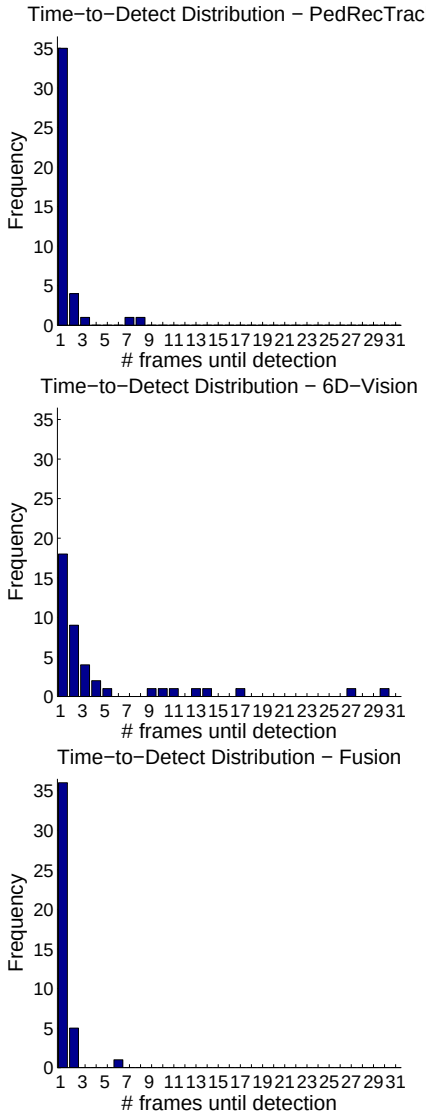


Figure 6.6: Distribution of the number of frames until a pedestrian is detected, from the first frame of full visibility, for PedRecTrack, 6D-Vision and Fusion, respectively. Distribution over occluded and non-occluded trajectories that were detected (42 in total).

Figure 6.7: An illustration of the complementary nature of PedRec with 6D-Vision. The grayscale image on the left displays the raw pedestrian detections (red box), 6D-Vision detections (small yellow box) and fusion results (blue box). The static fully visible pedestrian is detected by PedRec, the strongly occluded moving pedestrian is detected by 6D-Vision. Both are detected by the fusion approach. To the right of the grayscale image, three top views associated with Fusion, 6D-Vision and PedRec. Numbers denote distance to vehicle.

We experimentally determined the last possible brake time for the vehicle to come to a complete stop to correspond to a pedestrian distance of $20\ m$ (taking into account various device latencies). In scenario S01, the setup is such that the pedestrian is first fully visible at about 24 m distance ($3.8\ m$ lateral) to the vehicle. This means that the system has only about seven frames (corresponding to $4.1\ m$ driven) to determine pedestrian position and speed, perform situation analysis and make the correct decision to initiate braking.

In scenario S02, the pedestrian dummy was initially occluded by a parking van aside of the road. Thus, the pedestrian dummy was only detected at a distance of less than $20\ m$ and collision avoidance by braking was no longer possible. In the following example, the ego vehicle was driving with a constant speed of $45\ km/h$ and the pedestrian was first detected at a distance of $15.9\ m$ and a lateral offset of $-3.4\ m$. Figure 6.8 depicts the time-to-collision (TTC), time-to-brake (TTB), and time-to-steer (TTS) values provided by the situation analysis module of Section 6.3. As the pedestrian dummy becomes visible very late in this scenario, automatic braking can no longer avoid a collision and TTB $=-\infty$ throughout this measurement. As soon as TTS falls below a predefined total reaction time of the system ($200\ ms$ in our prototype system), an automatic steering maneuver is triggered and the TTX computation is stopped.

Figure 6.9 shows the commanded trajectory y_{trj} and the reconstructed lateral position y of the vehicle after the lateral controller has been started. The actual lateral position y was reconstructed using speed measurements and lateral acceleration measurements from odometry and inertial senors in our experimental vehicle. In this experiment, a fixed target lateral offset of $1\ m$ has been chosen. As can be seen from the measurement data, the time lag between actual and commanded trajectory position is approx. $200\ ms$. This time lag corresponds to the total reaction time of our system and is induced by our vehicle's dynamics, data processing time and the phase lag of the steering actuator.

Figure 6.10 show the measured lateral acceleration and vehicle speed during the automatic evasive maneuver. The maximum measured lateral acceleration is less than 10% higher than predefined limit of $a_{y,max} = 5\ m/s^2$. This performance is acceptable in our application. The absolute speed of the vehicle is reduced by $3\ km/h$ during the maneuver. We tested the integrated system by means of 20 runs on both scenarios S01 and S02. In all 40 runs, the prototype vehicle selected the correct action in time, not hitting the pedestrian dummy once. In the braking scenarios, the vehicle stopped approximately $30 - 150\ cm$ ahead of the dummy.

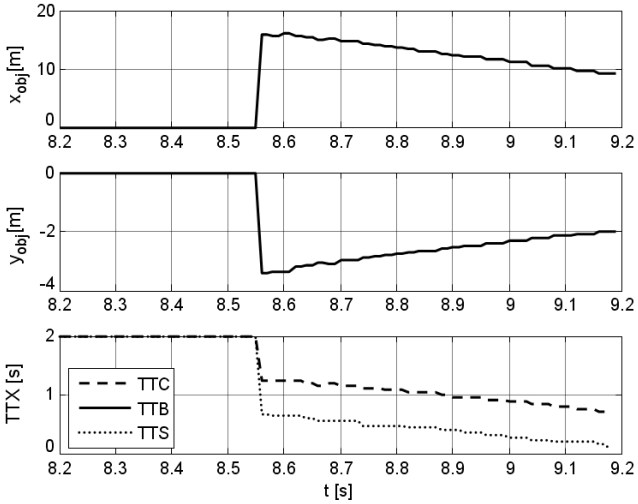


Figure 6.8: Position of detected pedestrian (top, middle) and corresponding time-to-x values (bottom). Note that time-to-brake (TTB) is $-\infty$ in this sequence and thus not visible. An evasion maneuver is triggered at $t = 9.18\ s$.

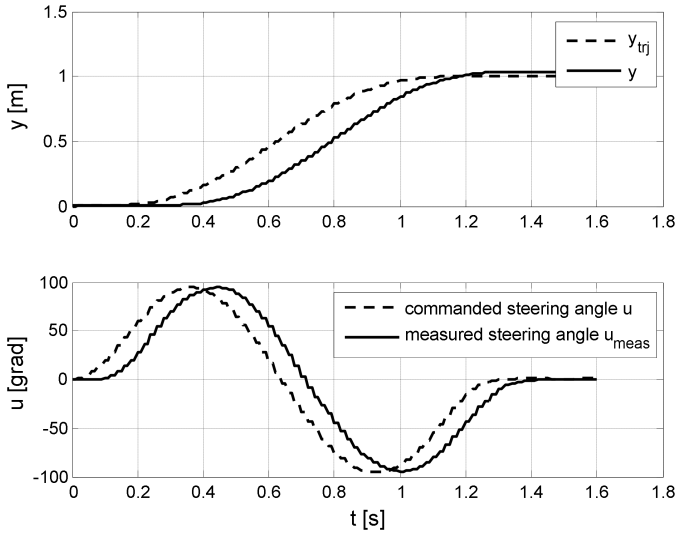


Figure 6.9: Commanded trajectory and measurement results after evasion has been triggered. Top: Lateral position of the vehicle. Bottom: Steering wheel angle. The upper plot shows a total reaction time of the vehicle of approx. 200 ms; this includes a steering actuator phase lag of about 70 ms as depicted in the lower diagram.

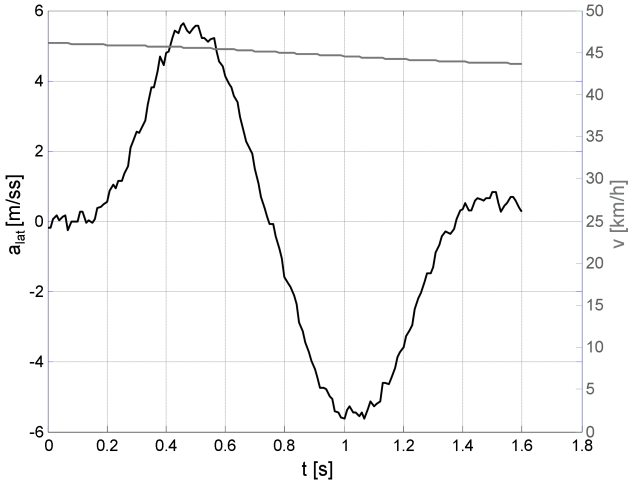


Figure 6.10: Measured lateral acceleration and vehicle speed.

6.5 Discussion

The previous section demonstrated a remarkably reliable vehicle system on the test track, that can detect pedestrians and make the right decision to brake or to evade, in a split of a second. There are a number of technical challenges associated with extending the flawless performance of the system on the test track to the real urban traffic environment,

Regarding the sensing component, note that for our experimental setting on the test track, it was easy to discard 6D-Vision detections on moving vehicles, based on speed considerations. Therefore, the remaining 6D-Vision detections, associated with realistic pedestrian speeds, were treated similarly to the PedRec detections in the fusion approach of Section 6.2.3. The decision whether to output a track was solely based on the number of detections, irrespective of their source. In a real traffic environment, there will be many other moving objects, which could be pedestrian-like. Future work will develop a probabilistic approach, which maps 6D-Vision and PedRec detections onto posteriors for pedestrians, taking into account bounding box sizes, locations, speeds and classifier decision values. The decision whether to initiate a track would be made by analyzing the

cumulative probability of observing a pedestrian.

It is paramount to avoid false system activations (i.e. unnecessary braking or evasion maneuver). For that, all system modules and in particular the sensing component (6D-Vision, PedRec) will need to be enhanced (e.g. better position and velocity estimation, recognition performance, recognizing pedestrians under partial occlusion [46]). Sensor fusion (e.g. with radar, laser scanners) can provide an additional level of robustness.

The sensing component might be extended to other traffic participants, such as bicyclists and cars, to match the capabilities of the current situation analysis component. The current evasion maneuver results in a lateral offset of 80-100 cm of the vehicle. Larger offsets are conceivable. This places demands that the sensing component also performs a free space analysis [8], to verify that the automatic evasion maneuver can indeed be safely performed. Being able to detect elements of the traffic infrastructure (e.g. lane markings, traffic lights) will furthermore enable more sophisticated situation analysis.

As a final note, we emphasize that the presented system is meant for emergency situations, in which the driver will likely not be in a position to still act properly. Vehicle control (and responsibility) rests, however, fully with the driver; at each time instant the driver can overrule the system, by either accelerating or maintaining a grip on the steering wheel.

6.6 Conclusion

This chapter presented a novel active pedestrian safety system, which combines sensing, situation analysis, decision making and vehicle control. The vision sensing component fuses two complementary approaches: generic motion-based object detection (6D-Vision) and pedestrian recognition (PedRec). Situation analysis was based on numerical simulation, which allowed to incorporate more complex, non-circular vehicle paths based on a polynomial model. Decision making involved the continuous monitoring of time-to-collision, time-to-brake and time-to-steer measures, and initiating a specialized control loop in case of an evasion maneuver.

Extensive pre-crash experiments with the system on the test track have been performed. We demonstrated that the benefit of adding 6D-Vision to a baseline PedRec(Track) system is that lateral moving pedestrians (2 m/s) can be detected earlier when partially occluded by a parked car, and furthermore, velocity estimation is more accurate. On two scenarios, requiring a split-second decision

between no action, automatic braking and automatic evasion, the system made in all runs (over 40) the correct decision [80, 92, 96, 183]. Despite the strong performance on the test track, additional challenges remain before this system can reliably be deployed in real urban traffic.

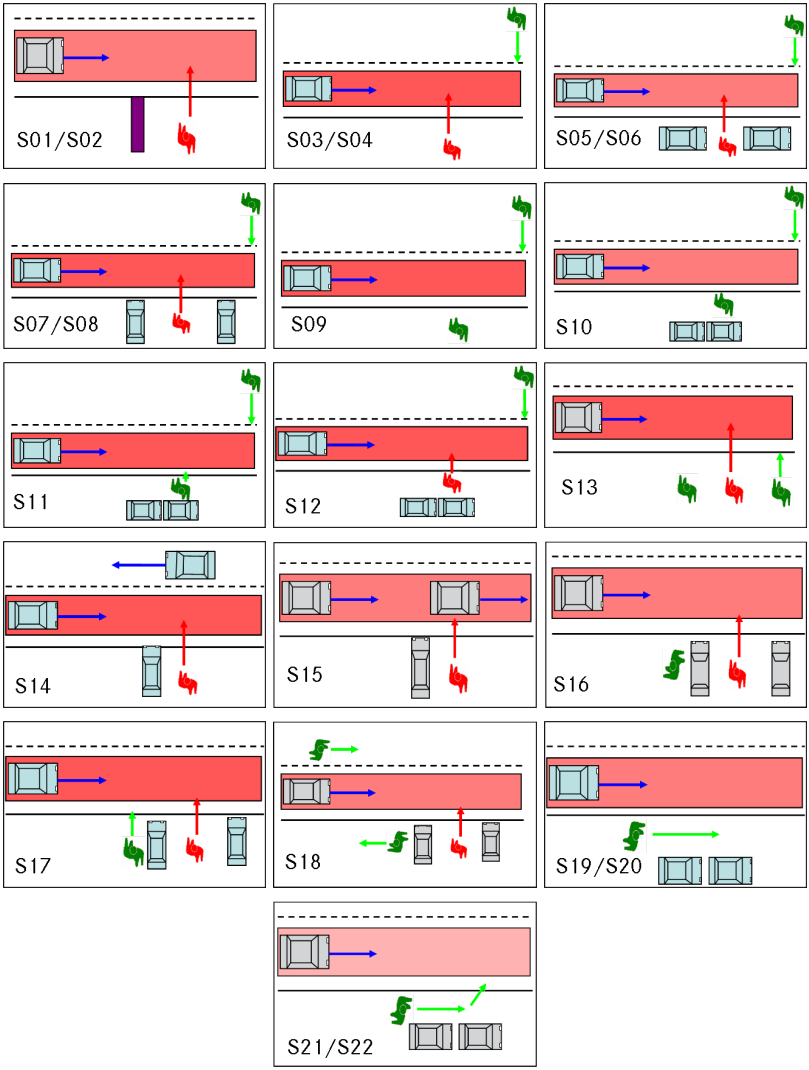


Figure 6.11: Illustration of the 22 different scenarios, performed with real pedestrians (green) and a pedestrian dummy (red). Scenario pairs associated with a single diagram were performed with different dummy/pedestrian speeds, i.e. either $1\ m/s$ or $2\ m/s$.

Figure 6.12: Braking scenario S01

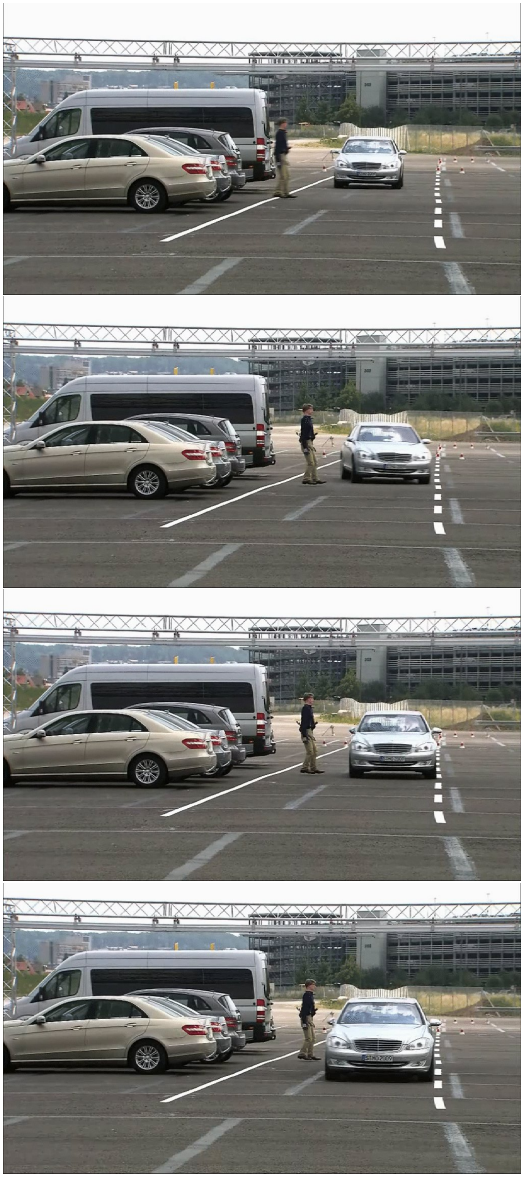


Figure 6.13: Evasion scenario S02

Chapter 7

Path Prediction and Action Classification

7.1 Introduction

Predicting the path of a pedestrian is important in several application contexts, such as robot control in human-inhabited environments and driver assistance systems for improved traffic safety. In this chapter, we consider the intelligent vehicles context, where strong gains have been in improving vision-based pedestrian recognition performance, see the previous chapters. However, the initiation of an emergency vehicle maneuver requires a precise estimation of the current and future position of the pedestrian with respect to the moving vehicle (see Chapter 6.3.1). A deviation of, say, $25\ cm$ in the estimated lateral position of the pedestrian can make all the difference between a "correct" and an "incorrect" maneuver initiation.

One major challenge is the highly dynamic behavior of pedestrians, which can change their walking direction in an instance, or start/stop walking abruptly. As a consequence, prediction horizons for active pedestrian systems are typical short; even so, small performance improvements can produce tangible benefits. For example, accident analysis [118] shows that being able to initiate emergency braking $0.16\ s$ (4 frames @ $25\ Hz$) earlier, at a Time-to-Collision of $0.66\ s$, reduces the chance of incurring injury requiring hospital stay from 50% to 35%, given an initial vehicle speed of $50\ km/h$.

This chapter focuses on the task of predicting the position of pedestrians walking towards the road curbside, when viewed from an approaching vehicle. A secondary question is whether the pedestrian will cross or stop. See Figure 7.1. This setting is inspired by an earlier human factors study by Schmidt and Färber [150], which had several test participants watch videos of pedestrians walking towards the curbside and decide whether the pedestrians would stop or cross, at various time instants. Their study varied the amount of visual information provided to the test participants and examined its effect on their classification performance.

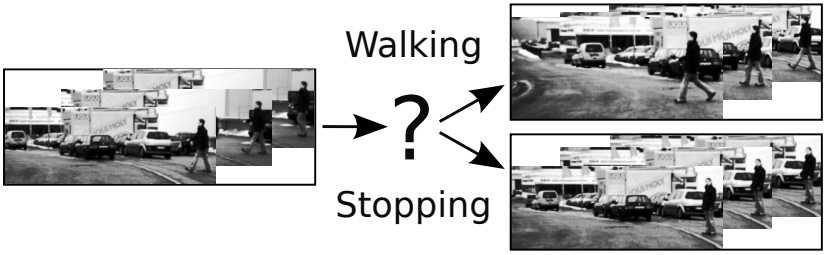


Figure 7.1: Pedestrian path prediction and action classification: Where exactly will the pedestrian be in the immediate future? Will the pedestrian cross?

In the baseline case, the pedestrian was fully visible, whereas in other cases, parts of the pedestrian body were masked out. Masking the complete pedestrian, and leaving only positional information (bounding box), turned out to decrease human accuracy markedly, showing the importance of augmented visual features for this prediction task.

We address the following questions in this chapter:

- at the short prediction horizons typical of the traffic safety context: can non-linear models outperform linear models, or alternatively, can higher-order Markov models outperform their first-order counterparts?
- do augmented visual features (optical flow) improve path prediction and action classification over the use of positional information only?
- how does measurement error (e.g. pedestrian localization error, vehicle ego-motion estimation error) affect the results? Can the more complex models still maintain an edge over the simpler ones?

In order to provide answers for the above questions, we consider a representative set of four approaches in this study. In the category of non-linear, first-order models with augmented visual features, we propose a novel pedestrian path prediction approach, based on Gaussian Process Dynamical Models (GPDM) [174] and dense optical flow features, see Section 7.2.1. An appealing aspect of this approach is that a low-dimensional, latent representation is learned from the data, which takes into account the process dynamics. In the category of non-linear, higher-order models with augmented visual features, we propose a novel

Probabilistic Hierarchical Trajectory Matching (PHTM) approach, based on a low-dimensional motion representation, see Section 7.2.2. Finally, in the category of first-order Markov models using positional information only, and mostly as a baseline, we consider the popular Kalman Filter (KF, linear model) and its extension Interacting Multiple Model Kalman Filter (IMM-KF, mixture of linear models) [10], see Section 7.2.3.

Experimental results on real traffic data are given in Section 7.3, with pedestrian image location obtained either from ground truth (optionally corrupted with noise) or obtained by a state-of-the-art pedestrian detection system, see Chapter 5. Several experimental cases are distinguished (pedestrian stopping vs. walking, ego-vehicle standing vs. moving). A discussion of the results, in terms of prediction performance and computational cost is given in Section 7.4. The chapter concludes in Section 7.5.

7.2 General Framework

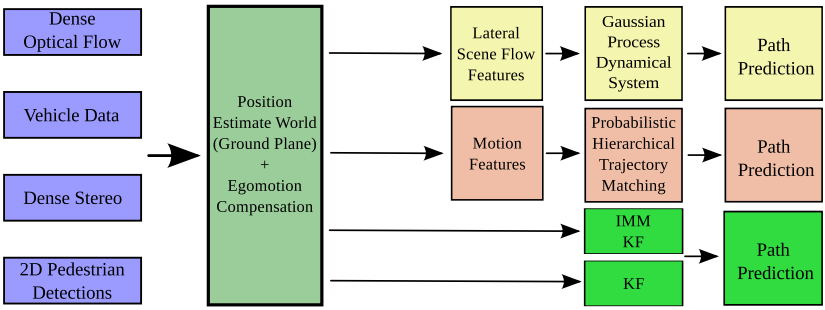


Figure 7.2: Overview of considered approaches for pedestrian path prediction.

We compare four different approaches for pedestrian path prediction, involving Gaussian Process Dynamical Models, Probabilistic Hierarchical Trajectory Matching, Kalman Filters and IMM Kalman Filters. See Figure 7.2 for an overview.

To allow meaningful comparisons among the systems, several pre-processing components are set equal. Bounding boxes containing pedestrians are supplied from the same detector module. Dense disparity is computed using the Semi-Global Matching stereo algorithm [89]. Pedestrian positions on the ground plane are obtained by considering the midpoint of the bounding box and the disparity

computed over the part of the bounding box that corresponds to the upper body (assuming typical human proportions). The latter involves clustering disparity values using mean-shift [30] and selecting the cluster with the largest weight; the median of the corresponding disparity values provides the desired pedestrian distance.

Vehicle ego motion is compensated by rotation and translation of pedestrian positions to a global reference point using a single track vehicle model [146] and velocity and yaw-rate measurements from on-board sensor data. The two approaches that use augmented visual features (GPDM, PHTM) compute dense optical flow [177] over the bounding boxes provided by the pedestrian detector; this flow is subsequently ego-motion compensated.

7.2.1 Gaussian Process Dynamical Model System

The first approach uses scene flow features describing the lateral movement of the pedestrian derived from the dense optical flow field and measured pedestrian distance in the world. Feature dimensionality is reduced by means of GPDM [174] with a dynamic model in the latent space. To overcome the absence of a direct mapping from feature space to latent space, the dynamic model is combined with a particle filter. GPDM models that capture the walking and stopping movement of a pedestrian are trained separately. The learned dynamical models provide optical flow fields at future time instants; future lateral positions can be derived by integration. Longitudinal position is estimated independently by means of a separate Kalman Filter for each action class (walking vs. stopping). Weighting lateral and longitudinal predictions using the probability of each action model results in future pedestrian positions.

Feature Extraction

Given the lateral component from dense optical flow and a pedestrian distance derived from dense stereo the lateral velocity of a pedestrian in the world is computed.

With the pedestrian distance (as disparity $disp$), the horizontal component of the optical flow field (V_u), the camera base width (b) and camera cycle time Δt the lateral speed v_X (m/s) of each pixel is computed using:

$$v_X = \frac{V_u \cdot b}{disp \cdot \Delta t} \tag{7.1}$$

To obtain only flow values located on the pedestrian body a mask image is generated from the thresholded disparity image and velocity values corresponding to background are set to zero. Applying this distance mask also adds rough pedestrian contour information to the feature. Figure 7.3 describes the feature extraction steps.

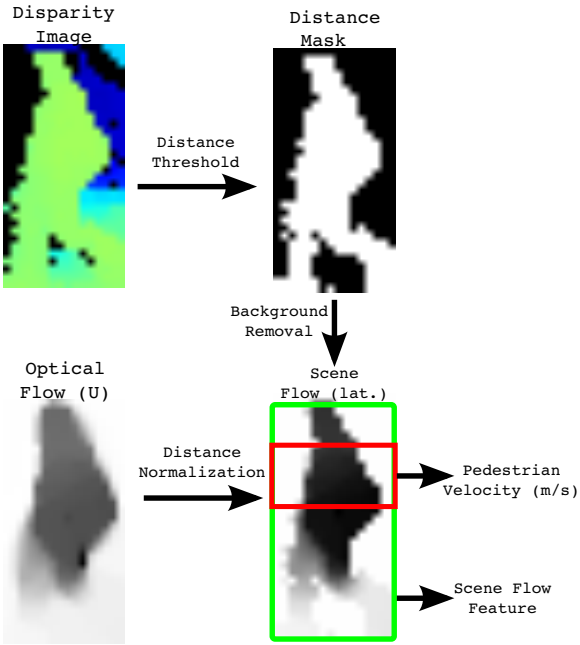


Figure 7.3: Feature extraction using dense optical flow and roughly estimated pedestrian contour from dense stereo.

For further use as a feature this scene flow image is rescaled to 32×16 pixel and concatenated to a feature vector $\mathbf{y_t} \in \mathbb{R}^D$ with $D = 512$. From the scene flow image (*SFlowX*) the lateral velocity of the pedestrian can directly be extracted using the median of velocity values located in the area of the pedestrian upper body (Figure 7.3 red box).

Dynamical Model

We are interested in a low dimensional representation $\mathbf{x}_t \in \mathbb{R}^d$ of features $\mathbf{y}_t \in \mathbb{R}^D$ from a pedestrian sequence with $d < D$. This dimensionality reduction is realized using the Gaussian Process Dynamical Model (GPDM) [167, 173, 174] which allows modeling the dynamics of the features over time t in the low dimensional space. For data in latent space $\mathbf{x}_t$ the relation to the input $\mathbf{y_t}$ can be described using:

$$\mathbf{y_t} = g(\mathbf{x_t}; \mathbf{B}) + \mathbf{n}_{y,t} \tag{7.2}$$

with zero-mean Gaussian noise $\mathbf{n}_{y,t}$ and mapping function g with parameters $\mathbf{B} = [b_1, b_2, \ldots]$. Assuming a first-order Markov model the dynamics of the data in the latent space $\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_N$ can be described using:

$$\mathbf{x_t} = f(\mathbf{x}_{t-1}; \mathbf{A}) + \mathbf{n}_{x,t} \tag{7.3}$$

with zero-mean Gaussian noise $\mathbf{n}_{x,t}$ and mapping function f with parameters $\mathbf{A} = [a_1, a_2 \ldots]$.

In a Gaussian process framework the parameters and basis functions of f and g are marginalized out and the positions of the latent coordinates are optimized.

Latent Mapping In the GPDM framework the conditional density for the data $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T$ given latent positions $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ is described using:

$$p(\mathbf{Y} \mid \mathbf{X}, \overline{\beta}, \mathbf{W}) = \frac{|\mathbf{W}|^N}{\sqrt{(2\pi)^{ND}|\mathbf{K}_Y|^D}} \exp\left(-\frac{1}{2} tr(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^T)\right) \tag{7.4}$$

with kernel matrix $\mathbf{K}_Y$ and kernel hyper parameters $\overline{\beta} = \{\beta_1, \beta_2, ...\}$ and $\mathbf{W}$. To equally weight all the feature dimensions the scale parameter is set to $\mathbf{W} = \mathbf{I}$ and is omitted in the following equations. Entries in the kernel matrix are defined using a kernel function $(\mathbf{K}_Y)_{i,j} = k_Y(\mathbf{x}_i, \mathbf{x}_j)$. For our data, we use a radial basis function (RBF) kernel with an additional noise term i.e.,

$$k_Y(\mathbf{x}_i, \mathbf{x}_j) = \beta_1 \exp\left(-\frac{\beta_2}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) + \frac{\delta_{\mathbf{x}_i,\mathbf{x}_j}}{\beta_3} \tag{7.5}$$

.

Dynamic Mapping The dynamics of the time-series data is incorporated using:

$$p(\mathbf{X} \mid \overline{\alpha}) = \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(N-1)d}|\mathbf{K}_X|^d}} exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_X^{-1}\mathbf{X}_{2:N}\mathbf{X}_{2:N}^T)\right) \tag{7.6}$$

with $\mathbf{X}_{2:N} = [\mathbf{x}_2, \ldots, \mathbf{x}_N]^T$, the kernel matrix $\mathbf{K}_X$ constructed from $\mathbf{X}_{1:N-1} = [\mathbf{x}_1, \ldots, \mathbf{x}_{N-1}]^T$ with dimensionality $(N-1) \times (N-1)$ and entries $(\mathbf{K}_X)_{i,j} = k_X(\mathbf{x}_i, \mathbf{x}_j)$. A combination of a RBF and linear kernel with an additional noise is used for the dynamics

$$k_X(\mathbf{x}_i, \mathbf{x}_j) = \alpha_1 \exp\left(-\frac{\alpha_2}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) + \alpha_3 \mathbf{x}_i^T \mathbf{x}_j + {\alpha_4}^{-1}\delta_{\mathbf{x}_i,\mathbf{x}_j} \tag{7.7}$$

with kernel hyper parameters $\overline{\alpha} = \{\alpha_1, \alpha_2, \ldots\}$.

Learning the GPDMs Combining the latent and dynamics mapping defines the model:

$$p(\mathbf{X}, \mathbf{Y}, \overline{\alpha}, \overline{\beta}) = \; p(\mathbf{Y} \mid \mathbf{X}, \overline{\beta}) p(\mathbf{X} \mid \overline{\alpha}) p(\overline{\alpha}) p(\overline{\beta}) \tag{7.8}$$

Learning a GPDM requires finding the latent positions $\mathbf{X}$ and kernel hyperparameters $\mathcal{H} = \{\overline{\alpha}, \overline{\beta}\}$ with respect to the features $\mathbf{Y}$ by minimizing the negative log-posterior $-\ln p(\mathbf{X}, \mathcal{H} \mid \mathbf{Y})$. Minimization can be done using a scaled conjugated gradient (SCG) method [173]. This requires the inverse of the kernel matrix with a complexity of $O(N^3)$ in each optimization iteration. We select $d = 3$ as the latent space dimensionality.

It is difficult to learn a generic GPDM that captures large variations in the data and different motions. Combining trajectory data where the pedestrian is walking and data where the pedestrian is stopping results in degenerated models. Urtasun *et al.* [166] introduce additional constraints to prevent the degeneration of models. Selecting the correct constraints for a model that captures the walking and stopping motion of a pedestrian for the used features is difficult, especially with noisy data. Additionally, the complexity when training the model is increased. To avoid these problems, we train two separate models. The first model is trained using trajectory data segments where pedestrians are walking. Stopping situations are selected to train the second model. Because the beginning of a stopping action is difficult to define, data from 20 frames $(0.91s)$ before the stopping of the pedestrians is used. Examples of the two models are plotted in Figure 7.4.

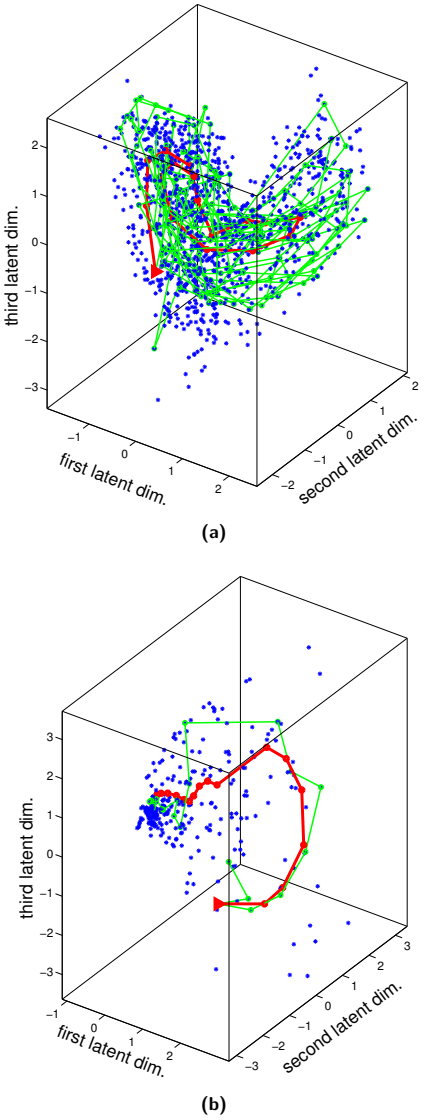


Figure 7.4: Traversal of a training trajectory (○) through the learned latent space (∗) and mean predictions (○) of a point (▶) for 17 frames ($0.77s$). Figures depict (a) the walking case and (b) the stopping case. All available training samples are shown.

Mean Prediction With the learned dynamic model a point $\mathbf{x}$ in the latent space is predicted and the most likely successor is derived using

$$\mu_{\mathbf{X}}(\mathbf{x}) = \mathbf{X}_{2:N}^{T}\mathbf{K}_{\mathbf{X}}^{-1}k_{\mathbf{X}}(\mathbf{x}) \tag{7.9}$$

with the vector $k_{\mathbf{X}}(\mathbf{x})$ containing at the i-th entry the results of $k_X(\mathbf{x}, \mathbf{x_i})$ using training sample $\mathbf{x}_i$.

Figure 7.4 illustrates this mean prediction of a point for several frames on the low dimensional space.

Latent Reconstruction A point $\mathbf{x}$ in the latent space is reconstructed using:

$$\mu_{\mathbf{Y}}(\mathbf{x}) = \mathbf{Y}^{T}\mathbf{K}_{\mathbf{Y}}^{-1}k_{\mathbf{Y}}(\mathbf{x}) \tag{7.10}$$

An example of the reconstructed scene flow feature is show in Figure 7.5.

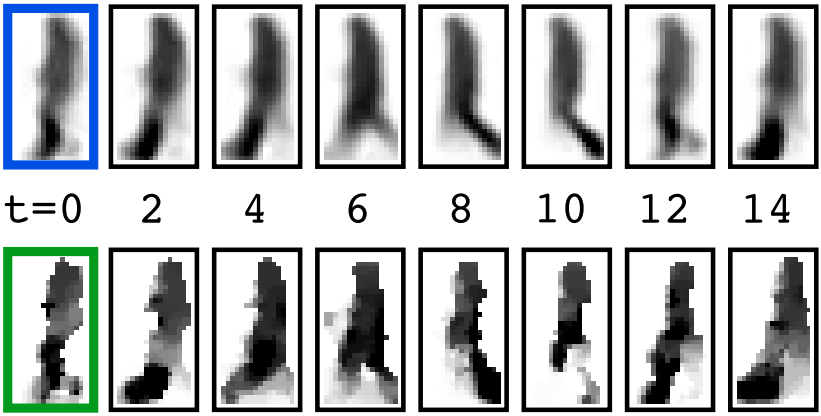


Figure 7.5: Observed ($t = 0$, bottom row) and optical flow features corresponding reconstructed features ($t = 0$, top row). (Top row) Reconstruction of the latent space prediction of the initial feature ($t = 0$, top row) for different prediction time-steps. (Bottom row) Flow that will be measured at the corresponding time-steps.

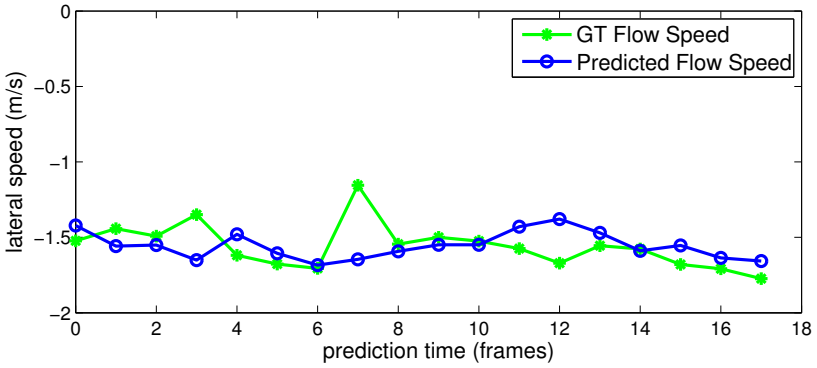


Figure 7.6: Predicted speed (∘) derived from predicted optical flow and corresponding measured optical flow speed (∗) for different prediction horizons.

Multiple Model Particle Filter

The state of a pedestrian at time t is described using $\phi_t = [\mathbf{x_t}, \mathcal{X}_t]$ where $\mathbf{x_t} \in \mathcal{R}^d$ is a point in the low dimensional space and $\mathcal{X}_t$ the lateral pedestrian position in the world. Given an observed motion feature $\mathbf{y_t}$ and observed lateral position $\mathcal{Y}_t$, the probability of a pedestrian state ϕ_t is computed by

$$p(\phi_t \mid \mathbf{y_t}, \mathcal{Y}_t) = \eta\, p(\mathbf{y_t}, \mathcal{Y}_t \mid \phi_t) \int p(\phi_t \mid \phi_{t-1}) p(\phi_{t-1} \mid \mathbf{y_{t-1}}, \mathcal{Y}_{t-1}) d\phi_{t-1} \tag{7.11}$$

with normalization constant η. The probability $p(\phi_t \mid \phi_{t-1})$ of observing a future state is computed from the GPDM latent space mean prediction.

This distribution is represented by a set of particles $\{\phi_t^{(s)} : s \in \{1, \ldots, S\}\}$ with corresponding weight $w_t^{(s)}$ that is propagated using a particle filter. Particles are predicted using the learned GPDM model with the predicted state $\hat{\phi}_t^{(s)} = [\hat{\mathbf{x}}_t^{(s)}, \hat{\mathcal{X}}_t^{(s)}]$ with $\hat{\mathbf{x}}_t^{(s)} = \mu_{\mathbf{X}}(\mathbf{x}_{t-1}^{(s)}) + n_x$, and $\hat{\mathcal{X}}_t^{(s)} = \mathcal{X}_{t-1}^{(s)} + s_{\mathcal{X}}(\hat{\mathbf{y}}_t^{(s)}, \Delta t)$. The predicted lateral position $\hat{\mathcal{X}}_t$ is estimated using $s_{\mathcal{X}}(\hat{\mathbf{y}}_t^{(s)}, \Delta t)$ which computes the traveled distance from the mean velocity derived from the reconstructed scene flow image $\hat{\mathbf{y}}_t^{(s)} = \mu_{\mathbf{Y}}(\hat{\mathbf{x}_t}^{(s)})$ and camera cycle time Δt. For each particle the noise term n_x is randomly sampled from $\mathcal{N}(\mathbf{0}, I_d \times \sigma_{n_x}^2)$, with an experimentally

derived $\sigma_{n_x} = 0.1$.

Scene flow feature similarity is computed using the Euclidean distance

$$d_f(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 \tag{7.12}$$

For the lateral position in the world the distance $d_p(\mathcal{Y}, \mathcal{X})$ is computed with:

$$d_p(\mathcal{Y}, \mathcal{X}) = \|\mathcal{Y} - \mathcal{X}\|^2 \tag{7.13}$$

Using the distances between the observed and predicted data the observation likelihood $p(y_t, \mathcal{Y}_t \mid \phi_t^{(s)}) \propto w_t^{(s)}$ is approximated using

$$w_t^{(s)} = exp\left(-\frac{d_f(\mathbf{y}_t, \hat{\mathbf{y}}_t^{(s)})^2}{2\sigma_f^2} - \frac{d_p(\mathcal{Y}_t, \hat{\mathcal{X}}_t^{(s)})^2}{2\sigma_p^2}\right) \tag{7.14}$$

with an empirically estimated $\sigma_f = 7$ for the feature similarity and $\sigma_p = 0.06$ for the deviation of the lateral position. The updated $\phi_t^{(s)}$ is obtained from $\hat{\phi}_t^{(s)}$ by reweighting the particle set.

For efficiency, an estimated state ϕ_T representing the pedestrian state in the future $T = t + \Delta T$ is derived from the weighted mean $\mathbf{x}^*$ of the particle set $\{\phi_t^{(s)}\}$ and iteratively predicted using $\mu_{\mathbf{X}}(\mathbf{x}^*)$. From the reconstructed predicted scene flow data (Figure 7.5) the pedestrian velocity in the future is computed (Figure 7.6). Integrating over the velocity predictions results in the predicted pedestrian position.

As mentioned in Section 7.2.1 we trained two models for the different pedestrian motions. Models are combined using an interacting multiple model particle filter (*MM-PF*) similar to [19]. For each model a fixed number of particles ($S = 200$) is used to represent the state. From the set of particles M_i in model i the model probability is derived using:

$$\gamma_i(t) = \frac{\sum_{\phi_t^{(s)} \in M_i} w^{(s)}}{\sum_{\phi_t^{(s)} \in M_1} w^{(s)} + \sum_{\phi_t^{(s)} \in M_2} w^{(s)}} \tag{7.15}$$

Model probabilities are updated similar to the *IMM-KF* scheme, described in Section 7.2.3. The conditional probability γ_{ij} of a transition from model i to j is computed using

$$\gamma_{ij}(t) = \frac{\Psi_{ij} \cdot \gamma_i(t)}{\sum_{k=1}^{2} \Psi_{kj} \cdot \gamma_k(t)} \tag{7.16}$$

with the state transition matrix Ψ.

We assume the lateral and longitudinal pedestrian dynamics to be weakly dependent. Longitudinal state estimation is decoupled and to each of the lateral models (GPDM) a Kalman Filter with corresponding constant velocity (CV) or constant position (CP) model is assigned to track the position. Longitudinal position $s^i_{\mathcal{Z}}(\Delta t)$ are linearly predicted with the estimated velocity of each filter. Mixing the lateral model predictions $s^i_{\mathcal{X}}(\Delta t)$ and the longitudinal KF prediction $s^i_{\mathcal{Z}}(\Delta t)$ with the state transition probabilities at t results in the pedestrian position:

$$s_{\mathcal{X},\mathcal{Z}}(\Delta t) = \sum_{i=1}^{2} \gamma_i \cdot \begin{pmatrix} s^i_{\mathcal{X}}(\Delta t) \\ s^i_{\mathcal{Z}}(\Delta t) \end{pmatrix} \tag{7.17}$$

In the following the approach using the Gaussian Process Dynamical Models in combination with scene flow features is abbreviated with *SFlowX/GPDM*.

7.2.2 Probabilistic Hierarchical Trajectory Matching System

The second approach uses motion features involving a low-dimensional histogram representation of optical flow. Measured pedestrian positions and motion features are subsequently used in a trajectory matching and filtering framework. From the filter state a future pedestrian position is derived by looking ahead on matched trajectories of the training set.

Motion Features

The low dimensional feature captures flow variations on the pedestrian legs and upper body. In order to operate from a moving vehicle additional invariance to pedestrian distance and vehicle motion is important. Features are designed to allow bounding box localization errors from a pedestrian detection system. Figure 7.7 illustrates the feature extraction steps. Flow vectors are normalized with the camera cycle time to account for asynchronous capture and frame drops. Flow vectors are further normalized with measurements from dense stereo for invariance to different pedestrian distances. The resulting normalized motion field is used to extract features given a bounding box detection and distance estimation z_{ped} from a pedestrian detection system. To ensure that the pedestrian is located in the box for all possible limb extensions and slight localization errors a bounding box aspect ratio of 4:3 is used. Motion vectors not belonging to the pedestrian body are suppressed by using only values at a depth similar to the estimated

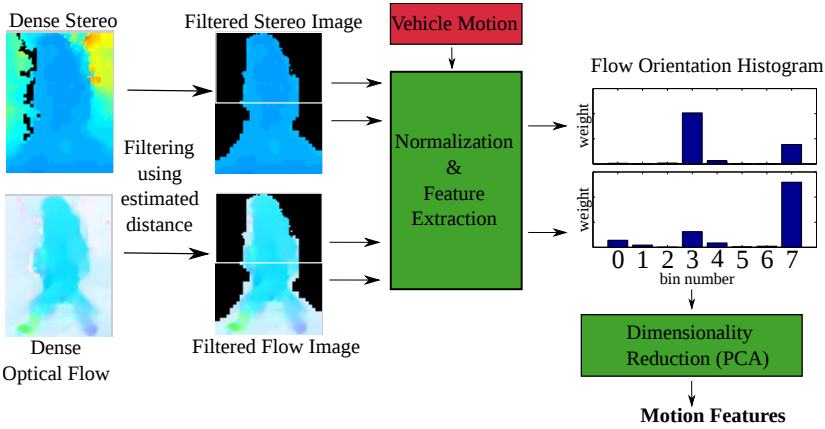


Figure 7.7: Motion feature extraction in the PHTM-based system

pedestrian distance. Remaining values in the motion field are used to compute the median object motion and extract orientation histograms. To capture motion differences between torso and legs the bounding box is split into an upper and lower sub-box. For each sub-box the median motion is removed to compensate the pedestrian ego motion. Resulting orientation vectors $v = [v_x, v_y]^T$ are assigned to bins $b \in [0, 7]$ using their 360^0 orientation $\theta = atan2(v_y, v_x)$ and bin index $b = \left\lfloor \frac{\theta}{\pi/4} \right\rfloor$. Bin contributions are weighted by their magnitude and resulting histograms are normalized with the number of contributions. A feature vector is formed by concatenating the histogram values and the median flow for the lower and upper box. Dimensionality reduction of the feature vector is achieved by applying PCA. The first three PCA dimensions with the largest eigenvalue are used as final histograms of orientation motion (*HoM*) features.

Trajectory Matching

A pedestrian trajectory Ω is represented using the ordered tuples $\Omega = ((\omega_1, t_1), \ldots, (\omega_N, t_N))$. For every timestamp t_i the pedestrian state ω_i consists of the lateral and longitudinal position of the pedestrian and additional features extracted from optical flow (Figure 7.8a). For path prediction, it is possible to compare an observed test trajectory with a history of H pedes-

trian states to each trajectory in a training database using a similarity measure. With the Quaternion-based Rotationally Invariant Longest Common Subsequence (QRLCS) metric [97] the optimal translation and rotation parameters to superimpose two trajectories are derived. The distance $\mathrm{dist}_{\mathrm{QRLCS}}(\Omega_i, \Omega_j) \in [0, 1]$ between two trajectories is given by the number of possible assignments determined by an ε area around each pedestrian state, normalized by the number of pedestrian states. Figure 7.8a illustrates this comparison process.

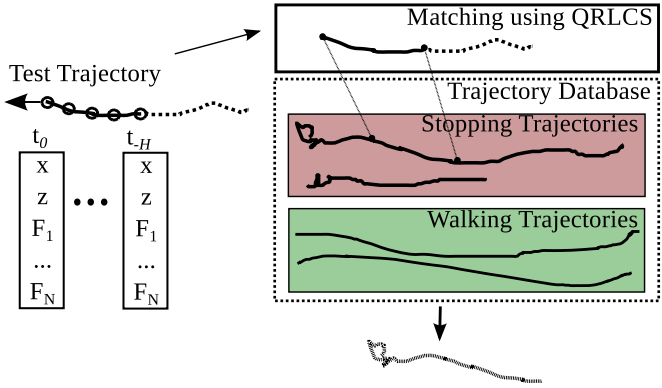
We replace this exhaustive search by a probabilistic search framework [97, 153]. A set of overlapping sub-trajectories (snippets, e.g. [90]) with fixed history of pedestrian states is created from a training database. Information of the snippet position in the origin trajectory and successor snippets are kept for later use. By piling the features for each state in a snippet into a description vector and applying the PCA method to these vectors, their principal dimensions can be ordered according to the largest eigenvalue. The resulting transformed description vector $\mathbf{c}$ is used to build a binary search tree. For each level l the snippet is assigned to the left or right sub-tree depending on the sign of c_l. Given N training snippets, the depth of the search tree, n, is $O(log(N))$. Figure 7.8b illustrates this search tree.

Given a trajectory $\Omega_{1:t}$ the probability of the state ϕ_t is computed by

$$p(\phi_t|\Omega_{1:t}) = \eta\, p(\Omega_{1:t}|\phi_t) \int p(\phi_t|\phi_{t-1})\, p(\phi_{t-1}|\Omega_{1:t-1})\, d\phi_{t-1} \tag{7.18}$$

with a normalization constant η. The distribution $p(\phi_t|\Omega_{1:t})$ is represented by a set of samples or particles $\{\phi_t^{(s)}\}$, which are propagated in time using a particle filter [17]. Each particle $\phi_t^{(s)}$ represents a snippet describing a pedestrian state with a history and an assigned likelihood. Our transition model $p(\phi_t|\phi_{t-1})$ is determined by a probabilistic search in the binary tree. Particle prediction is performed by a probabilistic search in the constructed binary tree and a lookup for the successor snippet in the training database. The distribution $p(\Omega_{1:t}|\phi_t)$ represents the likelihood that the measurement trajectory $\Omega_{1:t}$ can be observed given the current state. In the context of particle filters, this value corresponds to the weight of a particle and is approximated using $w^{(s)} = 1 - \mathrm{dist}_{\mathrm{QRCLS}}$ for each particle $\phi_t^{(s)}$.

An estimated state $\phi_T^{(s)}$ representing the pedestrian state in the future $T = t + \Delta T$ can be derived by looking ahead on the associated origin trajectory for



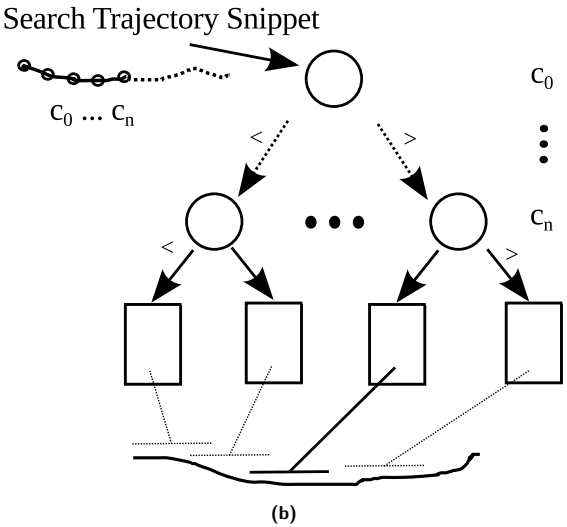


Figure 7.8: a) Test trajectory with history of length H containing position and feature information for every entry is matched to the training database. Resulting matching position and similarity distance to trajectories in the training database describe a possible trajectory course and class label. b) Tree representation of the trajectory training database. Leaf nodes represent trajectory snippets of fixed length. Similar trajectories are searched by traversing the tree using the trajectory descriptors for every level.

the current state $\phi_t^{(s)}$. This results in many hypotheses which are compensated using a weighted mean-shift algorithm [30] with a Gaussian kernel and weights $w^{(s)} \sim p(\phi_T^{(s)}|\Omega_{1:t})$. As the final predicted state ϕ_T^* the cluster center with the highest accumulated weight is selected.

The trajectory database contains two classes of trajectory snippets, the class $\mathcal{C}_s$ in which the pedestrian is stopping and the class $\mathcal{C}_w$ where the pedestrian continues walking. For the predicted object state ϕ_T^* derived using cluster members $L = \{\phi_t^{(l)}\}$ and the corresponding weight $w^{(l)}$ the stopping probability can be approximated using:

$$p(\mathcal{C}_s|L) \approx \frac{\sum_{\phi_t^{(l)} \in \mathcal{C}_s} w^{(l)}}{\sum_{\phi_t^{(l)} \in \mathcal{C}_s} w^{(l)} + \sum_{\phi_t^{(l)} \in \mathcal{C}_w} w^{(l)}}. \tag{7.19}$$

In the following the probabilistic hierarchical trajectory matching approach using histograms of orientation motion features is abbreviated with *HoM/Traj*.

7.2.3 Kalman Filter Based Systems

Kalman Filter

As a third approach a linear Kalman Filter (*KF*) [10] is used. The state $\hat{\mathbf{X}}$ of the filter is modeled as

$$\hat{\mathbf{X}} = [\ z\ x\ v_z\ v_x\ ]^T$$

with z/x being the longitudinal/lateral position of the pedestrian to the vehicle and v_z/v_x being its absolute longitudinal/lateral velocity in the world. Pedestrian positions are pseudo-measurements provided by the stereo pedestrian detection component, as described at the beginning of this Section. A constant velocity model (CV) is assumed as pedestrian motion model. Using this model means that all deviations from a constant pedestrian motion have to be captured as process noise. With the assumption that a pedestrian moving at $1.8\frac{m}{s}$ can stop in one second we select a process noise parameter $q = 1.8$ for the filter.

Interacting Multiple Model Kalman Filter

The fourth approach extends the previous KF with an additional constant position model (CP); this way, the Interacting Multiple Model Kalman Filter (*IMM-KF*) [10] is realized. The basic idea is to maintain a Kalman Filter for each possible motion model with state $\hat{\mathbf{x}}_j(t)$ and model probability $\gamma_j(t)$. This means

a steady walking pace is represented using a filter with the constant velocity (CV) model with process noise parameter q_{CV}. For non-moving pedestrians the constant position (CP) model with q_{CP} applies. Each iteration consists of three steps: interaction, filtering and mixing. The interaction step computes the mixing probability γ_{ij} from the current model probability γ_j and the state transition probability Ψ_{ij} (see Equation 7.16). From the mixing probability the mixed state mean $\hat{\mathbf{x}}_{0j}(t)$ and covariance matrix $\hat{\mathbf{P}}_{0j}(t)$ is computed as initial input for each filter in the filtering step using

$$\hat{\mathbf{x}}_{0j}(t) = \sum_{i=1}^{2} \hat{\mathbf{x}}_i(t)\gamma_{ij}(t). \tag{7.20}$$

See [10] for the computation of $\hat{\mathbf{P}}_{0j}(t)$. In the filtering step a *KF* predict/update step is done using the mixed state mean $\hat{\mathbf{x}}_{0j}(t)$ and covariance matrix $\hat{\mathbf{P}}_{0j}(t)$ derived in the interaction step. Given the likelihood function $\Lambda_j(t+1) = \mathcal{N}(r_j(t+1), \mathbf{S_j}(t+1))$ with residuum $r_j(t+1)$ and residual covariance $\mathbf{S}_j(t+1)$ the updated probabilities $\gamma_j(t+1)$ are computed using:

$$\gamma_j(t+1) = \frac{1}{c}\Lambda_j(t+1)\sum_{i=1}^{2}\Psi_{ij}\gamma_i(t) \tag{7.21}$$

with normalization factor c. An approximation of the resulting mixture model is then computed in the mixing step using

$$\hat{\mathbf{x}}(t+1) = \sum_{i=1}^{2} \hat{\mathbf{x}}_i(t+1)\gamma_i(t+1). \tag{7.22}$$

For the following evaluation $q_{CV} = 0.21$ and $q_{CP} = 0.41$ has been derived from the set of training trajectories, with respect to the positions minimum root-mean-square error (RMSE). The matrix Ψ describing the transition probabilities between the CV and CP model has experimentally been derived from the available training data $\Psi = [0.999, 0.001; 0.001, 0.999]$. Choosing larger values for the model transitions result in more frequent, undesired switches, especially with noisy measurements. The *IMM-KF* is said to be non-sensitive to improperly selected transition probabilities [18].

Sequences	**vehicle standing**	**vehicle moving**	**vehicle standing+moving**
ped. stopping	11	5	16
ped. walking	9	4	13

Table 7.1: The number of sequences with different pedestrian and vehicle actions in our dataset.

7.3 Experiments

Video data of two scenarios (Figure 7.1) was recorded using a stereo camera system (baseline $30\ cm$, $22\ fps$) mounted behind the windshield of a vehicle. The first scenario features the stopping of a pedestrian at the curbstone. In the second scenario, the pedestrian crosses the street. In both scenarios, the pedestrian was not occluded. In some test runs the vehicle is stationary while in others the vehicle is moving at speeds of $20-30\ km/h$. The dataset involved four different pedestrians in three different locations at a distance range of $5-34\ m$ to the vehicle. Table 7.1 provides some further statistics on the dataset. Figure 7.9 illustrates some test images.

The ground truth (GT) locations of the pedestrians in the world were obtained by manual labeling the pedestrian shapes in the images. The median disparity value on the pedestrian upper body and the center foot-point of the shape is used to obtain the longitudinal and lateral positions on the ground plane. In terms of alignment along the time axis, for each trajectory where the pedestrian is stopping the moment of the last placement of the foot is labeled as the stopping moment. The time-to-stop (TTS) value counts the number of frames until this event; frames earlier to the stopping event have positive TTS values, frames after the stopping event have negative TTS values. In sequences where the pedestrian continues walking, the closest point to the curbstone (with closed legs) is labeled. Analogous to the TTS definition, the latter is called time-to-curb value (TTC).

Performance evaluation is done using input data with different noise characteristics regarding the image bounding box positions. 2D bounding boxes derived from manually labeled pedestrian shapes (termed *label box*) are used as the most accurate input data for feature extraction and localization; it reflects the case of an "ideal" pedestrian detector. We further consider the case where these ideal 2D bounding boxes are perturbed by uniform noise; we add up to 10% of the original height of the bounding boxes to their height and center (the resulting bounding

Figure 7.9: Example images from the dataset showing showing the pedestrian action. Images show the labeled stopping (left) or walking (right) moment.

	veh. standing+moving	
	lat.	**long.**
GT	0.03	0.10
label box	0.05	0.22
jittered box	0.13	0.68
sys. detections	0.06	0.64

Table 7.2: Mean deviation (m) of the pedestrian position on the ground plane (lateral and longitudinal) compared to the smoothed ground truth data

boxes are termed *jittered*). Finally, we consider 2D bounding boxes provided by a state-of-the-art HOG/linSVM pedestrian detector [34] (termed *system detections*). Hereby, detection "gaps" are filled in by means of a standard correlation tracker. Considering data with artificial noise allows to abstract away from the noise bias of a particular pedestrian detector. As we will see shortly, the overall noise level added artificially is realistic, in the sense that it is similar to that of a state-of-the-art detector.

The lateral and longitudinal position errors on the ground plane for different input data are summarized in Table 7.2. In these experiments, we compared to a smoothed version of the GT ground plane positions. GT positions from walking trajectories, where we are certain that the pedestrian is moving with an approximately constant velocity ($-40 \leq TTC \leq 40$), were fitted with a curvilinear model, minimizing pedestrian velocity- and yaw-changes by non-linear least-squares. For the stopping trajectories, smoothing was only applied to the cases where the pedestrian is standing ($TTS \leq 0$), by simple averaging. Note that smoothed GT was only used for the purpose of Table 7.2. In the path prediction experiments, comparisons involved the non-smoothed GT. Following observations can be made from Table 7.2. As expected, the longitudinal error is larger than the lateral error, due to stereo vision characteristics. Adding afore-mentioned uniform noise on 2D bounding boxes results in a degradation of positional accuracy of about $10\ cm$ and $50\ cm$ in lateral and longitudinal direction, respectively. Positional errors are similar for the artificial noise and real detector case.

7.3.1 Parameter Settings and Evaluation Set-Up

We compare the four approaches using equal parameter settings, whenever possible. Lateral and longitudinal noise parameters for the *KF* and *IMM-KF* and longitudinal noise parameters for tracking the distance of the *SFlowX/GPDM* are selected from Table 7.2. Process noise parameters and state transition matrix were derived heuristically (see Section 7.2.3). The same state transition matrix is used for the *MM-PF* of the *SFlowX/GPDM* system and the *IMM-KF*.

The analysis of walking trajectories showed an average gait cycle of $10 - 14$ frames for different pedestrians. The trajectory database for the *HoM/Traj* contains sub-trajectories, generated in a sliding window fashion, with a fixed length of 10 frames. For test trajectories a history of 14 frames is used to capture gait cycle variations. Approximating the current probability density is done with $S = 400$ particles and a tree search deviation parameter of $\beta = 0.05$. The mean shift position procedure operates with a kernel width value $h = 0.1$.

Training and testing data has been processed using *leave-one-out* cross validation. This means that one sequence is used for testing and the remaining training sequences are used to learn the GPDM models *SFlowX/GPDM*, or for search tree generation (*HoM/Traj*).

7.3.2 Pedestrian Path Prediction

We are interested in the ability of each system to predict future pedestrian positions accurately. Tables 7.4 and 7.5 list ground plane localization accuracy (i.e. longitudinal and lateral dimensions combined) at different prediction horizons, for each system. Localization accuracy is measured in terms of mean and standard deviation of the per-sequence RMSE. Per-sequence RMSE is determined by comparing system predictions at various time horizons with the Ground Truth (GT), when the pedestrian is inside the frame range $[20, -10]$, where frame 0 denotes the manually labeled TTS/TTC moment. This corresponds to an evaluation time range of $[0.91, -0.45]$ seconds around the TTS/TTC event. Pedestrian positions are predicted up to 17 frames ($0.77\ s$) into the future. Tables 7.4 and 7.5 list the results for walking and stopping trajectories, respectively. Results are further differentiated based on whether the own vehicle is standing or moving, or whether all data is used (cf. Table 7.1).

On walking scenarios (Table 7.4) all approaches show a similar prediction performance when pedestrian bounding boxes are set precise (*label box*). In the more realistic case of inaccurate image localization (*jittered box*) and moving ve-

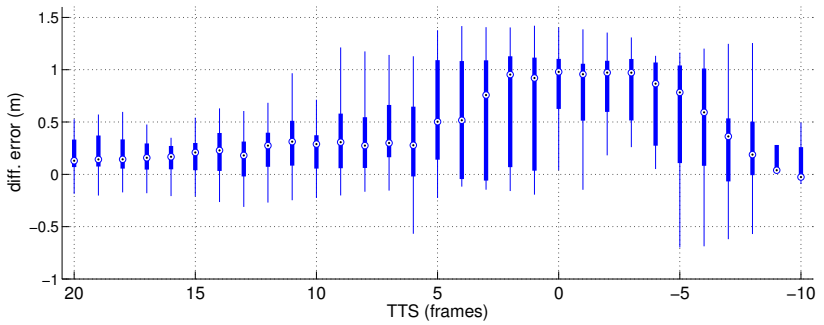


Figure 7.10: Distribution of the lateral prediction error difference (*IMM-KF* - *HoM/Traj*). Results for the *jittered* data, prediction horizon of 17 frames and stopping trajectories.

hicle, we see that *HoM/Traj*, unlike the other approaches, shows no performance degradation, and thus gains a slight edge. We attribute this to the robustness of trajectory matching to outliers in the longitudinal dimension. On stopping scenarios (Table 7.5), where the constant velocity assumption is violated, incorporation of motion features leads to a path prediction performance advantage of up to a factor of two for *HoM/Traj* and *SFlowX/GPDM* compared to the *KF* based variants (e.g. *jittered* data, vehicle standing and moving case). As before, the trajectory matching of *HoM/Traj* shows added robustness to noise caused by bounding box position errors and vehicle ego motion.

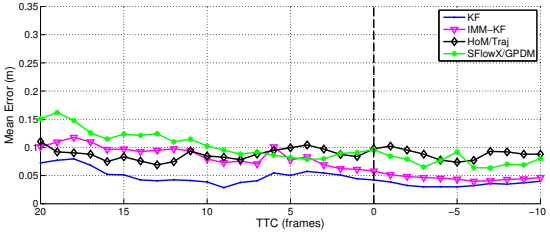
In the intelligent vehicle pedestrian safety context, the *lateral* component of the localization error is especially relevant; it determines whether the pedestrian enters the vehicle driving corridor and a collision potentially occurs. Figure 7.11 lists the mean lateral localization error at various time offsets to the labeled TTS/TTC moment (*jittered* data, vehicle standing and moving case). Separate plots are shown depending on the prediction horizon (0 or 17 frames) and whether the pedestrian is walking or stopping. We observe no significant performance difference between for walking trajectories (Figure 7.11a, 7.11c). For stopping scenarios (Figure 7.11b, 7.11d), the advantage of the additional motion model of the *IMM-KF* vs. the *KF* becomes visible (in Table 7.5 this advantage was averaged away over frame range [20, -10], due to the inclusion of time instants still involving walking). Stopping of the pedestrian leads to a switch to the CP model and lower localization error compared to the *KF* with CV model. Figure 7.11 also

		system detections			
		walking		stopping	
		0	**17**	**0**	**17**
KF	Mean	0.2	0.55	0.27	1.08
	± *Std*	*0.05*	*0.28*	*0.08*	*0.29*
IMM-KF	Mean	0.21	0.55	0.29	1.04
	± *Std*	*0.06*	*0.3*	*0.14*	*0.25*
HoM/Traj	Mean	0.14	0.39	0.14	0.63
	± *Std*	*0.03*	*0.12*	*0.04*	*0.22*
SFlowX/GPDM	Mean	0.15	0.43	0.21	0.52
	± *Std*	*0.06*	*0.27*	*0.06*	*0.19*

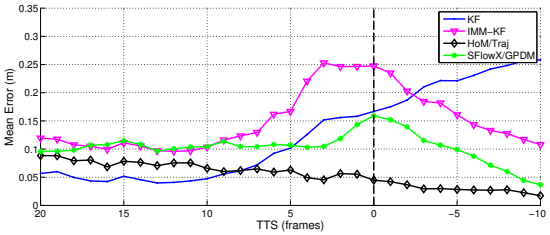
Table 7.3: Mean combined longitudinal and lateral RMSE (m) for *stopping and walking trajectories* using system detections with different prediction horizons (frames).

shows that *HoM/Traj* and *SFlowX/GPDM* are more quickly able to adjust to the change in the pedestrian motion, resulting in a lower lateral localization error than the KF-based approaches. Figure 7.10 illustrates the distribution of the lateral prediction error *difference* between *IMM-KF* and *HoM/Traj* for stopping trajectories. Performance differences are clearly visible close to the stopping event $TTS = 0$.

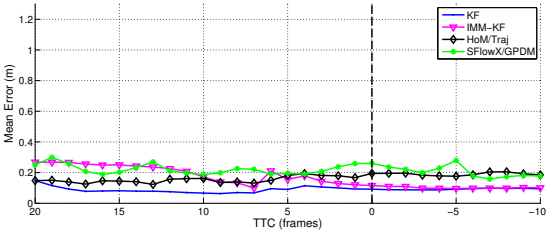
Results using tracked detections from a state-of-the-art HOG/linSVM pedestrian detector [34] are listed in Table 7.3. For this experiment, we used a subset of 7 walking and 13 stopping trajectories (5 trajectories with a moving vehicle) where the pedestrian detector had a decent performance in the first place (detection "gaps" no longer than 10 frames consecutively). We observe that using actual system detections, rather than simulated detections, does not change the performance ranking of the approaches considered (compare Table 7.3 with the entries of Tables 7.4 and 7.5 where the vehicle is standing and moving). In fact, performance with actual system detections is similar to that obtained with noise-perturbed, GT *jittered* data; this is not surprising given similar per-frame localization measurement error (cf. Table 7.2).



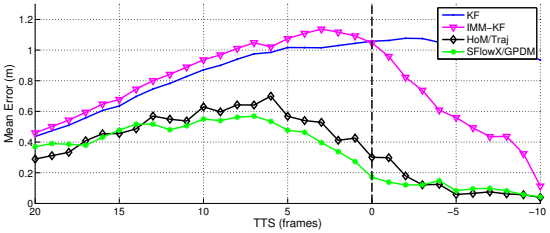
(a) pedestrian walking, prediction time 0 frames



(b) pedestrian stopping, prediction time 0 frames



(c) pedestrian walking, prediction time 17 frames



(d) pedestrian stopping, prediction time 17 frames

Figure 7.11: Mean lateral localization error at each time-step for *jittered* data and vehicle standing and moving (walking vs. stopping trajectories, prediction horizon 0 vs. 17 frames)

		veh. standing						veh. standing + moving				veh. moving			
		label box				jittered box		label box		jittered box		label box		jittered box	
		0	5	11	17	0	17	0	17	0	17	0	17	0	17
KF	Mean	0.15	0.19	0.22	0.28	0.24	0.35	0.17	0.38	0.24	0.45	0.21	0.62	0.24	0.69
	± *Std*	*0.07*	*0.07*	*0.09*	*0.12*	*0.2*	*0.2*	*0.07*	*0.25*	*0.17*	*0.28*	*0.05*	0.29	*0.06*	*0.29*
IMM-KF	Mean	0.14	0.17	0.2	0.25	0.23	0.34	0.19	0.41	0.26	0.5	0.29	0.77	0.33	0.86
	± *Std*	*0.07*	*0.08*	*0.1*	*0.12*	*0.21*	*0.22*	*0.1*	*0.3*	*0.19*	*0.34*	*0.09*	*0.26*	*0.09*	*0.27*
HoM/Traj	Mean	0.13	0.17	0.23	0.29	0.14	0.30	0.14	0.33	0.15	0.35	0.15	0.43	0.15	0.44
	± *Std*	*0.03*	*0.03*	*0.05*	*0.07*	*0.03*	*0.07*	*0.03*	*0.13*	*0.03*	*0.11*	*0.02*	*0.17*	*0.02*	*0.12*
SFlowX/GPDM	Mean	0.15	0.2	0.26	0.34	0.17	0.5	0.17	0.41	0.19	0.52	0.21	0.62	0.25	0.69
	± *Std*	*0.04*	*0.08*	*0.13*	*0.18*	*0.06*	*0.32*	*0.05*	*0.24*	*0.08*	*0.31*	*0.06*	*0.25*	*0.08*	*0.22*

Table 7.4: Mean combined longitudinal and lateral RMSE (m) for *walking trajectories* and different prediction horizons (frames).

		veh. standing						veh. standing + moving				veh. moving			
		label box				jittered box		label box		jittered box		label box		jittered box	
		0	5	11	17	0	17	0	17	0	17	0	17	0	17
KF	Mean	0.20	0.36	0.61	0.93	0.27	0.81	0.24	1.04	0.33	1.14	0.32	1.25	0.44	1.39
	± *Std*	*0.04*	*0.06*	*0.1*	*0.15*	*0.14*	*0.23*	*0.08*	*0.26*	*0.15*	*0.37*	*0.1*	*0.33*	*0.15*	*0.51*
IMM-KF	Mean	0.18	0.31	0.55	0.87	0.27	0.77	0.22	0.98	0.32	1.08	0.32	1.19	0.47	1.26
	± *Std*	*0.03*	*0.04*	*0.07*	*0.12*	*0.13*	*0.24*	*0.08*	*0.19*	*0.16*	*0.2*	*0.1*	*0.17*	*0.17*	*0.15*
HoM/Traj	Mean	0.12	0.19	0.34	0.58	0.12	0.56	0.13	0.63	0.12	0.62	0.15	0.74	0.15	0.76
	± *Std*	*0.02*	*0.04*	*0.11*	*0.17*	*0.02*	*0.10*	*0.02*	*0.21*	*0.02*	*0.18*	*0.02*	*0.23*	*0.03*	*0.19*
SFlowX/GPDM	Mean	0.23	0.27	0.35	0.51	0.29	0.62	0.16	0.53	0.23	0.64	0.21	0.66	0.41	0.89
	± *Std*	*0.07*	*0.07*	*0.06*	*0.07*	*0.08*	*0.22*	*0.05*	*0.23*	*0.26*	*0.29*	*0.05*	*0.32*	*0.42*	*0.36*

Table 7.5: Mean combined longitudinal and lateral RMSE (m) for *stopping trajectories* and different prediction horizons (frames).

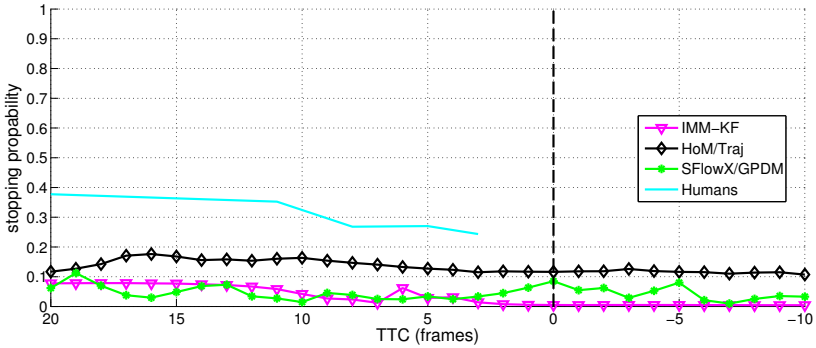
7.3.3 Pedestrian Action Classification

We also tested the ability of various systems to classify pedestrian actions, i.e. whether the pedestrian will cross or not. Figure 7.12 illustrates the performance of each system on stopping and walking test trajectories; depicted is the estimated probability of stopping, as a function of TTS or TTC. For the *SFlowX/GPDM* and *HoM/Traj* systems, this was achieved by means of Equations 7.15 and 7.19, respectively. For the *IMM-KF* filter, stopping was estimated by means of the probability of the CP model, following Equation 7.21.

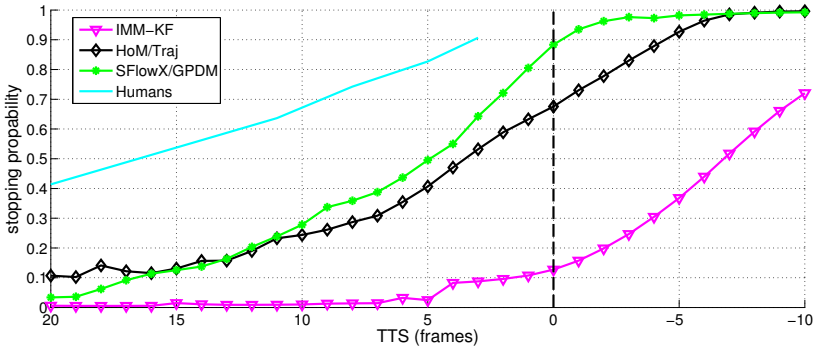
To put the performance of the systems in context, we also evaluated human performance. Video data was presented to several test subjects using a graphical user interfaces, where playback was automatically stopped at five different TTC or TTS moments (20, 11, 8, 5, 3). For each run, the test subjects had to decide whether the pedestrian will stop at the curbstone or cross the street and provide a probability (i.e. confidence) using a slider ranging from 0 to 1. Sequence and playback stopping point were randomly selected before being presented to the test subjects to avoid the effect of re-identification.

See Figure 7.12. On walking trajectories, all systems show a low and relatively constant stopping probability. On stopping trajectories, all systems initially start with a low stopping probability, since stopping is preceded by walking. But within a dozen frames before the stopping event, the stopping probability increases more markedly.

Class membership is determined at each time instant of an input trajectory assigned by thresholding the estimated stopping probability (cf. Figure 7.12). Based on the training set, we selected for each system and for the human group a threshold that minimizes its classification error (i.e. stopping classified as walking and vice versa) over all sequences and time instants. Figure 7.13 illustrates the resulting classification accuracy over time using these "optimal" thresholds. As can be seen, the humans outperform the various automatic systems at this action classification task. The humans reach an accuracy of 0.8 in classifying the correct pedestrian's action about 570 ms before the event. This accuracy is only reached about 230 ms before the event by the newly developed *SFlowX/GPDM* and *HoM/Traj* systems, which use augmented visual features. The baseline *IMM-KF* system does worst, reaching the corresponding accuracy only about 90 ms before the event.



(a) pedestrian walking



(b) pedestrian stopping

Figure 7.12: Estimated probability of stopping over time for (a) walking and (b) stopping test trajectory (averaged over all respective sequences).

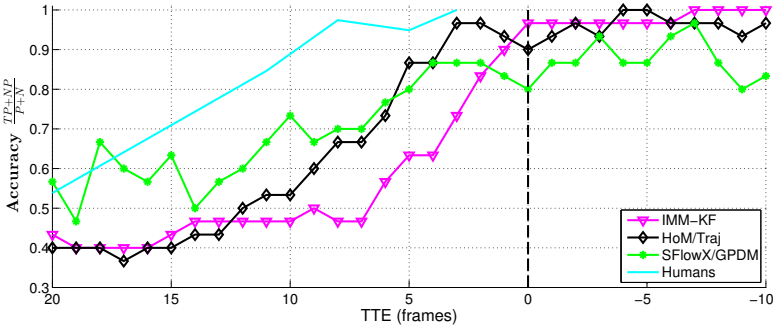


Figure 7.13: Classification accuracy of the different systems over time. Results for the *jittered* data.

7.4 Discussion

Table 7.3 indicates that the proposed, more advanced methods for pedestrian path prediction (*SFlowX/GPDM* or *HoM/Traj*) can achieve more accurate path prediction than basic approaches (linear KF or IMM extension thereof). The associated benefit, in terms of reduction of the combined lateral and longitudinal position error, is $10 - 50\ cm$ at a time horizon of $0 - 17$ frames (up to $0.77\ s$) around the stopping event. Figure 7.11(d) indicates that a $50\ cm$ improvement in lateral position estimation is reached at several time instants. Tables 7.4 and 7.5 also suggest that the vehicle ego-motion compensation is done reasonably but not perfectly. Further benefits can be obtained when localizing the pedestrian more accurately and improving upon the vehicle ego-motion compensation. Comparing the columns "vehicle standing, label box, 17" (ideal situation) and "vehicle moving, jittered box, 17" (currently achievable situation) shows that position prediction can be improved by approximately $15 - 81\ cm$ for the various systems.

These findings are encouraging in terms of the expected benefits that can be achieved, when integrating more sophisticated path planning in pedestrian safety systems that perform emergency vehicle maneuvers (braking, steering).

We now turn to computational cost issues. Popularity of the simple linear KF can be explained due to its relative effectiveness and its low computational requirements. Although the computational cost doubles for the two process model IMM-KF it remains moderate compared to the *HoM/Traj* and *SFlowX/GPDM* approaches. For the latter, the cost of motion feature extraction needs first to

be accounted for. Furthermore, for the *HoM/Traj* approach, a prediction step requires traversing the search tree for each particle and looking up the successor snippet. Computational costs to predict a snippet is linear in the depth of the search tree. To incorporate new measurements the *QRCLS* distance to each particle has to be computed to update the particle weights. Looking ahead pedestrian position requires applying the mean shift procedure to the predicted particle positions to find the main mode. Main computational costs of the *SFlowX/GPDM* can be subdivided into the costs of predicting a GPDM latent space position and reconstructing the feature to apply the particle weight update. To predict a single particle the mean prediction on the latent space has to be applied (Equation 7.9). Because the first part of the formula ($\mathbf{X}_{2:N}^T \mathbf{K_X}^{-1}$) can be precomputed, the online costs for a latent space prediction result from evaluating the kernel function $k_{\mathbf{X}}(\mathbf{x})$ between the particle latent position $\mathbf{x}$ and all inducing variables. Similarly, reconstructing the feature requires evaluation of Equation 7.10 with a precomputed $\mathbf{Y}^T \mathbf{K_Y}^{-1}$ and evaluation of the kernel function $k_{\mathbf{Y}}(x)$. Costs for an update and predict of a particle are limited by the number of inducing variables.

Using an unoptimized MATLAB implementation on a 2.53 GHz CPU the path prediction 17 frames into the future requires on average 0.003 s for the *KF* and 0.017 s for the *IMM-KF*. The MATLAB version of the *HoM/Traj* approach with an optimized version of the trajectory matching and mean shift procedure in *C* requires 0.6 s. Without code optimization the *SFlowX/GPDM* approach requires on average 5.4 s for the prediction. Processing times for both *HoM/Traj* and *SFlowX/GPDM* can much be improved by special hardware (i.e. GPU, DSP, FPGA) by parallelizing the particle computation.

In terms of scalability, learning a GPDM quickly becomes unfeasible for larger datasets (say, ≥ 1000 samples) without an approximation method. The fully independent training conditional (FITC) [112] method reduces the complexity from $O(N^3)$ for the SCG method [173] (cf. Section 7.2.1c) to $O(k^2 N)$, where k is the number of data points that remain in the computation of the covariance matrix. Our full dataset contains approximately 1700 training samples and we set $k = 100$. When using the FITC approximation with a fixed number of inducing variables k, the online computational costs do not increase when extending the size of the training set. Without an approximation method, kernel evaluations between all samples in the training set have to be applied. Regarding scalability with the number of pedestrian motion patterns considered, training a single model containing different motion patterns lead to degenerated models on our dataset. Degenerated models showed an insufficient latent space predic-

tion performance. Although methods exist to prevent model degeneration [166] when using sequences with a large variety of motion patterns the computational complexity during training increases. Extending the *SFlowX/GPDM* system with additional motion patterns requires training separate GPDMs for each motion pattern. In the online case, the computational costs increase linearly in the number of models.

Since the *HoM/Traj* systems is an instance based learning approach using a probabilistic search tree, different motion patterns can be added to the training set without complication. Adding additional snippets to the training set leads to an increase of the depth of the binary search tree. Online costs to predict the state of the particle filter are thus sub-linear (logarithmic) in the number of training samples.

7.5 Conclusions

We considered four approaches (*SFlowX/GPDM*, *HoM/Traj*, *KF*, *IMM-KF*) for stereo vision-based pedestrian path prediction from a vehicle. Two scenarios were considered: in one, the pedestrian walking towards the curbside, lateral to the vehicle driving direction, would stop, while in the other, the pedestrian would continue walking.

Experiments indicated similar path prediction performance of the four approaches on walking motion, with near-linear dynamics. During stopping, however, the newly proposed approaches (*SFlowX/GPDM* or *HoM/Traj*), with non-linear and/or higher-order models and augmented motion features, achieved a more accurate (longitudinal and lateral) position prediction of $10 - 50\ cm$ at a time horizon of $0 - 0.77\ s$ around the stopping event. During stopping, a $50\ cm$ improvement in lateral position prediction was reached at several time instants. Further benefits are possible when localizing the pedestrian more accurately and improving upon the vehicle ego-motion compensation: we obtained improvements in lateral position prediction of $15 - 81\ cm$ for the various systems.

These are encouraging results, indicating that more advanced pedestrian path prediction approaches can make a real difference, when integrated in the next-generation active pedestrian safety systems that perform emergency vehicle manoeuvres (braking, steering). But more work is necessary on improving pedestrian localization, enlarging the set of pedestrian motion patterns considered and increasing the size of the dataset, before these benefits can materialize.

Chapter 8

Conclusion and Outlook

The primary goal of this thesis was the development of vision-based methods that can be integrated in an active pedestrian protection system. These systems can prevent accidents in situations where the driver of a car is inattentive or does not (or can not) react fast enough (see Chapter 5). A strong focus has been placed on the use of dense stereo data in different modules of a pedestrian detection system, i.e. region of interest generation, classification, tracking and path prediction

A basic requirement for an on-board active pedestrian protection system is the capability to detect pedestrians in a complex, always changing environment with as few false detections as possible. In Chapter 4 we evaluate the performance gains with respect to a reduction in false positives when using dense stereo in the ROI generation and refining the pedestrian location. Two state-of-the-art baseline systems, both using a flat-world assumption for the ROI generation and a classifier (HOG/linSVM) operating on intensity image data, have been assessed. The first system solely depends on a monocular camera setup for detection and tracking. In contrast, the ROI generation of the second system is extended so that ROIs at a certain distance are only generated if there is enough depth support from stereo data. To improve the tracking performance the 3D world position of the pedestrian is derived from stereo measurements. At a detection rate of 60% the number of false positives was reduced by a factor of 4 compared to the purely monocular based system. The dataset was made publicly available to facilitate benchmarking.

In Chapter 5 the ROI generation has been further improved and stereo data has been used as additional input for the classification module. The ROI generation has been extended to recover scene geometry in terms of camera height, camera pitch and road profile from dense stereo data on a frame-by-frame basis. Compared to the baseline system with flat world assumption and fixed pitch a reduction of false positives by a factor of 2.3 at similar detection rates was

demonstrated. By fusing classifier responses from different modalities (intensity and depth), an additional reduction of false positives by a factor of 3.3 has been reached. The different characteristics of depth and intensity features help to improve the fused classification performance. Combining the proposed ROI generation and high-level fusion resulted in a reduction of false positives by a factor of 7.5 at classification-level. Hence, the overall performance of a pedestrian protection system should be enhanced by using dense stereo in the ROI generation and classification module.

Methods presented in Chapter 4 and Chapter 5 have been combined with a generic motion-based object detection (6D-Vision) and implemented in a demonstrator vehicle. The vehicle automatically triggered emergency measures to prevent an accident with a pedestrian. In situations where a collision can not be prevented by braking, an evasive steering maneuver was triggered. By fusing results from the pedestrian recognition and the generic motion-based object detection, partially occluded pedestrians can be detected early and a more accurate pedestrian velocity is estimated. On two scenarios, requiring a split-second decision between no action, automatic braking and automatic evasion, the system made the correct decision in all runs (over 40). Even though vehicles with emergency braking systems have been introduced by different car manufactures, further research is needed before systems that automatically initiate evasive steering maneuver will be available. In order to initiate an automatic evasive maneuver it is not only important to detect obstacles and other traffic participants (e.g. vehicles, pedestrians,...) but also understand their behavior and predict possible actions. Because an evasive maneuver can be a rigorous intervention, it is important to avert an unnecessary or false automatic evasive maneuver.

State-of-the-art situation analysis methods rely on the prediction of the current state into the future using appropriate motion models and accurate pedestrian velocity estimation. But due to the highly dynamic behavior, pedestrians can start/stop walking abruptly, simple motion models can fail to predict future states. Deciding if an automated emergency maneuver has to be initiated, or not can be difficult when model assumptions fail. One of the main contributions of this thesis is the introduction of two new approaches (*SFlowX/GPDM* and *HoM/Traj*) for pedestrian path prediction and action classification in Chapter 7.

Motivated by a human factors study [150] that analyzes the importance of visual cues to predict pedestrian behavior, the newly introduced approaches use motion features derived from dense optical flow to address the path and action prediction problem. Using dense optical flow and stereo as an input modality for

the feature derivation is inspired by the fact [38] that humans can recognize a person's action from the motion itself and do not need to reconstruct a three dimensional model of the person performing the action. Two scenarios have been considered. In both scenarios a pedestrian is walking towards the curbside lateral to the vehicle driving direction. In one scenario the pedestrian stops at the curbside, in the other scenario he continues walking on to the road. Experiments showed that during the pedestrian's stopping motion the newly proposed approaches with augmented motion features achieve a more accurate prediction by $10 - 50\ cm$ at a time horizon of $0 - 17$ frames (up to $0.77\ s$) compared to the baseline method. Additionally, the proposed system can predict the intended pedestrian action from the system state. To put the performance of the proposed systems in context, we evaluated human performance to classify pedestrian actions. Test participants were able to identify the pedestrian action with a certainty of 80% approximately $570\ ms$ in advance. The new systems with augmented visual features reached this accuracy only $230\ ms$ before the event. Initiating an emergency braking $230\ ms$ early to a collision with a pedestrian, at a velocity of $50\ km/h$, reduces the risk of sever injure from 50% to 25% and the risk of death from 25% to 10% [160].

Performance and robustness of the proposed method can be extended in the future using cameras with a higher image resolution and frame-rate. This will allow extracting pose-related features for path predicting that are lost in noisy and low resolution image data, e.g. a more accurate optical flow describing the pedestrian motion. An accurate pedestrian segmentation in images [59] is not only important to extract features located on pedestrian body parts but also for a refined estimation of the 3D position of the pedestrian. Besides features that capture pedestrian motion patterns, high level behavior information can be used to predict possible actions (e.g. head orientation, age). Up to now little research has been conducted in identifying if a pedestrian is distracted [155] and does not notice oncoming traffic, e.g. by cell phone use. Estimating the head pose [129] and body orientation [50] can help to identify these situation. Another factor that influences the road crossing behavior of a person is age [16, 138]. Classifying the age of a pedestrian from high resolution image data [107] can be an important cue to predict their behavior.

Besides analyzing the behavior of a pedestrian in isolation, the interaction of the different traffic participants and information about the infrastructure (e.g. crosswalks, traffic lights,...) is important to interpret the current situation. Generating a more complete model of the environment can be realized by extending

the existing on-board sensor infrastructure of a vehicle (e.g. extended camera setup, radar, lidar, ...) and/or using cooperative sensor technology (Car-2-X communication [3, 124]). In situations where a pedestrian can not be detected by on-board sensors (e.g. due to complete occlusion by a parked car) additional wireless transmitted information from a pedestrian or other vehicles that detected the pedestrian will allow an interpretation of the environment. Generating a model that describes the environment, the interaction of its different participants and their possible behaviors is important for the vision of accident-free driving and is crucial for autonomous driving [63].

Appendix A

This thesis has led to the following publications:

Journal Publications

- C. G. Keller, M. Enzweiler, C. Schnörr, M. Rohrbach, D. F. Llorca, and D. M. Gavrila. The benefits of dense stereo for pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 12(4):1096–1106, 2011
- C. G. Keller, T. Dang, A. Joos, C. Rabe, H. Fritz, and D. M. Gavrila. Active pedestrian safety by automatic braking and evasive steering. *IEEE Trans. on Intelligent Transportation Systems*, 12(4):1292–1304, 2011
- C. G. Keller and D. M. Gavrila. Will the pedestrian cross? A study on pedestrian path prediction. *IEEE Trans. on Intelligent Transportation Systems*, PP(99):1–13, 2013

Conference Publications

- C. G. Keller, D. F. Llorca, and D. M. Gavrila. Dense stereo-based ROI generation for pedestrian detection. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 81–90. Springer Berlin Heidelberg, 2009
- C. G. Keller, M. Enzweiler, and D. M. Gavrila. A new benchmark for stereo-based pedestrian detection. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 691–696, 2011
- C. G. Keller, C. Hermes, and D. M. Gavrila. Will the pedestrian cross? Probabilistic path prediction based on learned motion features. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 386–395. Springer Berlin Heidelberg, 2011

Bibliography

[1] Y. Abramson and B. Steux. Hardware-friendly pedestrian detection and impact prediction. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 590–595, 2004.

[2] I. P. Alonso, D. F. Llorca, M. A. Sotelo, L. M. Bergasa, P. Revenga de Toro, J. Nuevo, M. Ocana, and M. A. G. Garrido. Combination of feature extraction methods for SVM pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 8(2):292–307, 2007.

[3] L. Andreone, A. Guarise, F. Lilli, D. M. Gavrila, and M. Pieve. Cooperative systems for vulnerable road users: The concept of the WATCH-OVER project. In *Proc. of IEEE ITS World*, volume 4, 2006.

[4] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[5] G. Antonini, S. Venegas Martinez, M. Bierlaire, and J.P. Thiran. Behavioral priors for detection and tracking of pedestrians in video sequences. *Int. Journal of Computer Vision (IJCV)*, 69(2), 2006.

[6] H. Badino. A robust approach for ego-motion estimation using a mobile stereo platform. In *1st Int. Workshop on Complex Motion (IWCM04)*, October 2004.

[7] H. Badino, U. Franke, and D. Pfeiffer. The stixel world-a compact medium level representation of the 3d-world. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 51–60. Springer Berlin Heidelberg, 2009.

[8] H. Badino, R. Mester, T. Vaudrey, and U. Franke. Stereo-based free space computation in complex traffic scenarios. In *SSIAI*, pages 189–192, March 2008.

[9] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. H. Matthies. A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle. *Int. Journal of Robotics Research (IJRR)*, 28, 2009.

[10] Y. Bar-Shalom, X R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. A Wiley-Interscience publication. John Wiley and Sons, 2004.

[11] S. Bauer, U. Brunsmann, and S. Schlotterbeck-Macht. FPGA implementation of a HOG-based pedestrian recognition system. *MPC-Workshop*, pages 49–58, 2009.

[12] S. Bauer, S. Kohler, K. Doll, and U. Brunsmann. FPGA-GPU architecture for kernel SVM pedestrian detection. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 61–68. IEEE, 2010.

[13] A. M. Baumberg and D. C. Hogg. Learning flexible models from image sequences. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 299–308. Springer-Verlag, 1993.

[14] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2903–2910. IEEE, 2012.

[15] R. Benenson, R. Timofte, and L. Van Gool. Stixels estimation without depth map computation. In *Proc. of the Int. Conference on Computer Vision (ICCV)*, pages 2010–2017. IEEE, 2011.

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

[17] M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 909–924, 1998.

[18] S.S. Blackman and R. Popoli. *Design and analysis of modern tracking systems*, volume 685. Artech House Norwood, MA, 1999.

[19] Y. Boers and JN Driessen. Interacting multiple model particle filter. In *Proc. IET Radar, Sonar & Navigation*, volume 150, pages 344–349, 2003.

[20] A. Boudissa, Joo Kooi Tan, Hyoungseop Kim, and S. Ishikawa. A simple pedestrian detection using lbp-based patterns of oriented edges. In *Proc. of the Int. Conference on Image Processing (ICIP)*, pages 469–472, 2012.

[21] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Proc. of the Int. Conference on Computer Vision (ICCV)*, pages 1515–1522. IEEE, 2009.

[22] A. Broggi, P. Cerri, L. Gatti, P. Grisleri, H. G. Jung, and J. Lee. Scenario-driven search for pedestrians aimed at triggering non-reversible systems. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 285–291, 2009.

[23] A. Broggi, A. Fascioli, I. Fedriga, A. Tibaldi, and M. Del Rose. Stereo-based preprocessing for human shape localization in unstructured environments. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 410–415, 2003.

[24] J. Broughton, J. Knowles, C. Brandstaetter, N. Candappa, M. Christoph, M. M. Vis, M. Haddak, L. Bouaoun, E. Amoros, J.-F. Pace, C. Martínez-Pérez, J. Sanmartín, G. Yannis, P. Evgenikos, E. Argyropoulou, and P. Papantoniou. Traffic safety basic facts 2011. Technical report, European Commission, 2011.

[25] Y. Cao, S. Pranata, and H. Nishimura. Local binary pattern features for pedestrian detection at night/dark environment. In *Proc. of the Int. Conference on Image Processing (ICIP)*, pages 2053–2056, 2011.

[26] Y. Cao, S. Pranata, M. Yasugi, Z. Niu, and H. Nishimura. Stagged multi-scale lbp for pedestrian detection. In *Proc. of the Int. Conference on Image Processing (ICIP)*, pages 449–452. IEEE, 2012.

[27] C. Castro. *Human factors of visual and cognitive performance in driving*. CRC, 2008.

[28] A. Chavan and S.K. Yogamani. Real-time DSP implementation of Pedestrian Detection algorithm using HOG features. In *IEEE Int. Telecommunications Society Conference (ITST)*, pages 352–355, 2012.

[29] Z. Chen, D. Ngai, and N. Yung. Pedestrian behavior prediction based on motion patterns for vehicle-to-pedestrian collision avoidance. In *Proc. of IEEE ITSC*, pages 316–321, 2008.

[30] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(5):603–619, 2002.

[31] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):681–685, 2001.

[32] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[33] Tim F Cootes, Camillo J Taylor, et al. Statistical models of appearance for computer vision. *World Wide Web Publication, February*, 2001.

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

[35] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 428–441, 2006.

[36] T. Dang, J. Desens, U. Franke, D. M. Gavrila, L. Schäfers, and W. Ziegler. Steering and evasion assist. In *Handbook of Intelligent Vehicles*, pages 759–782. Springer London, 2012.

[37] DARPA. Urban challenge. In *http://www.darpa.mil/grandchallenge/*, 2007.

[38] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 928–934. IEEE, 1997.

[39] G. De Nicolao, A. Ferrara, and L. Giacomini. A collision risk assessment approach as a basis for the on-board warning generation in cars. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2002.

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

[41] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[42] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(4):743–761, 2012.

[43] R. P. W. Duin and D. M. J. Tax. Experiments with classifier combining rules. In *Proc. of the Int. Workshop on Multiple Classifier Systems (MCS)*, pages 16–29, 2000.

[44] A. Elgammal. Nonlinear generative models for dynamic shape and dynamic appearance. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 182–182, 2004.

[45] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[46] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila. Multi-Cue pedestrian classification with partial occlusion handling. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 990–997, 2010.

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

[48] M. Enzweiler and D. M. Gavrila. A mixed generative-discriminative framework for pedestrian classification. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[49] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(12):2179–2195, 2009.

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

[51] M. Enzweiler and D. M. Gavrila. A multi-level Mixture-of-Experts framework for pedestrian classification. *IEEE Trans. on Image Processing (TIP)*, 20(10), 2011.

[52] M. Enzweiler, P. Kanter, and D. M. Gavrila. Monocular pedestrian recognition using motion parallax. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 792–797, 2008.

[53] A. Ess, B. Leibe, and L. van Gool. Depth and appearance for mobile scene analysis. In *Proc. of the Int. Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[54] L. Fan, K.-K. Sung, and T.-K. Ng. Pedestrian registration in static images with unconstrained background. *Pattern Recognition*, 36(4):1019 – 1029, 2003.

[55] J. C. Fell. A motor vehicle accident causal system: the human element. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 19, pages 42–48. SAGE Publications, 1975.

[56] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2241–2248. IEEE, 2010.

[57] D. Fernandez, I. Parra, M. A. Sotelo, P. Revenga, S. Alvarez, and M. Gavilan. 3D candidate selection method for pedestrian detection on non-planar roads. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1162–1167, 2007.

[58] P. Fiorini and Z. Shiller. Time optimal trajectory planning in dynamic environments. *Proc. of the ICRA*, 2:1553–1558, 1996.

[59] F. Flohr and D. M. Gavrila. Pedcut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues. In *Proc. of the British Machine Vision Conference (BMVC)*, 2013.

[60] U. Franke, D. M. Gavrila, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler. Autonomous driving goes downtown. *Intelligent Systems and Their Applications, IEEE*, 13(6):40–48, 1998.

[61] U. Franke, S. Gehrig, H. Badino, and C. Rabe. Towards optimal stereo analysis of image sequences. In *Robot Vision*, pages 43–58, 2008.

[62] U. Franke and A. Joos. Real-time stereo vision for urban traffic scene understanding. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 273–278, October 2000.

[63] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. G. Herrtwich. Making Bertha See. In *ICCV Workshop on Computer Vision for Autonomous Driving*, 2013.

[64] U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6d-vision: Fusion of stereo and motion for robust environment perception. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 216–223. Springer Berlin Heidelberg, 2005.

[65] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[66] H. Fritz. Vorrichtung zur Durchführung eines Fahrspurwechsels. German Patent Disclosure DE 100 12 737 B4, 2001.

[67] H. Fritz. Verfahren und Vorrichtung zur Kollisionsvermeidung für ein Fahrzeug durch Ausweichen vor einem Hindernis. German Patent Disclosure DE 10 2009 020 648 A1, 2009.

[68] K. Fuerstenberg. Pedestrian protection based on laserscanners. In *Proc. of the ITS*, 2005.

[69] T. Gandhi and M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Trans. on Intelligent Transportation Systems*, 8(3):413–430, 2007.

[70] D. M. Gavrila. A Bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(2):1408–1421, 2007.

[71] D. M. Gavrila, J. Giebel, and S. Munder. Vision-based pedestrian detection: The PROTECTOR system. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 13–18. IEEE, 2004.

[72] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. Journal of Computer Vision (IJCV)*, 73(1):41–59, 2007.

[73] D. Gerónimo. *A global approach to vision-based pedestrian detection for advanced driver assistance systems.* PhD thesis, Computer Vision Center. Barcelona (Spain), February 2010.

[74] D. Gerónimo, A .M. López, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(7):1239–1258,, 2010.

[75] D. Gerónimo, A. D. Sappa, A. López, and D. Ponsa. Adaptive image sampling and windows classification for on-board pedestrian detection. *Proc. of the Int. Conference on Computer Vision System (ICVS)*, 2007.

[76] D. Gerónimo, A. D. Sappa, D. Ponsa, and A. M. López. 2d-3d based on-board pedestrian detection system. *Computer Vision and Image Understanding*, 114(5):583–595, 2010.

[77] J. J. Gibson and L. E. Crooks. A theoretical field-analysis of automobile-driving. *The American journal of psychology*, 51(3):453–471, 1938.

[78] J. Giebel, D. M. Gavrila, and C. Schnörr. A Bayesian Framework for Multi-cue 3D Object Tracking. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 241–252. Springer, 2004.

[79] G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe. 3d vision sensing for improved pedestrian safety. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 19–24, June 2004.

[80] S. Grundhoff. Auto Motor Sport — Der elektronische Fahrlehrer kommt, August 2009. [Online; accessed 25-August-2013].

[81] E. Guizzo. How google's self-driving car works. IEEE Spectrum, 10 2011.

[82] Y. Hattori, E. Ono, and S. Hosoe. Optimum vehicle trajectory control for obstacle avoidance problem. *IEEE/ASME Trans. on Mechatronics*, 11:507 – 512, 2006.

[83] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern recognition*, 42(3):425–436, 2009.

[84] S. Hezel, A. Kugel, R. Männer, and D. M. Gavrila. FPGA-based template matching using distance transforms. In *Proc. of IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 89–97, 2002.

[85] J. Hillenbrand, A. Spieker, and K. Kroschel. A multilevel collision mitigation approach. *IEEE Trans. on Intelligent Transportation Systems*, 7(4):528 – 540, 2006.

[86] B. L Hills. Vision, visibility, and perception in driving. *Perception*, 1980.

[87] M. Hiromoto and R. Miyamoto. Hardware architecture for high-accuracy real-time pedestrian detection with CoHOG features. *Proc. of The Fifth IEEE Workshop on Embedded Computer Vision*, pages 894–899, 2009.

[88] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 807–814, June 2005.

[89] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):328–341, 2008.

[90] N. Howe, M. Leventon, and W. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In *Advances in Neural Information Processing Systems (NIPS)*, pages 820–826, 2000.

[91] M. Hussein, F. Porikli, and L. Davis. A comprehensive evaluation framework and a comparative study for human detectors. *IEEE Trans. on Intelligent Transportation Systems*, 10:417–427, 2009.

[92] H. Ippen. Auto Zeitung — Gefahren Umfahren, August 2009. [Online; accessed 25-August-2013].

[93] IRTAD. International traffic safety data and analysis group. In *http://www.internationaltransportforum.org/home.html*, 2011.

[94] R. Isermann, M. Schorn, and U. Stählin. Anticollision system proreta with automatic braking and steering. *Vehicle System Dynamics*, 46:683 – 694, 2008.

[95] H. Ju, B. Kwak, J. Shim, and P. Yoon. Precrash dipping node (PCDN) needs pedestrian recognition. *IEEE Trans. on Intelligent Transportation Systems*, 9(4):678 – 687, 2008.

[96] D. Juchem. Die Welt — Mercedes lernt das automatische Ausweichen, August 2009. [Online; accessed 25-August-2013].

[97] E. Käfer, C. Hermes, C. Wöhler, H. Ritter, and F. Kummert. Recognition of situation classes at road intersections. In *Proc. of the Int. Conference on Robotics and Automation (ICRA)*, pages 3960–3965, 2010.

[98] C. G. Keller, T. Dang, A. Joos, C. Rabe, H. Fritz, and D. M. Gavrila. Active pedestrian safety by automatic braking and evasive steering. *IEEE Trans. on Intelligent Transportation Systems*, 12(4):1292–1304, 2011.

[99] C. G. Keller, M. Enzweiler, and D. M. Gavrila. A new benchmark for stereo-based pedestrian detection. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 691–696, 2011.

[100] C. G. Keller, M. Enzweiler, C. Schnörr, M. Rohrbach, D. F. Llorca, and D. M. Gavrila. The benefits of dense stereo for pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 12(4):1096–1106, 2011.

[101] C. G. Keller and D. M. Gavrila. Will the pedestrian cross? A study on pedestrian path prediction. *IEEE Trans. on Intelligent Transportation Systems*, PP(99):1–13, 2013.

[102] C. G. Keller, C. Hermes, and D. M. Gavrila. Will the pedestrian cross? Probabilistic path prediction based on learned motion features. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 386–395. Springer Berlin Heidelberg, 2011.

[103] C. G. Keller, D. F. Llorca, and D. M. Gavrila. Dense stereo-based ROI generation for pedestrian detection. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 81–90. Springer Berlin Heidelberg, 2009.

[104] U. Kiencke and L. Nielsen. *Automotive Control Systems: For Engine, Driveline and Vehicle*. Springer-Verlag, 2000.

[105] J. Klappstein, F. Stein, and U. Franke. Monocular motion detection using spatial constraints in a unified manner. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 261–267, 2006.

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

[107] Young Ho Kwon and N. da Vitoria Lobo. Age classification from facial images. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 762–767, 1994.

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

[109] S. M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Computer Science Dept, Iowa State University, 1998.

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

[111] N. D. Lawrence. Gaussian process latent variable models for visualization of high dimensional data. *Advances in Neural Information Processing Systems (NIPS)*, 16:329–336, 2004.

[112] N. D. Lawrence. Learning for larger datasets with the Gaussian process latent variable model. *Workshop on Artificial Intelligence and Statistics*, 2007.

[113] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks*, pages 255–258. MIT Press, 1998.

[114] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[115] D. F. Llorca, M. A. Sotelo, I. Parra, J. E. Naranjo, M. Gavilán, and S. Álvarez. An experimental study on pitch compensation in pedestrian-protection systems for collision avoidance and mitigation. *IEEE Trans. on Intelligent Transportation Systems*, 10(3):469–474, 2009.

[116] P. Marchal, M. Dehesa, D. M. Gavrila, M. M. Meinecke, N. Skellern, and V. Vinciguerra. Final report. *Deliverable 27, EU Project SAVE-U*, 2005.

[117] M. Martin and H. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University, 1996.

[118] M. M. Meinecke, M. Obojski, D. M. Gavrila, E. Marc, R. Morris, M. Töns, and L. Letellier. Strategies in terms of vulnerable road user protection. In *EU Project SAVE-U, Deliverable D6*, 2003.

[119] M. M. Meinecke, M. Obojski, M. Töns, and M. Dehesa. SAVE-U: First experiences with a pre-crash system for enhancing pedestrian safety. In *Proc. of the ITS*, 2005.

[120] M. Meuter, U. Iurgel, S.-B. Park, and A. Kummert. Unscented kalman filter for pedestrian tracking from a moving host. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 37–42, 2008.

[121] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 69–81, 2004.

[122] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, and M. Yoshimoto. Architectural study of HOG feature extraction processor for real-time object detection. In *IEEE Workshop on Signal Processing Systems (SiPS)*, pages 197–202, 2012.

[123] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(4):349–361, 2001.

[124] C. Morhart, E. Biebl, D. Schwarz, and R. Rasshofer. Cooperative multi-user detection and localization for pedestrian protection. In *German Microwave Conference (GeMiC)*, pages 1–5. IEEE, 2009.

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

[126] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(11):1863–1868, 2006.

[127] S. Munder and D. M Gavrila. An experimental study on pedestrian classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(11):1863–1868, 2006.

[128] S. Munder, C. Schnörr, and D. M. Gavrila. Pedestrian detection and tracking using a mixture of view-based shape-texture models. *IEEE Trans. on Intelligent Transportation Systems*, 9(2):333–343, 2008.

[129] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(4):607–626, 2009.

[130] S. Nedevschi, S. Bota, and C. Tomiuc. Stereo-based pedestrian detection for collision-avoidance applications. *IEEE Trans. on Intelligent Transportation Systems*, 10(3):380–391, 2009.

[131] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, T. Graf, and R. Schmidt. High accuracy stereovision approach for obstacle detection on non-planar roads. In *Proc. of the IEEE Intelligent Engineering Systems (INES)*, pages 211–216, 2004.

[132] L. Oliveira, U. Nunes, and P. Peixoto. On exploration of classifier ensemble synergism in pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 11(1):16–27, 2010.

[133] Paul L Olson. Vision and perception. *Automotive ergonomics*, 1993.

[134] F. Oniga, S. Nedevschi, M. M. Meinecke, and T. B. To. Road surface and obstacle detection based on elevation maps from dense stereo. In

IEEE Intelligent Transportation Systems Conference (ITSC), pages 859–865, 2007.

[135] M. Oren, C.P. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 193–99, 1997.

[136] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3258–3265, 2012.

[137] G. Overett, L. Petersson, N. Brewer, L. Andersson, and N. Pettersson. A new pedestrian dataset for supervised learning. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 373–378, 2008.

[138] J. A. Oxley, E. Ihsen, B. N. Fildes, J. L. Charlton, and R. H. Day. Crossing roads safely: An experimental study of age differences in gap selection by pedestrians. *Accident Analysis & Prevention*, 37(5):962 – 971, 2005.

[139] H. B. Pacejka. *Tyre and Vehicle Dynamics*. SAE International, 2002.

[140] C. Papageorgiou and T. Poggio. A trainable system for object detection. *Int. Journal of Computer Vision (IJCV)*, 38:15–33, 2000.

[141] D. Pfeiffer and U. Franke. Towards a global optimal multi-layer stixel representation of dense 3d data. In *Proc. of the British Machine Vision Conference (BMVC)*, pages 1–12, 2011.

[142] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances In Large Margin Classifiers*, pages 61–74, 1999.

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

[144] Clemens Rabe, Uwe Franke, and Stefan Gehrig. Fast detection of moving objects in complex scenarios. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 398–403, 2007.

[145] L. Raskin, M. Rudzsky, and E. Rivlin. Dimensionality reduction using a gaussian process annealed particle filter for tracking and classification of articulated body motions. *Computer Vision and Image Understanding*, 115(4):503 – 519, 2011.

[146] P. Riekert and T. E. Schunck. Zur Fahrmechanik des gummibereiften Kraftfahrzeugs. *Archive of Applied Mechanics*, 11:210–224, 1940.

[147] M. Rohrbach, M. Enzweiler, and D. M. Gavrila. High-level fusion of depth and intensity for pedestrian classification. In *Pattern Recognition*, Lecture Notes in Computer Science, pages 101–110. Springer Berlin Heidelberg, 2009.

[148] E. Sangineto, M. Cristani, A. Del Bue, and V. Murino. Learning discriminative spatial relations for detector dictionaries: An application to pedestrian detection. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 273–286, 2012.

[149] C. Schmidt, F. Oechsle, and W. Branz. Research on trajectory planning in emergency situations with multiple objects. In *Proc. of the IEEE ITSC*, pages 988 – 992, 2006.

[150] S. Schmidt and B. Färber. Pedestrians at the kerb - recognising the action intentions of humans. *Transportation Research Part F*, 12(4):300 – 310, 2009.

[151] N. Schneider and D. M. Gavrila. Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2013.

[152] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. *Proc. of the Int. Conference on Computer Vision (ICCV)*, pages 24–31, 2009.

[153] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. *Proc. of the European Conf. on Computer Vision (ECCV)*, 2350:784–800, 2002.

[154] M. Sivak. The information that drivers use: Is it indeed 90% visual? *Perception*, 25:1081–1090, 1996.

[155] D. Stavrinos, K. W. Byington, and D. C. Schwebel. Distracted walking: Cell phones increase injury risk for college pedestrians. *Journal of Safety Research*, 42(2):101 – 107, 2011.

[156] P. Sudowe and B. Leibe. Efficient Use of Geometric Constraints for Sliding-Window Object Detection in Video. In *Proc. of the Int. Conference on Computer Vision System (ICVS)*, 2011.

[157] N. Suganuma and N. Fujiwara. An obstacle extraction method using virtual disparity image. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 456–461, 2007.

[158] M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata. Pedestrian detection with convolutional neural networks. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 224–229. IEEE, 2005.

[159] J. Tao and R. Klette. Tracking of 2d or 3d irregular movement by a family of unscented kalman filters. *Journal of Information and Convergence Communication Engineering*, 2012.

[160] B. C. Tefft. Impact speed and a pedestrian's risk of severe injury or death. *Accident Analysis & Prevention*, 2012.

[161] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B*, 61(3):611–622, 1999.

[162] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University Technical, 1991.

[163] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on Riemannian manifolds. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[164] I. Ulusoy and C. M. Bishop. Generative versus discriminative methods for object recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 258–265. IEEE, 2005.

[165] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, M. Gittleman, S. Harbaugh, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson. Special Issue on the 2007 DARPA Urban Challenge. *Journal of Field Robotics (JFR)*, 2008.

[166] R. Urtasun, D. Fleet, and N. Lawrence. Modeling human locomotion with topologically constrained latent variable models. *Human Motion–Understanding, Modeling, Capture and Animation*, pages 104–118, 2007.

[167] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 238–245, 2006.

[168] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. Dimensionality reduction: A comparative review. Technical Report TiCC-TR 2009-005, Tilburg University, 2009.

[169] W. Van der Mark and D. M. Gavrila. Real-time dense stereo for intelligent vehicles. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 7(1):38–50, 2006.

[170] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *Int. Journal of Computer Vision (IJCV)*, 63(2):153 – 161, 2005.

[171] Xiaoyu W., T. X. Han, and Shuicheng Y. An HOG-LBP human detector with partial occlusion handling. In *Proc. of the Int. Conference on Computer Vision (ICCV)*, pages 32–39, 2009.

[172] C. Wakim, S. Capperon, and J. Oksman. A Markovian model of pedestrian behavior. In *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, pages 4028–4033, 2004.

[173] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1441–1448. MIT Press, 2006.

[174] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):283 – 298, 2008.

[175] X. Wang, T.X. Han, and S. Yan. A HOG-LBP human detector with partial occlusion handling. *Proc. of the Int. Conference on Computer Vision (ICCV)*, pages 32–39, 2009.

[176] A. Wedel, H. Badino, C. Rabe, H. Loose, U. Franke, and D. Cremers. B-spline modeling of road surfaces with an application to free-space estimation. *IEEE Trans. on Intelligent Transportation Systems*, 10(4):572–583, 2009.

[177] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers. Duality TV-L1 flow with fundamental matrix prior. In *Proc. of Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6, 2008.

[178] C. Wöhler and J. K. Anlauf. A time delay neural network algorithm for estimating image-pattern shape and motion. *Image and Vision Computing*, 17:281–294, 1999.

[179] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 794–801, 2009.

[180] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *Proc. of the Int. Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[181] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *Int. Journal of Computer Vision (IJCV)*, 75(2):247 – 266, 2007.

[182] B. Wu and R. Nevatia. Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[183] C. Wüst. Spiegel Online — Segensreicher Schlenker, August 2009. [Online; accessed 25-August-2013].

[184] L. Zhang, B. Wu, and R. Nevatia. Detection and tracking of multiple humans with extensive pose articulation. In *Proc. of the Int. Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[185] L. Zhao and C. Thorpe. Stereo and neural network-based pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 1(3):148–154, 2000.

[186] Q. Zhu, M. Yeh, K. Chen, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1491–1498, 2006.