CHALMERS

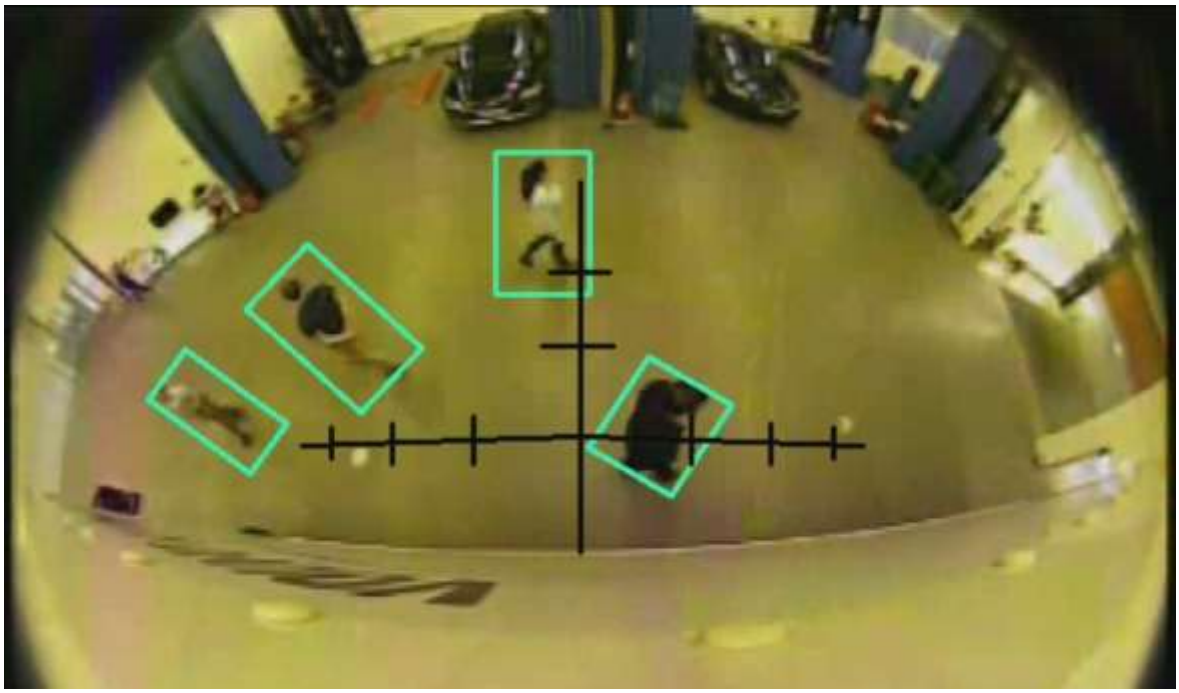


Image processing for pedestrian detection using a high mounted wide-angle camera

Master of Science Thesis in the Master's Degree Programme, Applied Physics

MARTIN LARS SVANTE JOHANSSON

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Division of Computing Science
Göteborg, Sweden, May 2014

Image processing for pedestrian detection using a high mounted wide-angle camera

Martin Lars Svante Johansson

Examiner: Devdatt Dubhashi

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover:
Detection output of four pedestrians by the side camera, presented in Chapter 10.6

Department of Computer Science and Engineering
Göteborg, Sweden May 2014

Image processing for pedestrian detection using a high mounted wide-angle camera
Master's Thesis in the *Master's programme Applied Physics*
Martin Lars Svante Johansson
Department of Computer Science and Engineering
Division of Computing Science
Chalmers University of Technology

Abstract

The purpose of this thesis has been to develop a automated pedestrian detection system for high mounted wide angle cameras on a Volvo truck, and prove its functionality for pedestrian detection in urban environments. Two cameras at different heights have been evaluated: one middle mounted forward facing camera; and one top mounted side facing camera. The thesis presents a framework of combining several independent algorithms fused into a unified decision for the presence of a pedestrian. For the forward facing camera a evaluation of image distortion was first made. Two background subtraction techniques were implemented and compared: the Mixture of Gaussians, and the Codebook method. Further, two object classification methods were compared: the Viola & Jones method where salient features of pedestrians are captured by an overcomplete set of Haar-like wavelet features and chosen by the gentle AdaBoost training algorithm; and the continuation of this method by Dollar *et al.*. To handle the in-plane distortion a rotational scheme was evaluated and setup that resulted in five different classifier regions.

The system was trained on pedestrian images captured around Lindholmen in Gothenburg, Sweden. Finally, tracking was incorporated in the form of Kalman filtering. For the side camera two new classifiers were trained based on the pedestrian bounding box viewing angle. An example interface was also developed that displays the final unified decision with a color bounding box warning system based on pedestrian proximity, the current active systems, a distance measure, and tracking history. Evaluation data was captured to test and verify the pedestrian detection and tracking method under normal urban environments. Experimental evaluation of the system on a conventional 2.16 GHz Intel Core2 CPU operating on 720×576 pixel images shows result of a robust detection and tracking system for pedestrians of different sizes, rotations and postures - with fast enough algorithms suitable for on-line operation.

Keywords: *Automotive safety, Computer Vision, Image Processing, Pedestrian detection, Wide-angle lens, Machine Learning, Classification,*

Acknowledgments

This master thesis has been carried out at Volvo Technology AB, Gothenburg, in cooperation with the department of Computer Science at Chalmers University of Technology. The thesis was performed between February and July 2012.

First and foremost, I would like to thank my supervisor at AB Volvo, Grant Grubb, for always taking the time to listen. A special thanks to Devdatt Dubhasi and Chiranjib Bhattacharyya for their support and insights as well as all the inspiring colleagues at Volvo.

Martin Lars Svante Johansson, Göteborg, August 2012.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

This chapter introduces the master's thesis by providing a general background of automotive safety. Further, the field of pedestrian detection is introduced and the research area motivated. This will be followed by a problem area section discussing practical and theoretical problems. The purpose and research questions of the master's thesis will follow accompanied by the delimitations. Finally, the outline of the thesis will conclude the chapter.

1.1 Background

1.1.1 Pedestrian detection

In the European Union the second source of traffic injuries and fatalities are pedestrian accidents. To detect pedestrians has therefore been the focus of a lot of recent research due to its importance within automotive safety to avoid such possible collisions. The major cause of these traffic accidents are due to human errors. This could be solved by introducing a pedestrian detection system that is able to analyze the vehicle surrounding and localize possible conflicts as to warn the driver for a possible collision. However, this has shown to be a complex problem due to: large pedestrian variability, dynamic environments, and real-time implementations. The field of machine learning has made these difficulties possible to overcome by using learning algorithms that learn from examples and allows avoiding the need for any hand-crafted solution. As a result, successful pedestrian detection systems have now been implemented in newer cars (for instance Volvo S60). However, larger commercial vehicles such as buses and trucks have not been researched for this implementation.

1.1.2 Large Vehicles and Automotive safety

Driven by increasing customer demand on safer vehicles for drives and other vulnerable road users (VRUs), the electronic vision systems market is rapidly growing. 94% of customers' regards safety in vehicles as a major concern shows a study conducted by the European New Car Assessment Program (Euro NCAP) [35]. There is also added demand with increasing legislations that emphasize on vision systems with the purpose to prevent fatalities of VRUs. For heavy goods vehicles there exists established driver assistance systems such as Adaptive Cruise Control or Lane Departure Warning for highway environments. But in the case of an urban environment the traffic situation is much more complex and additional driver support systems are needed.

A close investigation of truck accidents done by the Volvo Accident Research Team [88] revealed that the majority of truck accidents occur at low speeds and in the very near vicinity around the host vehicle - in the vehicles blind zone - and is due to a lack of visibility for the driver. A summary from the investigation that shows the most relevant accidents with VRUs can be seen in figure 1.1. Here the blind-zone refers to the area that the driver of the vehicle

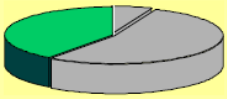
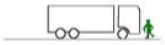
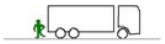
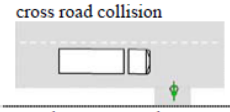
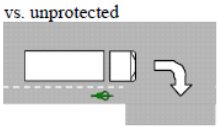
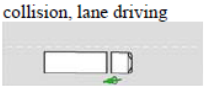
Unprotected 		Relative share	Description	Traffic environment	Speed	Typical cause
1.	Truck- unprotected collision, truck front vs. unprotected when taking off 	10%	Collision with unprotected, frontal part of truck in low speed manoeuvring or starting from stationary e.g. at crossroads or zebra crossings. (mostly pedestrians)	Urban areas, daylight.	Low	- Limited visibility; front of cab, right or left side of cab. - Limited driver knowledge of blind spots. - Lack of communication with other road user. - Driver stressed, inattentive or distracted.
2.	Truck- unprotected collision, truck vs. unprotected when reversing 	20%	Collision with unprotected, rear parts of truck/trailer in low speed reversing. Distribution trucks when delivering goods/ garbage collectors. (mostly pedestrians)	Urban areas, daylight. Most often elderly people, but also children	Low	- Limited visibility rear of truck. - External acoustic warning signal not enough. - Working routines not good enough. - Lack of knowledge. - Driver stressed, inattentive or distracted.
3.	Truck- unprotected collision, cross road collision 	20%	Collision with unprotected at intersection, moderate or high speed. (pedestrians, bicycles, mopeds)	Urban areas. The driver is often surprised and not prepared for the sudden situation.	High	Other road user: - Lack of judgement. - Misjudgement of speed of truck. - Drugs. - Truck speed above allowed.
4.	Truck- unprotected collision, truck side or front vs. unprotected 	20%	Collision with unprotected, most often right turn. (pedestrians, bicycles, mopeds)	Urban areas, low speed, narrow city streets, often a parallel bicycle lane or a zebra crossing.	Low	- Limited visibility side of truck. - Lack of knowledge about the blind spots. - Driver stressed, inattentive or distracted
5.	Truck- unprotected collision, lane driving 	10%	Collision with unprotected. Lane driving, e.g. lane change, merge, cut in, not keep in lane. (bicycles, mopeds, motorcycles)	Urban areas.	Me-dium	- Lack of visibility - Driver stressed, inattentive or distracted.
6	Other truck- unprotected collision,	20%				

Figure 1.1: Truck accident scenarios causing injuries to unprotected road users [88].

is incapable of seeing from a normal sitting position even when using the standard rear-view mirrors. In figure 1.2 a visualization of the front and side blind-zone area is seen for a standard Volvo truck.

It is therefor necessary to introduce an automatic detection system that observes the area in the very near vicinity of the truck, inhibiting the truck to drive in the case there is a pedestrian present. Given the size of the vehicle a higher mounted camera system is preferred as to cover a larger field of vision and eliminate the problem with occlusion of objects that frontal cameras suffer from. The occlusion problem is also so severe in crowded urban environments when using a frontal camera that there is no efficient tracking algorithm that can handle this even when using a multiple-camera approach. A normal camera lens has around 55° angular field of view, which is insufficient to cover the entire blind-zone for many LGV's [37]. Cameras utilizing wide angle lenses (also called fish-eye) is therefore required, having a 180° field of view or more.

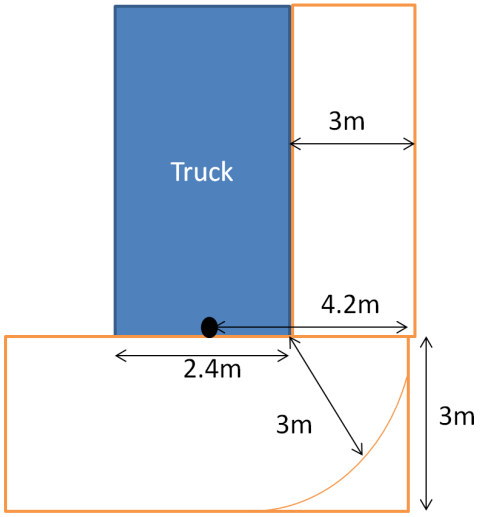


Figure 1.2: The front and side blind-zone area for a Volvo truck.

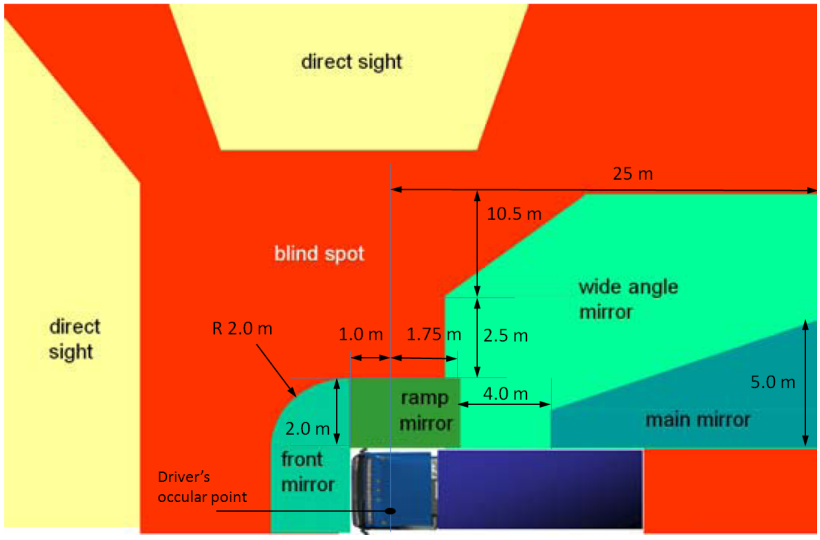


Figure 1.3: Blind-zone area for LGVs (left hand drive) that are required to be visible to the driver.

1.1.3 Legislations

As a result to prevent accidents the European Union introduced a legislation in the form of directive 2007/38/EC [20] (built upon Directive 2003/97/EC [19]) with requirements on the blind-zone area for an LGV that must now be visible to the driver with the use of indirect vision systems. The indirect vision system can be additional mirrors (internal or external) or more desirable a camera-monitor-device since it is almost impossible to cover the rear side of an LGV utilizing only a mirror system. The requirements can be seen in the visibility map in Figure 1.3. However, the indirect sight efficiency utilizing mirrors is limited and the wide angled side mirrors are very sensitive to miss-adjustments and will give a significant decreased observation area in the case of misalignment. Secondly, the mirrors are quite large and thus create an additional blind spot themselves. Lastly, to constantly observe all six mirrors, that also add distortion to all objects due to their wide-angle, is an almost impossible task for the driver, especially during complex traffic situations. Therefore, an automatic detection and warning system to aid the

driver in critical situations is greatly needed.

1.2 Challenges

1.2.1 Challenges for pedestrian protection systems

There has been an extensive amount of interest in the field of pedestrian detection over the past few years and covers a wide variety of applications such as surveillance, for instance a street being monitored, or intelligent vehicles with an on-board camera detecting possible collisions on the road ahead. From a human observer perspective the problem to identify a pedestrian from other objects, such as a garbage bin may seem simple, but from a machine vision perspective this is a difficult task due to the many parameters to take into account, such as clothing, pose, lighting, and background. The main challenges can be summarized as following:

- **Appearance variability** is very broad for pedestrians. A pedestrian is not a clearly specified term, for instance, the pedestrian appearance may differ in shape, color, pose, size, and include different accessories that complicates classification even further.
- **Outdoor urban environments** is where the detection and identification of a pedestrian is performed. This environment is usually heavily cluttered and complex with possible occlusion from other objects such as parked vehicles, which makes it difficult to segment a pedestrian from the background.
- **Dynamic scenes** with constant changing physical environments gives difficulties with different weather conditions that might decrease the quality of the sensed data in the form of lower contrast and shadows and changing illumination. This puts strict hardware requirements on the sensors being used.
- **Real-time** operation of the system is necessary to detect pedestrians fast enough so that the driver has time to react and the pedestrian is far ahead of the vehicle.

1.2.2 Challenges for a high mounted camera

Since the blind-zone is much larger for LGV's than for normal passenger cars the detection problem is inherently more difficult. One way to better capture the environment is to mount the camera in a higher position, and thereby capturing a bigger portion of the scene. Figure 1.4 shows the mounting position of the two cameras used in this master's thesis. However, there is a big difference between pedestrian detection using a horizontal view in comparison to a top-view. This is due to two main reasons:

1) Depending on the distance from the camera the aspect ratio of a pedestrian will change dramatically, and standard machine learning segmentation algorithms may not work correctly.

2) One can extract fewer features from the top-view compared to the horizontal view and it is thus harder to distinguish a human and a non-human. From a top-view only features such as the head and shoulder regions can be extracted.

1.2.3 Challenges for a wide-angled camera

For a wide-area coverage of the entire blind zone with only one camera the proposed camera system to be investigated in this master's thesis uses a 185° wide angled fish-eye lens. To add

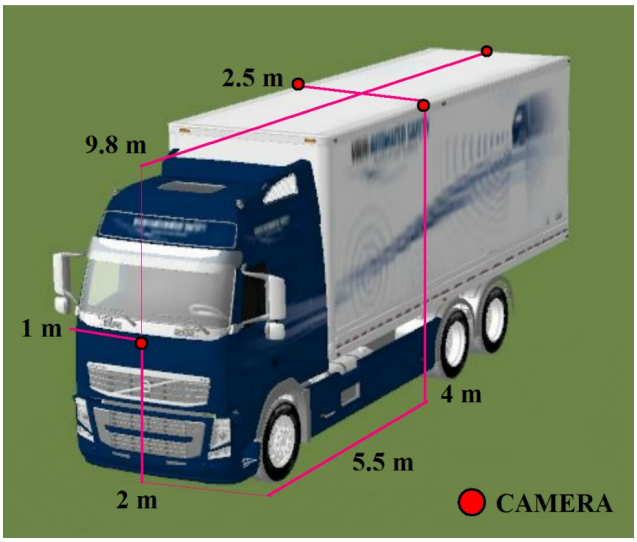


Figure 1.4: The positioning of the cameras on the truck.

further to the problem this camera type contributes a significant distortion to the image, and might need to be rectified or restored in order to analyze the image correctly.

In figure 1.5 we see an image captured with the camera system. The first thing to notice is the rotation of objects from the vertical line, as shown for the two pedestrians in figure 1.5, that can be as severe as $\pm 90°$. The second thing to notice is the horizontal and vertical bending of an object caused by the radial distortion, highlighted with fuchsia in Figure 1.5. Pedestrians most often occur as long vertical objects and are thus sensitive to horizontal bending, as also depicted in Figure 1.5 for the rightmost pedestrian, in contrast to the parking assistance lines or the edge of the truck which has a sensitivity to vertical bending. Overall this is an unstudied area and non-standard algorithms for pedestrian detection need to be considered.

1.3 Pedestrian Detection System Requirements

Given the problems discussed in the previous sections we can now formulate the four main criterions for a reliable pedestrian detection system.

1) **Real-time requirement:** If the system is not able to detect a pedestrian in time it will be of no use. This puts a constraint on the algorithms utilized to be of low complexity.

2) **Robustness:** All pedestrians in the very near vicinity of the vehicle needs to be detected, and false positives kept at a minimum. The system also needs to be able to detect pedestrians that are not in the direct path of the vehicle, but might be if the pedestrian steps out into the road.

3) **Variability:** Regardless of the size, number of accessories worn, or pose of the pedestrian, the system should be able to make a detection.

4) **Environment:** The system should be able to make a detection regardless of the surrounding scenery as well as the environmental condition. This includes handling different illumination conditions caused by clouds, heavy rain, or a backlight sky from a low standing sun. The system should also be able to handle all different light conditions that occur during a day from early morning to late night.

Figure 1.5: A captured frame from the camera system that displays the vertical rotation (highlighted in black) and the significant non-rectilinearity of objects (highlighted in fuchsia) caused by radial distortion.

1.4 Purpose and Research Questions

The main purpose of this master's thesis is to adapt and/or modify existing state-of-the-art road user detection algorithms to larger commercial vehicles that use a high mounted wide-angle lens (fish-eye) camera and thus having a top-view perspective. Given the described background and the highlighted problems in the challenges section the following research questions (RQs) are derived:

Main RQ: What are the considerations and challenges when using a top-view approach together with a wide-angle lens camera for pedestrian detection, and what machine learning and computer vision algorithms can be used to implement such an approach?

RQ 1) What are the current state-of-the-art pedestrian detection techniques?

RQ 2) What are the current state-of-the-art detection implementation techniques when using a "top-view" approach and what are the challenges and limitations?

RQ 3) What are the current state-of-the-art image analysis methods when using a wide angled lens within the field of automatic detection?

RQ 4) Which (machine learning) algorithms are suitable and realistic to be adapted for an implementation?

1.5 Delimitations

The scope of the master's thesis is to give a proof-of-concept of a pedestrian detection system using high mounted wide-angled cameras situated at two different heights: in the front and on

the side of a truck suggested by Volvo. Since this is a proof-of-concept the requirements to solve the problems described in section 1.2 will be low. The requirements that this master's thesis will focus on are:

- **Real-time.** To be able to detect pedestrian the system must run in real-time when deployed in a vehicle. So the algorithms presented in this thesis will have a limited complexity and be real-time algorithms. The developed system will however operate on pre-recorded videos and not in real-time
- **Variability.** The research will only consider detection of pedestrians and no other vulnerable road users such as bicyclists or motorcyclists. The variability requirement is mitigated to include various clothings and the most common poses; standing, walking, and running, excluding any accessories.
- **Lighting.** The system should be able to detect in normal day-time with even illumination lighting conditions. This constrains the problem to not cover night time, fast illumination changes, or extreme weather conditions.
- **Detection Range.** Since other sensors can detect a pedestrian from 5 meters and upwards, the system will cover the blind-zone region as described in Figure 1.2.
- **Scenarios.** The detection scenarios focuses on the most common accidents situations that occur, as described in figure 1.1. The scenarios are limited to a stationary vehicle, a moving vehicle in speeds up to 30 km/h, and start inhibit. The front camera will only focus on single pedestrians, while the side camera will deal with multiple pedestrians. A normal urban environment street will be used for the test scenarios.

1.6 Thesis outline

The chapters for this thesis are organized as follows:

Chapter 1 introduced a brief background of automotive safety and described the pedestrian detection problem. It then continued with the overall goals and delimitations of the thesis.
Chapter 2 consists of a survey of previous work and state of the art in object detection that will cover the main components involved in a monocular pedestrian detection system: camera calibration, how to generate the hypothesis (region of interest selection), the object detection, object classification, and finally tracking. The second part of the survey contains a study of the state-of-the-art systems with test criteria and data sets.
Chapter 3 describes a high-level overview and motivates the chosen approaches to pedestrian detection for experimental evaluation. No implementation level details are given but the overall detection framework is described.
Chapter ?? presents the camera calibration and undistortion theory with a method for solving for wide angle cameras.
Chapter 5 introduces the foreground-background subtraction algorithm theory of Mixture of Gaussians and the Codebook as object detection methods. Further, a connected component analysis is described.
Chapter 6 presents the theory of the two object classification methods used in the thesis. First, a detailed description of the feature extraction, the AdaBoost training algorithm, and the cascaded classifier. Following, is a description of the further developed generalized version of the same method.
Chapter 7 describes the Kalman filter tracking procedure.

Chapter 8 presents the performance evaluation methodology together with a description of the ROC-curve .
Chapter 9 describes the methods and procedures for the implementation. The test problem is specified and the performance profile is defined.
Chapter 10 gives the experimental results from the test sequences for the implemented algorithms, and the accuracy of the final system. The structure is such that, first a comparison between different methods are done, and the best performing methods are tested against a wide variety of different traffic scenarios.
Chapter 11 provides a discussion of the limitations and advantages of the proposed methods and algorithms. Further, a suggested direction for future research is provided
Chapter 12 summaries the master's thesis and concludes the approaches and obtained results.

Chapter 2

Survey and State Of the Art

This chapter describes the main components of a pedestrian detection system - the camera model, ROI selection, object classification, and tracking. Then a review is made of the current state-of-the-art, datasets and performance evaluation. The focus will be on computer vision algorithms related to the camera system, thus monocular cameras for rotated pedestrian detection using a sliding window approach.

2.1 Overview

In the last decade there has been a significant progress within pedestrian detection, but it is still many magnitudes away from the desired performance for most applications. As described in Chapter 1.2 there are numerous challenges within the area. This chapter will review the current state of the art in automatic object detection and localization, focused on pedestrian detection.

The problem to automatically detect a pedestrian in a video sequence can be divided into three parts: generating an initial hypotheses, also called the region of interest (ROI), and isolating foreground objects; classifying them as pedestrian or non-pedestrian for verification; and temporal integration in the form of tracking. A real detection system may also involve a pre-processing step with task such as gain adjustment, exposure time, or camera calibration, and a final high-level decision step based on the detection information and incorporates the field of human-machine interaction. A schematic overview of this pedestrian detection architecture is visualized in figure 2.1. Furthermore, different techniques within object detection can be classified according to how they treat the following

- Generation of an initial region of interest object hypothesis
- Descriptors and features that are extracted from the input
- Classifier used to evaluate the features

The survey will for practical reasons be limited to include the most influential work and those that are most relevant to the camera system to be used. This means a monocular camera system as a detector operating in the visible spectrum as a sensor in low to medium resolution, and the sliding window approach will be focused since it is the most promising for these conditions. The tracking methods will be constrained to methods based on objects and motion representations. Numerous amount of related surveys already exist with a similar focus, for a more extensive review and detailed implementations the reader is directed to the reference section [15, 17, 26, 31, 37].

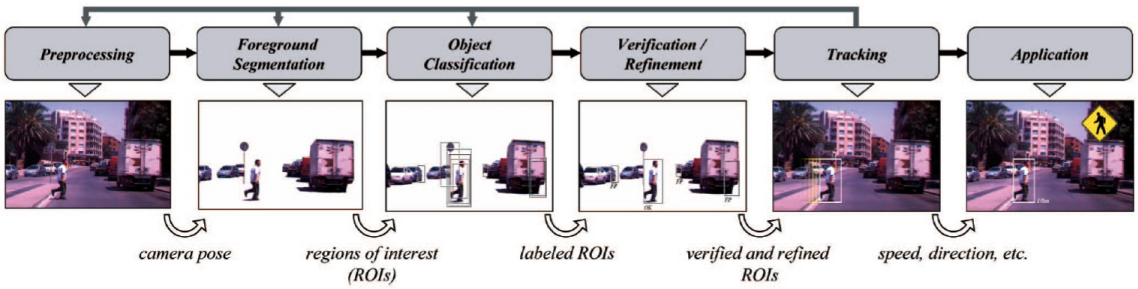


Figure 2.1: Proposed architecture for an on-board pedestrian detection system [31]. The gray arrows represent potential feedback processes between the blocks.

2.2 Dioptric Cameras (Preprocessing)

2.2.1 Dioptric Camera Models

Due to the miniaturization and decreasing market price catadioptric, dioptric, and polydioptric omnidirectional cameras have gained popularity in a wide variety of areas such as robotic vision tasks, surveillance, and automotive application due to their large field of view. In this thesis a dioptric camera is used with a wide-angle lens (also named fisheye lens). The drawback of this camera lens type is its inherent tangential distortion, radial distortion, and uneven illumination in the image that makes it difficult for pedestrian detection using the standard classification methods.

To compensate for the distortion a post-processing of the captured image is carried out. A distortion model is assumed and the model parameters are estimated from a calibration step. There are many proposed models for the fisheye camera compared to the central catadioptric camera where a unified model exist [69]. The radial distortion models can broadly be divided into three main areas for fisheye cameras: the pinhole model, the captured rays model, and the unified catadioptric model - where the first two can be discarded for this thesis. For the latter, Scaramuzza *et al.* [71, 72] proposed a general model in 2006 called the Taylor model that combines the central catadioptric camera and wide-angle camera. It has the benefit that both dioptric and catadioptric cameras are described using a Taylor polynomial.

2.2.2 Camera Calibration

When using a camera a calibration step is necessary. There are two camera calibration branches: automatic calibration, and photogrammetric calibration. Most commonly used are planar grids that are captured at different orientations and positions. There exist a few methods that handle self-calibration of both intrinsic and extrinsic parameters [71, 73]. A more common approach, known as the camera pose estimation, is to assume the intrinsic parameters to be constant and only make an initial computation and then have a continuously updating for the extrinsic parameters.

To initially calculate the intrinsic parameters and make the assumption that they remain constant, and let the extrinsic parameters be updated continuously, is the most widely used method commonly called camera pose estimation. This method is then divided into the two subgroups monocular-based and stereo-based, where the base of the monocular approach is to study visual features.

2.3 Region of Interest Selection

The Region Of Interest (ROI) step, also called foreground segmentation, restricts the scene sent to the classification module. This is an important step as to lower the number of candidates by avoiding evaluating non-important regions such as the sky.

To obtain an initial region of interest hypotheses for an object, the simplest technique is the sliding window approach, where a window for detection is shifted over all locations in the image at various scales and selects candidates according to the pedestrian size constraints. Since an image can contain thousands of detection windows, this process often have a too high computational cost to allow real-time processing [8, 9, 56, 65, 70, 82]. A significant speedup can be obtained if a classifier cascade of increasing complexity is coupled with the sliding window method [55, 61, 77, 83, 87, 92, 96].

Another approach, if one has prior knowledge of the target object class and the camera geometry, is to limit the search space in the image [18, 30, 43, 58, 76, 95]. This is done based on, for example, the flat-world assumption or the object aspect ratio, named the pedestrian size constraints. The pedestrian size constraints have a certain threshold that the ROIs must fulfill to be considered as a pedestrian candidate, which can significantly reduce the computational time needed.

Further techniques to generate an initial ROI hypotheses is to use object motion as a cue, which will be further described in detail in section 2.4.3. This is widely used for static cameras within surveillance after a background subtraction step [59, 79, 95].

2.4 Object Detection

The different techniques for object detection related to our pedestrian detection problem can roughly be divided into three categories: feature based, template based, and motion based. The entire taxonomy is depicted in Figure 2.2. The pros and cons of each area will now be further described.

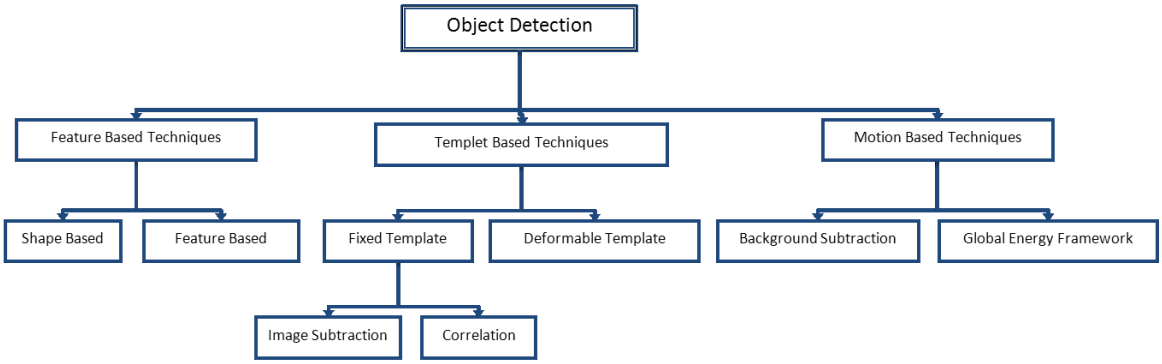


Figure 2.2: Taxonomy over object detection methods [1].

2.4.1 Feature based

The first step in these algorithms is to extract features, and then to classify the features from a trained recognition system. The ideal feature detector should be able to find features on an object that are independent of changes in lighting, perspective, and scale. The feature selection step is an important step since the rest of the detection algorithm is dependent solely on these features [21]. The main challenge of feature based algorithms is the feature selection, where

different approaches have different drawbacks. For instance, color based features must be able to handle all color variations in clothing, while shape based features must be able to handle the variations in poses and additional accessories. In general these approaches are usually computationally expensive. A brief description of the most commonly used features will now follow.

Point Detectors

There are numerous techniques to get a sparse representation of an image by using salient points. The aim of point based detection is to find a generalized detection system given various local image points, called *key points*, and create a feature vector based on these key points to use for object detection. Some commonly used key point detectors are the Harris corner detector [33], Harris-Laplace [54], or the Difference of Gaussians [48].

One of the most common and most popular approaches is the Scale-Invariant Feature Transform (SIFT) image based descriptor [47, 48]. In the SIFT method the image content is transformed into local feature coordinates and uses the dominant orientation and local scale obtained from the key point detector to vote into orientation histograms with weighting based on gradient magnitude, and thus attaining translation, rotation, and scale invariance for the SIFT descriptor - visualized in Figure 2.3.

The histogram of oriented gradient (HOG) features introduced by Dalal & Triggs [7–9] was popularized due to its substantial improvements over intensity based features. This was built upon by Zhu *et al.* [96] with the introduction of integral histograms resulting in a speed-up. With its low false-positive ratio it is still one of the most popular pedestrian detection methods. But at its disadvantage is a higher computational cost, with a real-time feature loss.

Sparse key point detectors have the advantage of a compact representation that gives a speed up in the classification process. However, there is a limitation in generalizing to object classes or categories.

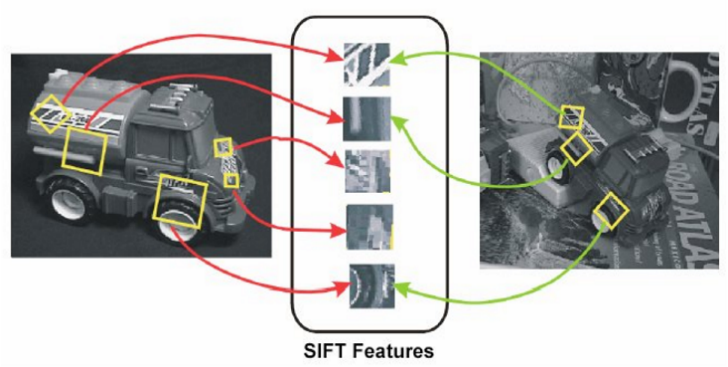


Figure 2.3: An example of the SIFT method showing corresponding features for an object in a scene that differs in orientation and translation.

Part and Limb Detectors

For part-based object detection the pedestrian shape is decomposed into semantically meaningful parts (legs, torso, or head) [25,26]. The spatial relations are restrained between parts, and integrate the responses from each local part into a final detection [5]. A discriminative classifier is trained for each part and a larger model is used to represent the geometric relations between the parts.

Wavelet Based Detectors

One common feature set for many detectors evaluates image regions with operators similar to Haar wavelets. Papageorgio & Poggio [65, 66] popularized the over-complete dictionary of multiscale nonadaptive Haar wavelet features that have been frequently adapted by many others [56, 78, 86, 87]. The overcomplete features are calculated by taking the difference between sums of intensity values for pixel regions at different orientations, scales, and positions. The features are simple and extremely fast evaluated due to the integral image technique used [46, 87].

These over-complete Haar-like features was extended in 2002 by Lienhart & Maydt [46] to include features rotated by 45°. Like the original features, these new features can also be rapidly computed with the integral image method. This made it possible to capture new domain knowledge much better, for example the arms of a walking pedestrian, but with the drawback of a slower, over-fit, and increased complexity in the training process of the classifier due to the additional features. To further improve detection accuracy Viola & Jones also added temporal information to their person detector [64].

Edge and Gradient Based Detectors

Other techniques model the local edge structure by calculating discontinuities in the image brightness function. Gavrila and Philomin [29] popularized a pedestrian detection system based on extracted edge images and matched these by using the chamfer distance to a set of learned examples. This has later been extended to a practical real-time pedestrian detection system [28].

Shape Models.

By using shape cues one can eliminate the need to consider clothing and lighting properties of the pedestrian appearance. Real-time matching algorithms has been realized with space-transforms combined with pre-computed hierarchical models [27, 30]. To model the 2D pedestrian shape space (disregarding more complex 3D human shape models) both continuous and discrete approaches have been proposed. A discrete representation of the shape is based on a set of exemplar shapes [22,23,67,70]. The drawbacks of this method is an increase in false alarms when non-pedestrian objects appear/disappear in a scene.

2.4.2 Template Based

Template based detection is the technique to find features in a image that match a template image, see Figure 2.4. A simple description of the template matching technique is to represent the image as a bi-dimensional array of intensity values that is compared to a template representing the object using a suitable metric (such as the Euclidean distance).

To learn the features there is a initialization process where the algorithm sweeps the frame and tries to find the matching features. Occlusion is possible to detect and appear as missing template features in the frame, however they are not detected while occluded but as they reappear in the image. These algorithms are less suitable for multiple object tracking [85]. The drawback of this type of detection algorithms is that the computational complexity is often high, and occlusion of objects is a challenge.

2.4.3 Motion Based

Background subtraction is the process to detect a moving object in a video frame based on its deviation from a reference background frame. A large enough deviation indicates a moving object.

Figure 2.4: An example of template based detection.

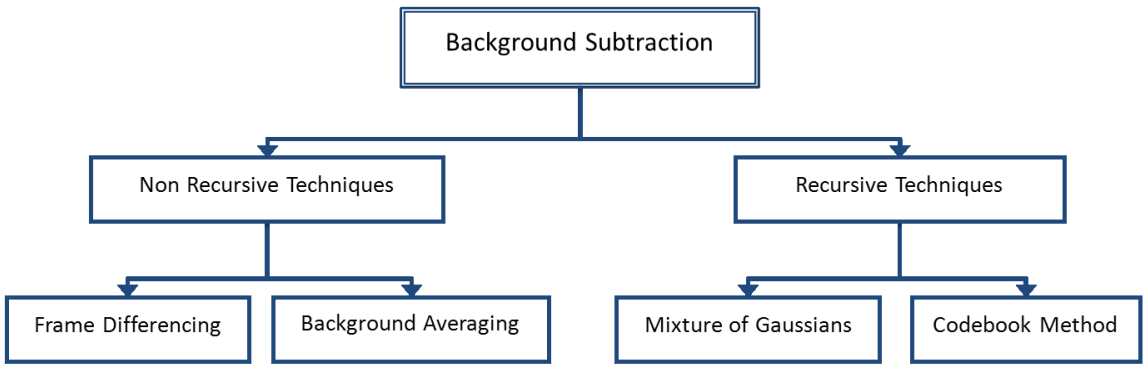


Figure 2.5: Taxonomy over background subtraction methods [1].

Background subtraction using a fixed camera for pedestrian detection, as in the field of surveillance, is the most effective and common method of segmenting foreground objects from a background. The basics of background subtraction can be described using four principles:

1) Background subtraction should segment objects of interest as they appear in a scene

2) Pixels that satisfy a certain stationary criterion should be declared as background and ignored

3) The background model must be able to adapt to gradual as well as sudden changes in the background

4) A background model should be able to take into account changes in different spatial scales.

The major four steps in a background subtraction algorithm are: preprocessing, background modeling, foreground detection, and data validation. This is done by storing a background model and locating the areas in a new frame image which has a sufficient difference from the background given a threshold value.

However, this has obvious drawbacks for pedestrian detection from a automobile since the moving vehicle gives a constantly changing background. So, for a camera mounted on a moving car, motion-based pedestrian detection algorithms does not work as the main detection method. Generalization to camera motion is possible, assuming only translatory motion, by calculating the optical flow deviation from the expected flow field of the ego-motion [18, 67]. Another solution to this has been tried by interpolating motion data between frames to evaluate subsequent background frames from their motion patterns [34] and [87]. Still, the drawback is a several frames delay before any detection is possible.

Background subtraction algorithms can roughly be divided into two categories: non-recursive and recursive, as seen in Figure 2.5. The simplest method for background subtraction is to take a naive frame differencing, where the last previous frame will represent the background model.

The segmentation is done by comparing the color value difference between the previous and current frame and compare it to a threshold value. This is a memory inexpensive method, but will not work well in scenes with changing backgrounds, or when the object stops moving. For the second category the scene history is taken into account and a more dynamic background model is learned.

The other approach is to use a recursive learning of the background by multimodal probability density functions. The intensity value for each pixel can be represented by a Gaussian mixture model, and a simple heuristic will decide which intensities belong to the background. The pixels that do not correspond to the background intensity will be labeled as foreground. A common method was developed by Stauffer and Grimson where each pixel is modeled with a mixture of K gaussian [80]. The codebook method developed by Kim *et al.* is another multimodal background approach [40]. A codebook stores a series of key color values, or codewords, for each background pixel given a training sequence. The codewords will then determine what color each pixel will have over a certain period of time and can from that decide the background pixels. The advantage for this method is that it can handle scenes that have moving backgrounds and varying illumination (such as highlights and shadows). For the recursive methods the foreground pixels will be combined in a final step to object shapes by a 2D connected component analysis.

2.5 Classification Methods

Once the initial object hypotheses set has been obtained with any of the aforementioned detection methods, these need to be classified as either pedestrian or non-pedestrian by some classifier model. Classifiers can be divided into two broad categories of learning models [84]: discriminative approaches, such as Support Vector Machines (SVMs) that model the decision boundary between object and non-object classes given training examples; and generative approaches, such as graphical models where the pedestrian appearance is modeled in terms of its class-conditional density function.

2.5.1 Discriminative Models

For the discriminative classification techniques, the goal is to separate pattern classes in a feature space by a decision boundary. These technique have become popular due to their superior performance to automatically select relevant features from large feature sets. A description of the most common architectures will now be presented.

Support Vector Machines

Support Vector Machines are one of the most popular methods to build a discriminate classifier for object detection [74]. A separating boundary is found between the two object classes so that it attains the maximum margin in the feature space. The kernel employed during training implicitly determines this feature space.

Dalal *et al.* used linear SVMS for classification of dense features computed with gradient orientation histograms [8, 9] . Using non-linear SVMs as classifiers with polynomial or radial basis functions to map the input features to higher dimensional spaces has shown to increase detection quality [6, 65]. However, compared to using linear kernels these give a higher cost of computation.

Cascaded AdaBoost

Boosting is a powerful method to form discriminative classifiers [56]. A linear combination of weak classifiers is generated, by sequentially choosing the classifier that minimizes the weighted error on the training set, and finally produce a strong classifier. The boosting method is often used to build a cascade of classifiers, where each stage have a strong classifier, that give a high performance, rapid evaluation, and serves as the base for many modern detectors.

Viola & Jones [86] built upon the ideas proposed by Papageorgiou *et al.* [65] and proposed an algorithm in 2001 for boosted cascades with Haar-like wavelets capable of achieving a rapid image processing of 15 frames per second by introducing integral images. The algorithm used for training was based upon AdaBoost for automatic feature selection. Zhu *et al.* [96] used weak classifiers as linear SVMs of gradient orientation histogram blocks together with AdaBoost to train a strong classifier cascade. Viola *et al.* [87] further built on the boosting detector and combined the wavelets with motion information.

Neural Network

Multiple layers of neurons have also been used to create a decision boundary between classes, such as feed-forward multilayer neural networks [39]. In the area of pedestrian detection, multilayer neural networks have been utilized particularly together with adaptive local receptive field features as nonlinearities in the hidden network layer [25, 30, 57, 82, 91]. Classification and feature extraction is combined with a single model with this architecture.

2.5.2 Generative Models

In a generative classifier the joint distribution $p(x,y)$ is learned and using Bayes rule to calculate $p(y|x)$ to predict the label y which is most likely, in contrast to discriminative classifier that learns the conditional probability distribution $p(x|y)$. Commonly used algorithms are Linear Discriminant Analysis (LDA), and the Naive Bayes classifier.

2.6 Multiple Orientation Detection

The cascaded Haar Classifier methods described above have been optimized to detect objects with similar rotations as the objects used during training. As discussed in chapter 1.2.3, pedestrians can occur in a lot of different poses and rotations given a wide-angle camera. The rotations can be divided into two categories: in-plane rotation, and out-of-plane rotation. To train a single classifier for all different orientation variations and poses is possible, but would result in a low performance. Viola & Jones [64] showed that training a classifier for all poses is "hopelessly inaccurate". They could however detect faces at different orientations by training 12 different classifiers, all with a specific in-plane face orientation between 0° and 330° and perturbed randomly by ±15° (a range of 30°), and then combining their outputs. To increase speed, but resulting in a lower accuracy, they trained a decision tree to find the best possible orientation for each candidate region. Kölsch and Turk [42] investigated this further for hand detection and observed an efficient detection for one detector for about 15° of rotations, and stressed the importance of rotated example images within those rotation limits for the training data.

Rowley *et al.* [68] successfully used neural networks for rotated face detection. The first network matched the most suitable face detection orientation for a region, which was rotated according to that orientation and passed to a second normal detection network for upright faces. The latter network used faces over a range of 20° (±10°) during training and showed successful face detection within the same range.

Messom & Barcak [53] used another approach to train a single cascade that is rotated to its closest operational range. Previous methods have a wide spread for the integral images of 0° and 45°, but they proposed 26.5° and 63.5° integral image arrays. The results for the rotated cascades were accurate but did not perform as well as the original classifier. Horton extended these methods for rotational detection of sea creatures [36].

It has been shown that training multiple classifiers to handle the different rotations is a sub-optimal approach compared to training a single classifier and applying this to different rotations of the image [36, 64].

2.7 Object Tracking

A complete pedestrian detection system also needs to include a tracking system to predict and analyze dynamics and behaviors. Real-time tracking of objects can be found in various of applications: traffic monitoring, monitoring and surveillance, vehicle navigation, and motion-based recognition. The goal of an object tracker is to produce a trajectory over time by localizing the object position in all frames. An object tracker could also give the objects occupied region in a frame at every time instant. A complete pedestrian detection system should incorporate tracking after the detection has been made to predict collision possibilities; or if there is missed detection, a temporal trajectory of a pedestrian. Tracking visual features is an important task within computer vision but it is a difficult task, especially in a complex environment. Challenges are: abrupt changes in the object motion, appearance pattern of the object and scene, occlusion, and camera motion.

To detect and create a correspondence for an object between frames can be performed in two ways: separately or jointly. In the separate method an object detection system deals with the detection of the object frame-by frame and connects the object across frames. In the joint method an iterative updating of the object position and information of the region from previous frames determines the correspondence and the object region. In figure 2.6 the different tracking categories are depicted and will be presented in the following sections.

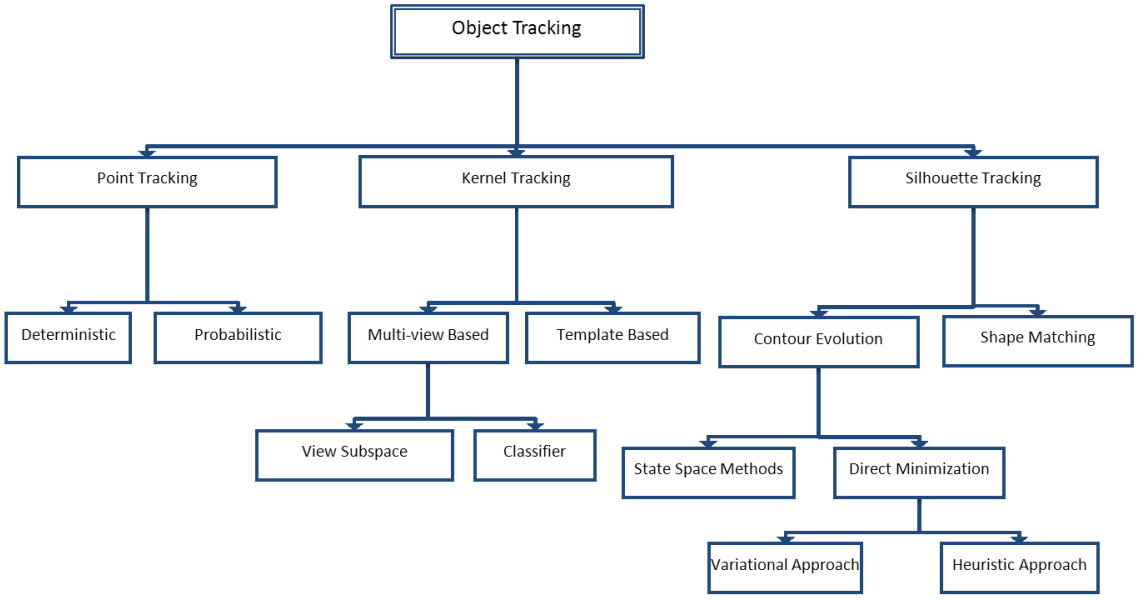


Figure 2.6: Tracking methods taxonomy [93].

2.7.1 Point Tracking

For these algorithms the object is represented as a point and the association of points is given from the objects previous position and motion state. This method requires another system to detect the object in every frame.

2.7.2 Kernel Tracking

Kernel refers to the object shape and appearance. For example, the kernel can be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames [93].

2.7.3 Silhouette Tracking

For silhouette tracking methods the object region is estimated for each frame and tracked. The information in the object region is used, such as the appearance density and shape models in the form of edge maps. The silhouettes are then tracked by shape matching or contour evolution from the object models. These are essentially object segmentation methods for the temporal domain utilizing priors from previous frames.

2.8 Benchmarking

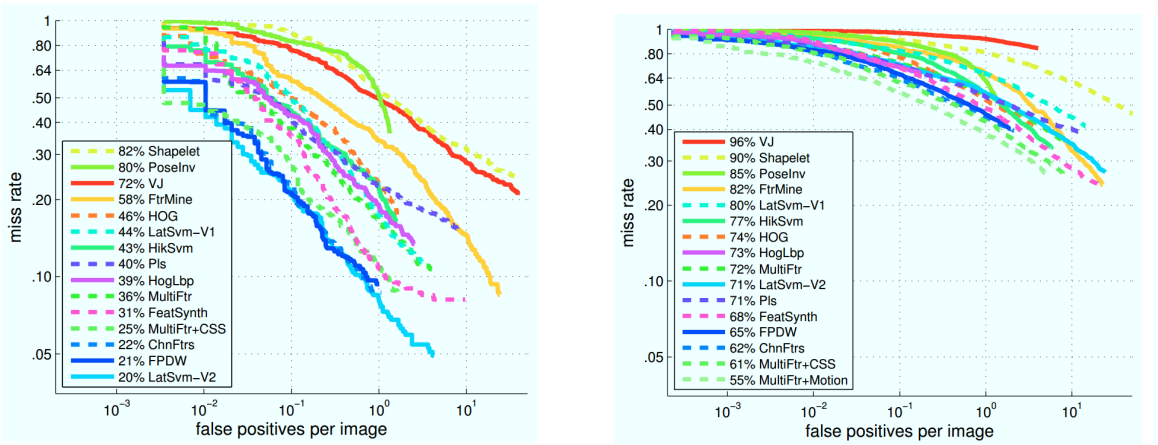
In this section a benchmark comparison is presented of the performance for the sixteen most common pre-trained pedestrian detectors using monocular images with a focus on sliding window approaches. Note that this is not an exhaustive list.

In Figure 2.7 the detector performance results are shown for the INRIA [8] and Caltech [15] datasets. Performance is better on the INRIA datasets, which contains grayscale, static high resolution pedestrian images, with CHNFTRS and its continuation FPDW scoring a log-average miss rate of 22-22%. The Caltech dataset is more challenging, with a resulting log-average miss rate between 51-55%. The ranking of the detector has been seen to be consistent across datasets, meaning that evaluation is independent on the dataset used.

In Figure 2.8 a runtime analysis is presented for the detectors on the Caltech dataset on 640 $\times$ 480 pixel images for pedestrians over 100 pixels. Detector speeds show a range between .02 fps to 7 fps achieved by FPDW (a faster version of CHNFTRS). No correlation between runtime and accuracy is seen. Some frame rates might seem low; note however that all detectors can be implemented as part of a full system containing ground plane constraints and region-of-interest selection that will reduce the runtime drastically.

2.9 Performance Evaluation and Datasets

To measure performance of a pedestrian detection system is non-obvious as no standardized test exists. One can separate the most commonly used approaches to characterize the performance of a specific object detection algorithm into two methods: pixel-based methods, and template/object-based methods. The pixel based methods assumes a detection of all the active pixels in a image, while the object detection methods can be formulated as a classic binary detection problem, thus a set of independent pixel detection problems. The Receiver Operating Characteristics (ROC) curve is commonly used to show the performance ratio of a system. It displays the number of correctly identified pedestrians (true positives) versus the falsely identified pedestrians (false negatives).



(a) INRIA dataset. **(b)** Caltech dataset.

Figure 2.7: Detector accuracy on the (a) INRIA and (b) Caltech datasets under the reasonable evaluation setting [15].

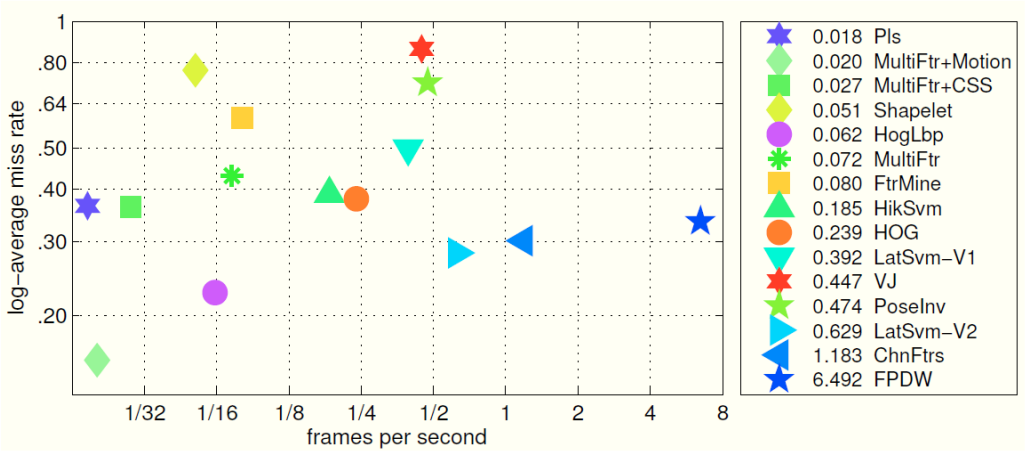


Figure 2.8: Log-average miss rate versus runtime (frames per second) for each detector on 640 × 480 pixel images ([14, 15].

Even though there are many proposed techniques in the general architecture of the pedestrian detection systems, the experimental side suffers from a lack of attention. The performance reported can differ in magnitude by several orders (within the same study) [64]. This is due to the variety of image data used, the different size of the test data sets, and unsatisfactory evaluation criteria (cover area, tolerance, etc.).

One can group current datasets into two categories: 1) "person" datasets covering a large number of different domains with unconstrained poses for a person and 2) "pedestrian" datasets with upright people possibly in motion. In figure 2.9 a overview of current pedestrian datasets is presented. It is organized into three sections, where the first one contains older or more narrow datasets, the second contains more extensive datasets, and the third contains the Caltech Pedestrian Dataset, a more comprehensive dataset that also includes occlusion data.

		Training			Testing			Height			Properties						
	imaging setup	# pedestrians	# neg. images	# pos. images	# pedestrians	# neg. images	# pos. images	10% quantile	median	90% quantile	color images	per-image eval.	no select. bias	video seqs.	temporal corr.	occlusion labels	publication
MIT	photo	924	–	–	–	–	–	128	128	128	✓						2000
USC-A	photo	–	–	–	313	–	205	70	98	133		✓					2005
USC-B	surv.	–	–	–	271	–	54	63	90	126		✓					2005
USC-C	photo	–	–	–	232	–	100	74	108	145		✓					2007
CVC	mobile	1000	6175†	–	–	–	–	46	83	164	✓		✓				2007
TUD-det	mobile	400	–	400	311	–	250	133	218	278	✓	✓					2008
Daimler-CB	mobile	2.4k	15k†	–	1.6k	10k†	–	36	36	36			✓				2006
NICTA	mobile	18.7k	5.2k	–	6.9k	50k†	–	72	72	72	✓		✓				2008
INRIA	photo	1208	1218	614	566	453	288	139	279	456	✓						2005
ETH	mobile	2388	–	499	12k	–	1804	50	90	189	✓	✓	✓	✓			2007
TUD-Brussels	mobile	1776	218	1092	1498	–	508	40	66	112	✓	✓	✓				2009
Daimler-DB	mobile	15.6k	6.7k	–	56.5k	–	21.8k	21	47	84		✓		✓	✓		2009
Caltech	mobile	192k	61k	67k	155k	56k	65k	27	48	97	✓	✓	✓	✓	✓	✓	2009

Figure 2.9: An overview of existing datasets [15].

Chapter 3

Selected pedestrian detection approaches

This chapter presents and motivates the chosen approaches and algorithms used in the master's thesis.

There are of course numerous interesting approaches to the different problems raised, but this thesis will focus on the most widely used approaches to solve these.

A camera calibration will be performed to undistort the image and evaluate its efficiency. Only radial distortion will be covered, other geometric effects such as center of distortion, tangential distortion, and uneven illumination will not be considered in the calibration model. The omnidirectional camera calibration technique developed by Scaramuzza [71, 72] will be used since it is able to deal with the specific camera type used in this thesis. To perform the central omnidirectional camera calibration procedure a Matlab Toolbox will be used [73].

Motion will be used as a first cue for the existence of a pedestrian. This is a normal task when faced with a static background. Two common background subtraction techniques will be evaluated: the Mixture of Gaussian, and the Codebook method. However, when there is camera movement the background will be subjected to ego-motion, depending on the camera motion and scene structure. Hence, when the pedestrian is moving longitudinally and thus parallel to the ego-motion, these methods might fail to pick up on the motion cues. The Mixture of Gaussian method will therefor also be evaluated for a moving camera to see if it is possible to adapt quickly enough to the scene changes to detect pedestrians.

A machine learning approach will be pursued to capture both stationary and moving ego-vehicle detection. Due to its simplicity, robustness, and high speed - a 95% detection accuracy at around 17fps - Lienhart *et al.*'s extension [44–46] of the Viola & Jones Haar wavelet-based cascade classifier architecture [86] will be trained with the AdaBoost learning algorithm, and tested utilizing a sliding window approach. This approach will also be used to obtain a baseline for further development and evaluation. To capture the state-of-the-art detection performance, and due to its close implementation resemblance of the Viola & Jones detector, the multiscale detector developed by Dollar *et al.* [12] will be implemented.

A temporal integration with a 2D bounding box tracker in the form of the Kalman tracking algorithm will be implemented due to its simplicity, low computational cost, and inherent smoothing.

A performance evaluation framework and setup will be explored. Since we are not interested in the point target detection but rather the object regions, the pixel-based methods are of little use, and a adaptive rotated and scalable bounding box will be used to region match hand labeled annotated ground truth data. The ROC curve will be used to evaluate the classifier efficiency.

The Daimler-DB dataset was chosen since the data was collected from a mobile record-

ing setup, which has varying scenery, without constraint on illumination, poses and limits the selection bias. It contains fully visible pedestrians in a upright position with 15,660 positive training examples, and 6,744 full negative image samples. It is a classification dataset containing cropped pedestrian windows which makes it useful for training. This data set was chosen due to its complexity in dynamically changing backgrounds and overall realism which grants a robust classifier.

Chapter 4

Camera Calibration

In this chapter the unified Taylor model for the calibration and undistorting of the fisheye camera will be presented.

4.1 Background

There are two primary distortion types caused by a wide-angle camera: radial distortion, and tangential distortion. The major distortion originates from radial distortion, which is the only distortion that will be covered in this thesis as the tangential distortion is negligible [10, 49].

To define how real-world objects will be projected on the image plane the camera uses parameters, where the most important are: the possible displacement from the optic axis to the image center, and the focal length all in x, y direction. These parameters can be summarized in the intern camera matrix. Instead of choosing the coordinate system with origo at the cameras center of projection but a real-world coordinate system we get the extern camera matrix, which specifies the transformation between the two coordinate systems. This latter matrix specifies rotations in x, y, and z direction, as well as translation.

4.2 Calibration Theory

In this section the calibration model will be described, which is based upon the omnidirectional camera calibration technique presented in 2006 and 2008 by Scaramuzza *et al* [71, 72]. This techniques is a unified model for catadiptric and wide-angle cameras and is therefore justified to be used.

To calibrate a wide-angle camera successfully one wants to find how the 2D pixel point p with coordinates (u,v) relates to the 3D vector P with coordinates (x, y, z) emanating from the mirror plane viewpoint. To simplify there are a few assumptions that the model is based on:

1) There exist a axis of origin point of the camera coordinate system XYZ where every reflected ray is intersected in for the mirror (a central system).

2) There exist only small rotation deviations, i.e. the mirror axis is well aligned with the camera axis.

3) There is rotational symmetry for the mirror with respect to its axis.

4) Since this thesis is considering a fish-eye lens the camera lens distortion will not be considered since the projection function f will integrate this.

Assumption 2 above then gives that u and v will be proportional to x and y according to

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \begin{bmatrix} u \\ v \end{bmatrix}, \qquad \alpha > 0 \tag{4.1}$$

The searched function that should be estimated during calibration is one that maps p in the image into the vector P in 3D space

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha u \\ \alpha v \\ f(u,v) \end{bmatrix} \tag{4.2}$$

Since P is a vector equation 4.2 can be rewritten to include α into the function f. Assumption 3 above then gives that the function $f(u,v)$ only have one dependence, the distance $\rho = \sqrt{u^2 + v^2}$ to a given point from the image center.

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \tag{4.3}$$

To make a successful calibration all we need to find is the function $f(\rho)$, and is in this model described by a a taylor polynomial, where the degree and coefficients are extracted during the calibration process. The center of distortion as well as calibration points are found. The polynomial description is:

$$f(\rho) = a_0 + a_1\rho + a_2\rho^2 + a_3\rho^3 + a_4\rho^4 \ldots \tag{4.4}$$

where the coefficients $a_0, a_1, a_2, a_3, a_4, \ldots$ are the parameters to be estimated during calibration.

Chapter 5

Background Subtraction

In this chapter the background subtraction as a object detection method is presented. The Mixture of Gaussian and the Codebook method will be described and finally a description of the needed connected component analysis is introduced.

5.1 Mixture of Gaussian Method

Since the variance of the pixel intensity levels vary between pixels we want to estimate reasonable values for this variance in an image. A complex distribution and elaborate model is then needed in the form of a Gaussian mixture model (GMM). The most common approach for updating a GMM will serve as a basis and is presented in the following theory section [80].

The Mixture of Gaussian (MoG) method is able to track multiple Gaussian distributions simultaneously, and maintains a density function for each pixel. This gives a multi-modal background distribution handling and allows an adaptive parameter model update as it is parametric. The pixel process of a pixel is the set of the last t values at the pixel. The MoG model presented in the next section is adapted from [97] which is an improved algorithm based on the results in [98]. The number of components of the mixture, and not only the parameters of the mixture, is adapted constantly for each pixel. The algorithm is able to fully adapt to a scene by choosing the components number for each pixel in an on-line procedure.

Given a color-space (RGB for instance) at time t, we denote the pixel value as $\vec{x}^{(t)}$. A Bayesian decision R is made to decide if the pixel belongs to a foreground (FG) object or the background (BG) for the pixel-based background subtraction.

$$R = \frac{p(BG|\vec{x}^{(t)})}{p(FG|\vec{x}^{(t)})} = \frac{(\vec{x}^{(t)}|BG)p(BG)}{p(\vec{x}^{(t)}|FG)p(FG)} \tag{5.1}$$

In order to adapt to gradual changes in a scene due to changing weather conditions or new objects that are brought in, new samples need to be added and older ones discarded. A time period T is chosen, and at time t we get the estimated background model from a training set χ according to $\chi_T = \{x^{(t)}, \ldots, x^{(t-T)}\}$. The training data set χ is updated for every new sample and $\hat{p}(\vec{x}^{(t)}|\chi,BG)$. But since there could be foreground objects included in this value we let the estimation be $\hat{p}(\vec{x}^{(t)}|\chi,BG+FG)$. Using a M components GMM:

$$\hat{p}(\vec{x}^{(t)}|\chi,BG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I) \tag{5.2}$$

where $\hat{\sigma}_1, \ldots, \hat{\sigma}_M$ are the mean estimates and $\hat{\vec{\mu}}_1, \ldots, \hat{\vec{\mu}}_M$ are the variance estimates describing the Gaussian components. $\hat{\pi}_m$ is the non-negative mixing weights and add upp to one. Proper dimensions for the identity matrix I and a diagonal covariance matrix are assumed.

Given a time t and a new data sample $\hat{x}^{(t)}$ the recursive update equations, implemented from the paper by Zikovic [97], are

$$\hat{\pi}_m \longleftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) \tag{5.3a}$$

$$\hat{\vec{\mu}}_m \longleftarrow \hat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\hat{\vec{\delta}}_m \tag{5.3b}$$

$$\hat{\sigma}_m^2 \longleftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}^T\vec{\delta}_m - \hat{\sigma}_m^2) \tag{5.3c}$$

where $\vec{\delta}_m = (\vec{x}^{(t)} - \hat{\vec{\mu}}_m)$, and α is a exponentially decaying constant to limit the old data influence and can be approximated with $\alpha = 1/T$. For a 'close' component (the Mahalanobis distance from the component is less than some standard deviation) the ownership $o_m^{(t)}$ with largest $\hat{\pi}_m$ and all others are set to zero. If no 'close' components exist a new one is generated with $\hat{\pi}_{M+1} = \alpha, \hat{\vec{\mu}}_{M+1}) = \vec{x}^{(t)}$ and $\hat{\sigma}_{M+1} = \sigma_0$ where σ_0 is a initial variance. The components with the smallest $\hat{\pi}_m$ is discarded if the maximum components number is reached. This gives an on-line clustering algorithm. The background model can be approximated with the B largest clusters, to exclude some small weights $\hat{\pi}_m$ foreground clusters:

$$p(\hat{x}|\chi_T, BG) \sim \sum_{m=1}^{B} \mathcal{N}(\vec{x}; \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I) \tag{5.4}$$

Sorting the components according to descending weight $\hat{\pi}_m$ then give:

$$B = \arg\min_b \Big(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f) \Big) \tag{5.5}$$

where c_f is the maximum data amount that can be associated to foreground objects without affecting the background model. For instance, if a new object enters and becomes stationary in the scene it will generate a stable cluster. This causes a constant increase for the weight π_{B+1} of the new cluster. If the object remains stationary long enough, c_f will at some point become smaller than the weight and will be considered background. From equation 5.3a we find the needed time to be about $\log(1 - c_f)/\log(1 - \alpha)$ frames. For values of $\alpha = 0.001$ and $c_f = 0.1$ this equals 105 frames.

5.2 The Codebook Method

In this method a background model is created by considering the changes in brightness and color. A codebook, built up by one or more codewords, will be assigned to each pixel and will represent different states in the background. It can adapt and compress background models and capture motion over long periods of time with constrained memory, together with unconstrained training that allows foreground objects to be moving in the scene during the training phase. This allows handling of scenes with moving backgrounds such as a waving tree or a scenes with illumination variations. It has the advantage over the MOG in that it can handle fast variation backgrounds that a few Gaussians are not able to model.

The algorithm to build the background model is based on a quantization (clustering) technique [41] where every pixel is sampled and clustered to form set of codewords. The encoding of the background is performed on a pixel by pixel basis.

To construct a codebook let a single pixel with training sequence χ with N RGB-vectors be: $\chi = \{x_1, x_2, \ldots, x_N\}$, and L be the codewords that builds the codebook $C = \{c_1, c_2, \ldots, c_L\}$ for that pixel. There is a different codebook size for every pixel based on its sample variation. There

is a RGB vector $v_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ and a 6-tuple $\mathbf{aux}_i = \langle \check{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i \rangle$ for each codeword $c_{i,i=1,\ldots,L}$. The tuple represents the intensity values and time variables as follows [40]:

$\check{I}, \hat{I}$: accepted minimum and maximum brightness that is accepted for the codeword;
f : occurrence frequency of the codeword;
λ : the longest interval in the training period where the codeword has not recurred;
p,q : the first and last access time, respectively, that the codeword has occurred.

During the training process every sampled value x_t at time t will be evaluated to the present codebook and compared if there is a matching codeword, denoted c_m. c_m will then be used as the encoding approximation for the sample. The algorithm to find c_m that matches the best, a measure of color distortion and brightness bounds are used as described below in algorithm 1.

Algorithm 1 Algorithm for Constructing the Codebook

$L \leftarrow 0$, $C \leftarrow \emptyset$
for $t = 1$ to N **do**
  $x_t = (R,G,B), I \leftarrow R + G + B$
  Find c_m in $C = \{c_i | 1 \leq i \leq L\}$ corresponding to x_t given condition a) and b).
    a) $colordist(x_t, v_m) \leq \epsilon_1$
    b) $brightness(I, \langle \check{I}_m, \hat{I}_m \rangle) =$ **true**
  if $C = \emptyset$ or no match **then**
    $L \leftarrow L + 1$ and create new codeword c_L
    $v_L \leftarrow (R,G,B)$
    $\mathbf{aux}_L \leftarrow \langle I, I, 1, t-1, t, t \rangle$.
  else {Update matching codeword c_m consisting of its RGB vector v_m and its 6-tuple $\mathbf{aux}_m$}
    $v_m \leftarrow (\frac{f_m R_m + R}{f_m + 1}, \frac{f_m G_m + G}{f_m + 1}, \frac{f_m B_m + B}{f_m + 1})$
    $\mathbf{aux}_m \leftarrow \langle min\{I, \check{I}_m\}, max\{I, \hat{I}\}, f_m + 1, max\{\lambda_m, t - q_m\}, p_m, t \rangle$.
  end if
end for
for each $c_i, i = 1$ to L **do**
  $\lambda_i \leftarrow max\{\lambda_i (N - q_i + p_i - 1)\}$.
end for

5.3 Foreground Cleanup and Connected Components Analysis

Connected-component analysis is used to clean up the foreground to get a better segmented picture. The first step is to input the noisy mask image and perform two morphological operations; an opening (a more complex erosion) to erase smaller areas of noise, followed by a closing (a more complex dilation) to rebuild the remaining surviving components.

The remaining (outer) contours are then retrieved, following the method derived by Suzuki *et al.* [81]. The length of each contour is calculated and if they have the length corresponding to a pedestrian i.e. are above a specific threshold, they are kept. To be able to analyze the contour further an approximation is made to reduce the number of vertices, and the heuristic Douglas-Peucker (DP) method is utilized [16]. The DP algorithm is a high-quality curve simplification algorithm that takes a contour and picks the two extreme points and connects them with a line. The contour is then searched to find the point that is furthest frown the drawn line, which will

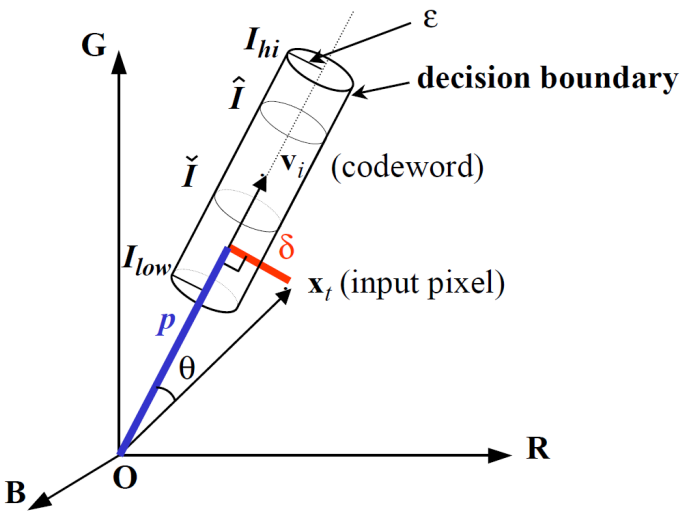


Figure 5.1: Visualization of the codebook color model - a separate evaluation of color distortion and brightness distortion [40].

then be added to the approximation. This will then be repeated, adding the next most distant point to the approximation, until the distance is below a specified threshold. Finally, the smallest rotated rectangle that encloses the contour is found and returned as the detected bounding box. There exist faster algorithms than DP but these are considered to produce inferior results [52].

Chapter 6

Object Detection

In this chapter two object detection methods will be described: the method proposed by Viola & Jones and improved by Lienhart together with a theoretical description of integral images and how to construct a multi-stage boosted classifier with Adaboost; and the generalized Viola & Jones method proposed by Dollar et al.

6.1 Cascade of Classifiers by Adaptive Boosting

Introduced in 2001 by Viola and Jones [86] this visual object detection framework was capable of extremely rapid image processing. The framework consists of three key methods: a fast computing of haar-like features through the image representation technique called Integral Image, the Adaptive Boosting (AdaBoost) learning algorithm for visual feature selection; and a "cascaded" combination method of the classifiers to allow rapid evaluation of the image by quickly discarding background regions and focus the computation on more likely object regions. These three methods will now be described.

6.1.1 Features

Haar-like Features (Image features)

The motivation behind using features as the input to a learning algorithm instead of raw pixel values directly is that a pixel value of a pedestrian body only reveals the chromaticity and brightness information, which does not help in process to classify the object. A pedestrian detection method that classifies images based on the value of simple feature descriptors can distinguish between the variance of body parts as well as the geometric relation between different areas. Another reason is that the pixel-based system operates much slower than the feature-based system and it enables to learn ad-hoc domain knowledge that are otherwise difficult to learn with a finite set of training data. A large and general haar-like feature pool will therefore increase the learning algorithm's capacity. The shape, scale, and position within the interest region determine which feature is going to be used in a specific classifier.

The Haar Classifier Cascades used in this thesis consist of two-dimensional Haar Wavelet Features, i.e. white and black rectangular patterns. In figure 6.1 we see the set of 14 simple Haar-like feature prototypes that are adopted in this thesis: edge features which represent the boundary information, line features representing lines, and center-surround feature which provides encirclement information. The white areas have positive weight and the black areas have negative.

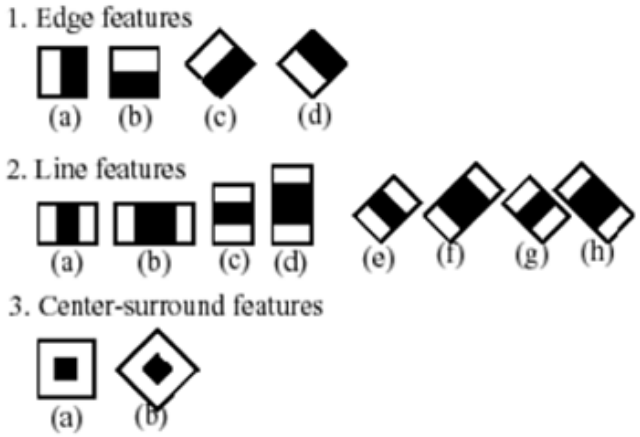


Figure 6.1: The simple haar-like features and center-surround features used in the classifiers. White areas have positive weight and black areas have negative weight.

Integral Image Calculation

The total number of features for a specific prototype is large, with a sum total of 117,941 features for a 24×24 window. The integral image allows for rapid linear time processing of features over millions of possible locations in an image using a set of over-complete Haar-like features. To compute a up-right rectangular feature we create a auxiliary image called Summed Area Table $SAT(x,y)$, which is the pixel sum values ranging from the bottom right corner at (x,y), to the top left corner at (0,0):

$$SAT(x,y) = \sum_{x' \leq x, y' \leq y} I(x', y') \tag{6.1}$$

Consider the blue marked rectangle in figure 6.2 with an image array I with all the pixel intensity values stored. Once the elements above and to the left is known, every element in the integral image $SAT(x,y)$ can be calculated with only one pass over the given image. The value at location (x,y) in the SAT is therefore found by adding the original pixel value $I(x,y)$ and the pixels directly to the left and above, found from $SAT(x-1,y)$ and $SAT(y-1,x)$, and subtracting the value at the top left of $i(x,y)$ found from $SAT(x-1, y-1)$ (to account for the double calculation)

$$SAT(x,y) = SAT(x, y-1) + SAT(x-1, y) + I(x,y) - SAT(x-1, y-1) \tag{6.2}$$

with boundary values $SAT(-1,y) = SAT(x, -1) = SAT(-1, -1) = 0$. This gives that only four table lookups are needed to calculate the pixel sum for any upright rectangle r with width w and height h described by $r = (x,y,w,h)$

$$RecSum(r) = SAT(x, y) + SAT(x+w, y+h) - SAT(x, y+h) - SAT(x+w, y) \tag{6.3}$$

Given the rectangle in Figure 6.2 this becomes, $RecSum(r) = SAT(A) + SAT(D) - SAT(B) - SAT(C)$. For instance, the response for feature (2b) in Figure 6.1 is calculated by first summing the image pixels under the entire rectangular feature, and then subtracting the sum of the image pixels under the black stripe multiplied by 3 (compensating for the size of area difference). Since the image edges need to be parallel with the rectangle edges this is limited to upright rectangles. Lienhart and Maydt extended this to include 45° tilted rectangles [46]. The calculations are done in a similar fashion where the Tilted SAT ($TSAT$) calculation need two passes, the first from left to right and top to bottom (with increasing x and y):

$$TSAT(x,y) = TSAT(x-1, y-1) + TSAT(x-1, y) + I(x,y) - TSAT(x-2, y-1) \tag{6.4}$$

with boundary values $TSAT(-1,y) = TSAT(-2,y) = TSAT(x,-1) = 0$, and the second pass from right to left and bottom to top (with decreasing x and y):

$$TSAT(x,y) = TSAT(x,y) + TSAT(x-1,y+1) - TSAT(x-2,y) \tag{6.5}$$

Given this tilted integral image, any 45° tilted rectangle pixel sum can be calculated with again only four table lookups. For a tilted rectangle described by $r = (x,y,w,h)$, the pixel sum can be calculated as

$$RecSum(r) = TSAT(x+w,y+w)+TSAT(x-h,y+h)-TSAT(x,y)-TSAT(x+w-h,y+w-h) \tag{6.6}$$

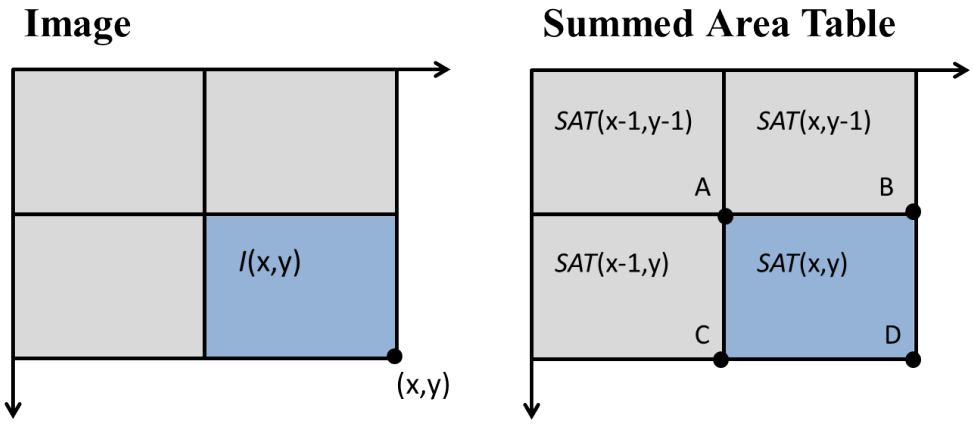


Figure 6.2: Calculating the values of a upright rectangular subset in a grid through the use of a Summed Area Table.

6.1.2 Adaptive Boosting, Classifier Cascade, training

Since the number of features for a given sub-window is extremely large it is not feasible to compute the complete set. However, AdaBoost can be used to find a smaller set of features to form an effective classifier and train the classifier [23]. AdaBoost is a learning algorithm that given a simple learning algorithm (weak learner) can boost its classification performance. It does this by an iterative process to find an accurate strong classifier, $H(x)$, based on a linear combination of T weak classifiers, $h_t(x)$.

$$H(x) = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big) \tag{6.7}$$

where α_t is the weight of the training data in round number t. After each round all examples are re-weighted, where the weights are increased on the incorrectly classified examples (and correspondingly decreased the weights on the correctly classified examples) as to emphasize them in the next round of learning. The weak classifier therefore only needs to perform slightly better than chance, which gives an inexpensive computation. The full algorithm is listed in algorithm 2. There exist many different boosting variants such as Real AdaBoost, Discrete AdaBoost, and Gentle AdaBoost. They all have the same computational complexity, but have different learning algorithms.

Algorithm 2 The boosting algorithm AdaBoost

Given training samples $(x_1,y_1),\ldots,(x_n,y_n)$ where $x_i \in X, y_i \in Y = \{-1,+1\}$ for negative and positive examples.
Initialize weights $w_1(i) = \frac{1}{n}, i = 1, \ldots, n$.
for $t = 1$ to T **do**
  From the set of weak classifiers $\mathcal{H}$, find classifier h_t that minimizes the error ϵ_j with respect to the distribution w_t.
    $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j$, where $\epsilon_j = \sum_{i=1}^{n} w_t(i)[y_i \neq h_j(x_i)]$
  if $\epsilon \geq 0.5$ **then**
    return
  end if
  Set $\alpha_t = \frac{1}{2}\log(\frac{1+r_t}{1-r_t})$, where ϵ_t is the weighted error rate of classifier h_t
  Update the weights
    $w_{t+1}(i) = \frac{1}{Z_t} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$, where Z_t is a normalization factor
  Output the final classifier
    $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
end for

6.1.3 Classifier Cascade

The multi-stage classification method will substantially reduce the processing time by starting with the simplest classifiers that rejects a majority of the data and gradually increases the classifier complexity. This procedure will rapidly reduce the amount of classification tests per data point, and thus reducing the number of images that will need to be processed, since the majority of the negative data images will be rejected in early stages of the cascade and gives an improved overall performance. The weak classifier is then only needed to perform better than chance, and leads to being simple and computationally inexpensive.

The cascade process is similar to a decision tree with a depth of four, where a series of classifiers is used to to test an image. The computation is as following:

1) The cascade has several stages/classifiers; if the stage is tested true it continues to be tested against the second stage in the three and so on until it fails a classifier and will then be labeled ”false”, or been evaluated true in all classifiers and then labeled as ”true”

2) In each stage of the cascade a false negative rate threshold, or the sum of its output features, needs to be satisfied for that stage to return ”true”, and if chosen to a small value it will result in a correct classification of a large percentage of the positive data for the cascade

3) A feature will return the total sum of its rectangle outputs

4) A rectangle will return the pixel values sum for the image region bounded by that rectangle an multiplied with a given weight

The framework of the Haar wavelet-based cascade uses the approach of sliding windows with increasingly complex detector layers in a degenerate decision tree [87]. There is a set of nonadaptive Haar wavelet features used for each layer [56]. The trained cascade of classifiers forms a degenerate decision tree, where a classifier at each stage will detect almost all objects of interest and rejecting a certain fraction of non-pedestrians. To detect a pedestrian in a image, the final detector evaluates a sub-window and then shifts the window Δ pixels to the subsequent locations until the entire image has been scanned. The detector is then scaled up with a factor s and the image is rescanned. This gives that areas in the image that clearly do not contain

a pedestrian will be discarded in early stages of the cascade. When a sub-image has passed through all stages of the cascade without being rejected, it is identified as a pedestrian.

The K number of needed cascade layers can be determined from the goal of the overall false positive rate F, and detection rate D

$$F = \prod_{i=1}^{K} f_i \tag{6.8}$$

$$D = \prod_{i=1}^{K} d_i \tag{6.9}$$

For instance, if a overall detection rate of 0.9 is the goal then a 10 stage classifier with a detection rate of 0.99 for each stage would be set during training (since $0.99^{10} \approx 0.9$).

There are two types of tradeoffs for the training process: involving more features gives a higher detection rate and lower false positive rate, but requires more computation. To find a optimum can be difficult, so a simple framework is used based on the specified acceptable rates f_i and d_i. At each layer of the cascade the detection and false positives rates are evaluated by testing the current detector on a specified validation set. The number of features is then increased until the target rates are satisfied for that layer. Once satisfied, the overall target false positive rate is evaluated; if it is still above the threshold then another layer is added to the cascade. The collection of false detection made while running the current detector on a subset of given negative examples are then placed in the negative set for subsequent layer training. A more detailed description of the steps is given in algorithm 3.

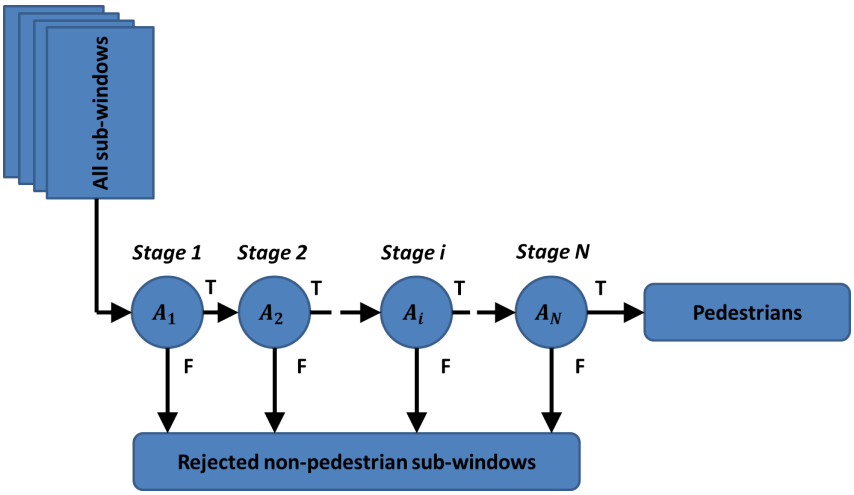


Figure 6.3: The degenerate decision tree, or 'cascade', for the detection process.

The feature-based classifier that best can classify the given weighted training samples will be added at each round of boosting. A N stages cascade structure is visualized in figure 6.3, where A_i is an AdaBoost classifier at stage i. As seen, this corresponds to a degenerated decision tree where at each stage the classifier detects nearly all pedestrians while at the same time discarding a certain pre-set fraction of non-pedestrians. Sub-images that clearly do not contain a pedestrian is therefore rejected at an early stage of the cascade, and a object is labeled pedestrian if it successfully passes through all stages.

Figure 6.4 shows the complete flow diagram for this pedestrian detection system, which includes the offline training with AdaBoost and a online detection as described above.

Algorithm 3 Training algorithm for constructing a Cascade of Classifiers

Define $F = \Pi_{i=1}^{K} f_i$, $D = \Pi_{i=1}^{K} d_i$
Specify maximum acceptable false positive rate f per layer, minimum acceptable detection rate d per layer, and target overall false positive rate F_{target}
Input set of positive examples P, and negative examples N
Initialize $F_0 = 1.0$, $D_0 = 1.0$, $i = 0$
while $F_i > F_{target}$ **do**
  $i \leftarrow i + 1$
  $F_i = F_{i-1}$, $n_i = 0$
  while $F_i > f \times F_{i-1}$ **do**
    $n_i \leftarrow n_i + 1$
    Train a classifier with n_i features with AdaBoost using P and N
    Evaluate current cascaded classifier on validation set to determine F_i and D_i
    Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least $d \times D_{i-1}$
  end while
  $N \leftarrow \emptyset$
  if $F_i > F_{target}$ **then**
    Evaluate current cascaded detector on the set of non-pedestrian images and put any false detections into set N
  end if
end while

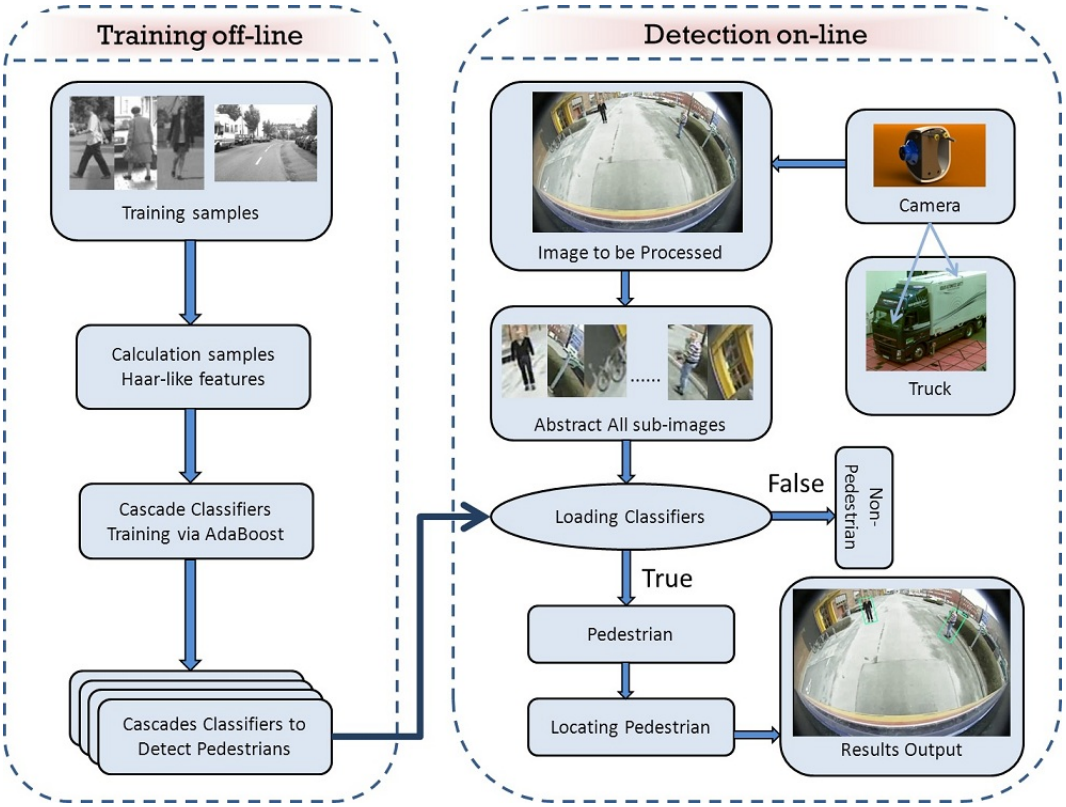


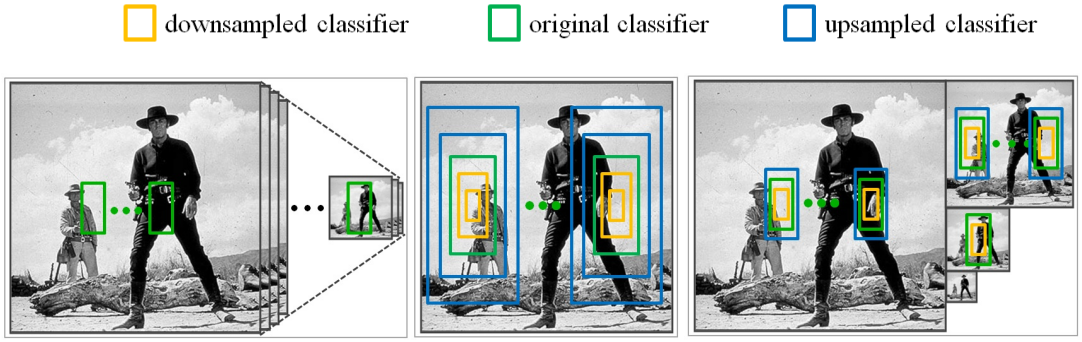
Figure 6.4: Flow-chart of the proposed pedestrian detection system.

6.2 The Fastest Pedestrian Detector in the West

The limitation of many modern multiscale detectors is the image pyramid construction, often finely sampled at 8-18 scales per octave and the following computation of features at every scale. The insight in the proposed technique by Dollár *et al.* [12, 13] is that the a calculation of the feature response at a single scale for a broad family of features can be used as an approximation for the feature response at nearby scales that gives a speedup of 10-100 times of similar methods and with only a small loss in accuracy (around 1-2% for the Caltech Pedestrian dataset). This detection method is a generalization of the Viola & Jones method discussed in section 6.1, where instead of utilizing the integral image over the original intensity image I with Haar-like features, multiple channels are used.

Gradients computed at one scale can give a good approximation to gradient histograms at a different scale and can thus avoid the extensive gradient computation over the finely sampled image pyramid. This approximation is valid over an entire scale octave. This method can therefore run at near real time (around 7 fps on 640x480 images).

In figure 6.5 three different methods for multiscale detection is presented. Subfigure 6.5b represents a dense image pyramid that is created and features are computed for each image scale, and for each fixed scale the sliding window classification paradigm is used. In subfigure 6.5c the scale invariant features proposed by Viola and Jones is shown, discussed in section 6.1, a fast method but few features are scale invariant yielding a generality loss. The proposed technique from Dollár *et al.* is a hybrid between the two earlier approaches and uses a sparsely sampled image pyramid that approximates the features within an entire octave and within each octave uses a classifier pyramid.



(b) dense image pyramid **(c)** classifier pyramid **(d)** Dollar et al. hybrid approach

Figure 6.5: Three different pipelines for multiscale detection. (b) the time consuming features are computed at each scale of the image pyramid together with a sliding window classification. (c) The scale invariant method presented in section 6.1. (d) a hybrid approach between the other two [12].

Chapter 7

Tracking with Kalman Filtering

This chapter describes the final step for a complete pedestrian detection system - pedestrian tracking. The Kalman filtering technique is used in this master's thesis and will be described in this chapter.

7.1 Kalman Filter Principles

To complete the pedestrian detection system a tracking procedure should follow after a detection is made to increase accuracy and alert for possible future collisions. The Kalman filtering technique will be used to approximate future pedestrian locations given its current location. The Kalman filter algorithm takes measurements over time, which can contains noise and other uncertainties, and outputs a approximation of the process state that is more accurate than only a single measurement. This results in a temporal trajectory of a pedestrian that makes it possible to predict the pedestrian motion and probability of conflicting courses.

There are two steps in the Kalman filter algorithm [89]: a time update that projects the current state and estimation of the error covariance to attain *a priori* estimate for the concurrent time; and a measurement update that takes new measurements and incorporates them with the *a priori* estimates, resulting in an improved *a posteriori* estimate. The algorithm is capable of real-time operation due to the recursion used, where only the previous state and present input measurements are used and no further information from the past is required.

Let the state equation of the system $X \in R^n$ be approximated as a linear stochastic difference equation [32]

$$X_t = \Phi_{t-1} X_{t-1} + W_{t-1} \tag{7.1}$$

and a measurement $Z \in R^n$ as

$$Z_t = H_t X_t + V_t \tag{7.2}$$

where W_t is the state noise, V_t is the measurement noise, Φ_t is the state transformation matrix (and relates the state at time $t-1$ to the present state at time t), and H_t is the measurement matrix. The random variables W and V are assumed to be independent and zero mean Gaussian white noise

$$p(W)\tilde{N}(0, Q) \tag{7.3}$$

$$p(V)\tilde{N}(0, R) \tag{7.4}$$

where Q is the state noise covariance,R is the measurement noise covariance. Let $\hat{X}_t^- \in R^n$ be the *a priori* estimate at time t, and $\hat{X}_t \in R^n$ be the *a posteriori* state estimate at time t given measurement Z_t. The state update equations of the Kalman filtering are then

$$\hat{X}_t^- = \Phi_{t-1}\hat{X}_{t-1} \tag{7.5}$$

$$P_t^- = \Phi_{t-1}P_{t-1}(+)\Phi_{t-1}^T + Q \tag{7.6}$$

and the measurement update equations are

$$K_t = \frac{P_t^- H_t^T}{H_t P_t^- H_t^T + R} \tag{7.7}$$

$$\hat{X}_t = \hat{X}_t^- + K_t[Z_t - H_t\hat{X}_t^-] \tag{7.8}$$

$$P_t = (I - K_t H_t)P_t^- \tag{7.9}$$

where K_t is the Kalman gain matrix (chosen), and P_t is the estimate error covariance. The full prediction-correction cycle with time and measurement equations can be seen in Figure 7.1.

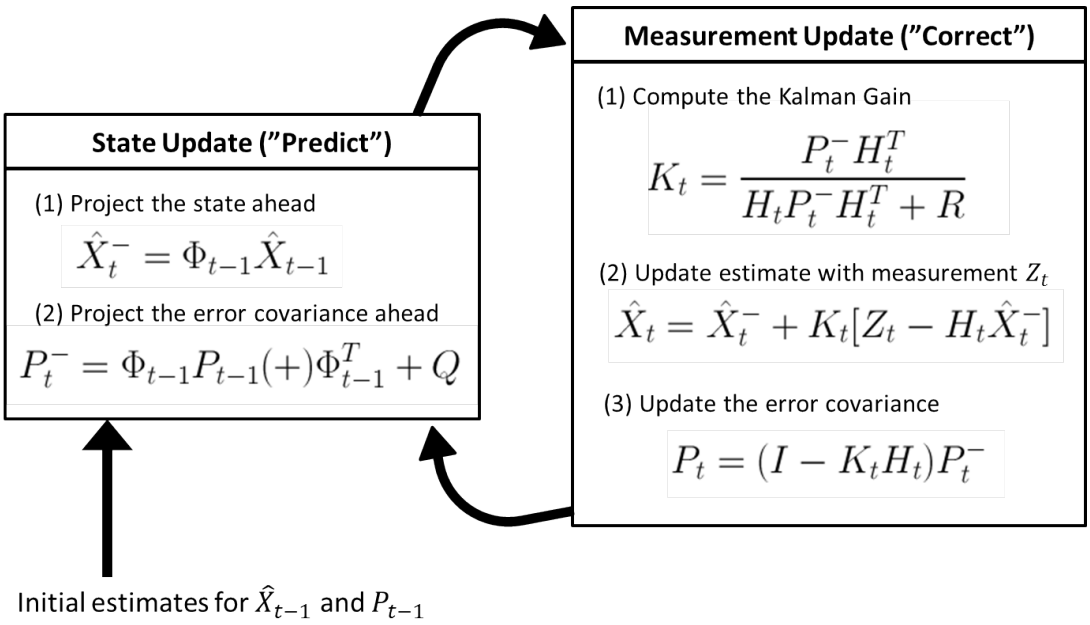


Figure 7.1: The Kalman filter cycle with time update and measurement update equations. The time update estimates future states from the current state, and the measurement update corrects the estimation by an actual measurement. Extended from [89].

7.2 Tracking Pedestrians with Kalman Filtering

In this thesis the pedestrian representation is a n-gon bounding box, and the state at a time t is characterized by the center point of the bounding box x_t, y_t and their corresponding velocities $(\Delta x_t, \Delta y_t)$, modeled together with the mean height and width (h_t, w_t) together with their velocity $\Delta h_t, \Delta w_t$. The corresponding state vector of the pedestrian at time t is then

$$X_t = (x_t, y_t, h_t, w_t, \Delta x_t, \Delta y_t, \Delta h_t, \Delta w_t)^t \tag{7.10}$$

Assuming that the velocity of a pedestrian and the vehicle does not change significantly between two subsequent frames, we can represent the state transformation matrix as

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.11}$$

In the same way we represent the measurement vector as $Z_t = (x_t, y_t, h_t, w_t)^t$, and the measurement matrix as

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.12}$$

Chapter 8

Performance Evaluation

In this chapter the evaluation methodology to be used in this master's thesis is presented - the object region match method and the Receiver Operating Characteristic curve.

8.1 Overview

Since object detection is not a classic binary detection problem it still has major drawbacks. For instance, instead of just take into account missed detections and false alarms one needs to consider several different other errors that can occur such as a ground truth region is split into several regions, or the opposite, two ground truth regions are merged as one detection. Also, there can exist many valid segmentations that corresponds to a ground truth pedestrian, and can be considered as a correct classification. The term "ground truth pedestrian" denotes the rotated ground-truth bounding box manually marked out around the pedestrian. The detections are the output boxes from the detection algorithms.

In this thesis the different object detection methods will be naively evaluated by the objective metrics proposed by Nascimento *et al.* [50, 60]. The evaluation considers all different detection errors and compares these with the manually labeled ground truth data. The parameters that are considered are:

Correct Detection (CD): The detected region corresponds to one and only one ground truth region (a 1-1 match).

False Detection (FD): The detected region has no corresponding region in the ground truth.

Missed Detection (MD): The ground truth region has no correspondence in the detection region.

Merge Region (M): The detected region has many corresponding ground truth regions.

Split Region (S): The ground truth region has many corresponding detection regions.

Split-Merge Region (SM): The condition (M) and (S) are simultaneously satisfied.

In order to be be able to make a performance evaluation of a detection method a few steps need to be followed. Firstly, a test sequence need to be selected and a ground truth for the containing pedestrians need to be marked. The detection method to be evaluated is then compared with the ground truth as to classify the errors in one of the classes described above.

8.2 Region Match

To match the detected object to the ground truth a binary correspondence matrix C^t is constructed with the size NxM, where N is the number of ground truth regions $\tilde{R}_i$, and M is the number of detected regions R_j. C^t will then have the following definition

$$C^t(i,j)(\mu) = \begin{cases} 1, & \text{if } \frac{\#(\tilde{R}_i) \cap R_j}{\#(\tilde{R}_i) \cup R_j} > T \\ & \forall_{i \in \{1,\ldots,N\}, j \in \{1,\ldots,M\}} \\ 0, & \text{if } \frac{\#(\tilde{R}_i) \cap R_j}{\#(\tilde{R}_i) \cup R_j} < T \end{cases} \tag{8.1}$$

where T is the threshold requirement. Defining two auxiliary vectors to represent number of ones in each column and row as

$$L(i) = \sum_{j=1}^{M} C(i,j) \qquad i \in \{1,\ldots,N\} \tag{8.2}$$

$$C(j) = \sum_{i=1}^{N} C(i,j) \qquad j \in \{1,\ldots,M\} \tag{8.3}$$

The ground truth regions can now be associated with the detected regions from the following classification rules

$$\textbf{CD} \quad \exists_i : L(i) = C(j) = 1 \wedge C(i,j) = 1 \tag{8.4a}$$
$$\textbf{FD} \quad \exists_i : C(j) = 0 \tag{8.4b}$$
$$\textbf{MD} \quad \exists_i : L(i) = 0 \tag{8.4c}$$
$$\textbf{M} \quad \exists_i : C(j) > 1 \wedge C(i,j) = 1 \tag{8.4d}$$
$$\textbf{S} \quad \exists_i : L(i) > 1 \wedge C(i,j) = 1 \tag{8.4e}$$
$$\textbf{SM} \quad \exists_i : L(i) > 1 \wedge C(j) > 1 \wedge C(i,j) = 1 \tag{8.4f}$$

The six different scenarios can be seen in Figure 8.1 together with its corresponding binary correspondence matrix.

To capture different aspects and further quantify the performance of the different object detection systems five of the performance evaluation metrics proposed by Mariano *et al.* [50] has been implemented. All of the metrics will have a value range from zero to one (perfect). In the following paragraph a summary of one of the metrics is presented - a pixel-count-based metric that measures the coverage of the algorithms output over the ground-truth, named Area-Based Recall.

Let $U_{G^{(t)}}$ and $U_{D^{(t)}}$ be the spatial union of the boxes in the ground-truth $G^{(t)}$ and output detection $D^{(t)}$ for a single frame t:

$$U_{G^{(t)}} = \bigcup_{i=1}^{N_{G^{(t)}}} G_i^{(t)} \qquad U_{D^{(t)}} = \bigcup_{i=1}^{N_{D^{(t)}}} D_i^{(t)} \tag{8.5}$$

The ratio $Recall(t)$ between ground truth areas that are detected and total ground truth is then defined as:

$$Recall(t) = \begin{cases} \text{undefined} & \text{if } U_{G^{(t)}} = \emptyset \\ \frac{|U_{D^{(t)}} \cap U_{G^{(t)}}|}{|U_{G^{(t)}}|} & \text{otherwise} \end{cases} \tag{8.6}$$

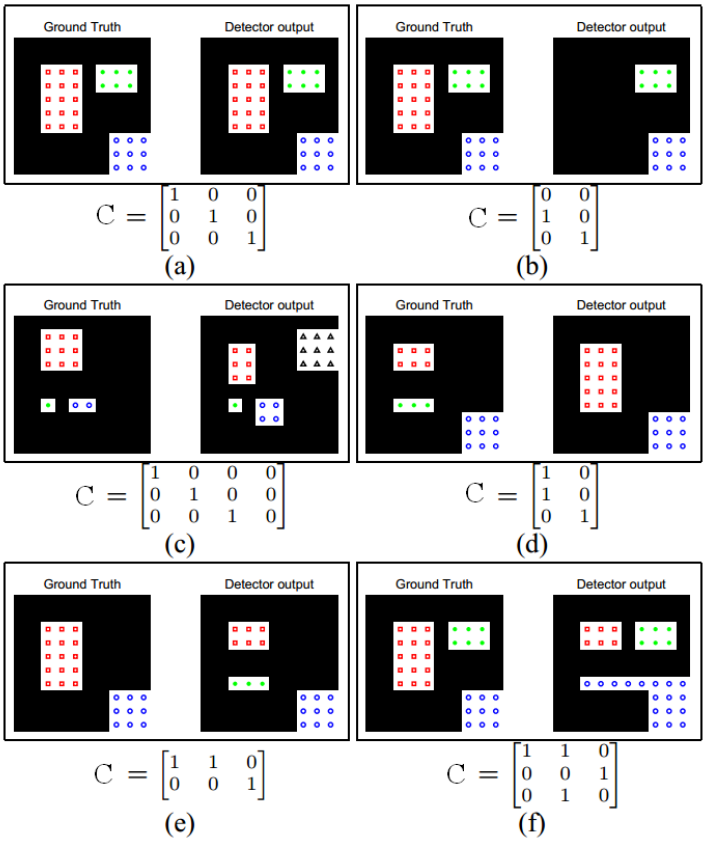


Figure 8.1: Matching cases than can occur during evaluation: a) Correct Detection; b) Missed Detection; c) False Detection; d) Merge; e) Split; f) Split Merge [60].

This metric evaluates the frame as a binary pixel map (pedestrian/non-pedestrian; detected/non-detected) and gives a good recall for comparison between algorithms, but will miss the individual pedestrian evaluation in the ground-truth as well as being biased towards pedestrians with larger ground-truth bounding boxes. But since the test scenarios covered in this thesis only deal with detection of single pedestrians this will not be an issue.

8.3 Receiver Operating Characteristic Curve

The above accuracy measure is only justified when a correct classification is equal in importance for all classes, and incorrect classification is equally bad for all classes. For object detection this is not the situation, where thousands of negative sub-regions pass the detector for each true region. Always returning false could therefore result in an excess accuracy of 99% with this metric. This also makes it difficult to comparing algorithms [86]. A better technique to evaluate an object detection algorithm is to build a Receiver Operating Characteristic (ROC) curve [90]. A discrimination threshold is varied and one calculates the true positives and false positives for each different threshold and plot the two different counts against each other. If the false positive count is on the x-axis and the true positive count on the y-axis, the ideal ROC curve would lie towards the top left graph corner; representing a 100% true positive rate and 0 false positives.

Implementing the ROC curve for the Haar Classifier Cascade the threshold to be varied is selected to be the number of neighbours, described in Chapter 9.5. The rightmost curve point represents the true positive and false positive counts for all frames when no merging of neigbour rectangles are performed. The second rightmost curve point represents the merging of at least

two neighbours to count as a detection. The nth point therefore corresponds to the detection performance when at least n neighbour regions are merged. This continues until n becomes infinite and correspondingly zero true positives and zero false positives are obtained.

Chapter 9

Methodology and Implementation

This chapter describes the details on training the cascaded detector, the different test sequences used, and presents the method used for the combined detection system and the implementation of all the components in the final system.

9.1 Experimental Platform

All algorithm implementation was conducted on a standard laptop computer using the Intel OpenCV 2.3 library [62] together with the described algorithms and implemented in C/C++ and MathWorks Matlab 7.12. The experiments were run on a Dual Intel 2.4 GHz PC with 2 GB of RAM. All test and training data sequences were captured around Lindholmen, Gothenburg, Sweden, over 3 sessions. The camera system, mounted on the front and side of the truck, were captured at a resolution of 720×576 pixels.

9.2 Training the Cascade of Classifiers with AdaBoost

All training samples were divided into positive samples (containing a pedestrian) and negative samples (not containing a pedestrian). The positive samples are cut outs of pedestrians with different clothing, poses, and sizes. To crop the captured images faster the *image clipper* tool was extended and used [38]. The negative image set contains complete background images of a variety of typical urban environments and are randomly sampled during training. At every stage during training the negative images are scanned for false positive classified regions that are then used as negative training samples for the next training stage.

The training samples cover upright pedestrians with all out-of-plane orientation variations. In this thesis all features are required to completely be within the training samples. Apart from this there were no imposed requirements on wavelet locations or scales. All detectors are composed of a cascade of weak classifiers composed of the over-complete set of basic and rotated Haar-like features, as presented in Chapter 6.1.1, resulting in a set of 266319 features used. It has been shown that Gentle AdaBoost with small CART trees as a base classifiers outperform Discrete AdaBoost and stumps [44], and were thus chosen for training. Empirical studies has shown there is a saturating performance after the 15th stage in the training phase, so the total number of stages were set to 18 to capture this. A minimum hit rate of 0.995 and a maximum false alarm of 0.5 were set for each stage. In accordance with equation (6.8) and (6.9) a detection rate of $\sim 92\%$ and a false positive rate of $\sim 3.8 \times 10^{-6}$ should therefore be achieved for the complete cascaded classifier.

9.2.1 Multi-view Detection

The input image is split into five different rotational regions as to classify pedestrian and non-pedestrian in each of these separate regions with the same classifier, see Figure 9.5 for a schematic of the setup. The exact rotation separation and division were found by analyzing the maximum and minimum detection angles possible for a pedestrian at each location in the scene, see Chapter 10.3.1. To best capture this rotational behavior it has been shown that during the training of the cascade the positive image samples should be perturbed with an approximate angle range equal to the step size between the image rotation [42], which in this case is set to a value of $\pm 15°$, see Chapter 10.3.1.

9.2.2 Front Camera

(a) Positive training samples.

(b) Negative training samples.

Figure 9.1: A subset of the (a) positive training samples, and (b) negative training samples for the front camera.

Two separate pedestrian detectors were trained: one at an early phase in the thesis with lower thresholds and fewer positive and negative training samples- 1100 and 1000 respectively - and one at a later stage of the thesis with more training samples and higher accuracy thresholds.

For the final cascaded classifier 2000 positive images, resized to 18×36 pixels, were hand labeled for training. 1000 samples collected by me and 1000 from the Daimler Benchmark dataset **??**. Correspondingly 3000 negative samples were used, 1500 collected by me and 1500 from the Daimler dataset. A subset of the positive and negative training samples used can be seen in figure 9.1.

9.2.3 Side Camera

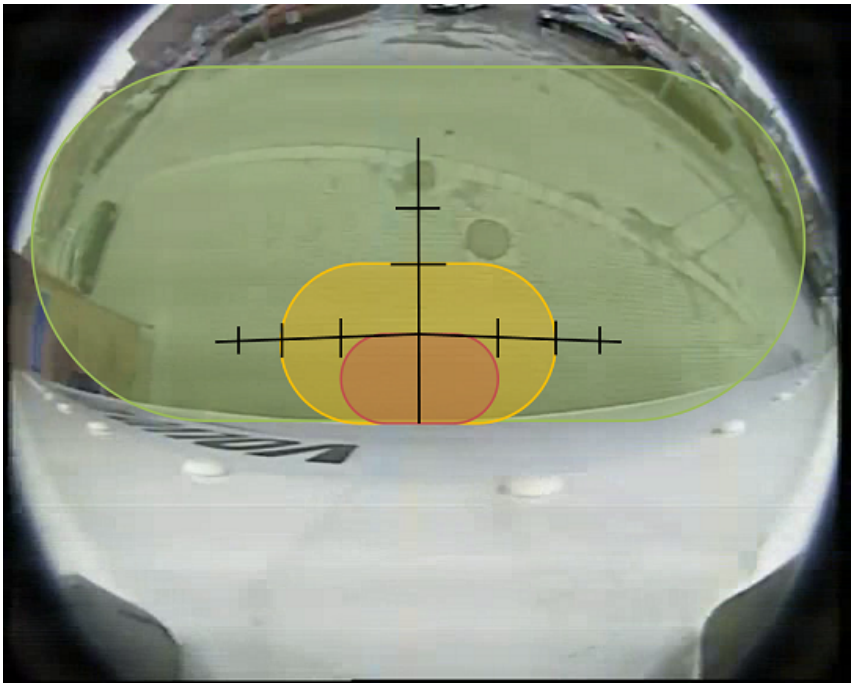


Figure 9.2: Classifier division and distance markings (in meters) for the side camera.

Training a classifier for the side camera has the added complication of self-occlusion due to the top-view that needs to be take into account. Inspired by the recent work of Yuan *et al.* [94] the pedestrian training samples were divided into different classes and each class were trained separately in accordance with the amount of radial distortion. In this scenario the scene has been divided into three different classes according to the bounding box ratio of the pedestrian. In figure 9.2 we see how the scene is divided between the three different classifiers for the side camera. Each colored area represents at what feet location of a pedestrian each specific classifier is active. Classifier 1 is active approximately within a one meter radius region with width center at the midway edge of the trucks long side. Classifier 2 is similarly active approximately within a two meter radius, and classifier 3 is active for the rest of the scene. The division depicted in figure 9.2 is not exact but in reality contains overlaps between the active regions as to not lose detection.

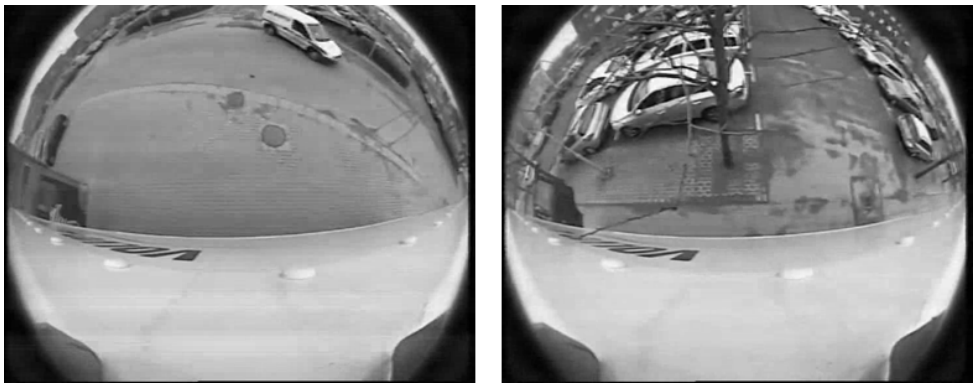
Since there currently does not exist a dataset which covers top-view pedestrians all training samples, positive and negative, were captured and hand labeled by me. Classifier 1 used 475 positive images resized to 24 by 24 pixels, and 325 negative images; classifier 2 used 565 positive 20 by 30 pixels positive images, and 475 negative images; and classifier 3 is the same classifier trained for the front camera as described in the section above. Figure 9.3 depicts a subset of the positive and negative training samples for the side camera. The same in-plane rotation scheme as described in section 9.2.1 is used.

9.3 Scene Setup and Region Testing

Since the pedestrian size is strongly correlated to its position in the scene, the starting scale of the detection sub-window was increased for the lower half of the image frame. The starting scale of the Haar Classifier Cascades was set to 1.1 and increased with a scale step size of 1.1–1.2 until the scanning window exceeds the image size. The scanning starts at the top left corner and

(a) Positive training samples for classifier region 1.

(b) Positive training samples for classifier region 2.



(c) Negative training samples.

Figure 9.3: A subset of the training samples used for the side camera. Subfigure (a) and (b) show positive training samples for the two different classifier regions, and subfigure (c) shows negative training samples.

continues in steps of $max(2.0, scale)$ pixels (rounded to nearest pixel). Figure 9.4 shows how a 20 $\times$ 20 pixels detection window cascade tests a 640 $\times$ 480 pixel image, resulting in 533,126 evaluated regions.

Due to the heavy distortion the pedestrian size constraint will vary across the scene, and hence the ROIs is implemented to satisfy certain thresholds to be considered for further evaluation. For instance, a detection output at the left side of the image that is rotated clockwise can be discarded since that is not possible to contain a pedestrian. The same reasoning goes for large scale detections far back in the scene, detections at sky level, and detections with large deviations from the standard pedestrian bounding box ratio. There is also a communication interface between the different methods to improve the detection accuracy. For example, a ROI found with the background subtraction model will be analyzed deeper with the classifier than other regions.

Scale	Step (pixels)	Region size (pixels)	Regions tested
1.00	2.00	20 × 20	82,810
1.20	2.00	24 × 24	81,648
1.44	2.00	29 × 29	80,160
1.73	2.00	35 × 35	78,540
2.07	2.07	41 × 41	71,416
2.49	2.49	50 × 50	48,100
2.99	2.99	60 × 60	32,163
3.58	3.58	72 × 72	21,228
4.30	4.30	86 × 86	14,058
5.16	5.16	103 × 103	9,085
6.19	6.19	124 × 124	5,704
7.43	7.43	149 × 149	3,626
8.92	8.92	178 × 178	2,146
10.70	10.70	214 × 214	1,260
12.84	12.84	257 × 257	680
15.41	15.41	308 × 308	325
18.49	18.49	370 × 370	144
22.19	22.19	444 × 444	33
		Total	533,126

Figure 9.4: Number of evaluated regions for a 640 × 480 pixel image by a 20 × 20 unit cascade [36].

9.4 Combining the systems

As discussed the distortion is severe close to the camera, and hence the classifier will not be able to function in this region. However, the background subtraction methods perform well in this area and makes for a better candidate to deal with this region. This give a partitioning of the scene where the background subtraction methods will have the biggest role of the near vicinity detection region and the classifier the biggest role for the further away regions. The background subtraction methods are still active in the far away regions as well, but with a much higher detection threshold. There is also significant overlap between all regions as to sustain a smooth transition between the regions. The Figure 9.5 shows the schematic of the final detection setup between the classifiers and background subtraction methods. Since the pedestrian bounding box size is large in this area, this knowledge is utilized by setting a high threshold for the contour size in the connected component analysis to filter out noise and false detections from the ground plane (such as that from rain puddles and man holes). The scene is restricted so only pedestrians within the right size will be evaluated at the correct position. This means that the pedestrian size close to the camera will be set high and thus clear away smaller detections and false alarms.

In the final combined system the methods are divided into three different setups depending on the ego-vehicle speed and outdoor illumination. When the truck is stationary the MoG (with a long learning rate) and the haar-classifier is switched on. The codebook model is also activated and constructs the background model. When the codebook method has completed the background model learning the MoG method is switched off and the codebook takes over its role. When the truck starts moving the Codebook method is then switched off and the MoG is

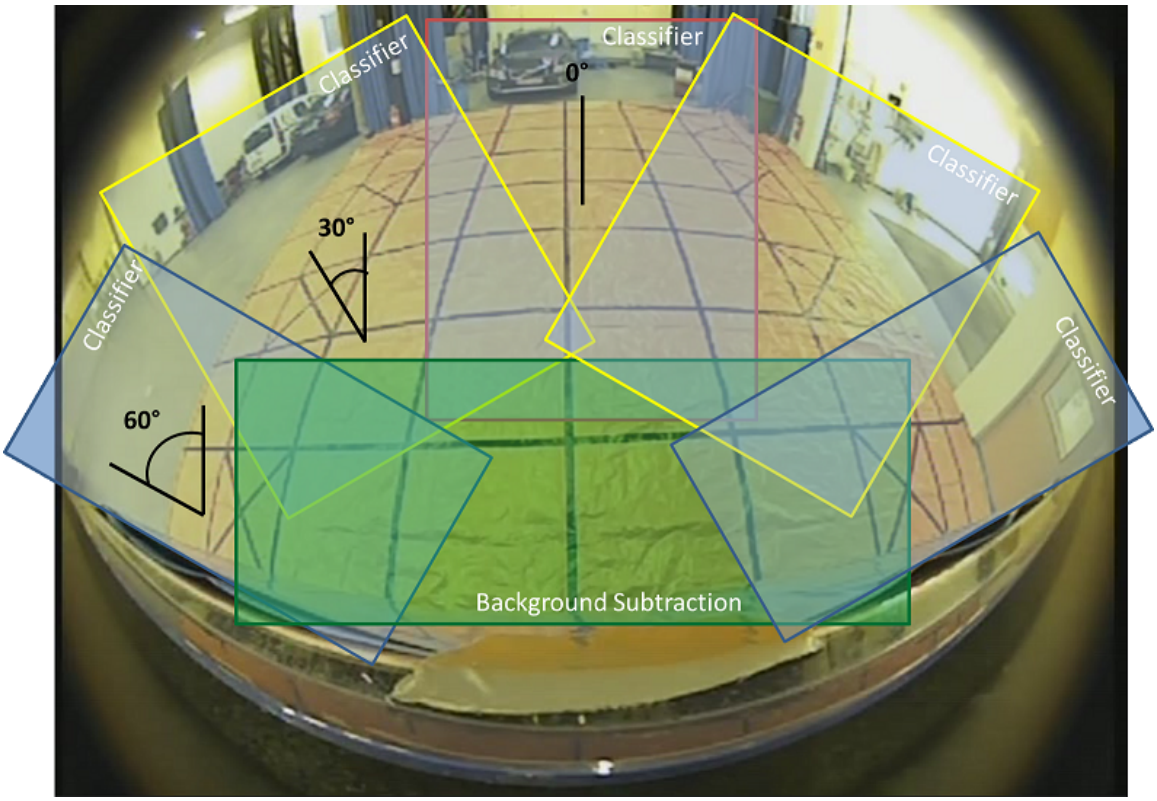


Figure 9.5: Classifier region setup and segmentation areas between the classifier and background subtraction method for the final system.

turned on again, but this time with a faster learning rate as to adapt quicker.

9.5 Fusion and Merging Multiple Detections

As seen in Figure 10.10 the output from the classifier results in many overlapping detections. Since the different classifier regions are overlapping as well as output from the background subtraction method will add further detection regions and give even more overlapping detection. A post-process is needed that combines overlapping detection and returns one final detection per pedestrian, both for the detected sub-windows that the classifier outputs and for all the candidate rectangles from each system. This is performed by clustering the rectangles using the rectangle equivalence criteria, which partitions the set of detection rectangles into disjoint subsets. A set of detections are in the same subset if they are similar in location and size, i.e. their bounding regions overlap. Detections α and β are a part of the same set if

$$\alpha_x - \frac{1}{5}\alpha_{width} \leq \beta_x \leq \alpha_x + \frac{1}{5}\alpha_{width} \tag{9.1}$$

$$\beta_y - \frac{1}{5}\alpha_{height} \leq \beta_y \leq \alpha_y + \frac{1}{5}\alpha_{height} \tag{9.2}$$

$$\frac{5}{6}\alpha_{width} \leq \beta_{width} \leq \frac{6}{5} \leq \alpha_{width} \tag{9.3}$$

The final rectangle from each partition is computed as the average of all input rectangles in the subset. For each merge set the number of neighbours are stored to approximate the confidence measure - more merged rectangles represents a higher likelihood that the region

contains a pedestrian. A threshold is set according to the number of neighbours needed for the region to be kept, eliminating detection regions with few merged rectangles. Also, detection within another detection that contains more neighbours will be erased.

9.6 Tracking

Tracking with the Kalman filter is activated if a pedestrian is detected in five consecutive frames. The state vector is then initialized from the last two frames $t-1$ and t described as:

$$X_0 = [x_t, y_t, h-t, w_t, (x_t - x_{t-1}), (y_t - y_{t-1}), (h_t - h_{t-1}), (w_t - w_{t-1})]^{t-1} \tag{9.4}$$

For the initial state X_0 the covariance matrix P_0 is initialized with larger values since it is updated iteratively. It is assumed that the center point of the bounding box has a $\pm$ 10 pixel deviation in x and y, the velocity has $\pm$ 5 pixels deviation, the height $\pm$ 5 and its velocity ± 3 pixels deviation, and the width $\pm$ 10 and its velocity $\pm$ 5 pixels (the width deviation is naturally larger than the height due to the pedestrian leg stride). The error covariance matrix P_0 is thus initialized as

$$P_0 = \begin{bmatrix} 10^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5^2 \end{bmatrix} \tag{9.5}$$

The state and measurement noise covariance matrices are initialized in the same manner. The standard deviation for the center point is set to 5 pixels and its velocity to 2 pixels, the standard deviation for the height is set to 3 pixels and its velocity to 1 pixel, and the standard deviation for the width is set to 5 pixels and its velocity to 2.

$$Q = \begin{bmatrix} 5^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2^2 \end{bmatrix} \tag{9.6}$$

$$R = \begin{bmatrix} 9 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 9 \end{bmatrix} \tag{9.7}$$

The correspondence for the pedestrian across frames is established by the distance measure of the bounding box described by equation (9.3). To eliminate as many false positives as possible the threshold for the detection system is set high. This will also lead to a higher number of missed detections, and it is the trackers job to fill in the gaps and connect the pedestrian position between detections. If the system is unable to detect a pedestrian the Kalman filter will continue to predict its position for the next 20 frames; taking the predicted position as measured position.

9.7 Test Sequences and Scenarios

All components were evaluated on numerous sequences with different difficulties. The test scenarios focused on the cases discussed in Chapter 1.2.1 and are limited according to the delimitations in Chapter 1.5.

To show the resulting performance of the trained detectors a ROC plot was produced from a set of 1300 hand labeled images containing pedestrians. The two different trained classifiers were tested on this set to display the value of the given amount of training data used.

9.7.1 Front Camera

Table 9.1 summarizes the sequences used for the front camera. The sequences are divided according to increasing difficulty, with a stationary truck for sequence 1 and 2, where sequence 2 contains a more challenging walking pattern, and sequence 3–5 contains a moving truck. There is also a segmentation made depending on where in the frame a pedestrian is situated based on the camera position according to **a)** critical ($\pm \approx 5$ m to the sides, and up to 2.5 m in the front) corresponding to the blind-spot region, **b)** middle (± 5 m to the sides, and 2.5-4.5 m in the front), and **c)** far (± 5 m to the sides, and above 4.5 m in the front).

The MOG and Codebook method will only be evaluated and compared against each other given a stationary truck (Sequence 1 and 2). Since the Codebook is not adapted to a moving camera it will not be evaluated for Sequence 3 and 4.

9.7.2 Side Camera

Similarly table 9.2 summarizes the test sequences used to evaluate the side camera system. Sequence 1–3 have a stationary truck, while sequence 4 have a moving truck. Sequence 1 contains a single pedestrians, and sequence 2–4 contain multiple pedestrians.

9.8 Performance Evaluation

This thesis considers a generic test scenario since it is a proof-of-concept and hence aims to evaluate the inherent potential of the pedestrian detection method. The matching criterion is based on a 2D bounding box overlap, as described in Chapter 8, since it has no prior scene knowledge. Furthermore, apart from practical feasibility, there is no constraints placed on the processing times that are allowed.

The pedestrian detection system performance is evaluated from the hit rate from the different test set sequences with ground-truth that has been manually hand labeled with rotated bounding box locations corresponding to the location of a pedestrian. Using simple bounding boxes allows for an inexpensive ground-truth annotation and has been carried out with a modification of the ViPER interface [11], as seen in Figure 9.6 To account for the pedestrian rotation across the scene a four point polygonal box model is used instead of the normal upright or rotated bounding boxes.

9.9 Applications

Among the numerous applications possible, a measure of the exact instantaneous distance from the truck to a pedestrian is one of the most important features for an automatic driver assistance system. This was implemented by utilizing the already calculated camera undistortion matrix to map the feet position of a detected pedestrian into this coordinate system to ensure a better (less radial distorted) interpolation to the real world position. An 8 by 7 meter grid pattern

Stationary Truck

Name	Description	Length [s]
Sequence 1:	A pedestrian walking and running straight across the scene, walking straight towards the camera, and walking diagonally across the screen.	97
Sequence 2:	Walking across the scene and stopping for 5 seconds and continues walking again. Walking across the scene and stopping in the middle tying shoelaces for 10 seconds and then continue walking.	39

Moving Truck

Name	Description	Length [s]
Sequence 3:	Pedestrian walking/running across the scene at different distances, start inhibit scenarios, and stationary pedestrians on sidewalks.	60
Sequence 4:	Same as Sequence 3 but the pedestrian has brighter clothes.	56
Sequence 5:	A drive in an urban environment.	230

Table 9.1: Test sequences for the front camera.

was placed on the ground (see figure 10.8a) and the four corner pixel points, together with their real world position, was used to calculate the perspective transform for each of the 56 squares. The perspective transform for the square which center point minimize the Euclidean distance to the pedestrian feet position is then used together with the calculated extrinsic parameters of the camera to transform the 2D pixel coordinates to the 2D real world x, y coordinates. The origin (0, 0) in the real world system is set to be in the middle of the front of the truck. The resulting distance is printed next to the detection box in each frame, and if -3 m $< x < +3$ m$\wedge y < 1.5$ m the detection box will be marked red as to alert the driver. For visualization purposes the current active systems is depicted with a filled green circle if activated, otherwise a red circle. If the codebook is activated but in the background learning phase, this is depicted with a flashing green/red circle. To see in what method state the system is the ego-vehicle motion is also depicted.

Name	Description	Length [s]
Sequence 1:	A single pedestrian walking around the scene. The truck is stationary.	158
Sequence 2:	Two pedestrians walking around the scene wearing bright clothes. The truck is stationary.	134
Sequence 3:	Multiple pedestrians walking around in the scene. Occlusion is occurent. The truck is stationary.	143
Sequence 4:	At most two pedestrians walking alongside the truck. The truck is moving.	129

Table 9.2: Test sequences for the side camera.

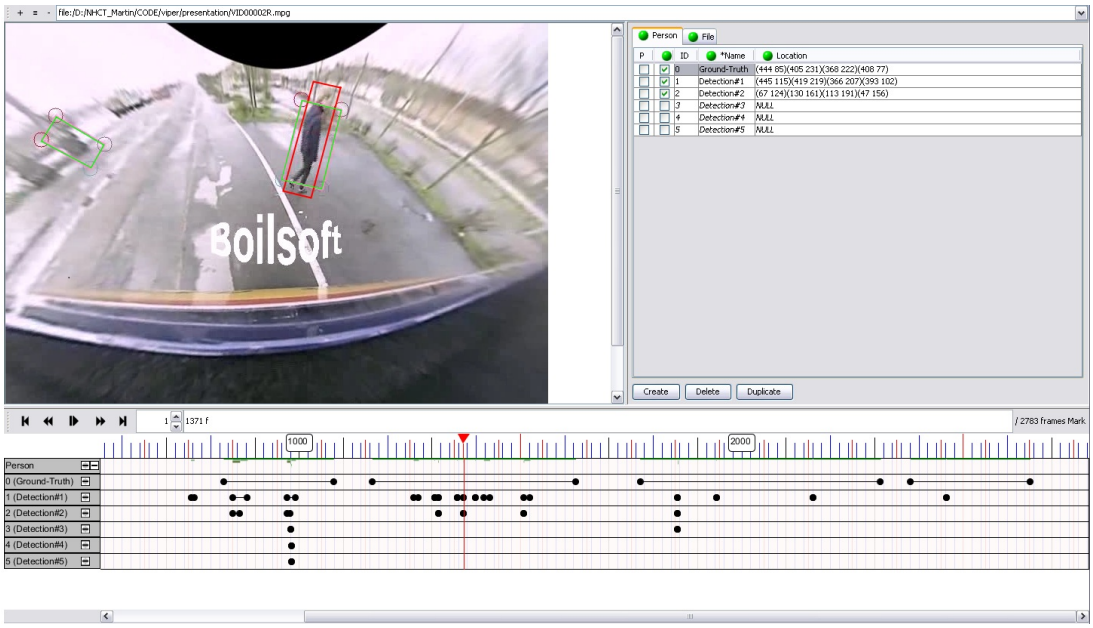


Figure 9.6: The toolbox for annotating the pedestrians in the ground truth (the red box) and also visualizing the detection outputs (green box).

Chapter 10

Results

The objective of this chapter is to present the results from the implemented algorithms discussed in previous chapters. First, the calibration process is evaluated, followed by the comparison of the background subtraction techniques, the classifier evaluation and undistortion comparison, the secondary classification method, the final system, and the extended final system for the side camera. Lastly, the time consumption and an example interface is presented.

10.1 Camera calibration

In Figure 10.1 a few of the used calibration images are shown. In Figure 10.2a we see the extrinsic parameters as a plot of every position used during calibration with the grid pattern, and in Figure 10.2b the final pixel error for each grid position. This gives a good display of the transition from the 2D pixel image in Figure 10.1 and their corresponding 3D real-world position in Figure 10.2a. In the calibration a fourth order polynomial was used. From Figure 10.2b we observe that this yields a error within ± 2 pixels of the real pixel value, where the more tilted grid patterns contain the highest error.

The resulting output after applying the radial distortion correction that results from the camera calibration of the image frame can be seen in Figure 10.3. Now the previously horizontal and vertical distorted objects appear straight, but at the cost of resolution loss at the boarders and stretched out features. The black border areas are a result of the correction procedure and can be avoided by changing the extrinsic zoom factor.

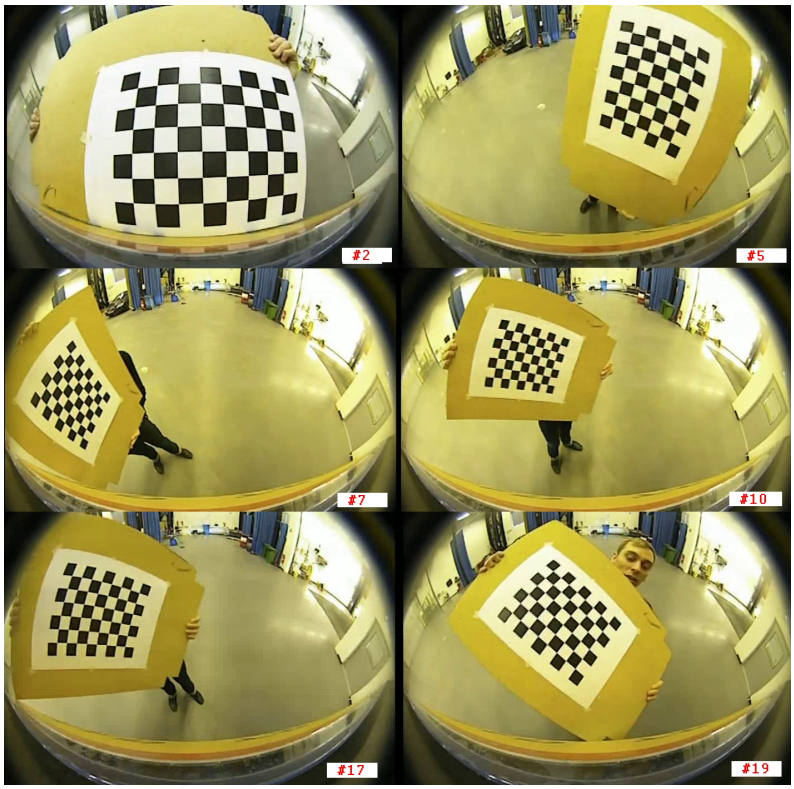
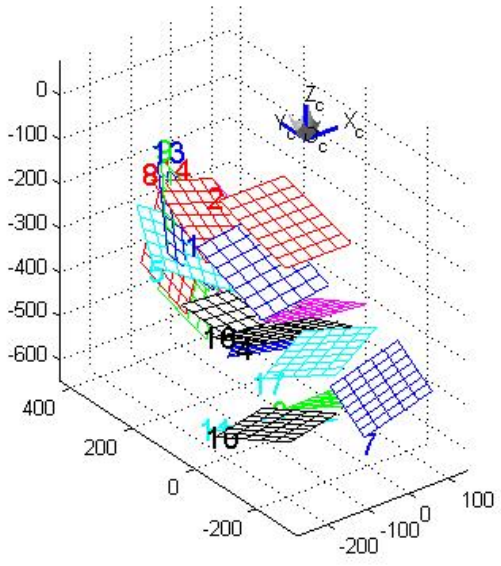
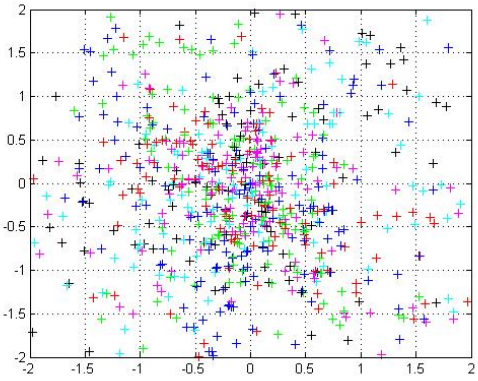


Figure 10.1: A few of the calibration images used.

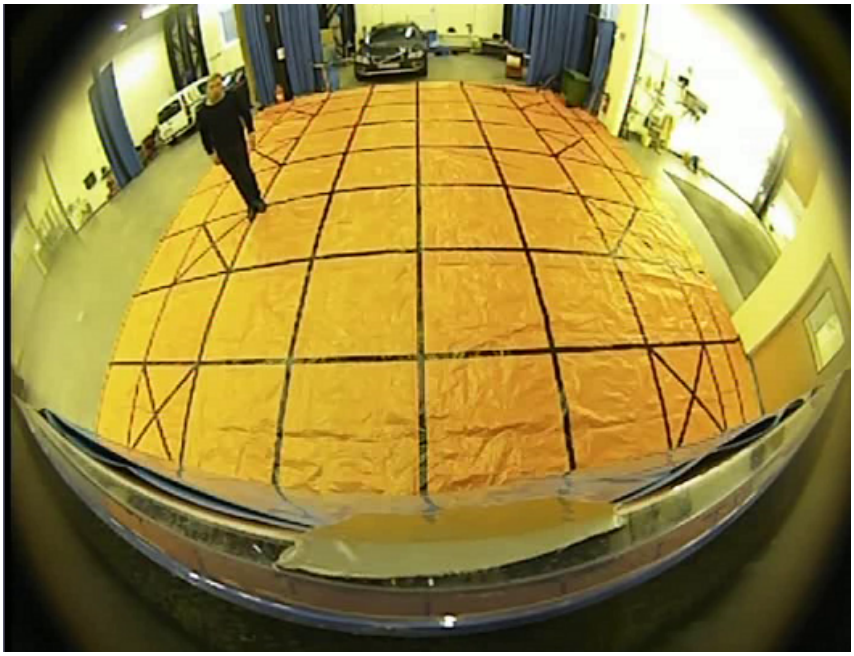


(a) The extrinsic camera parameters. The number next to each grid pattern can be connected to the images in Figure 10.1.

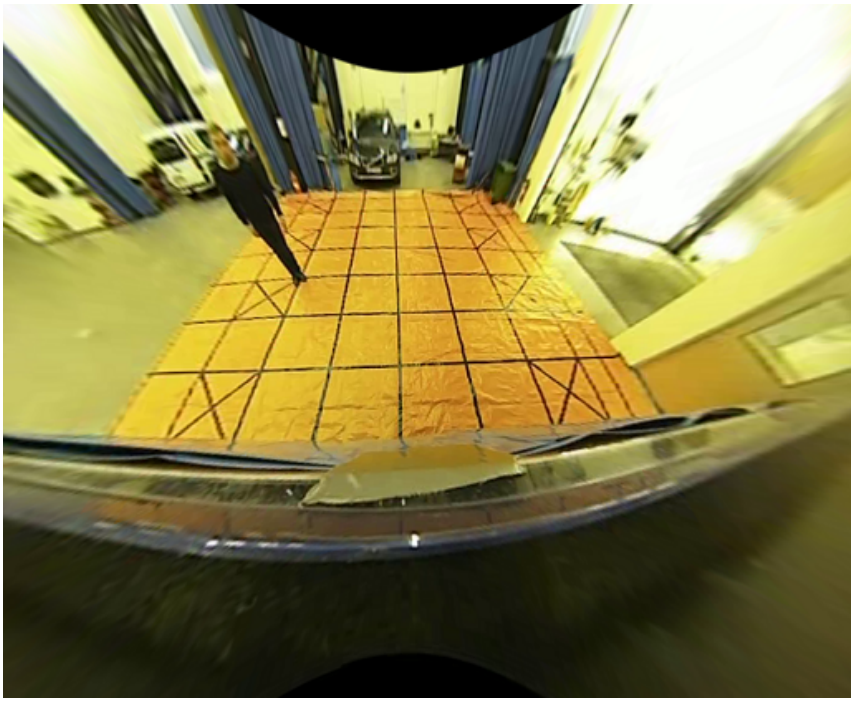


(b) Pixel error.

Figure 10.2: The resulting extrinsic parameters (a), and the pixel error (b) from the calibration procedure.



(a) Original frame



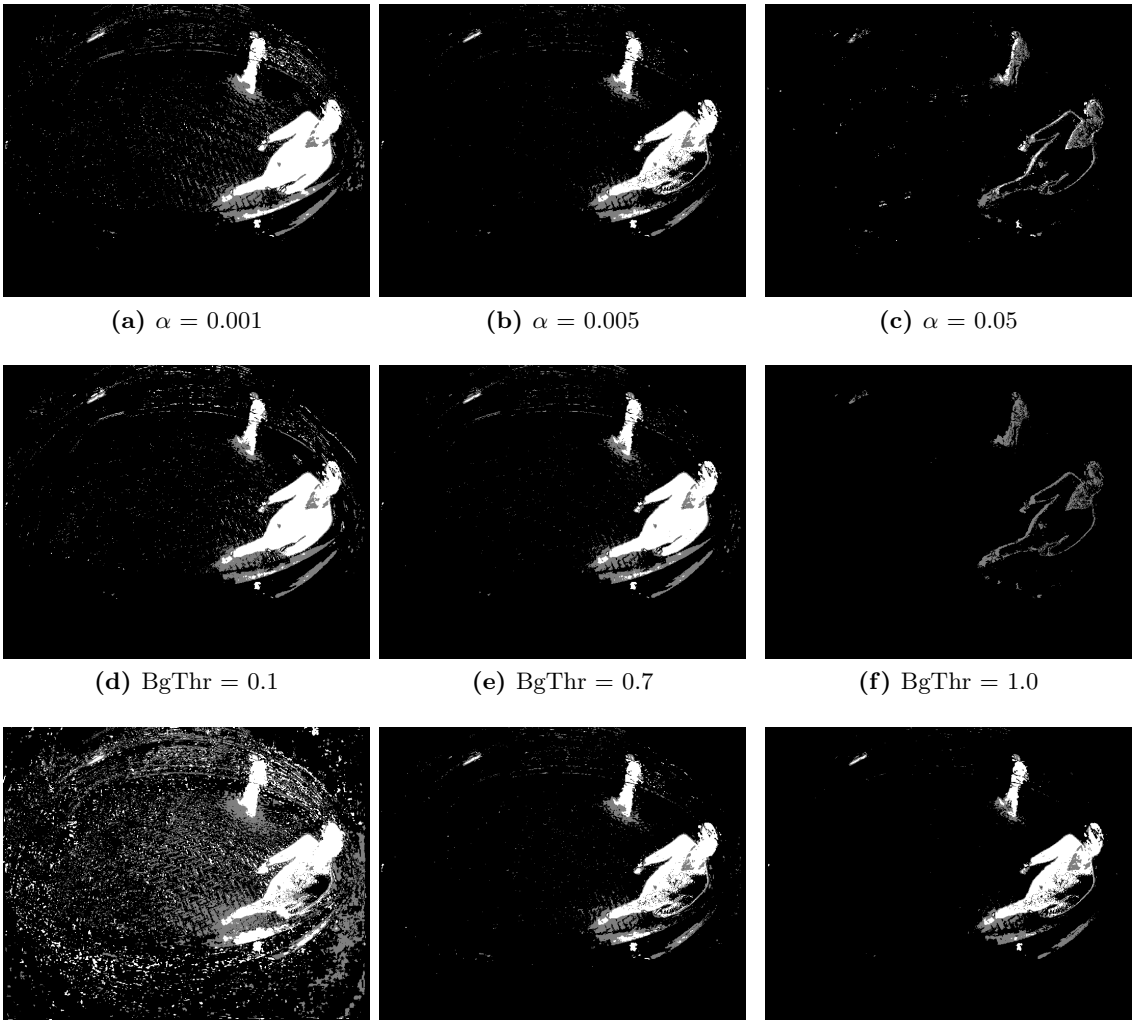
(b) Frame after radial distortion correction

Figure 10.3: Before and after the undistorting of a frame.

10.2 Background Subtraction

10.2.1 Mixture of Gaussians

As described in chapter 5.1 there are numerous parameters which controls the result of the foreground segmentation for the MOG. Figure 10.4 shows the result of varying three of the most important parameters: the learning rate α (defined as 1/window size), the background threshold (BgThr), and the standard deviation threshold (StdDevThr). First thing to notice is the grey area which corresponds to lower intensity regions such as the shadow from a person and can be filtered out with a intensity threshold. From subfigure 10.4a – (c) we see that for a higher α value a foreground object is more quickly considered as a background object. Subfigure 10.4d–(f) shows that a lower background threshold includes more noise, but a too high value yields an incomplete segmentation. Lastly, subfigure 10.4 (g) – (i) shows that setting a lower standard deviation threshold gives a large amount of noise, and a higher value on the contrary gives little noise but still keeps the foreground object intact.



(a) $\alpha = 0.001$ **(b)** $\alpha = 0.005$ **(c)** $\alpha = 0.05$

(d) BgThr = 0.1 **(e)** BgThr = 0.7 **(f)** BgThr = 1.0

(g) StdDevThr = 1.5 **(h)** StdDevThr = 3.5 **(i)** StdDevThr = 5.5

Figure 10.4: Different Mixture of Gaussian parameters and their affect to the foreground segmentation. (a) - (c) describes changes in the learning rate α; (d) - (f) describes changes in the background threshold; (g) - (i) describes changes in the standard deviation threshold.

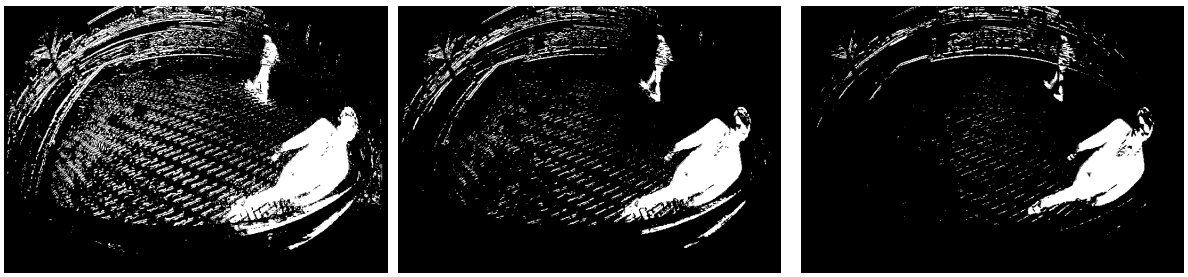
From extensive analysis performed as above the learning rate was set to 0.005, the number of gaussians to 5, the background threshold to 0.7, and the standard deviation threshold to 4.5.

(a) Foreground segmentation **(b)** Connected components **(c)** Detection output

Figure 10.5: Mixture of Gaussians foreground segmentation (a), connected components (b), and output detection result (c).

10.2.2 The Codebook

As described in chapter 5.2 the codebook method depends on the low and high intensity threshold of the color axes to separate foreground objects from the background. In figure 10.6 these threshold values are varied according to low minimum and maximum values (a), mid ranged values (b), and high values (c). In subfigure (a) we observe full object detection but includes a lot of noise, in subfigure (b) the foreground objects are still segmented correctly and the noise is significantly reduced, and in subfigure (c) the noise is reduced further but the objects has lost some of its information. As a result, a medium ranged value biased towards the low side is

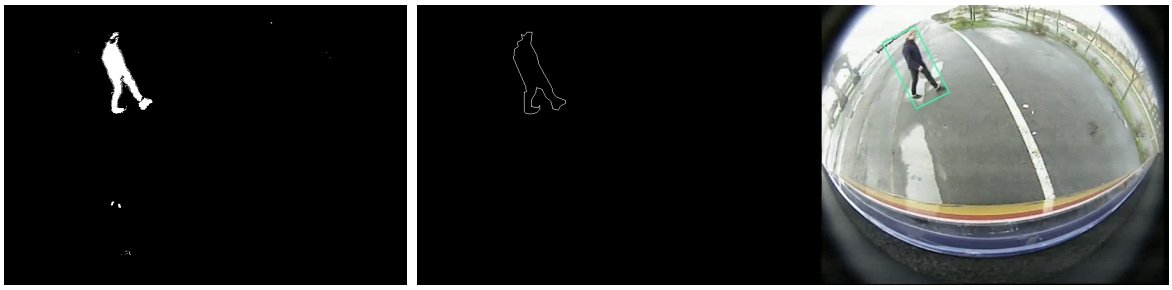


(a) Low minimum and maximum threshold. **(b)** Mid valued minimum and maximum thresold. **(c)** High minimum and maximum threshold.

Figure 10.6: The Codebook method changing the the intensity parameters for foreground segmentation. (a) corresponds to low minimum and maximum threshold, (b) to mid values, and (c) to high minimum and maximum thresholds.

chosen for the minimum and maximum threshold values since it is more important to keep as much information about the foreground object as possible in contrast of eliminating noise which can be filtered out in subsequent steps.

Figure 10.7 shows the foreground segmentation (a) and the resulting connected components (b) with the final detection output (c). It shows minimal noise and a good capture of the entire pedestrian.



(a) Foreground segmentation **(b)** Connected components **(c)** Detection output

Figure 10.7: The Codebook foreground segmentation (a), connected components (b), and output detection result (c).

10.2.3 Comparison

Table 10.1 presents the results from evaluating the two background subtraction methods for the sequences 1–2, with a stationary truck. The Codebook method outperforms the MoG over all detection ranges with no missed detections and a very low number of false positives. Since the Codebook is not adapted to a moving camera it is not evaluated for sequence 3 and 4.

Sequence 1		True Positives	False Negatives	False Positives
MOG	Near	765	0	21
	Middle	860	24	0
	Far	743	20	0
Codebook	Near	765	0	12
	Middle	884	0	0
	Far	763	0	0

Sequence 2		True Positives	False Negatives	False Positives
MOG	Near	468	4	24
	Middle	502	7	0
Codebook	Near	472	0	1
	Middle	509	0	1

Table 10.1: Comparison between the MOG and the Codebook background subtraction method. Both methods show good results for near field detection, with the Codebook as a superior method.

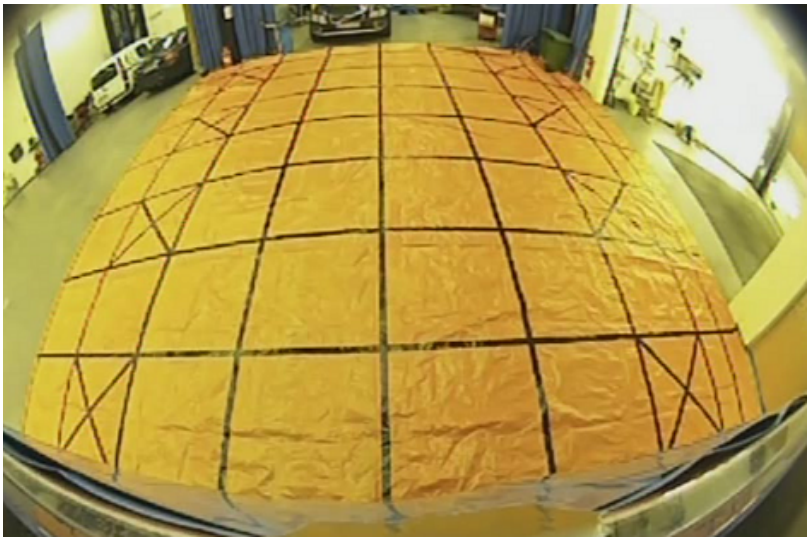
Sequence 3		True Positives	False Negatives	False Positives
MOG	Blind zone	292	4	74
	Outside	797	399	462
Sequence 4				
MOG	Blind Zone	657	70	198
	Outside	510	150	218

Table 10.2: Resulting performance for the Mixture of Gaussian method for sequence 3 and 4. It shows a better detection performance for the blind zone region than outside.

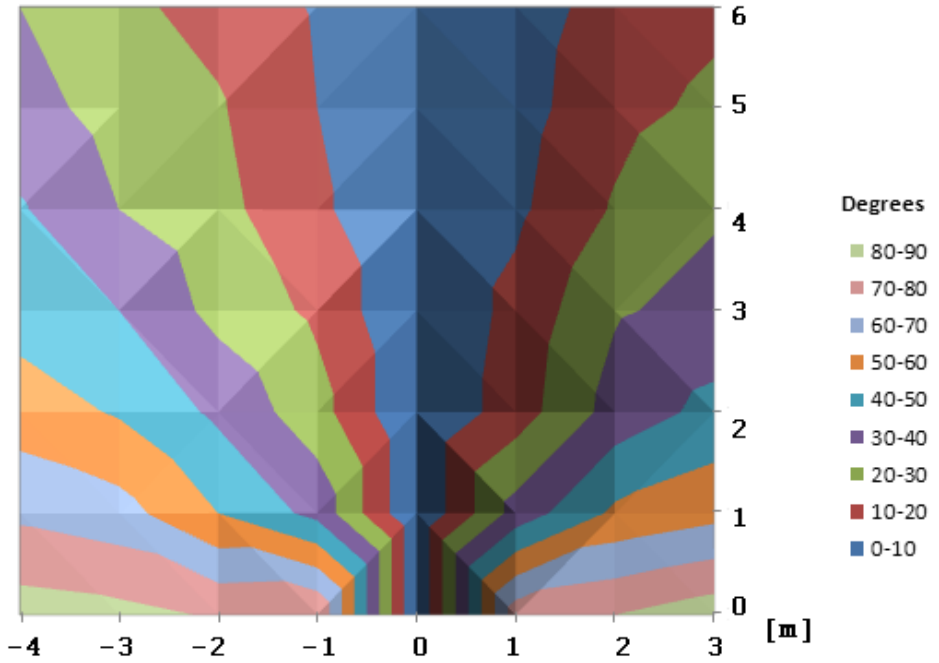
10.3 Classifier Evaluation

10.3.1 Classifier Detection Setup

Figure 10.8a shows the evaluation grid that was used to determine the vertical rotational deviation for a pedestrian at each grid position. In figure 10.8b the measured vertical rotation deviation for a pedestrian is then plotted, where (0, 0) is the position of the middle of the front truck. We observe that a vertical displacement close to the truck will give a large deviation in the rotation, and further away in the scene the rotational deviations occurs slower.



(a) The 8 by 8 meter grid pattern used for evaluation.



(b) The rotation distribution in the grid pattern.

Figure 10.8: Based on the grid pattern in (a) the distribution of a pedestrians in-plane angle deviation was measured and is plotted (b).

To find the optimal segmentation of the frame for the classifier, the detection range as a

function of pedestrian rotation was found. For steps of 15° rotation of the entire image, the rotational deviation range was noted for where the classifier was able to make a successful pedestrian detection. The results from this analysis are found in table 10.3. We see that the classifier can still detect a pedestrian rotated about 20 degrees in either direction. A decreased detection range is observed for higher rotational degrees due to the increased distortion for these border positions. This result implies that an image rotation step of 30° will be able to give good accuracy, which agrees well with the findings of Kölsch and Turk [42].

Image rotation	Detection range
0°	±26°.
15°	±23°
30°	±22°
45°	±22°
60°	±21°
75°	±19°
90°	$-15° \rightarrow 0°$

Table 10.3: Pedestrian detection range for a classifier at different image rotations.

10.3.2 Detector Cascade

The final detector is a 18 layer cascade of classifiers with a total amount of 154,190 features used. The two trained classifiers were each run on one overall test sequence and the resulting ROC curves plotted in Figure 10.9. Each point on the curve corresponds to an increase by one in the number of neigbours needed for a candidate rectangle to be retained. The best overall classifier is the one with a larger training sample size. The optimum number of neighbours to satisfy a high detection rate and low false positives can be seen to be around 16 neighbours.

The resulting bounding boxes from a single frame computation with the classifier is seen in Figure 10.10. We observe a large certainty with many detections for the pedestrians and a low amount of false positives. In the middle of the scene the overlapping detection regions can be seen as both vertical and rotated rectangles appear correctly.

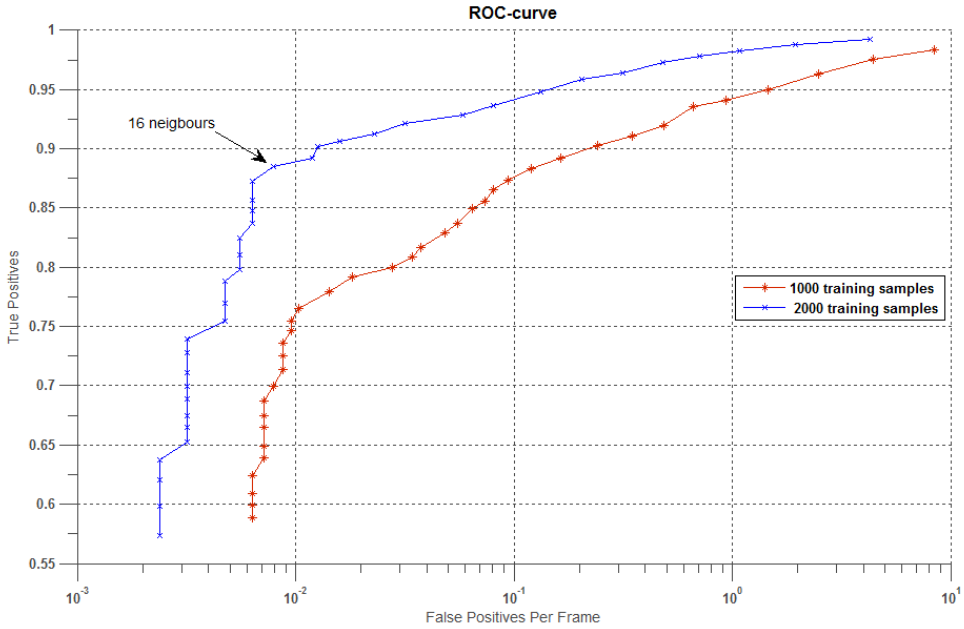


Figure 10.9: Classifier Receiver Operating Characteristic (ROC) curve for two different trainings samples sizes

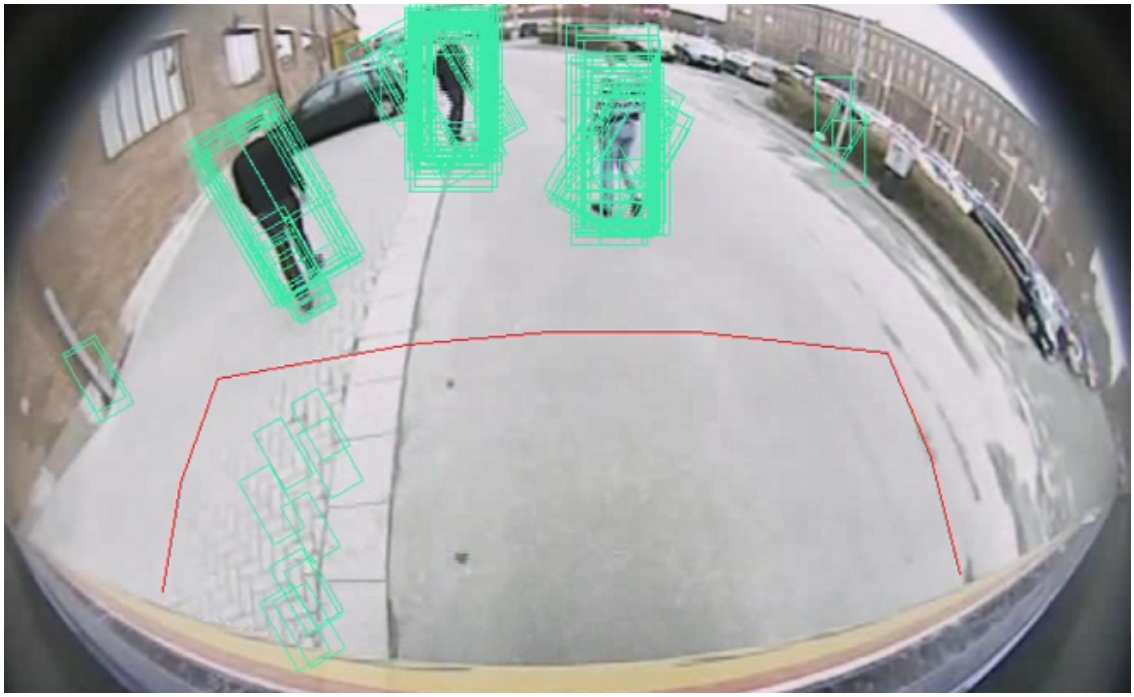


Figure 10.10: All bounding boxes from possible pedestrians deteted with the haar-classifier.

10.3.3 Undistortion Comparison

In table 10.4 the detection result is shown for the cascade classifier described earlier given all four test sequences. This initial results shows promising performance with high accuracy and a reasonably low number of false alarms. As expected the classifier fails at the near regime of the truck where a pedestrian is significantly distorted as well as at the more difficult scenarios during sequence 4. This result is compared against the undistorted camera output also presented in the table. We see that undistorting the image does not help the detection accuracy (number of true positives) and also leads to a significant increase in false positives. The reason for the high number of false alarms is because objects that used to be slightly crooked, such as lamp posts or trees, now appears straight and resembles more a pedestrian and therefore falsely gets classified as a pedestrian by the classifier.

Sequence 1		True Positives	False Negatives	False Positives
Distorted	Near	576	189	96
	Middle	881	3	6
	Far	735	28	9
Undistorted	Near	437	328 (86)	64
	Middle	876	8 (0)	2732
	Far	726	37 (0)	15

Sequence 2		True Positives	False Negatives	False Positives
Distorted	Near	116	356	11
	Middle	507	2	4
Undistorted	Near	147	325 (29)	5
	Middle	499	10 (0)	1020

Sequence 3		True Positives	False Negatives	False Positives
Distorted	Blind Zone	221	75	33
	Outside	1163	33	31
Undistorted	Blind Zone	272	24 (3)	12
	Outside	1119	77 (28)	776

Sequence 4		True Positives	False Negatives	False Positives
Distorted	Blind Zone	173	550	193
	Outside	292	372	149
Undistorted	Blind Zone	142	585 (18)	266
	Outside	333	327 (12)	941

Table 10.4: Comparison between the undistorted and the distorted setup for the classifier. The numbers in parenthases represent the number of pedestrians not possible to detect due to the extrinsic zoom.

10.3.4 The Fastest Pedestrian Detector in The West

In table 10.5 the result for the FPD classifier is presented for all sequences. It clearly outperforms the standard haar classifier in detection accuracy, and equally has a much lower amount of false positives. However, as for the earlier classifier it fails to detect the distorted near field region.

Sequence 1	True Positives	False Negatives	False Positives
Near	500	265	5
Middle	883	0	1
Far	742	21	0

Sequence 2	True Positives	False Negatives	False Positives
Near	120	360	2
Middle	509	0	1

Sequence 3	True Positives	False Negatives	False Positives
Blind Zone	105	191	1
Outside	1165	31	2

Sequence 4	True Positives	False Negatives	False Positives
Blind Zone	97	626	1
Outside	460	204	3

Table 10.5: Results for the FPTW classifier for sequence 1, 2, 3, and 4.

10.4 Combined system

The results of the background subtraction and classifier based final combined system setup as described in Chapter 9.5 is presented in table 10.6. All thresholds have been further increase in all systems to reduce the number of false positives at the cost of a few more missed detections. The system shows high accuracy in all detection regions and now has a significantly lower amount of false positives.

Sequence 1	True Positives	False Negatives	False Positives
Near	764	1	28
Middle	883	1	0
Far	734	29	0

Sequence 2	True Positives	False Negatives	False Positives
Near	472	0	0
Middle	501	8	0

Sequence 3	True Positives	False Negatives	False Positives
Blind Zone	265	31	10
Outside	1154	42	10

Sequence 4	True Positives	False Negatives	False Positives
Blind Zone	493	172	116
Outside	565	123	98

Table 10.6: Combining the background subtraction methods and the classifiers.

10.5 Tracking

As motivated in chapter 9.6 all thresholds have been increased, leading to a much lower amount of false positives, but also to a higher degree of missed detection that the Kalman tracking method will fill then fill in. In table 10.7 the results are shown for the complete system with the added Kalman tracking. As expected, all sequences and for all pedestrian positions show a higher degree of correct detections. The visual result from the tracking phase can be seen in Figure 10.12 where the tracking history is shown as a green line that represents the center point for the tracked pedestrian.

Sequence 1	True Positives	False Negatives	False Positives
Near	765	0	14
Middle	884	0	0
Far	758	5	0

Sequence 2	True Positives	False Negatives	False Positives
Near	472	0	0
Middle	509	0	0

Sequence 3	True Positives	False Negatives	False Positives
Blind Zone	286	10	11
Outside	1167	29	0

Sequence 4	True Positives	False Negatives	False Positives
Blind Zone	576	89	80
Outside	629	59	44

Table 10.7: Detection results for the final system with added tracking.

10.6 Side-Camera

In Figure 10.11 two detection frames are shown from the side camera detection evaluation. For both frames a metric grid is also plotted for distance clarification. Here we can clearly observe the different aspect ratios of the bounding boxes depending on the distance from the container side. The results show smooth detection between the different classifier and rotation regions without any loss of generality for different pedestrians appearances and poses.

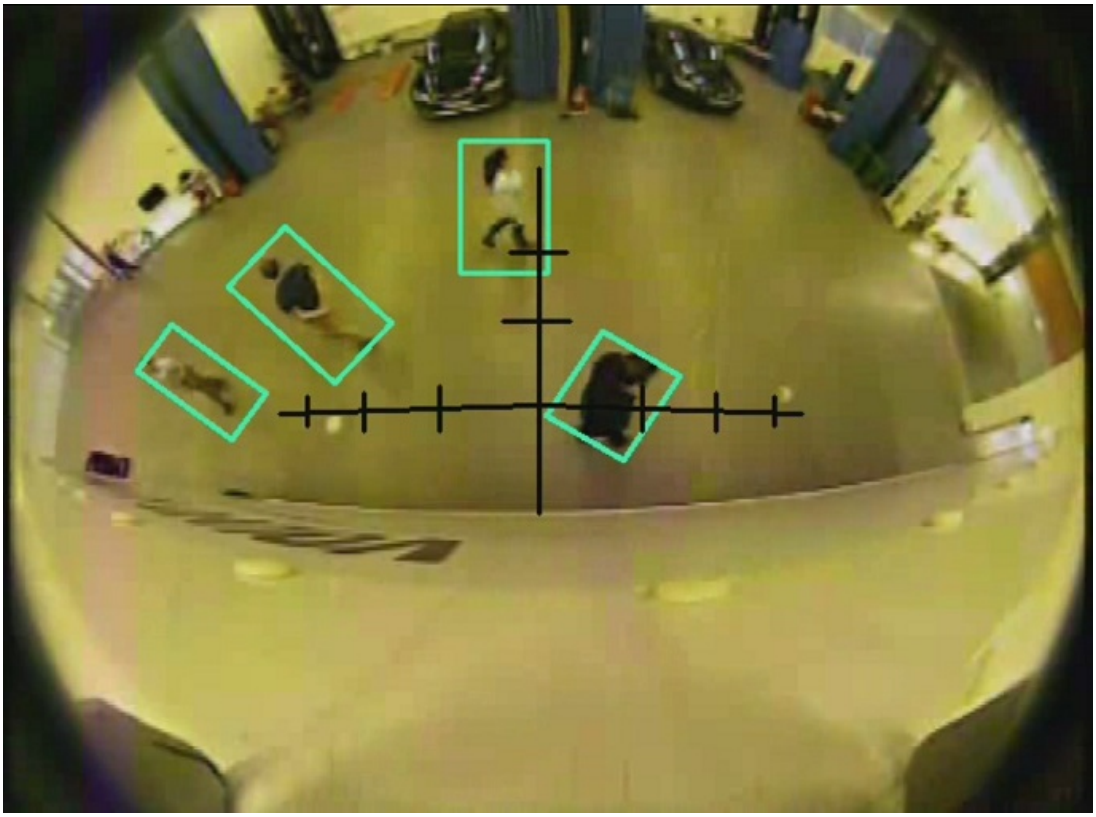
	True Positives	False Negatives	False Positives
Sequence 1	3201	83	120
Sequence 2	4920	645	151
Sequence 3	12336	1503	319
Sequence 4	3662	946	1198

Table 10.8: Detection results for the side camera.

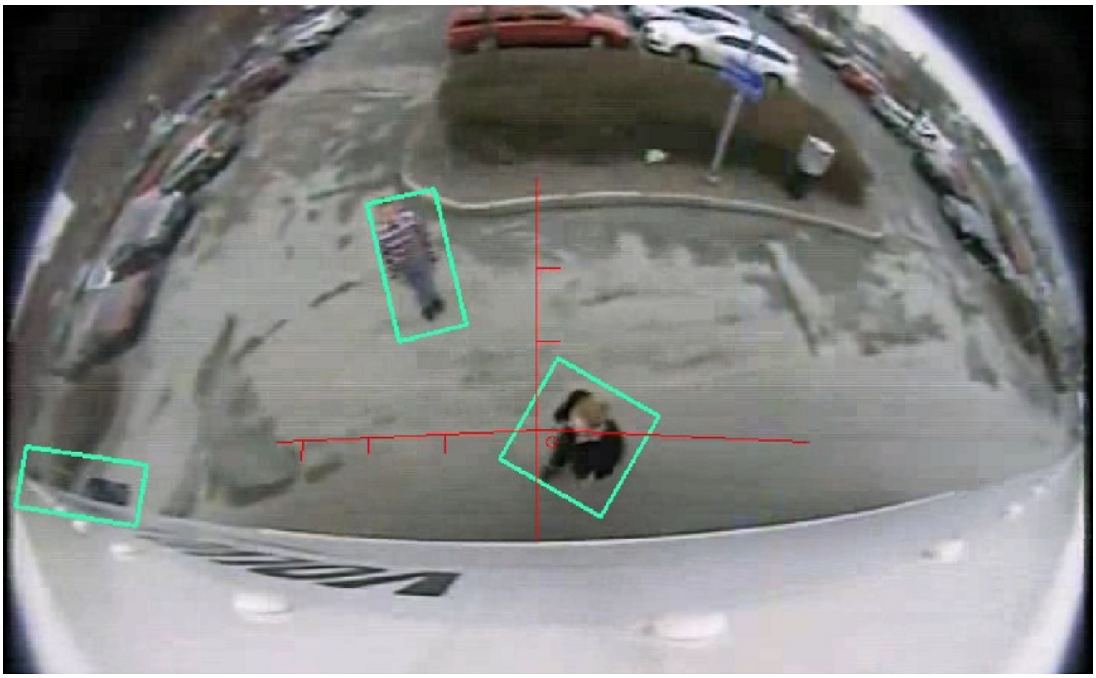
Table 10.8 presents the frame hit detection results for the final side camera setup for all the test sequences. Sequences 1-3 show a high detection rate and a low number of false positives. For sequence 4 we see that it misses about every fifth detection. This is because one person in the scene is wearing black (almost always detected), and the other is wearing white (rarely detected), and the environment and illumination is also very bright (see figure 10.11b). For all sequences a detection radius up to 6 meters was possible.

The false alarms that are observed appear in the blending zone between the different classifier regions. The classifier only sees half a person and mistakenly labels the bottom half as a pedestrian. This could be resolved by setting up a better allowed bounding box values at the border of a classifier region. The rest of the observed false alarms are due to the substantial noise present from the camera throughout all the sequences. This noise originated from a lack of power supply and if corrected could give even further increase in accuracy.

Note that no tracking was used to obtain these results, so the accuracy can be improved further including this - since the false negatives appear continuously during testing that makes it easier to ”fill in the gaps”.



(a) Detection result from Sequence 3.



(b) Detection result from Sequence 4.

Figure 10.11: Detection examples for the side camera. Subfigure (a) shows an detection frame from Sequence 3, and subfigure (b) shows a detection frame from Sequence 4.

10.7 Time Consumption

In table 10.9 the time consumption for the various stages of the foreground-background subtraction methods are listed.

Time measure [ms]	MOG	Codebook
Initialization	39.2	54.4
Learn background	0	29.6
Create Background Model	–	27.9
Segmenting BG and FG	47.5	13.8
Foreground cleanup and CC analysis	13.3	8.83

Table 10.9: Time consumption for the Mixture of Gaussians and the Codebook[s].

In table 10.10 the runtime effect of restricting various aspects of the classifier can be seen. The window step size is set as to capture 75% of the size of the smallest pedestrian and scales by a factor of 1.1. The pedestrian size restriction is set as to detect the smallest and largest possible pedestrian in the scene and sizes outside of that can be discarded. Similarly, the scene restriction is set so the window only searches the locations where a pedestrian might appear, i.e. we can discard the sky and other unrealistic locations. Adding these restrictions gives a speedup by about a factor of 20 without loosing any detection accuracy. Note that the time consumption for the classifier is non-fixed since it is dependent on how many stages are evaluated for each subregion. If an image region contains many objects that resembles pedestrians it will go through more stages in the cascade that increase the computation time.

	Time [ms]
Baseline	2570
Restricting Window step size	826
Restricting pedestrian size	1160
Restricting Scene	1010
Final setup	143

Table 10.10: [s].

The runtime for the different steps of the described approaches and the final system, using the V&J classifier, can be seen in table 10.11.

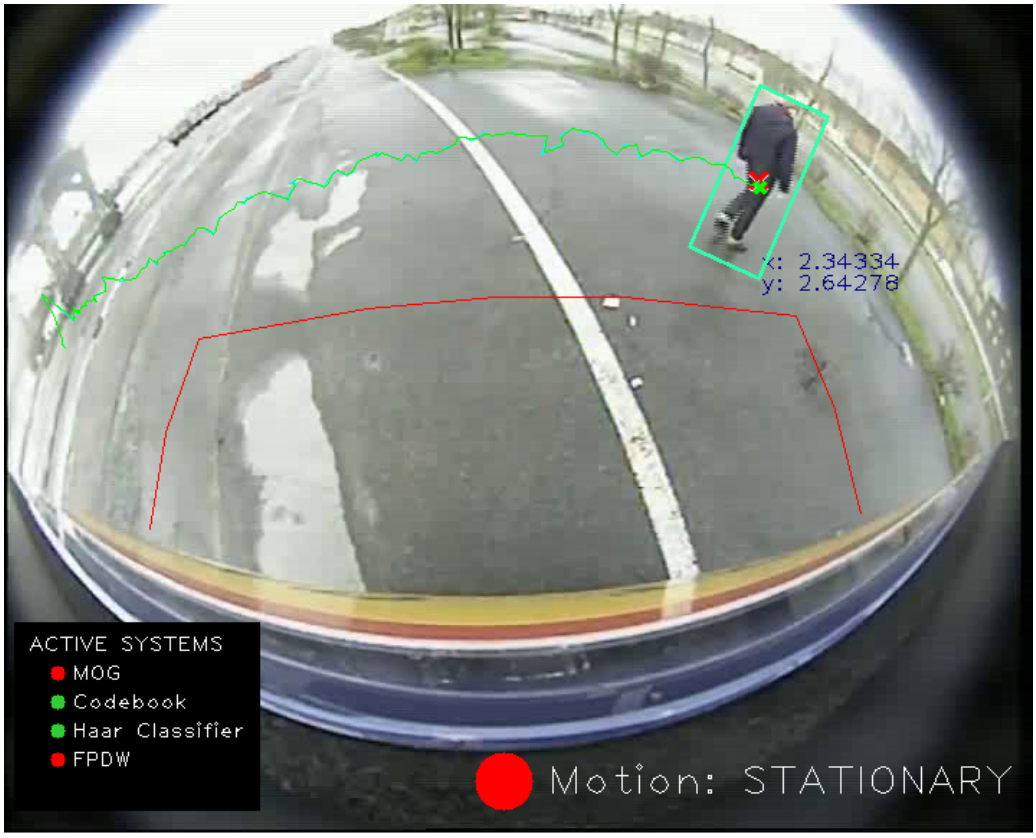
10.8 Application and interface example

An example of an interface for the front camera when the truck is stationary is shown in Figure 10.12, and when the truck is in motion in Figure 10.13. The detected pedestrian is marked with a green rotated rectangle when outside of the blind-zone region, and marked with a red rotated rectangle when positioned inside the blind-zone region. The critical blind-zone boarder is shown with red lines. Next to the bounding box rectangle is the x and y distance from the midpoint of the truck to the midpoint of the bottom edge of the bounding box- with the positive x-axis to the right. The motion of the vehicle is displayed at the bottom right and divided into three

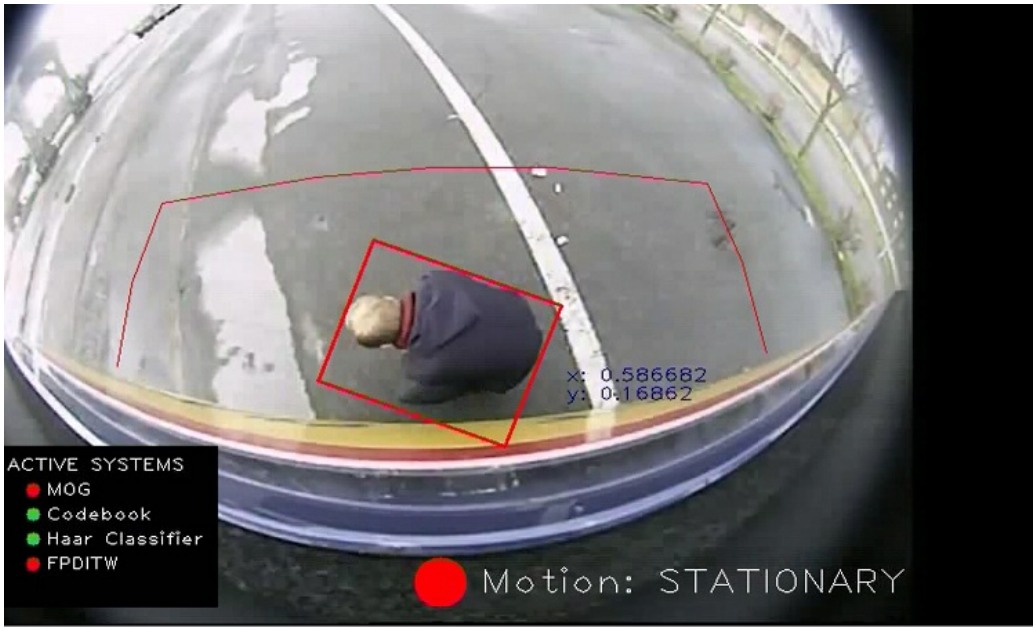
Function	**Time [ms]**
Background Subtraction	42.8
Image rotation	89.7
Classifier (V&J)	83.5
Classifier (FPDW)	19.8
Tracking	–
Total	228

Table 10.11: Time consumption of the main bottlenecks and the final running time. [ms].

segments: stationary (2 < km/h), slow (2 - 10 km/h), and fast (10> km/h). The current active methods are shown with a green marking at the bottom left. When a method is initializing it is shown blinking in red and green.



(a) Tracking with history displayed.



(b) A pedestrian is kneeling in the critical area.

Figure 10.12: Example of the interface when the truck is stationary. It displays the pedestrian position in relation to the front of the truck, the vehicle motion, and the activated methods. In (a) Kalman filter tracking with history is displayed for a pedestrian crossing the scene from left to right. In (b) a pedestrian is kneeling in the critical blind zone region.

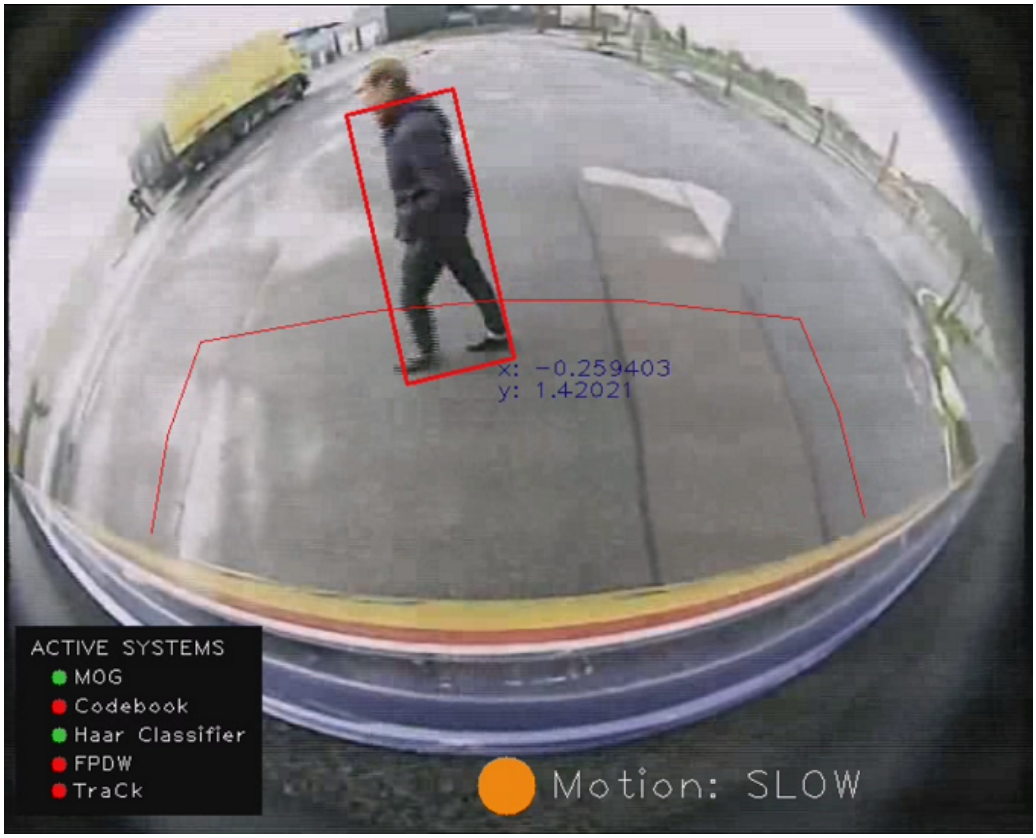


Figure 10.13: Example of the interface when the truck is in motion.

Chapter 11

Discussion and Future Work

This chapter discusses the obtained results and compares it to the research questions and hypothesis, and provides a ground for future research.

11.1 Overview

The results obtained in this master's thesis are in line with previous research and have a promising outlook. The reason for why there is a lower detection rate in certain situations is that the shape of the pedestrians used for training are not equivalent to the pedestrians used in the testing for the classifier. The general conclusion for the field of automatic pedestrian detection is that there has been tremendous progress in the last decade, however there is still a long way to go for a ideal system. This can be seen from Figure 11.1 that depicts an evaluation of detectors for smaller scale pedestrians (a) and with heavier occlusion (b), where minimal detection is observed.

11.2 Improve Accuracy

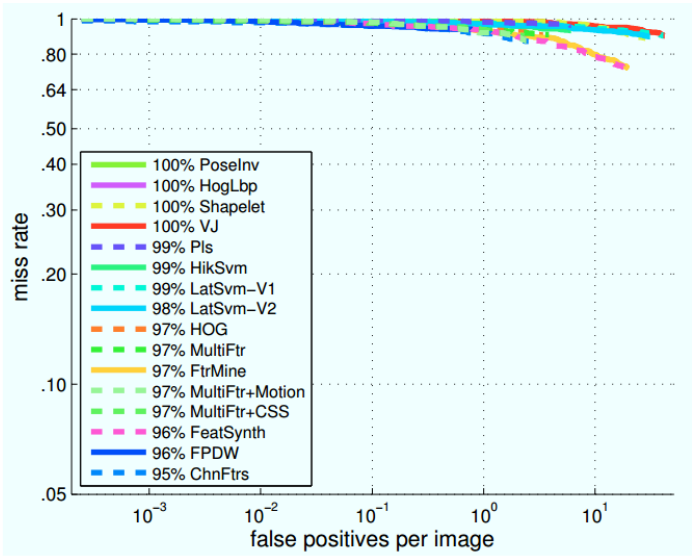
11.2.1 Increased training data

As shown, more training data gives better performance. The trained detector used a very small amount of training data, so using a much larger dataset should boost performance. This can be seen from Figure 11.2.

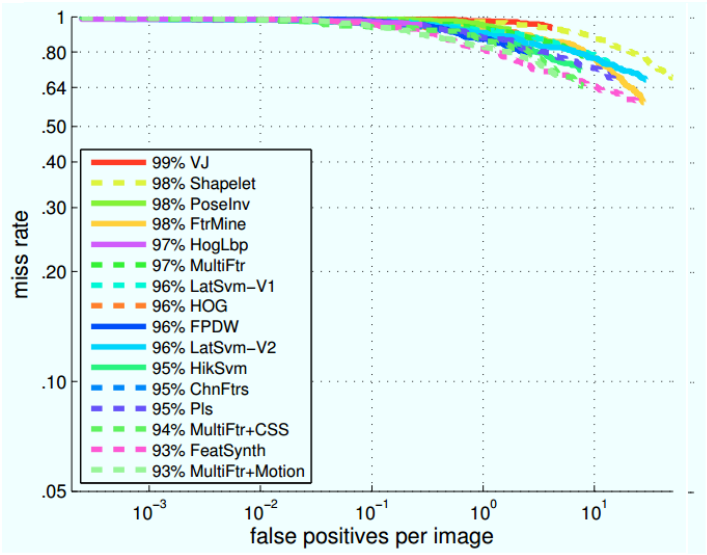
11.2.2 Alternative Techniques

To further reduce false positives the ground plane assumption, especially at lower resolutions, could be extended to more advanced approaches to utilize context better. The system has been focused on detection in the near vicinity of the truck, therefore small scale pedestrians in the below 100 pixel range will be needed to be addressed. This is important when the vehicle is moving at higher speeds since if detection is made to close there is no time for the driver or the avoidance system to react. The current feature representation is tolerant to scale variations but not to intensity variations, which limits its performance over different lighting and pedestrian appearance conditions.

The background subtraction approach used in this thesis is not purposed for a moving camera, but a static camera. To generalize this to a moving camera the deviation from the observed optical flow from the ego-motion flow field could be calculated, assuming a translatory camera motion [18, 67].



(a) Small scale pedestrians.



(b) Heavy occlusion.

Figure 11.1: Detector results here the state-of-the-art detectors break down. (a) shows far-scale pedestrian detection and (b) shows heavily occluded detection.

11.2.3 Increased Field-of-view by Sensor Fusion

Using a camera system as sensor have many benefits due to the strong visual features, plentifulness of texture and color cues, and good spatial resolution. However, the performance can easily be degraded due to cluttering and varying illumination. It is therefore important to include additional sensors to complement each other and provide a stronger overall system performance. Though the aim of this thesis was to see what is possible to achieve by only using a single camera system at two different height positions, in a real scenario multiple cameras would be used. There is also the question of what to do if the camera fails. Examples of additional sensors could be active sensors such as ultrasonic or radar sensors - radar has a better detection on larger distances compared to the camera system.

As seen from the accidents scenario displayed in Figure 1.1 many accidents, especially with

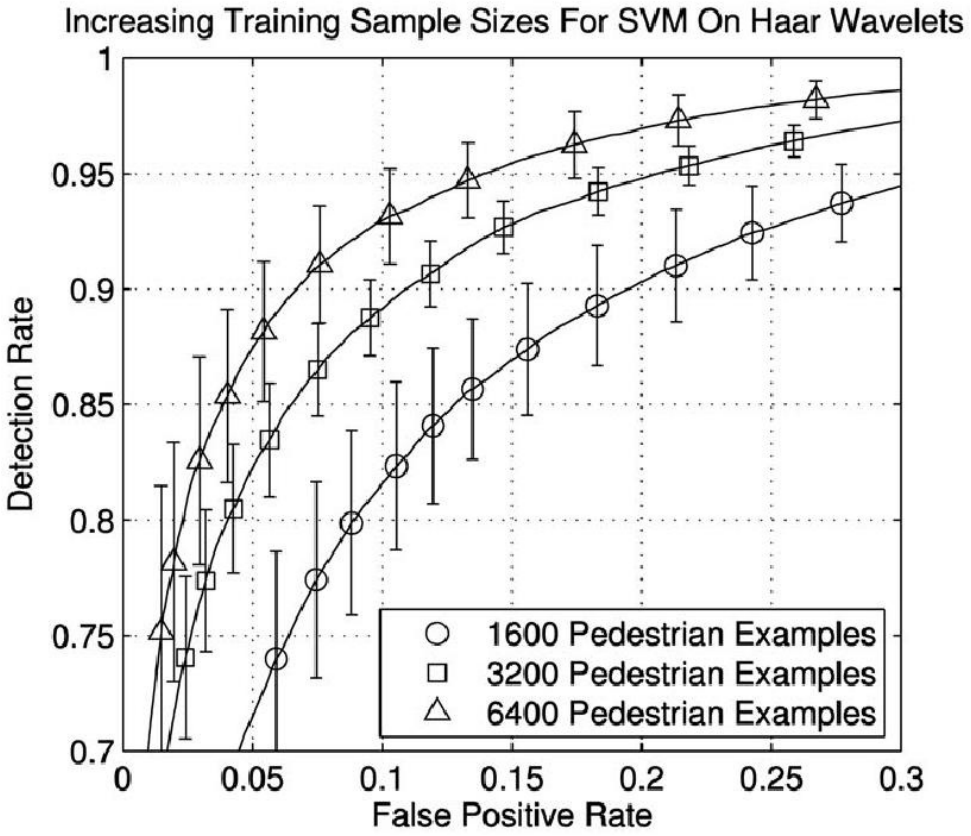


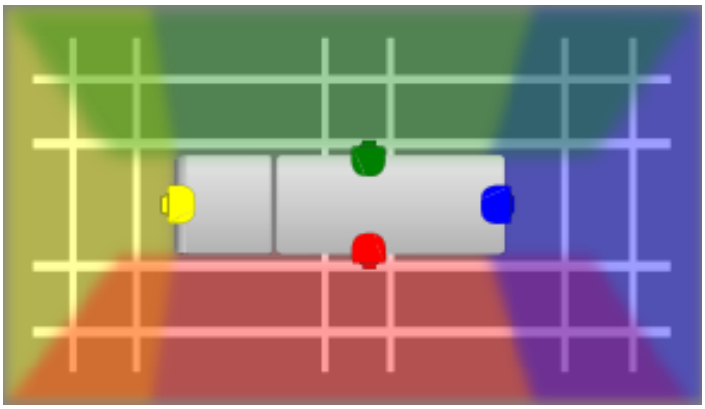
Figure 11.2: ROC for various training sample sizes. Increasing the number of training examples gives better detection.

bicyclists, occur when the truck is turning right. To prevent these accidents the system needs to detect at the vicinity of the camera range where the distortion is severe. To increase the accuracy in these regions a data fusion step is needed to connect the information between the front camera and side camera. To increase the field-of-view a camera system can be placed on every edge of the truck container to provide a 360 degrees surround view system. By doing image warping and stitching a birds-eye view projection can be made by. Figure 11.3 shows some initial results to this approach for the same camera system and truck as used in this master's thesis. To avoid blind wedge-shaped voids over the stitching between the camera projection areas a blending stitching has been used instead of a normal simple stitching (see Figure 11.4). Further views such as "tail gate" to visualize the pick up area of the truck should also be researched further.

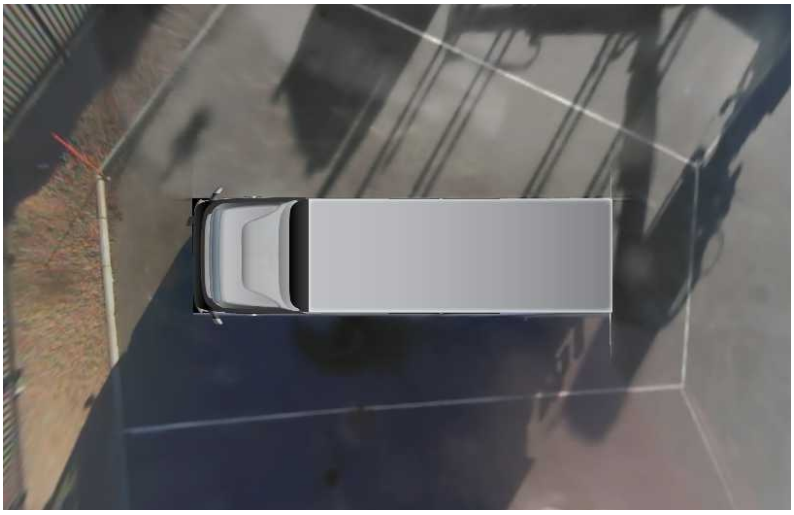
Two additional sensors mounted on the left and right side mirror can be introduced to provide even further visibility (as seen in Figure 11.5) and increased accuracy for pedestrian detection. Figure 11.6 shows a detection example from the side camera and the corresponding birds-eye view (same camera system as used in this thesis is utilized). To combine all of these sensors with a sensor fusion step is therefore a natural continuation from this master's thesis work.

11.2.4 Extend the Methods and Scenarios

The Haar Classifier Cascades are not invariant to rotation, so a rotation of the image (or multiple rotated cascades) is required. In chapter 10.3.1 a test of the image rotation was carried out, but the perturbation of training samples at different random angles was not evaluated, which



(a) Camera placement



(b) Birds-eye view

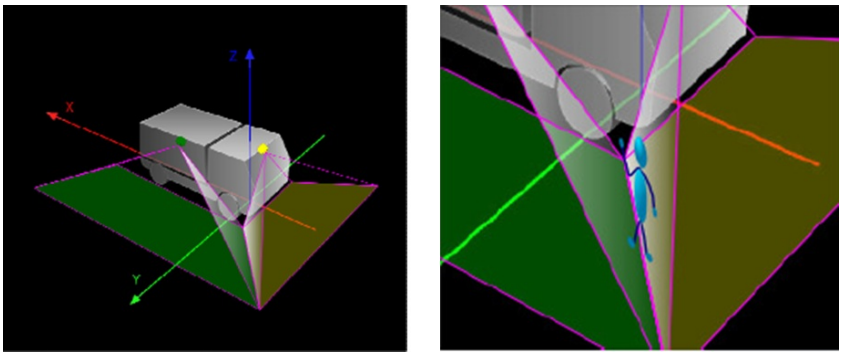
Figure 11.3: 360 degrees surround view. (a) shows the camera placement on top of the container and overlapping detection areas. (b) shows the resulting birds-eye view.

is problem specific and has been suggested to yield a more accurate detector [36]. To expand on the background subtraction techniques used, namely the MOG for a moving vehicle, the rotational motion of the truck could be connected to the existing gyrosensors as to compensate for that ego-motion [18, 67] since the scene objects is independent of distance for this motion. The current tracking method implemented is only developed for a single pedestrian and also needs to be extended to be able to track multiple pedestrians - which will need an extra data association step.

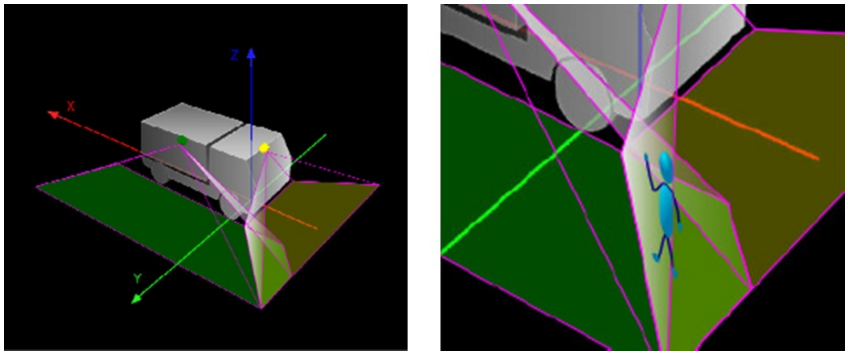
More advanced test sequences need to be captured to ensure the system independence for pedestrian occlusion, road condition, time of the day, and weather conditions. It could also be interesting to test how well the detection system will perform when detecting children, since the system is trained for upright adults. Since many accidents occur with other VRUs such as bicyclists and motorcyclist, the system needs to be expanded to handle these as well.

11.3 Real-time

A reasonable continuation of this master's thesis would be to improve the speed as to satisfy the real-time requirement. The bottleneck for the speed of the detector is no longer the computation



(a) Simple stitching



(b) Blended stitching

Figure 11.4: Camera view stitching. (a) shows the simple stitching method. (b) shows blended stitching method.

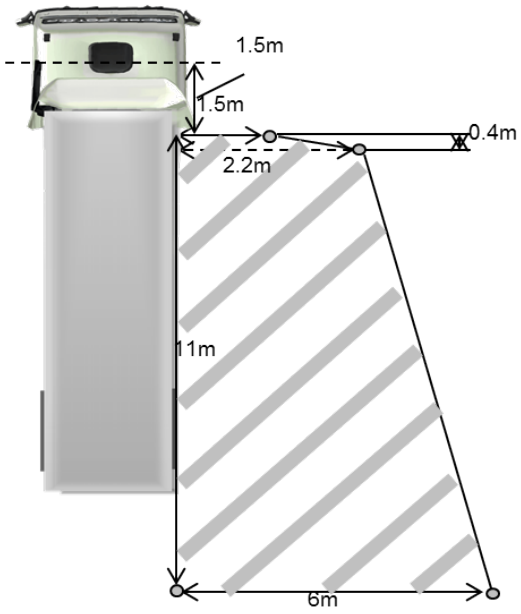


Figure 11.5: Field-of-view for the camera when placed on the side-mirror.

of the image pyramid, and to achieve further increase in speed fewer detection windows could be sampled (with a small accuracy loss). Compute Unified Device Architecture (CUDA), a C-based programming model created by NVIDIA, has recently received much attention due to its general purpose computing on Graphics Processing Units (GPGPU) which enables great

Figure 11.6: Birds-eye view and side-mirror detection for a bicyclist.

parallelism from available graphic units. For instance the Haar-based classifier used in this thesis could be designed with a parallel processing pipeline. This has been shown to run at 35 fps for a high-definition 1080p video using the sliding window approach with a one pixel step size [63], and 12 to 38 times faster than the CPU version for lower resolutions [75]. Similar results have been shown for a HOG detector with a 13 times speed up compared to the CPU version [3]. Finally, an evaluation to see if the detection module is reasonable to be install in a truck with its limited ECU capacity.

11.4 Further Applications

This master's thesis has focused on VRU detection for accident avoidance, but could be extended to other areas. One interesting area could be burglary detection, where the system detects and alerts when a person remains close to the truck for a longer period of time, that results in a wireless transmission of the camera output to the drivers' smartphone for inspection.

Chapter 12

Conclusion

In this master's thesis a framework has been proposed for pedestrian detection using a top-view wide angle camera at two different height positions. Two different background subtraction techniques were evaluated, The Mixture of Gaussians method and the Codebook method, and compared for accuracy, efficiency and time requirements. Two different machine learning techniques were evaluated: a cascaded classifier trained with the AdaBoost algorithm with a large set of haar-like features, and a generalized version that uses multiple intensity channels and scale approximated gradient histograms. A rotational scheme was evaluated for the classifier to detect all in-plane rotations of pedestrians, which resulted in a division into 5 different orientation regions. A unwarping transformation was evaluated as a response to the camera distortion for the classifier, but did not result in a improved detection accuracy and also significantly increased the number of false alarms and was left out of the final system.

The final system consists of a background subtraction technique for the close to front area, and a classifier evaluation for the mid-close to far areas. The background subtraction and classifier technique used in evaluation is based on the ego-vehicle speed and environmental illumination. A final tracking step in the form of Kalman tracking was also implemented. The final method does not have any unwarping transforms. An interface was developed to display which system is active and the current meter position from the front of the truck the pedestrian is located.

For the higher mounted side camera two additional classifiers were trained based on the variation of the pedestrian scale appearance, giving three different classifier evaluation regions. An extended version of the classifier rotational scheme was implemented for all of the three classifier regions.

Both of the cameras show robustness to various motions such as someone kneeling tying their shoes, walking, running, as well as partial occlusion. It also shows very good detection in the blind-zone region. The biggest limitation was seen to be extreme outdoor illuminations, and bright clothed pedestrians that have little contrast to the surrounding environment. Operating on 720 by 576 pixel images, the system is capable of detection at 11 frames per second on a conventional 2.16GHz Intel Core2.

Bibliography

[1] Bandarupalli, V. Evaluation of video based pedestrian and vehicle detection algorithms (2010). Master's thesis, UNLV Theses/Dissertations/Professional Papers/Capstones. Paper 757.

[2] Bertozzi, M., Broggi, A., Grisleri, P., Graf, T., and Meinecke, M. Pedestrian detection in infrared images. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE* (june 2003), pp. 662 – 667.

[3] Bilgic, B. Fast human detection with cascaded ensembles (2010). Master's thesis, MIT.

[4] Bradski, G., and Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008.

[5] Bradski, G., and Kaehler, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008.

[6] Comaniciu, D. An algorithm for data-driven bandwidth selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 25*, 2 (feb 2003), 281 – 288.

[7] Dalal, N. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble, 2006.

[8] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 1, pp. 886–893 vol. 1.

[9] Dalal, N., Triggs, B., and Schmid, C. Human Detection Using Oriented Histograms of Flow and Appearance Computer Vision – ECCV 2006. In *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3952 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2006, ch. 33, pp. 428–441.

[10] Devernay, F., and Faugeras, O. Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured enviroments. *Mach. Vision Appl. 13*, 1 (Aug. 2001), 14–24.

[11] Doermann, D., and Mihalcik, D. Tools and techniques for video performance evaluation. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on* (2000), vol. 4, pp. 167–170.

[12] Dollar, P., Belongie, S., and Perona, P. The Fastest Pedestrian Detector in the West. In *Procedings of the British Machine Vision Conference 2010* (2010), British Machine Vision Association, pp. 68.1–68.11.

[13] Dollár, P., Tu, Z., Perona, P., and Belongie, S. Integral channel features. In *BMVC* (2009).

[14] Dollar, P., Wojek, C., Schiele, B., and Perona, P. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), pp. 304–311.

[15] Dollar, P., Wojek, C., Schiele, B., and Perona, P. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34*, 4 (april 2012), 743 –761.

[16] Douglas, D. H., and Peucker, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization 10*, 2 (Oct. 1973), 112–122.

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

[18] Enzweiler, M., Kanter, P., and Gavrila, D. Monocular pedestrian recognition using motion parallax. In *Intelligent Vehicles Symposium, 2008 IEEE* (june 2008), pp. 792 –797.

[19] European Parliament and Council. Directive 2003/97/ec of the european parliament and of the council of 10 november 2003 on the approximation of the laws of the member states relating to the type-approval of devices for indirect vision and of vehicles equipped with these devices, amending directive 70/156/eec and repealing directive 71/127/eec.

[20] European Parliament and Council. Directive 2007/38/ec of 11 july 2007 on the retrofitting of mirrors to heavy goods vehicles registered in the community.

[21] Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W. Efficient and effective querying by image content. *Journal of Intelligent Information Systems 3*, 3-4 (1994), 231–262.

[22] Fang, Y., Yamada, K., Ninomiya, Y., Horn, B., and Masaki, I. A shape-independent method for pedestrian detection with far-infrared images. *Vehicular Technology, IEEE Transactions on 53*, 6 (nov. 2004), 1679 – 1697.

[23] Freund, Y., and Schapire, R. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, P. Vitányi, Ed., vol. 904 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1995, pp. 23–37.

[24] Friel, M., Hughes, C., Denny, P., Jones, E., and Glavin, M. Automatic calibration of fish-eye cameras from automotive video sequences. *Intelligent Transport Systems, IET 4*, 2 (june 2010), 136 –148.

[25] Fukushima, K., Miyake, S., and Ito, T. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *Systems, Man and Cybernetics, IEEE Transactions on SMC-13*, 5 (1983), 826–834.

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

[27] Gavrila, D. A bayesian, exemplar-based approach to hierarchical shape matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 8 (aug. 2007), 1408 –1421.

[28] Gavrila, D., Giebel, J., and Munder, S. Vision-based pedestrian detection: the protector system. In *Intelligent Vehicles Symposium, 2004 IEEE* (2004), pp. 13–18.

[29] Gavrila, D., and Philomin, V. Real-time object detection for ldquo;smart rdquo; vehicles. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (1999), vol. 1, pp. 87–93 vol.1.

[30] Gavrila, D. M., and Munder, S. Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. J. Comput. Vision 73*, 1 (June 2007), 41–59.

[31] Geronimo, D., Lopez, A. M., Sappa, A. D., and Graf, T. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 7 (July 2010), 1239–1258.

[32] Guo, L., Li, L., Zhao, Y., and Zhang, M. Study on pedestrian detection and tracking with monocular vision. In *Computer Technology and Development (ICCTD), 2010 2nd International Conference on* (nov. 2010), pp. 466 –470.

[33] Harris, C., and Stephens, M. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference* (1988), pp. 147–151.

[34] Heisele, B., and Wohler, C. Motion-based recognition of pedestrians. In *Proceedings of the 14th International Conference on Pattern Recognition-Volume 2 - Volume 2* (Washington, DC, USA, 1998), ICPR '98, IEEE Computer Society, pp. 1325–.

[35] Hobbs, A. Euro ncap/mori survey on consumer buying interests. In *Proc. Euro. NCAP Conf.* (2005).

[36] Horton, M. P. Multiple prediction combination and confidence measures for marine object detection, *phd* thesis, Feb 2009.

[37] Hughes, C., Glavin, M., Jones, E., and Denny, P. Wide-angle camera technology for automotive applications: a review. *Intelligent Transport Systems, IET 3*, 1 (march 2009), 19 –31.

[38] Image-Clipper-Tool. Image clipper tool. `http://code.google.com/p/imageclipper/`. Accessed: 05/03/2012.

[39] Jain, A., Duin, R., and Mao, J. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 1 (jan 2000), 4 –37.

[40] Kim, K., Chalidabhongse, T., Harwood, D., and Davis, L. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging 11*, 3 (June 2005), 172–185.

[41] Kohonen, T. The handbook of brain theory and neural networks. MIT Press, Cambridge, MA, USA, 1998, ch. Learning vector quantization, pp. 537–540.

[42] Kolsch, M., and Turk, M. Analysis of rotational robustness of hand detection with a viola-jones detector. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03* (Washington, DC, USA, 2004), ICPR '04, IEEE Computer Society, pp. 107–110.

[43] Leibe, B., Cornelis, N., Cornelis, K., and Van Gool, L. Dynamic 3d scene analysis from a moving vehicle. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (june 2007), pp. 1 –8.

[44] Lienhart, R., Kuranov, A., and Pisarevsky, V. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. 2003, pp. 297–304.

[45] Lienhart, R., Liang, L., and Kuranov, A. A detector tree of boosted classifiers for real-time object detection and tracking. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on* (july 2003), vol. 2, pp. II – 277–80 vol.2.

[46] Lienhart, R., and Maydt, J. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on* (2002), vol. 1, pp. I–900 – I–903 vol.1.

[47] Lowe, D. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (1999), vol. 2, pp. 1150–1157 vol.2.

[48] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2 (Nov. 2004), 91–110.

[49] Mallon, J., and Whelan, P. Precise radial un-distortion of images. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (aug. 2004), vol. 1, pp. 18–21.

[50] Mariano, V. Y., Min, J., Park, J.-H., Kasturi, R., Mihalcik, D., Li, H., Doermann, D., and Drayer, T. Performance evaluation of object detection algorithms. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3 - Volume 3* (Washington, DC, USA, 2002), ICPR '02, IEEE Computer Society, pp. 30965–.

[51] Matsuda, R., Tan, J. K., Kim, H., and Ishikawa, S. Detection of pedestrians employing a wide-angel camera. In *Control, Automation and Systems (ICCAS), 2011 11th International Conference on* (oct. 2011), pp. 1748 –1751.

[52] McMaster, R. B. Automated line generailization. *Cartographica 24*, 2 (1987), 74–111.

[53] Messom, C., and Barczak, A. Fast and efficient rotated haar-like features using rotated integral images. `http://www.araa.asn.au/acra/acra2006/papers/paper_5_63.pdf`, 2006.

[54] Mikolajczyk, K., and Schmid, C. Indexing based on scale invariant interest points. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (2001), vol. 1, pp. 525 –531 vol.1.

[55] Mikolajczyk, K., Schmid, C., and Zisserman, A. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision* (2004), vol. I, pp. 69–81.

[56] Mohan, A., Papageorgiou, C., and Poggio, T. Example-based object detection in images by components. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 4 (Apr. 2001), 349–361.

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

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

[59] Nakajima, C., Pontil, M., Heisele, B., and Poggio, T. Full-body person recognition system. *Pattern Recognition 36*, 9 (Sept. 2003), 1997–2006.

[60] Nascimento, J. C., and Marques, J. S. New performance evaluation metrics for object detection algorithms. In *PETS ECCV, 6th International Workshop on Performance Evaluation for Tracking and Surveillance* (Prague, Czech Republic, May 2004), pp. 7–14.

[61] Okuma, K., Taleghani, A., Freitas, N. D., Freitas, O. D., Little, J. J., and Lowe, D. G. A boosted particle filter: Multitarget detection and tracking. In *In ECCV* (2004), pp. 28–39.

[62] OpenCV. Open computer vision library. `http://opencv.willowgarage.com`. Accessed: 01/02/2012.

[63] Oro, D., Fernandez, C., Saeta, J., Martorell, X., and Hernando, J. Real-time gpu-based face detection in hd video sequences. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* (nov. 2011), pp. 530 –537.

[64] P. Viola, M. J. Fast multi-view face detection. Tech. rep., 2003.

[65] Papageorgiou, C., and Poggio, T. A trainable system for object detection. *Int. J. Comput. Vision 38*, 1 (June 2000), 15–33.

[66] Papageorgiou, C. P., Oren, M., and Poggio, T. A general framework for object detection. *Sixth International Conference on Computer Vision* (1998), 555–562.

[67] Polana, R., and Nelson, R. Low level recognition of human motion (or how to get your man without finding his body parts). In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on* (nov 1994), pp. 77 –82.

[68] Rowley, H., Baluja, S., and Kanade, T. Rotation invariant neural network-based face detection. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on* (jun 1998), p. 963.

[69] Rufli, M., Scaramuzza, D., and Siegwart, R. Automatic detection of checkerboards on blurred and distorted images. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* (2008), pp. 3121–3126.

[70] Sabzmeydani, P., and Mori, G. Detecting pedestrians by learning shapelet features. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (june 2007), pp. 1 –8.

[71] Scaramuzza, D. *Omnidirectional Vision: From Calibration To Robot Motion Estimation, PhD Thesis no. 17635*. PhD thesis, ., 2008.

[72] Scaramuzza, D., Martinelli, A., and Siegwart, R. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems* (Washington, DC, USA, jan. 2006), ICVS '06, IEEE Computer Society, pp. 45–.

[73] Scaramuzza, D., Martinelli, A., and Siegwart, R. A toolbox for easily calibrating omnidirectional cameras. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (2006), pp. 5695–5701.

[74] Schapire, R. E. The boosting approach to machine learning, an overview. In *In MSRI Workshop on Nonlinear Estimation and Classification* (2002).

[75] Sharma, B., Thota, R., Vydyanathan, N., and Kale, A. Towards a robust, real-time face processing system using cuda-enabled gpus. In *High Performance Computing (HiPC), 2009 International Conference on* (dec. 2009), pp. 368 –377.

[76] Shashua, A., Gdalyahu, Y., and Hayun, G. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In *Intelligent Vehicles Symposium, 2004 IEEE* (june 2004), pp. 1 – 6.

[77] Shet, V., Neumann, J., Ramesh, V., and Davis, L. Bilattice-based logical reasoning for human detection. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (june 2007), pp. 1 –8.

[78] Shimizu, H., and Poggio, T. Direction estimation of pedestrian from multiple still images. In *Intelligent Vehicles Symposium, 2004 IEEE* (june 2004), pp. 596 – 600.

[79] Spengler, M., and Schiele, B. Towards robust multi-cue integration for visual tracking. In *Machine Vision and Applications* (April 2003), vol. 14, p. 50.

[80] Stauffer, C., and Grimson, W. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* (1999), vol. 2, pp. 2 vol. (xxiii+637+663).

[81] Suzuki, S., and Be, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing 30*, 1 (Apr. 1985), 32–46.

[82] Szarvas, M., Yoshizawa, A., Yamamoto, M., and Ogata, J. Pedestrian detection with convolutional neural networks. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE* (june 2005), pp. 224 – 229.

[83] Tuzel, O., Porikli, F., and Meer, P. Human detection via classification on riemannian manifolds. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (june 2007), pp. 1 –8.

[84] Ulusoy, I., and Bishop, C. Generative versus discriminative methods for object recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (june 2005), vol. 2, pp. 258 – 265 vol. 2.

[85] Vazquez, C., Ghazal, M., and Amer, A. Feature-based detection and correction of occlusions and split of video objects. *Signal, Image and Video Processing 3*, 1 (2009), 13–25.

[86] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (2001), vol. 1, pp. I–511 – I–518 vol.1.

[87] Viola, P., Jones, M., and Snow, D. Detecting pedestrians using patterns of motion and appearance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (oct. 2003), pp. 734 –741 vol.2.

[88] Volvo-Accident-Research-Team. In *Volvo 3P Accident Research Safety Report 2007* (2007).

[89] Welch, G., and Bishop, G. An introduction to the kalman filter. Tech. rep., Chapel Hill, NC, USA, 1995.

[90] Witten, I. H., and Frank, E. *ROC curves in Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

[91] Wohler, C., and Anlauf, J. An adaptable time-delay neural-network algorithm for image sequence analysis. *Neural Networks, IEEE Transactions on 10*, 6 (1999), 1531–1536.

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

[93] Yilmaz, A., Javed, O., and Shah, M. Object tracking: A survey. *ACM Comput. Surv. 38*, 4 (Dec. 2006).

[94] Yuan, X., Song, Y., and Wei, X. Automatic surveillance system using fish-eye lens camera. *Chin. Opt. Lett. 9*, 2 (Feb 2011), 021101.

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

[96] Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 2, pp. 1491 – 1498.

[97] Zivkovic, Z. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (aug. 2004), vol. 2, pp. 28 – 31.

[98] Zivkovic, Z., and van der Heijden, F. Recursive unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 26*, 5 (may 2004), 651 –656.