Pedestrian detection for automotive night vision

Brouwer, E.J.A.

Award date:
2006

Link to publication
Pedestrian Detection for automotive night vision.

E.J.A. Brouwer

Coach: Dr. P.M. Hofman (Philips Research)
Supervisor: Ir. R.A.C. Braspenninng (Philips Research)
Date: February 2006
Concerns: Masters thesis

Period of Work: 2005-2006

Notebooks: None

Authors' address E.J.A. Brouwer nlv11184@natlab.research.philips.com
The last decade, safety became increasingly important in car industry. This resulted in the introduction of various safety measures, like electronic car stabilisation, crumple zone and air bags. With the introduction of night vision systems a new type of driver assistance is presented, which can compensate the shortcomings of the human visual system after sundown. A new extension to this system will be pedestrian detection, in order to warn a driver when approaching a pedestrian. Within Philips, a pedestrian detection system was developed for far infrared videos by the Video Processing group (ViPs group). Far infrared night vision is based on thermal imaging and pedestrians appear therefore as white areas (hot spots) in an image. Improvements in this system are presented both in the candidate selection and classification part. Furthermore, the ViPs system contains an horizon detection algorithm. Unfortunately, the position of the detected horizon can be inaccurate. In this report, a reliability measure is proposed to take the inaccuracies into account in the candidate classification. Finally two methods are evaluated for calculation of the distance between the camera and a pedestrian. The distance can be used for a speed-distance adaptive detection system.
Conclusions: One of the two proposed gradient based candidate selection methods, the gradient matching algorithm, gains high detection rates. The achievable detection rate is higher than 93% in all available night vision videos, if an overlap of 50% is required between the detected region of interest (ROI) and the pedestrian region available in ground-truth data. For the same amount of overlap, the ViPs system scored a detection rate between the 50 and 85%, depending on the type of sequence. The object splitting algorithm is able to increase the detection rate of the ViPs candidate selection method with 10 to 45%, depending on the type of sequence.

The two discussed distance calculation algorithms can estimate the distance between the camera and a pedestrian with an accuracy of approximately 20 metres. If a pedestrian is within a range of 50 metres of the camera, the motion based method can achieve estimates with inaccuracies of ±7 metres.

Seven new symmetry filters are proposed. The most promising filter is the Edge Distance filter. This filter is able to decrease the false rate up to ±20%, with a minimal decrease in detection rate. The proposed modification of the aspect ratio filter of the ViPs system preserves the detection rate better than the original filter (±5-10%, depending on the type of sequence), while the false rate is further decreased.

A new pedestrian detection system was made, based on the gradient matching candidate selection algorithm and the new and modified feature filters. Detection rates varied between the 53 and 97%, depending on the type of sequence. The number of false detections per frame was at most 0.6. The ViPs system scored a detection rate between the 34 and 68% with also a maximum false rate of 0.6 false detections per frame.

The false rate is too high for implementation in night vision systems. New feature filters have to be designed and training based methods have to be considered.
# Contents

1 Introduction .................................................. 3
   1.1 Night vision ............................................... 3
   1.2 Objectives .............................................. 5
   1.3 Outline of this thesis .................................... 6

2 Available methods ............................................... 7
   2.1 Candidate selection ...................................... 7
      2.1.1 Thresholding ....................................... 7
      2.1.2 Connected Component Analysis .......................... 9
      2.1.3 Horizontal segmentation and bodyline detection .... 9
   2.2 Candidate classification .................................. 10
      2.2.1 Inertia-based classification .......................... 10
      2.2.2 Contrast-based classification ........................ 11
      2.2.3 Probabilistic template matching ........................ 12
      2.2.4 Hyperpermutation Network ............................. 12
   2.3 Performance evaluation method ............................ 13

3 Pedestrian detection system ViPs ............................... 16
   3.1 Overview of the detection system ......................... 16
   3.2 Detection algorithms ...................................... 17
3.2.1 Horizon detection ..................................................... 19
3.2.2 Road detection .......................................................... 20
3.3 Feature filters .............................................................. 21
  3.3.1 Aspect ratio, filling ratio and minimum height filter ............ 22
  3.3.2 Vertical edge strength, above horizon and distance to road filter ... 22
  3.3.3 Relative size, pinhole camera model and symmetry filter ........ 23

4 Candidate selection ....................................................... 25
  4.1 Night vision luminance data statistics and thresholding .............. 25
    4.1.1 Luminance data statistics and ViPs threshold .................. 25
    4.1.2 Threshold parameter optimisation ................................ 28
  4.2 Gradient based segmentation ......................................... 29
    4.2.1 Gradient statistics ............................................... 30
    4.2.2 Gradient based local thresholding ................................ 32
    4.2.3 Gradient matching ............................................... 37
  4.3 Candidate selection performance .................................... 42

5 Candidate classification .................................................. 45
  5.1 Pedestrian distance ................................................... 45
    5.1.1 Camera parameters ............................................... 46
    5.1.2 Size-based object distance ....................................... 47
    5.1.3 Motion-based object distance ..................................... 50
  5.2 Reliability of the horizon detection ................................ 53
  5.3 Object splitting ..................................................... 59
  5.4 Symmetry filters ..................................................... 61
    5.4.1 Edge based symmetry ............................................... 62
    5.4.2 Luminance based symmetry ........................................ 66
Glossary

d Distance
f Focal length of the camera
l, m Pixel positions within an ROI or block
n Frame number
p Probability
r ROI number
vi Linear velocity along axis i
x, y Pixel positions within an image
xw, yw, zw Pixel positions in real world coordinates
C Camera
D Discriminant
DR Detection ratio
E(x, y) Edge pixel at location (x, y)
F(x, y) Luminance value of the pixel at position (x, y)
Ffr Number of false detections per frame
Fmean Mean luminance of a frame
Fstd Standard deviation of a frame
FR False ratio
Hi Height of i
I Inertia
J Jacobian matrix
Kim Intensity image
Km Intensity margin
Mi Mask of i
MH Modified Histogram
Ph Horizontal projection
Pr Frame rate
Ps Vertical projection
PR Performance rating
$Q$  Kernel
$R$  Reliability of horizon line
$R_{asp}$  Aspect Ratio
$R_{fill}$  Filling Ratio
$R_i$  Ratio of $i$
$S_i$  Area of $i$
$T_i$  Threshold of $i$
$W_i$  Width of $i$
$Y$  Image
$\varepsilon$  Error
$\theta_i$  Rotation angle around axis $i$
$\mu_i$  Mean of $i$
$\sigma_i$  Standard deviation of $i$
$\phi_i$  Angular velocity around axis $i$
$\mathcal{V}$  Horizon line
$R$  Rotation matrix
$T$  Translation matrix
$V$  Linear velocity matrix
$\Phi$  Angular velocity matrix
$\nabla$  Gradient
$\#$  Number of elements

Table 1: Conventions that are used throughout this thesis.

FIR  Far infrared
FOV  Field Of View
GBDRopt  Gradient Based Detection Rate optimised system
GBFRopt  Gradient Based False Rate optimised system
HPN  Hyperpermutation Network
HVS  Human Visual System
IR  Infrared
LM  Levenberg-Marquardt
NIR  Near infrared
poi  Point of inflection
ROI  Region Of Interest
SSA  Segmentation Side Accuracy
SSE  Segmentation Side Efficiency
SVM  Support Vector Machine
VES  Vertical Edge Strength
ViPs  Video Processing and Visual Perception group within Philips

Table 2: Abbreviations that are used throughout this thesis.
Chapter 1

Introduction

The last few decades, the number of cars has grown explosively. This resulted in heavy traffic and an increasing amount of traffic signs and rules. Driving became a complex job and the number of accidents increased every year. To reduce the impact of an accident, car manufacturers improved their cars with the introduction of better brakes, tires and cages. This mainly reduced the consequences of accidents, but it did not really deal with the main cause of accidents: the driver. Modern traffic demands considerably experienced drivers and visibility and the ability to react quickly became key features for safe driving. To help the driver understand and control the surrounding, parking assistance and distance measuring tools were built into the cars. Together with electronic car stabilisation systems, they form a good addition on the human driving capabilities and can reduce driving errors. However, only a few systems are available for improving the visibility (like better lighting), which influences the drivers reaction for 90%. Especially in driving after sundown this becomes important, the Human Visual System (HVS) is then less able to see depth and to recognise colours. In addition to these shortcomings of the HVS, a 50-year old driver needs twice as much light as a 30-year old driver for seeing the same object. It will be no surprise that more than three times as many deaths occur in accidents during nighttime than during daytime driving [1]. Therefore, night vision systems have become of interest to many researchers, governments and car manufacturers. Night vision covers topics like road sign-, lane-, obstacle- and pedestrian detection. In this report pedestrian detection is studied.

1.1 Night vision

Several types of cameras can be used for the enhancement of visibility, each with their own pros and cons. The simplest and cheapest solution is a normal CCD or CMOS camera, which gives a high resolution colour image. These cameras are typically used in trucks for observation of the blind spot. These cameras are unfortunately less suitable for detecting and classifying objects like pedestrians, due to all different colours in clothing, the highly detailed scenes and a, by the headlights, limited range of the visible scene at night. For this reason, industry is studying the more expensive infrared (IR) cameras, which have some attractive properties
for nighttime driving. Two major classes can be distinguished, Near InfraRed (NIR) and Far InfraRed (FIR) cameras, which produce different types of images (see Fig. 1.1).

Near infrared, also called active night vision, is based on infrared light with a wavelength of 0.7 till 1.3 \( \mu \text{m} \). The shorter the wavelength of the light, the higher the emitted energy. An infrared light beams forward and all reflected light is measured with an infrared detector. In this way it is possible to get a clear image of the surrounding of the camera. Infrared light is invisible for the human eye, therefore a high light intensity can be used and it becomes possible to see three times as far as with normal headlights (see Fig. 1.2). Near infrared gives high contrast images and can typically be used for lane- and road sign detection.

Far infrared (passive night vision) is based on thermal imaging. The wavelength of FIR varies between 3 and 30 \( \mu \text{m} \), but is mostly situated around the 10 \( \mu \text{m} \). In this band, there is a minimum atmospheric absorption, which makes it suitable for detection at a large distance. Each object emits infrared light with an intensity depending on the temperature of the object. Objects with a temperature of 300K have a maximum energy emission in the FIR band. The luminance of an object in a FIR image depends on the temperature of the object. On a dry evening it is possible to visualise objects to the horizon, but even with fog it can give an improved visibility [2]. A typical application of FIR is pedestrian detection, because pedestrians appear as white spots in the images, even if they are far away. FIR imaging also has some disadvantages, it has a lower grey level dynamic range than NIR images and there are a lot of factors outside which can affect the visible temperature. In winter people wear thick insulating clothes and in summer a traffic sign can become very hot. Moving cars produce heat in the motor and the tires. And rain, dust and fog can heavily influence the visibility. This makes it complicated to detect to which object a white spot belongs and to classify the type of object. In Figure 1.4, a compilation is shown of possible pedestrian appearances in FIR night vision data.

Some car manufacturers already implemented a night vision system. Cadillac introduced their on passive infrared based system in 2000, but removed it in their 2005 models [3]. The infrared images were shown to the driver with a head-up display. Honda introduced a (passive) stereo night vision in the autumn of 2004 and integrated an LCD display in the dashboard [4] (see Fig. 1.3). BMW introduced in the autumn of 2005 also a passive night

Figure 1.1: Example of a far- (a) and a near (b) infrared image. In the FIR image only objects producing heat are visible. In the NIR image all reflecting objects within the IR beam are visible, like lane signs, the roadside and the traffic light.
1.2 Objectives

Recently BMW organised a night vision concept competition for pedestrian detection with passive infrared. Within Philips, the Video Processing and Visual Perception (ViPs) group subscribed to this competition. The ViPs group has a lot of experience with image enhancement, object detection and motion estimation in videos. Applications of their algorithms can be found in TVs from low end to high end. However, night vision is a complete new domain for this group. A new project was started to explore the possibilities in this field of research. Via the concept competition, there was the possibility to find out what the major difficulties in night vision object recognition are and to see how Philips is positioned with respect to the other competitors. The ViPs group delivers software and not a complete camera solution. The night vision sequences were provided by BMW. This graduation project is part of the ViPs night vision project. A lot of approaches for pedestrian detection are available in the literature, from simple threshold algorithms to more complex methods with symmetry detection and pattern matching. None of these algorithms is able to get a 100% detection ratio with a small false ratio. For most algorithms, either a training process is required, or much computation time is needed. In our project, we want to find an approach that outperforms other algorithms in detection- and false rate and detection distance. A monocular camera is used without calibration and most camera parameters are therefore unknown. The detection system does not use training based methods, because of the limited number of available night vision data. To speed up the algorithm, no pattern matching is used [8, 9].

In general, the object detection problem can be divided into two sub-problems, candidate selection and candidate classification. In candidate selection, regions of interest are searched where the desired object is possibly present. The aim is to mark all desired objects in an image as a candidate and to minimise the number of false candidates. In practice, it will be a trade-off between the detection rate and the number of false candidates. In the candidate classification problem, filters try to mark all false candidates and remove them from the can-
didate list. The filters can for example be feature based (based on pedestrian properties) or based on pattern matching. Here, also a trade-off exists, filters can be set very strict to filter out all false candidates, but this will be at the expense of a certain number of positive candidates. The division of pedestrian detection in candidate selection and candidate classification is also used in this report.

1.3 Outline of this thesis

For the graduation project, a literature study is performed to obtain a global overview of the available detection algorithms. In Chapter 2, a short summary is given of published algorithms. The available methods will be presented into three sections, candidate selection, candidate classification and performance analysis. In Chapter 3, the non-motion based part of the pedestrian detection system of the ViPs group will be described in detail. This detection system will serve as the reference system for presented algorithms and is partly the basis for a method introduced in this report. In Chapter 4, four candidate selection algorithms are described and their performance is measured. A statistical basis is searched for the ViPs candidate selection method and a modification is made. Furthermore, two new selection methods are proposed. In Chapter 5, new candidate classification methods will be introduced and modifications to known filters will be proposed. In Chapter 6, the performance of the non-motion based part of the ViPs system will be measured and compared with a modified version and a new proposed detection system. In Chapter 7, conclusions are drawn and a discussion is given for future work.

Figure 1.4: Compilation of possible pedestrian appearances in night vision videos. Pedestrians are taken from videos of two different camera types. Videos in dry weather and rainy weather conditions were used. Note the different shapes the pedestrians have and the effect of insulating clothes.
Chapter 2

Available methods

For more than ten years, researchers are looking for appropriate methods for reliable and fast pedestrian detection in night vision data [10]. The used systems can be divided into two categories: monocular and stereo camera solutions. The stereovision systems have the disadvantage of the extra costs of the second camera and they need calibration, which makes the system more complex. Cost price of a system is an important issue for mass production and acceptance of the product by consumers. Therefore, we take only monocular camera systems into account. In this chapter, some recently proposed methods are briefly reviewed. For detailed descriptions, we refer to the original papers. Each method starts with a candidate selection, followed by a candidate classification. This division will also be followed in this chapter. In the first section, some candidate selection methods are described. The second section explains some candidate classification methods. The last section describes a performance evaluation method for the detectors. In this chapter, most recently proposed methods are covered, with the exception of Support Vector Machines (SVM). SVM is a training based method where data are mapped to a higher dimension feature space, where the data are clustered and ordered in classes [11]. The theory is too complex to describe here in a single section. The pedestrian detection algorithms of the ViPs group are also not mentioned, they will be covered in Chapter 3.

2.1 Candidate selection

2.1.1 Thresholding

A popular candidate selection method is thresholding, since pedestrians appear as hot spots in a night vision image. It is a computationally simple method, which is an important property for the intended application. Thresholding will always be followed by one or more other segmentation steps. Biggest problem is the selection of an appropriate threshold. Mostly, adaptive thresholds are chosen that change with the scene dynamics. A disadvantage of this method is that multiple hot objects in a frame can change the frame statistics dramatically and influence the threshold value.
Fang et al. [12] propose the following threshold:

$$T(n) = F_{\text{max}}(n) - K_m,$$  \hspace{1cm} (2.1)

where $T$ is the luminance threshold, $F_{\text{max}}$ the maximum luminance value in a frame, $n$ the frame number and $K_m$ a fixed intensity margin. A large constant is selected for $K_m$, to ensure that all pedestrians in a frame will lie above the threshold. This approach will generally give hot spots wider and higher than a pedestrian. This is solved with horizontal segmentation and bodyline detection (see Subsection 2.1.3).

Xu et al. [13] define the threshold as:

$$T(n) = 0.2F_{\text{mean}}(n) + 0.8F_{\text{high}},$$  \hspace{1cm} (2.2)

where $F_{\text{mean}}$ is the mean luminance of the frame and $F_{\text{high}}$ is the highest possible luminance value (255 for an 8-bit image). The thresholding is followed by a connected components analysis (see Subsection 2.1.2).

A fixed threshold is used by Nanda et al. [14]. The threshold value is calculated with a training set of 1000 rectangular pedestrian boxes. The mean and the standard deviation of pixels belonging to the background, $(\mu_1, \sigma_1)$, and to the pedestrians, $(\mu_2, \sigma_2)$, are calculated. Then the threshold is calculated:

$$T = \frac{\sigma_1 \sigma_2}{\sigma_1 + \sigma_2} \ln \left( \frac{\sigma_1}{\sigma_2} \right) + \frac{\sigma_1 \mu_2 + \sigma_2 \mu_1}{\sigma_1 + \sigma_2}$$  \hspace{1cm} (2.3)

After thresholding, a probabilistic template is made, which can be used for pixel classification (see Subsection 2.2.3).

In the ViPs system, the threshold is made adaptive to the mean ($F_{\text{mean}}$) and standard deviation ($F_{\text{std}}$) of the luminance of a frame [15]:

$$T(n) = k_1 F_{\text{mean}}(n) + k_2 F_{\text{std}}(n),$$  \hspace{1cm} (2.4)

where $k_1$ and $k_2$ are constants and chosen as 1 and 2. The thresholding is followed by a connected component analysis.

Other proposed threshold methods are thresholding a background subtracted image [16], thresholding by using the luminance value of the last local minimum in the histogram of an image [17] and thresholding with help of the P-tile (Percentile) method [18]. For the P-tile method, the size of the object to be recognised has to be known.

In Fig. 2.1, the average luminance of pedestrian regions of interest (ROIs) of two sequences is plot, together with the threshold values of the methods of [12], [13] and [15]. The threshold value may be slightly higher than the average ROI luminance, because of the included background in a pedestrian ROI (see Fig. 2.2). The threshold value of Xu et al. is much too high, which is probably caused by an optimised threshold for a camera with different parameters than the one used by BMW. The threshold of Fang et al. is sometimes lower than zero, which means that a whole image will pass the threshold. Likely, their algorithm was designed for a camera where the pedestrians appear only in the upper part of the gray scale. The threshold of the ViPs system is almost equal to the average pedestrian ROI luminance. Their threshold was designed for the used video sequence and performs therefore the best.
2.1. CANDIDATE SELECTION

Average ROIs and threshold values

Figure 2.1: Average luminance of pedestrian ROIs (blue line) from a rain sequence (first 708 ROIs) and a dry sequence. The calculated threshold values of Eq. 2.1, 2.2 and 2.4 are plot in black, magenta and red, respectively.

2.1.2 Connected Component Analysis

For candidate classification, a switch must be made from pixel based processing to object based. A popular way to do this is a connected component analysis [8]. For the connected component analysis, a binary image is needed as input. After thresholding, a binary image can be made with a 1 representing pixels higher than the threshold and a 0 for pixels below the threshold. All 1’s connected to each other are now labelled with an unique number higher than zero. In the resulting image, all zeros represent the background and all pixels with the same number represent an object (see Fig. 2.3). For the search process, a 4- and 8-connected component search can be used. In a 4-connected component search, only the horizontal and vertical neighbours are considered, in a 8-connected component search also the diagonal neighbours are taken into account.

Each obtained object is considered as an ROI and will be passed to the candidate classification step. Disadvantage of this method is the possibility that objects can be split into two or more objects or may be connected to other objects, due to poor thresholding. For example, a connected component analysis on the right pedestrian in Fig. 2.2 can result in separate objects for the head and the legs. Post-processing is required in these cases to cluster the separate objects again to one object or to split connected objects.

2.1.3 Horizontal segmentation and bodyline detection

In Fang et al. [12], the thresholding is followed by a horizontal segmentation and bodyline detection. For the horizontal segmentation, a vertical projection curve is made of all pixels higher than the calculated threshold (see Eq. 2.1). It is assumed that pedestrians will cause a peak in the projection curve. The horizontal segments are derived by searching the starting- and end points of all rising and falling curves. The resulting image stripes are passed to the bodyline detection for vertical segmentation within each stripe.
For the vertical segmentation, the gradients are calculated for each pixel in an image stripe. Then, for each row within a stripe, the highest positive and most negative gradient are selected as boundaries for the bodyline. Using almost 1000 pedestrian ROIs histograms, a histogram variation curve was made. For each selected bodyline, a search area can be defined, where the boundaries are defined by aspect ratio restrictions. Within the search area Fang et al. search for the area with the lowest difference with the histogram variation curve. Finally in one image stripe, just one bodyline with corresponding bounding box is chosen as possible pedestrian ROI and passed to the classification step. It is assumed that each image stripe contains at most one pedestrian.

A disadvantage of this method is that in a worst case scenario, a lot of histograms have to be calculated. A second problem is the horizontal segmentation. It is not a trivial job to find the exact starting- and end point of the rising- and falling curves. Sometimes you want to take a local minimum into account (for example if two or more pedestrians are walking next to each other) and sometimes the local minimum is just noise. Fig. 2.4a shows an example of a vertical projection curve. In order to remove noise, the curve is low-pass filtered. If an image stripe is too wide, this can be corrected by the bodyline search. But, it is also possible that the start- and end point search algorithm get stuck into local minima, which will divide one pedestrian in two or more image stripes. Then the same problem arises as with the connected component analysis and post-processing is required. This is shown in Fig. 2.4b, where the pedestrian is divided into two rectangles. The vertical segmentation in the figure is performed by taking the highest and lowest pixel above the threshold in an image stripe.

2.2 Candidate classification

2.2.1 Inertia-based classification

The inertia-based classification feature is a shape independent method proposed by Fang et al. [12]. Pedestrians have more or less the same inertia, no matter the pose. The inertia is defined as the luminance value of a pixel, multiplied with the distance of the pixel to the centre point of an ROI. Pedestrians have typically the brightest pixels in the middle of the ROI. Therefore, just one template is needed as a reference for the inertia.
2.2. CANDIDATE CLASSIFICATION

The inertia, $I_{ROI}$, of an ROI is calculated as follows:

$$I_{ROI} = \frac{\sum_{x,y} F(x, y)d(x, y)^2}{\sum_{x,y} F_{\text{template}}(x, y)d_{\text{template}}(x, y)^2},$$

(2.5)

where $x$ is the horizontal coordinate, $y$ the vertical coordinate and $d$ the distance of a pixel to the centre of the ROI. Since just one template pedestrian is used, all found ROIs have to be resized to the template size before calculation of the inertia. For pedestrian ROIs, the inertia will be close to 1.

From computational point of view, this method is less interesting, because every candidate ROI has to be resized, followed by a pixel based calculation. This can eventually be solved by a lookup table with inertias of the template pedestrian of most occurring pedestrian sizes. Furthermore, the assumption that pedestrian ROIs have the brightest pixels in the middle of the ROI is not always valid. In winter sequences and rainy weather sequences, where people wear well-insulating coats, it is possible that only the head and legs are visible (see Fig. 2.2).

2.2.2 Contrast-based classification

A pedestrian appears as a white spot in infrared images, which gives a high contrast with neighbouring objects. This information is used by Fang et al. [12] for contrast based classification. In each ROI, the vertical edges are calculated using a modified Sobel method in combination with a constant threshold. With these edges, the row-edge index is calculated, which is defined as the average number of vertical edge pixels in each row of the ROI. For each ROI an upper and lower vertical neighbourhood is defined as the region above/under the ROI with the same width and half the height. For these regions also the row-edge indexes are calculated, called upper- and lower row-edge index. The method is based on the assumption that beneath a pedestrian is road with at most one vertical edge from a lane mark. Above a pedestrian is no other pedestrian, but sky, trees, buildings etc., which do not create two

![Figure 2.4: Example of a vertical projection curve, an unfiltered (blue line) and filtered (red line) curve (a). In (b), the corresponding segmentation result is shown.](image)

In (b), the corresponding segmentation result is shown.
or more long vertical edges within one small image stripe. With these two assumptions the following two rules were defined to classify an ROI as a non-pedestrian ROI:

- The lower row-edge index is larger than 1
- Both upper- and lower ROI row-edge index are close to, or larger than 1.5

Purpose of this classification method is to remove ambiguity between non-pedestrian ROIs containing light poles and ROIs in front of a light pole. A more detailed description can be found in [12].

2.2.3 Probabilistic template matching

Probabilistic template matching is a non-feature based classification method proposed by Nanda et al. [14]. For pedestrian detection, a probabilistic template is made with help of a training set of 1000 pedestrian ROIs, with the pedestrians in different poses. An intensity image, $K_{im}$, is created from the training data by thresholding the data (see Eq. 2.3), pixels with a luminance below the threshold will be clipped to zero, pixels with a higher luminance than the threshold will keep their luminance value. The probabilistic template consists of probabilities $p$ (for each pixel one) for belonging to a pedestrian or not. The probability $p$ is determined by the number of times that a pixel appears as 1 in the training data. For the final pedestrian detection, a probability $p_c(x, y)$ (for each pixel) is calculated, which indicates if the $l \times m$ window around the pixel contains a part of a pedestrian or not. The probability $p_c(x, y)$ is calculated as the sum of contributions of the same $l \times m$ window in the intensity image:

$$p_c(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{l} (K_{im}(i, j) - 127) * (p(i, j) - 0.5), \quad \text{(2.6)}$$

where $K_{im}$ is the thresholded image, $p$ is the probability calculated in the probabilistic template and $l$ and $m$ are the dimensions of the window. The window is moved over the entire image, which results in a new probability map with a probability $p_c$ for each pixel. From the training set of 1000 pedestrian ROIs, these $p_c$ maps are determined and the mean and standard deviation can be calculated. These mean and standard deviation are used for calculating a general threshold for the $p_c$ probability maps with help of Eq. 2.3.

Since we do not consider training based methods in our system, this method cannot be used.

2.2.4 Hyperpermutation Network

Meis et al. propose to use a Hyperpermutation Network (HPN) for pixel classification [19]. Pixels can be classified into two groups, "object of interest" and "background". Pixel classification is based on the neighbourhood of a pixel and not on the luminance of a pixel itself. The neighbourhood of each pixel is sub-sampled with a random pattern, resulting in 27 pixel values. A network of trained look-up tables tries to describe these 27 values in a 3 bit word.
This is repeated for all pixels. For all pixels in a frame the same pattern is used. The result is a new image with 3 bits per pixel. The trained filters give each sub-sampled group of pixels a confidence level. This procedure is repeated ten times, where for each iteration a different random sampling pattern is selected. At the end of the tenth iteration, the pixel classification takes place by thresholding the pixel values. Grouping of "object" pixels is performed with help of connected component analysis (see Subsection 2.1.2).

Since we do not consider training based methods in our system, this method can also not be used.

2.3 Performance evaluation method

After the implementation of a new candidate selection or candidate classification method, the performance of the algorithm has to be measured. In the literature, not much is available about this topic. Most papers present the detection rate, the false-positive rate and computation time of their algorithm. However, it is not mentioned how these numbers are measured. Are pedestrians only counted if they have a certain minimum size? When is a detection counted as positive, if just 5% of a person is marked as pedestrian or only if more than 80% is marked? Are groups of people counted, or only single persons? Which type of video material is used, summer, winter, rain, dry weather, urban or sub-urban? All these questions are important if we want to benchmark different algorithms and compare them with each other.

Unfortunately, only a few papers mention exact details about their measurements and there is no public accessible night vision video database with standard sequences for benchmarks. This makes it more or less impossible to judge the quality difference between developed and published algorithms. A performance measure can usefully be applied within our own test set. The effectiveness of different filters can be compared with each other and the detection and false rate can be calculated for our own desired constraints (size, distance, etc.).

To calculate the detection rate, two properties are important: which percentage of the detected area of a ROI is a pedestrian and which percentage is non-pedestrian. A detection must not be counted as positive, if only a small part of a pedestrian is detected. If the detected pedestrian area is much larger than the actual pedestrian, the detection must also be discarded. Fang et al. [12] proposed a criterion which combines both properties. In this report, we will use this performance measure. The method will be summarised in this section. Definitions and symbols were taken from the source article.

For the performance measure, it is supposed that ground-truth data are available for the used sequences. In practice, this means that a list must be available with the coordinates of manually selected pedestrians. In this selection process, choices can already be made about the minimum detection size, partly occluded pedestrians, etc.

The performance criterion consists of two parts, the segmentation side accuracy (SSA) and the segmentation side efficiency (SSE). The SSA is a measure for the amount of overlap between the pedestrian ROI (ground-truth) and the detected ROI, calculated as the ratio between the amount of overlap and the pedestrian ROI size. The SSE is a measure for the precision of the detected ROI size. The SSE is calculated as the ratio of the amount of overlap between the pedestrian ROI and the detected ROI and the total size of the detected ROI.
The indices are defined in the source article as follows:

- **Segmentation Side Accuracy (SSA):** The square root of the ratio of the detected pedestrian region area $S_{\text{overlap}}$ over the entire pedestrian area $S_{\text{pedestrian}}$.

- **Segmentation Side Efficiency (SSE):** The square root of the ratio of the detected pedestrian region area $S_{\text{overlap}}$ over the entire ROI area $S_{\text{ROI}}$.

Formally:

$$SSA = \sqrt{\frac{S_{\text{overlap}}}{S_{\text{pedestrian}}}} \quad (2.7)$$

$$SSE = \sqrt{\frac{S_{\text{overlap}}}{S_{\text{ROI}}}} \quad (2.8)$$

where $S_{\text{pedestrian}}$ is the area of the pedestrian bounding box, $S_{\text{ROI}}$ the area of the detection bounding box and $S_{\text{overlap}}$ the area where $S_{\text{pedestrian}}$ and $S_{\text{ROI}}$ overlap each other (see Fig. 2.5). The indices give values between 0 and 1. The higher the index, the better the quality of the detection. The SSA will be very low and the SSE very high, if a very small ROI is detected within a pedestrian ROI. If the detected ROI is much bigger than the pedestrian ROI, the SSA will be high and the SSE will be low. If a detection is almost perfect, both will be almost 1. The root in the definitions is less intuitive. Probably, Fang et al. wanted to model that an increase between for example 40 and 50% is more important than an increase between 90 and 100%. In this report, the definition with root is used. When presenting results, we have to keep in mind that the interesting domain in SSA and SSE will be between 0.2 and 0.6 (44.7% and 77.5% overlap).

![Figure 2.5: Indication of the SSA and SSE variables.](image)

A detection can now be counted as positive if the indices SSA and SSE exceed the fixed thresholds $T_{SSA}$ and $T_{SSE}$. The detection and false rate will in performance evaluations be calculated for a range of threshold values. In this report, the detection rate ($DR$) is defined as the ratio between the total number of detected pedestrians in a sequence and the total number of pedestrians present in the ground-truth data. The false rate ($FR$) is defined as the ratio between the number of false detections in a sequence and the total number of detections.
in the sequence:

\[
DR = \frac{\# \text{detected pedestrians}}{\# \text{pedestrians in ground truth data}} \tag{2.9}
\]

\[
FR = \frac{\# \text{false detections}}{\# \text{detections}} \tag{2.10}
\]
Chapter 3

Pedestrian detection system ViPs

The ViPs group recently started their research in the night vision domain. The night vision data was provided by BMW and contained eleven sequences. Two batches of sequences were available during our research (see Appendix C for a list of the available sequences and for the details of each sequence). The batches have different characteristics, possibly caused by using two different cameras (see Fig. 3.1). The first batch consists of eight videos, recorded in the winter in an urban and countryside area. One video is recorded in rainy weather and the rest in dry weather. Each video is available for a camera mounted very low on the car ("lower" sequences) and for a camera mounted on the roof ("upper" sequences). The second batch of sequences consisted of three videos recorded in the winter in a countryside area with dry weather. The algorithms of the ViPs group are based on the first batch of sequences. The camera parameters are unknown and no calibration is performed for determining the depth in scenes. The resolution of the videos is 320×240 (width×height) and the frame rate is 30 frames/second. The camera is mounted at a height of 33 cm for the "lower" sequences and at 1.65 m for the "upper" sequences. For the sequences of the first batch, meta-data is available with additional data like the speed and the yaw rate (angle of the front wheels) of the car at each frame. At this moment, this information is not used for pedestrian detection. It was decided to use feature based detection methods and no training algorithms, because of the limited number of test sets. Furthermore, the algorithms had to be computationally efficient to guarantee real-time processing. The detection system consists of motion and non-motion based pedestrian detection algorithms. In this report, only the non-motion based pedestrian detection algorithms are described. Besides pedestrian detection, also image enhancement is performed like flare reduction and road colouring. These topics are not covered in this report. In Section 3.1 an overview will be given of the complete system. In Section 3.2, the detection algorithms will be explained and in Section 3.3 the feature filters will be described.

3.1 Overview of the detection system

The non-motion based part of the pedestrian detection system of the ViPs group consists of four detection algorithms and a cascade of feature filters. In Fig. 3.2, a block diagram is given
3.2 Detection algorithms

The pedestrian detection system contains four detection algorithms: hot spot, edge, horizon and road detection. The hot spot detection is a thresholding of luminance data as explained in Subsection 2.1.1. The used threshold is calculated with Eq. 2.4, where $k_1 = 1$ and $k_2 = 2$. The thresholding is followed by a connected component analysis, which clusters the pixels above threshold to hot spots. $k_1$ and $k_2$ are constants, but in future they can be chosen adaptively.

of this part of the system. The ROIs are detected with the hot spot detection, followed by a connected component analysis which creates a list of ROIs. From this point the cascade of feature filters (Red boxes) start to eliminate the non-pedestrian ROIs. After each filter, non-pedestrian candidates are removed from the ROI list and only the remaining ROIs are passed to the next filter. Therefore, it is important to start the cascade with powerful filters, which filter on pedestrian properties that are generally valid and present in all the night vision data. Another possibility is to not remove the ROIs from the list, but to decrease their likelihood to be a pedestrian. Disadvantage of this method is that all ROIs have to be inspected by all filters, which is computationally expensive. In the current system, this possibility has not been implemented. The edge, horizon and road detection (green boxes) are used to adapt filters.

At the end of the feature filter cascade, all detected pedestrians are known to the system. With the help of motion tracking (not covered in this report), the predicted coordinates of pedestrians detected in the previous frame, are calculated. If a predicted coordinate corresponds with a pedestrian detected in the current frame, the confidence level is set to maximum. If a motion tracked pedestrian is not present in the detected pedestrians list of current frame, the confidence level is decreased. After a few frames of mismatches, the confidence level will be zero. With the motion tracker, it is now possible to detect pedestrians that were missed by the cascade filtering. The number of frames that a missed pedestrian can be re-included in the pedestrian list depends on the user defined thresholding of the confidence level. A low threshold can fill in a large detection gap, but also raises the false detection rate.
Figure 3.2: Block diagram of the detection algorithms and feature filters of the pedestrian detection system of the ViPs group.
to the temperature, or to other weather characteristics. During the research period, there was not enough night vision material available for determining such relationships. A disadvantage of the thresholding method is the sensitivity for insulating clothes. The luminance of the clothed body of a pedestrian is often lower than the threshold and the pedestrian will be split into multiple pieces.

The edge detection block is an implementation of the well-known Prewitt edge detection algorithm [20]. The horizon and road detection algorithms are more complicated and will be explained in the next sections.

3.2.1 Horizon detection

The horizon is an important feature in pedestrian detection, while every bottom of an object that is on the road will be below the horizon line. Hence, it can be perfectly used as a feature filter. The horizon will be defined as the end of the road, which is slightly different from the common definition: the line that separates the earth from the sky. The description will follow the original article of A.Ekin [21], however, values of variables and thresholds are taken from the implemented C++ code and can differ from the article.

The horizon detection is based on two assumptions. The first assumption is that the road and sky are a significant part of the scene, large enough to define the mean luminance of a frame. The second assumption is that the road and sky are typically colder than objects like cars, pedestrians and trees. This implies that on the horizon line, where the road is small, the luminance will be higher than in the sky and thus higher than the mean frame luminance ($F_{\text{mean}}$).

The first step in the algorithm is to create a binary road-sky mask ($M_{RS}$) by thresholding the input frame. The threshold $T_{RS}$ is defined by Eq. 2.4 with parameters $k_1 = 1$ and $k_2 = 0$. $M_{RS}$ is obtained by:

$$
M_{RS}(x, y) = \begin{cases} 
1 & \text{if } F(x, y) \leq T_{RS} \\
0 & \text{if } F(x, y) > T_{RS}
\end{cases}
$$

(3.1)

The road-sky mask will typically have the shape of an hourglass (see Fig. 3.3a). The road and sky can now be separated by taking a horizontal projection (projection on y-axis), which gives a V-shaped line (see Fig. 3.3b). The horizontal projection is defined by:

$$
P_h(y) = \sum_x M_{RS}(x, y)
$$

(3.2)

From the horizontal projection, the derivative $P_{Dh}$ is calculated:

$$
P_{Dh}(y) = \begin{cases} 
P_h(y) - P_h(y - N) & \text{if } y > N \\
0 & \text{if } y \leq N
\end{cases},
$$

(3.3)

where $N$ is the derivative step size and set to 3. The horizon detection is based on two estimations, one by using the horizontal projection $P_h$ and the other by using the derivative $P_{Dh}$. The search for the first horizon estimation, $\mathcal{V}_1$, starts at the point $y_{up} (=50)$. $\mathcal{V}_1$ is now defined as the first point after $y_{up}$, where $P_h$ is larger than $P_h(y_{up}) + T_{\mathcal{V}_1}$ (see Fig. 3.3b):

$$
\mathcal{V}_1 = \arg \min_{y>y_{up}} (P_h(y) > P_h(y_{up}) + T_{\mathcal{V}_1}),
$$

(3.4)
where \( y \) is the line number starting from the sky and \( T_{V1} \) is a fixed threshold (\( =30 \)). The second horizon estimation, \( V_2 \), is defined as the first time the derivative \( P_{Dh} \) is larger than \( T_{large} \), where \( T_{large} \) is a constant set to 50:

\[
V_2 = \arg\min_y (P_{Dh}(y) > T_{large})
\]  

(3.5)

The final horizon estimation, \( V \), is calculated by averaging \( V_1 \) and \( V_2 \):

\[
V = \frac{V_1 + V_2}{2}
\]  

(3.6)

An example of the result of the horizon detection is given in Fig. 3.4. The first estimation \( V_1 \) is plot in yellow, \( V_2 \) in blue and the final result \( V \) in red. To refine the result of the horizon detection, one additional step is added. If the top of the road from the road detection algorithm (see Subsection 3.2.2) is below the horizon line, than the horizon will be replaced by the top line of the road, \( y_{TOR} \):

\[
V = \arg\max(V, y_{TOR})
\]  

(3.7)

Although this detection method seemed to work rather effective, some disadvantages have to be mentioned. For the calculation of \( V_1 \), it is assumed that the right part of the V-shape is higher than the left part. However, new material from BMW (the second batch of sequences) showed that this assumption is not always valid. Even the V-shape is not always present, which was the underlying principle of the algorithm.

### 3.2.2 Road detection

Road detection is used for both pedestrian detection and enhancement of the image. The description of the algorithm is again according to the article of A. Ekin [21]. The basic idea of road detection is almost the same as the horizon detection. The road is a relatively large, cold part of the image which results in a luminance equal to, or lower than, the mean luminance of
3.3 Feature filters

In the hot spot detection and connected component analysis, a lot of ROIs were selected. These ROIs contain pedestrians and other objects. In the classification step, a cascade of feature filters test which ROIs contain pedestrian properties. In the following sections, a short explanation is given of each filter. The order of the block diagram of Fig. 3.2 will be followed.
3.3.1 Aspect ratio, filling ratio and minimum height filter

The aspect ratio ($R_{asp}$) filter restricts the tolerated proportions of the ROIs. The width of a pedestrian will always be smaller than the length (it is assumed that a pedestrian is standing straight). This property is used in the $R_{asp}$ filter. The $R_{asp}$ filter is defined straightforward:

$$R_{asp} = \frac{H_{ROI}}{W_{ROI}},$$

where $H_{ROI}$ is the height and $W_{ROI}$ the width of the ROI. In the literature, acceptable aspect ratios can be found between 2 and 5. The values in the ViPs project lie between 1.25 and 5. The difference can mainly be ascribed to the distance at which a pedestrian has to be detected. Observations showed that the aspect ratio becomes smaller if the distance to the pedestrian becomes larger. This is probably caused by flare or the limited camera resolution. The $R_{asp}$ filter is very powerful and filters out many false detections. However, if an ROI contains two pedestrians walking close to each other, or if a pedestrian is connected to an other hot object, this test will fail.

The filling ratio ($R_{fill}$) filter assumes that a pedestrian will always fill a certain part of its ROI. An L-shaped object for instance will have a much lower $R_{fill}$ than a person. The filling ratio is defined as:

$$R_{fill} = \frac{\text{#bright pixels in bounding box}}{S_{ROI}},$$

where $S_{ROI}$ is the area of the ROI and a bright pixel is defined as a pixel with a luminance higher than the threshold $T$ (see Eq. 2.4). The threshold for the minimum required filling ratio $T_{fill}$ is set to 0.3.

The minimum- and maximum height filter impose some size constraints to the ROIs. The minimum tolerated height is 5 pixels and the maximum is 200 pixels. All ROIs outside these bounds are rejected, irrespective of their aspect ratio.

3.3.2 Vertical edge strength, above horizon and distance to road filter

Pedestrians contain always at least two vertical edges, one rising edge for the left side and a falling edge for the right side of the body. The vertical edge strength filter calculates the number of vertical edges that are present on each scan line of an ROI. The output of the edge detection is used as input for this filter. The edge strength is calculated by computing the ratio of scan lines with more than one vertical edge and the height. A ROI is rejected if less than 50% of the horizontal scan lines contain more than one edge.

The bottom of a pedestrian bounding box must always be below the horizon line (see Fig. 3.4). The above-the-horizon filter therefore simply checks if the bottom of a bounding box is below the horizon line plus 5 pixels. The 5 pixels are taken as fault margin for the horizon and bounding box. If an ROI does not satisfy this rule, it is removed from the pedestrian list.

The distance-to-road filter assumes that the feet of a pedestrian are always in the same plane as the road. To validate this property, the vertical distance between the bottom of a bounding box and the first road pixel is calculated. If this distance is larger than 20 pixels, or if it is larger than the height of the bounding box itself, than this ROI will be rejected. This filter can prevent false detections in structured objects.
3.3.3 Relative size, pinhole camera model and symmetry filter

In the relative size filter, perspective congruency \([8, 22]\) is used to remove non-pedestrian candidates. For this filter, a flat road is assumed. At a larger distance from the camera, the bounding boxes must be smaller in size, and the centre coordinate of a bounding box must be higher than the centre coordinate of a bounding box nearby. For the non-pedestrian candidate elimination process, the first bounding box from the ROI list is taken as reference. From the reference box \((RB)\), the height, \(H_{RB}\), and vertical coordinate of the centre, \(y_{c,RB}\), are calculated. A height threshold \(T_H\) is calculated based on the reference box size:

\[
T_H = \max(0.5H_{RB}, H_{RB} - 7)
\]  

(3.10)

From all other bounding boxes \((BB)\) on the ROI list, also \(H_{BB}\) and \(y_{c, BB}\) are calculated. The object in a bounding box is situated on a larger distance than the reference bounding box, if \(H_{BB}\) is smaller than the height threshold \(T_H\). The centre coordinate \(y_{c, BB}\) must then be higher than the centre coordinate \(y_{c, RB}\) of the reference box. If this is not the case, a relative size inconsistency exists. The relative size filter, therefore, is defined by:

\[
BB_i = \begin{cases} 
\text{pedestrian} & \text{if } (y_{c, BB_i} > y_{c, RB}) \land (H_{BB_i} < T_H) \\
\text{non-pedestrian} & \text{otherwise}, 
\end{cases}
\]  

(3.11)

where \(i\) is the number of the bounding box. All bounding boxes that are classified as non-pedestrian are removed from the ROI list. The test is repeated with the second bounding box on the ROI list as a reference. This process will be repeated until all bounding boxes have served as reference. A necessary condition for this filter is that the bounding boxes are on the road, this is guaranteed by the distance-to-road filter.

The pinhole camera model calculates a multiplication factor that describes the perspective size relation between different bounding boxes. This filter needs the bounding boxes to be on the road and actually the biggest bounding box is assumed to be a pedestrian. For calculation of the multiplication factor, the largest bounding box on the ROI list is selected. The height, \(H_{RB}\), and the vertical coordinate of the centre, \(y_{c, RB}\), of the largest box are calculated. The multiplication factor \(R_{mult}\) is calculated as follows:

\[
R_{mult} = \frac{H_{RB} - H_{\min}}{y_{c, RB} - V},
\]  

(3.12)

where \(H_{\min}\) is a minimal height constant which is set to 5. \(V\) is the horizon line, calculated in the horizon detection algorithm. The value of \(R_{mult}\) is clipped between 1/12 and 1. Other pedestrian bounding boxes have to satisfy this multiplication factor. The required height of the other bounding boxes is calculated with help of their vertical centre coordinate and the calculated \(R_{mult}\) factor, using Eq. 3.12. This calculated height, \(H_{BBcalc}\), is compared with the actual height \(H_{BB}\). A bounding box is rejected if it satisfies one of the following two conditions:

\[
BB_i = \begin{cases} 
\text{non-pedestrian} & \text{if } H_{BB_i} < H_{BBcalc} \\
\text{non-pedestrian} & \text{if } H_{BB_i} > 5(y_{c, BB_i} - V) \\
\text{pedestrian} & \text{otherwise}, 
\end{cases}
\]  

(3.13)

A drawback is that this filter depends on the horizon estimate, which is not always correct. The perspective model is not valid, if the horizon is calculated incorrectly. Furthermore, the
multiplication factor $R_{\text{multi}}$ will vary with the slope of the road and will often be higher than 1, to which value it is clipped.

The symmetry filter is designed to filter out tires. Tires radiate frictional heat and will therefore appear as white area in night vision data. A tire has typically a hot spot in a triangular form in the bottom side of the bounding box, while a pedestrian has a more uniform distribution of bright pixels over the complete box. To determine if a ROI is a tire, the filling ratios of each side of the diagonal starting from the bottom left corner to the top right corner are calculated. If the area at the left of the diagonal is defined as $S_l$ and the other half as $S_b$, then the filter condition is defined as:

$$BB_i = \begin{cases} 
\text{non-pedestrian} & \text{if } (R_{\text{fill},S_l} < R_{\text{fill},S_b}) \land (R_{\text{fill},S_l} < 0.4) \\
\text{pedestrian} & \text{otherwise,} 
\end{cases}$$

where $R_{\text{fill}}$ is the filling ratio. It is not taken into account that an approaching car from the opposite direction will have the bright pixels of a tire in the top left corner.
Chapter 4

Candidate selection

Candidate selection in the ViPs system, is performed by thresholding the luminance data. It is well known in the literature to describe the content of a frame by its mean and standard deviation [22]. The values of $k_1$ and $k_2$ (see Eq. 2.4) were determined experimentally and chosen very accurately. The thresholding performs rather good, however, there was no theoretical argument for choosing these values. Another popular method in the literature for object segmentation is thresholding in the minimum of a bimodal histogram [22]. In the first section of this chapter the bimodal thresholding method is considered and we want to find out if there is a statistical basis for the values of $k_1$ and $k_2$, or if other better values could be chosen. In Section 4.2, two other thresholding algorithms are proposed, both based on gradient analysis. Section 4.3 provides a performance analysis of the candidate selection methods.

4.1 Night vision luminance data statistics and thresholding

4.1.1 Luminance data statistics and ViPs threshold

To evaluate the chosen threshold parameters, statistics from the frame and pedestrian luminances are required. To obtain data of pedestrians, more than 3000 rectangular pedestrian ROIs were selected manually from 6 different sequences. At the moment of selection, only the first batch of BMW videos was available (see Chapter 3 and Appendix C). Five dry weather sequences and one rain sequence from the "upper" cameras (mounted on the roof) were used. Because of the big differences between the dry weather sequences and rainy weather sequence, these sequences were dealt with separately. ROIs were selected as pedestrian if a reasonable part of a pedestrian was visible. A consequence of rectangular bounding boxes is some covered background in each ROI (see for example Fig. 2.2).

First the luminance distribution of each frame containing a pedestrian is calculated for the dry weather sequences and rainy weather sequence and put into a histogram for visualisation (see Fig. 4.1). Pixels with luminance value zero are removed from the histogram in order to obtain a better scaling. Both the dry weather and rainy weather histogram do not have a
bi-modal structure. Although, the histogram of the dry weather sequences has some peaks at the right side of the histogram. For determining a relationship between the luminances of pedestrians and frame luminances, histograms of pedestrian ROIs were made (see Fig. 4.2). The luminance of pedestrians in dry weather has an almost uniform distribution in the lower part of the histogram and a peak at the end. The rain sequence has a Gaussian or chi like distribution [23]. However, both histograms do not have a significant influence on the frame luminance histograms ($10^3$ versus $10^6$). If a bi-modal structure is found in the histogram of a frame, it cannot be concluded that a pedestrian is present in that frame or not. If a pedestrian is present, thresholding in the minimum will result in a detection loss of almost 60% of the pedestrian pixels. It can be concluded that the histogram shape does not give any basis for hot spot detection.

Figure 4.1: Histograms of luminance distribution for dry weather (a) and rainy weather (b) for frames containing a pedestrian.

Figure 4.2: Histograms of luminance distribution for dry weather (a) and rainy weather (b) for pedestrian ROIs.

The thresholding of the ViPs system is performed with help of the mean and standard deviation of a frame. To validate the chosen parameters $k_1$ and $k_2$, the average over all frames of the mean frame luminance and the standard deviation were calculated for both the dry
4.1. NIGHT VISION LUMINANCE DATA STATISTICS AND THRESHOLDING

weather sequences and the rainy weather sequence:

\[
\begin{align*}
\mu_{F_{\text{mean dry}}} &= 19.37 & \mu_{F_{\text{std dry}}} &= 19.81 \\
\sigma_{F_{\text{mean dry}}} &= 0.48 & \sigma_{F_{\text{std dry}}} &= 2.32 \\
\mu_{F_{\text{mean rain}}} &= 23.62 & \mu_{F_{\text{std rain}}} &= 16.06 \\
\sigma_{F_{\text{mean rain}}} &= 0.46 & \sigma_{F_{\text{std rain}}} &= 0.38,
\end{align*}
\]

where \( F_{\text{mean}} \) is the mean luminance and \( F_{\text{std}} \) the standard deviation of a frame, \( \mu \) and \( \sigma \) are the mean and standard deviation over all frames. It is assumed that \( \mu \) and \( \sigma \) (both of \( F_{\text{mean}} \) and \( F_{\text{std}} \)) have a Gaussian distribution. We checked this assumption by inspection of the histograms (which are not presented here). These two distributions are mutually independent and can therefore be combined to one new Gaussian distribution, characterised by a mean \( \mu_{\text{comb}} \) and standard deviation \( \sigma_{\text{comb}} \). The new mean and standard deviation can be derived from the old distributions:

\[
\begin{align*}
\mu_{\text{comb}} &= \mu_{F_{\text{mean}}} + \mu_{F_{\text{std}}} \\
\sigma_{\text{comb}}^2 &= \sigma_{F_{\text{mean}}}^2 + \sigma_{F_{\text{std}}}^2 \\
\sigma_{\text{comb}} &= \sqrt{\sigma_{F_{\text{mean}}}^2 + \sigma_{F_{\text{std}}}^2},
\end{align*}
\]

With help of the new combined distribution and taking into account that 95% of the data is situated between \( \mu - 2\sigma \) and \( \mu + 2\sigma \), the boundaries of a threshold of the form \( T = k_1\mu + k_2\sigma \) can be calculated:

\[
T_{\text{comb}} = k_1\mu_{F_{\text{mean}}} + k_2\mu_{F_{\text{std}}} ± 2\sqrt{(k_1\sigma_{F_{\text{mean}}})^2 + (k_2\sigma_{F_{\text{std}}})^2},
\] (4.1)

which results in the following thresholds and boundaries for the dry weather sequences and the rainy weather sequence:

\[
\begin{align*}
T_{\text{dry}}(n) &= 58.99 ± 9.33 \\
T_{\text{rain}}(n) &= 55.74 ± 1.78
\end{align*}
\] (4.2)

For evaluation of the performance of the threshold, there are several choices for the error criterion to be used. A first possibility is the average luminance of a ROI. This criterion has some disadvantages, each rectangular ROI contains some background, which lowers the average luminance. Furthermore, clothing can result in a very high variance and therefore in a useless value of the average. On the other hand, having a threshold higher than the average luminance will lead to missed parts of a pedestrian, no matter the variance. Another possible criterion could be the number of pixels in a ROI, higher than the threshold. However, also for this criterion the included background can make the results unreliable. An additional problem with this criterion is that optimising to the filling ratio is difficult. Optimising to a 100% filling ratio should lead to much too small bounding boxes, optimising to a lower ratio can introduce unnecessarily large boxes. A problem for both methods is that the number of detected pixels is not always representative for the usefulness of the result. For example, a pedestrian can be detected for 90%, but if on one (random) scan line no pixels are detected, the connected component analysis will divide this pedestrian in two ROIs. A correct evaluation of a thresholding can only be performed by visual inspection. In our performance evaluation, the average luminance of pedestrian ROIs is used, since this criterion is more suitable for optimisation.

From the average ROI luminances, the mean and standard deviation are calculated:

\[
\begin{align*}
\mu_{F_{\text{ROI mean dry}}} &= 66.30 & \mu_{F_{\text{ROI mean rain}}} &= 50.94 \\
\sigma_{F_{\text{ROI mean dry}}} &= 16.91 & \sigma_{F_{\text{ROI mean rain}}} &= 9.67
\end{align*}
\]
Figure 4.3: Histograms of average luminance distribution for dry weather (a) and rainy weather (b) in pedestrian ROIs.

The histograms of the dry weather sequences and the rainy weather sequence are shown in Fig. 4.3. The histogram of dry weather will be approximated by a Gaussian distribution (in fact, there is a little skew to the right, which makes it more a Chi like distribution). There is a mismatch between the histogram of the threshold $T_{dry}$ (see Eq. 4.2) and the histogram of the average ROI luminance in dry weather (see Fig. 4.3a). The upper bound of the threshold for dry weather ($T_{dry} = 68.32$) can cover only $\pm 50\%$ of the average ROI luminances.

### 4.1.2 Threshold parameter optimisation

To optimise the parameters $k_1$ and $k_2$, the Levenberg-Marquardt (LM) least square optimisation algorithm [24, 25, 26] has been used (see Appendix A). To optimise the parameters $k_1$ and $k_2$, a cost function is defined as a weighted sum of squared differences between the threshold and the average ROI luminance (Fig. 4.4(a)):

$$
\varepsilon(r, n) = \frac{k_1 F_{mean}(n) + k_2 F_{std}(n) - F_{ROI mean}(r, n)}{F_{ROI mean}(r, n)}
$$

$$
sign(\varepsilon) = \begin{cases} 
1 & \text{if } \varepsilon < 0 \\
0 & \text{otherwise}
\end{cases}
$$

$$
cost\ function = \sum_{n=1}^{N} \sum_{r=1}^{R} \frac{1}{2} (1 + \frac{1}{2} sign(\varepsilon)) \varepsilon(r, n))^2.
$$

where $n$ is the frame number, $r$ the ROI number in frame $n$, $\varepsilon$ the error, $F_{ROI mean}$ the average luminance of the ROI, $F_{mean}$ the mean and $F_{std}$ the standard deviation of the frame luminance. By taking the squared differences between the threshold and the average ROI luminance, the error of a threshold higher than the average ROI luminance will be as important as a threshold lower than the average ROI luminance. However, a higher threshold than the average ROI luminance is unwanted. To avoid these higher thresholds, an extra weight of 0.5 is added to the cost function when the thresholds become larger than the average ROI luminance. The error function has a "small valley" shape (Fig. 4.4) and can therefore best be solved with the Levenberg-Marquardt optimisation algorithm [24, 25, 26]. As input for the LM algorithm,
a data set is created with the average ROI luminances of 3 sequences, one rainy weather sequence (±700 ROIs) and 2 dry weather sequences (together ± 700 ROIs), all from the "upper" cameras. The optimised values of $k_1$ and $k_2$ were calculated to be -0.74 and 4.07, respectively. The boundaries of the thresholds for dry weather and rainy weather are then according to Eq. 4.1:

\[
T_{opt, dry} = 66.29 \pm 18.90 \\
T_{opt, rain} = 47.89 \pm 3.17
\]

The threshold range for dry weather covers the complete range of average ROI luminances. In theory, it is possible to have 100% detection. The calculated thresholds for each frame of the test set are shown in Fig. 4.5(a). The jumps in the figure are caused by the concatenation of data of different sequences. 38.5% of the ViPs thresholds are lower than, or equal to, the average ROI luminance. For the LM optimised parameters, this percentage is 57.1%. For verification of the parameters, another data set is taken and again the ROI luminance and thresholds are plot (see Fig. 4.5(b)). 62.3% of the ViPs thresholds and 52.3% of the LM calculated thresholds are lower than, or equal to, the average ROI luminance. The ViPs parameters perform better in the validation set, but the LM optimised parameters give a more constant result. To find a better threshold, a larger data set is required, with both "lower"- and "upper" camera sequences, and sequences with different outside temperatures, both rural and urban.

![Cost function Levenberg-Marquardt optimisation algorithm](image)

Figure 4.4: Cost function for the Levenberg-Marquardt optimisation method. The data tips denote the optimal parameters and the parameters of the ViPs system.

4.2 Gradient based segmentation

Two types of thresholding can be distinguished, global and local thresholding. In the first case, the threshold depends on properties like statistics and individual pixel values of a complete frame. In local thresholding, the threshold depends on the properties of the neighbourhood.
of a pixel or group of pixels. The previous section showed the difficulties to find an appropriate general global threshold for all night vision data. Therefore, we want to find another thresholding method based on local data properties. A typical local object property is the presence of edges. In the next subsection, the statistics of the gradients in night vision data are examined. In Subsection 4.2.2 a thresholding algorithm is presented based on local gradient thresholding and luminance thresholding. In Subsection 4.2.3, a thresholding algorithm is presented, based on edge detection and positive and negative gradient matching.

4.2.1 Gradient statistics

In the literature, the gradient of a discrete signal is defined in many different ways. A popular method is to take the difference between a pixel and its neighbouring pixel. This results in a shift of 0.5 pixel of the gradients. In this report, another popular definition is used:

\[
\begin{align*}
\nabla_x f(x, y) &= \frac{1}{2} \sum F(x + 1, y) - F(x - 1, y) \\
\nabla_y f(x, y) &= \frac{1}{2} \sum F(x, y + 1) - F(x, y - 1),
\end{align*}
\]

(4.6)

where \(x\) and \(y\) are the horizontal and vertical pixel coordinates and \(\nabla_x f(x, y)\) and \(\nabla_y f(x, y)\) the horizontal and vertical gradients at pixel location \((x, y)\). The combined gradient or contrast is then defined as:

\[
\nabla f(x, y) = \sqrt{(\nabla_x^2 f) + (\nabla_y^2 f)}
\]

(4.7)

In this case, the luminance of a pixel, \(F(x, y)\), itself is not used.

Two dry weather sequences and one rainy weather sequence from the first BMW batch were used for determining information about the gradients. For each frame containing a pedestrian, the combined gradient distribution was calculated. The histograms of the frame gradients are shown in Fig. 4.6. Both, dry weather and rainy weather sequences, have mainly
4.2. GRADIENT BASED SEGMENTATION

low gradients with an average gradient of 3.58 and 3.15, respectively. The main difference is the variance, which is in dry weather two times as high as in rainy weather (4.88 against 2.27). For the ROIs, we are only interested in the gradients of the edges. Gradients in the pedestrian body’s will be very low and therefore not interesting. For evaluation, the manually selected rectangular pedestrian ROIs were used (see Section 4.1). In order to find the edges of the pedestrians in the ROIs, the Canny edge detector as implemented in Matlab was used [27]. The Canny edge detector needs two thresholds, a high and a low threshold. Edges exceeding the high threshold are always passed to the output edge image. Edges exceeding the low threshold, but not the high threshold, are only present in the output image if they are connected to an edge of the high threshold edge image. In this way both, strong and weak edges, are detected and noise of weak edges is suppressed from appearing in the output image. The high threshold in Matlab is defined as the value in the gradient histogram of a frame, where 70% of all gradients are lower than this value. The low threshold is defined as 0.4 times the high threshold.

With help of the detected edges of the pedestrians, the edge gradients were calculated and plot in the histograms in Fig. 4.7. The histogram of dry sequences resembles a Gaussian distribution with a mean of 35.02 and a standard deviation of 10.06. The rainy weather pedestrians have a more Chi like distribution with a mean of 22.27. If we compare the frame gradient histograms with the ROI gradient histograms, we can see that the ROI and frame gradients are almost complementary to each other. The histogram of the ROI gradients starts at a gradient value, where the histogram of the frame gradients almost ends. To visualise the differences between the frame and ROI gradients, the cumulative percentage, $p_{\text{cum}}$, of the gradients are plot in Fig 4.8. The function is defined as:

$$p_{\text{cum}}(i) = p_{\text{cum}}(i-1) + \left( \frac{1}{N} \sum_{x,y} g(x,y,i) \right) \times 100\%,$$

where $x, y$ are the pixel coordinates and $N$ the number of pixels in a frame. $g(x, y, i)$ is defined as:

$$g(x, y, i) = \begin{cases} 
1 & \text{if } \nabla f(x, y) = i \\
0 & \text{otherwise}
\end{cases}$$
CHAPTER 4. CANDIDATE SELECTION

For clarity, the percentage of frame gradients is plotted as $100\% - P_{cum}(i)$. Gradient thresholding can give a large data reduction. For example, the number of dry weather frame gradients are reduced with 97%, if for the dry weather sequences a gradient threshold is chosen of 13, while 97% of the pedestrian gradients are preserved (see Fig. 4.8a). In Fig. 4.8 can be seen that for both, dry weather and rainy weather, thresholding on the gradients can give a data reduction of approximately 92%-95% without significantly losing pedestrian data.

4.2.2 Gradient based local thresholding

The previous Subsection showed that thresholding on the gradients can give a large data reduction, without significantly losing information about the pedestrians. For the local thresholding algorithm, this property is a good starting point. In the algorithm to be presented, fixed thresholds were used for dry weather sequences and the rainy weather sequence. The pedestrian gradients (and therefore also the gradient thresholds) will have no relation to the
mean frame gradient or the standard deviation, since these values heavily depend on the content. At the time of research, it was not possible to derive a relation between the gradient threshold and for instance outside temperature or weather type. All available night vision videos were recorded with an outside temperature between -0.5 and 5 °C and just one sequence showed rainy weather. In Fig. 4.9, a block diagram is shown of the local thresholding algorithm.

![Block diagram of local thresholding algorithm](image)

The gradient image is derived with help of Eq. 4.7. For our algorithm, we need as much pedestrian edges as possible, but of course, the number of non-pedestrian edges may not be too high. For dry weather, we set the fixed gradient threshold, $T_{G,\text{dry}}$, to $13 (\mu - 2\sigma$ for the almost Gaussian distribution of dry weather gradients), which in practice means that 97% (95% with a true Gaussian distribution) of the pedestrian edges are covered (see Fig. 4.8a). For the rainy weather sequence, the threshold $T_{G,\text{rain}}$ is set to 6, which covers ±96% of the pedestrian edges (see Fig. 4.8b). With the thresholds, a binary gradient mask $M_G$ is created:

$$M_G(x, y) = \begin{cases} 1 & \text{if } |\nabla f(x, y)| > T_G \\ 0 & \text{if } |\nabla f(x, y)| \leq T_G \end{cases}$$

(4.8)

Because only thresholding is performed and no line or edge detection, the resulting gradient mask $M_G$ suffers from salt and pepper noise, especially in the rainy weather sequence (see Fig. 4.10a). Before proceeding to the local thresholding step, this noise has to be reduced. Salt and pepper noise can typically be reduced with help of a 3 points median filter [28]. However, a median filter is expensive in hardware, since a pixel ordering of the support pixels is required. To avoid this, there is chosen to use the morphological operations image erosion and image dilation [29]. For both operations, a very small kernel, $Q_{DE}$, is used:

$$Q_{DE} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

(4.9)

First the erosion operation is performed, to remove single gradient pixels. By this action, edges are thinned, or partly removed. The erosion will be followed by a dilation, to recover removed edges. The kernel is chosen small to prevent losing complete edges during erosion and connecting different objects during dilation. The result is a binary image mask $M_{G,\text{NR}}$ with smoothed edges and little speckle noise (see Fig. 4.10b).

After noise removal in the binary image, we want to return to luminance images for local thresholding. The threshold will be adaptive to the luminance of the selected gradients in the binary image. The gradient of a pixel is calculated with help of its neighbouring pixels, its own
value is not taken into account (see Eq. 4.7). An edge pixel will therefore not be found on the lowest luminance (background) or highest luminance (pedestrian), but somewhere in between. This results in an edge pixel with a luminance higher than the surrounding background and lower than the object itself. The image is divided in blocks of 8 by 8 pixels. If a block does not contain edge pixels, the luminance threshold for that block is set to 255. If the block contains edge pixels, the mean luminance of the edge pixels is taken as local threshold:

\[
T_{\text{block}} = \begin{cases} 
255 & \text{if } \sum_{m,n} M_{G,\text{NRb}}(m,n) F_b(m,n) = 0 \\
\frac{1}{N_b} \sum_{m,n} M_{G,\text{NRb}}(m,n) F_b(m,n) & \text{otherwise, (4.10)}
\end{cases}
\]

where \(T_{\text{block}}\) is the local threshold for a block, \(m\) and \(n\) are the pixel coordinates in a block, \(N_b\) is the total number of pixels in the block \((m \times n)\), \(F_b(m,n)\) is the luminance of the block pixel at coordinate \((m,n)\) and \(M_{G,\text{NRb}}(m,n)\) is the value of the gradient mask after noise reduction at block pixel \((m,n)\). The block size can be chosen differently, in a higher resolution image it is useful to increase the size. The size may not become too large, because more than one object can appear in a block and influence the local threshold negatively. A small block size will not always cover a complete object, this is not necessarily a problem since the final connected component analysis only needs an outline of the object to obtain the right size. However, a small block size reduces the chance to close gaps in the edge of an object. The local block thresholds are used to threshold the corresponding blocks in the original luminance image. The result is a gray scale image \(Y_g(x,y)\):

\[
Y_g(x,y) = \begin{cases} 
0 & \text{if } F(x,y) < T_{\text{block}}(x,y) \\
F(x,y) & \text{if } F(x,y) \geq T_{\text{block}}(x,y)
\end{cases}
\)

The resulting image \(Y_g\) is depicted in Fig. 4.11 (for clarity, the image is showed as binary image).

The gradient calculation uses only pixel differences, which has the advantage that also objects with a lower luminance can pass the gradient threshold. On the other hand, non-pedestrian objects can also easily pass the thresholding. In Subsection 4.1.1, we have seen

![Figure 4.10: The binary mask after thresholding. Salt and pepper noise appears especially in rainy weather images (a). Most noise can be removed by image erosion and dilation actions (b). A binary 1 is represented by a black pixel and zeros with white pixels. The pedestrians are marked with a green box.](image)
4.2. GRADIENT BASED SEGMENTATION

Figure 4.11: The resulting image after the local thresholding step (a) and the final result of the algorithm (b). Images are shown as binary images for clarity, originally this are grey scale images. The pedestrians are marked with a green box.

that pedestrian luminances were not a significant part of the total frame luminance. Yet, in the gray scale image \( Y_g(x, y) \), the majority of the pixels (mostly background) are already dismissed. This gives the possibility that pedestrians are now significantly visible in the luminance histograms. For validation, histograms were made of the pixel luminances of the \( Y_g(x, y) \) frames containing pedestrians and of the pedestrians in those frames (see Fig. 4.12). The same dry weather sequences and rainy weather sequence were used as in Subsection 4.1.1. For the dry weather sequences, we have a bimodal histogram with a peak at luminance ±50 and ±135. The pedestrian histogram also has a peak at luminance 135, which has a height of \( \pm \frac{1}{3} \) of the peak in the frame histogram. The pedestrians are now a significant part of the frame histogram and a threshold can be defined between the two frame luminance peaks. Unfortunately, this is not true for the rainy weather histograms (not shown here). In the frame histogram of the rainy weather sequence is no bimodal structure. However, the pedestrian histogram covers almost the complete last part of the frame histogram and a threshold can be described with help of Eq. 2.4.

To calculate the global luminance threshold, \( T_{glob} \), a histogram with 32 bins is made of image \( Y_g(x, y) \). To remove local maxima, the histogram is low-pass filtered (an example is shown in Fig. 4.13a). We are not interested in the small peaks, therefore the mean of the histogram is subtracted from the histogram and resulting negative values are clipped to zero. Now, a simple maxima hunting algorithm can be run to find all maxima in the modified histogram \( (MH) \). A maximum is defined straightforward as a point where the magnitude is higher than the previous and next point. Since the histogram was low-pass filtered, we have a smooth signal where maxima comply to this definition. There are three possible outcomes, there are 1, 2 or more maxima. In case of one maximum, we have to deal with a rain sequence, a frame without pedestrians or small pedestrian ROIs on a large distance. In this case, the threshold is set to the mean plus standard deviation of \( MH \) (these values were chosen arbitrary). If two maxima are found, the threshold is set to the mean of the maxima. In case of three or more maxima, a simple clustering algorithm is used to decrease the number of maxima to two. The clustering algorithm calculates the distances between the
Figure 4.12: Histogram of frame luminances in $Y_g(x, y)$ (a) and of the pedestrians (b) in dry weather sequences.

indices of the $x$-coordinates of the maxima. The two maxima with the shortest distance are removed from the list of maxima and replaced by the average of these two. This process is repeated until two maxima remain, the problem is then simplified to a two maxima problem and the average is taken as threshold. The luminance threshold algorithm can be summarised in pseudo code as follows:

$$T_{glob} = \begin{cases} \mu(MH) + \sigma(MH) & \text{if } \#I_{max} = 1 \\ \frac{1}{2}(I_{max}(1) + I_{max}(2)) & \text{if } \#I_{max} = 2 \\ \frac{1}{X} & \text{if } \#I_{max} \geq 3, \end{cases}$$

\[
\begin{align*}
\text{cluster: } & \{ \\
\text{temp} & = \text{min}(d) \\
I_{max} & = \text{remove}(I_{max}(\text{temp}), I_{max}(\text{temp} + 1)) \\
I_{max} & = \text{add}(\frac{1}{2}(I_{max}(\text{temp}) + I_{max}(\text{temp} + 1))) \\
\}
\]

\[
X = \begin{cases} \frac{1}{2}(I_{max}(1) + I_{max}(2)) & \text{if } \#I_{max} = 2 \\ \text{cluster} & \text{if } \#I_{max} \geq 3, \end{cases}
\]

where $I_{max}(i)$ is an array with the indices of the maxima in increasing order, min($d$) searches the minimum in array $d$. "remove" is a routine that removes its arguments from the $I_{max}$ array and "add" adds its arguments to the array. A complete pedestrian can never spontaneously appear or disappear in a frame if it was (not) present in the previous frames. To reduce the effect of outliers in the global luminance threshold, the threshold is temporally filtered by an averaging filter over three frames.

Fig. 4.13(a) shows the histogram for a frame of a dry weather sequence after low-pass filtering. The histogram has the earlier mentioned bimodal structure. The vertical red line indicates the global luminance threshold as calculated by the algorithm (bin 12 or luminance value 91). The final result of the complete gradient local thresholding algorithm is shown in Fig. 4.11(b) for an example of a rainy weather frame and in Fig. 4.13(b) for an example of a dry weather frame.
4.2. GRADIENT BASED SEGMENTATION

Figure 4.13: Luminance histogram of \( Y_g(x, y) \) after low-pass filtering for a frame of a dry weather sequence (a). Note the typical bimodal structure. The vertical red line indicates the final global luminance threshold. The final result after the complete gradient local thresholding algorithm for the dry weather frame is shown in (b). The pedestrian is marked with a green box.

The temporal filter block is followed by a noise removal block. Due to the algorithm of the bimodal/luminance thresholding block, again small spots were created. There are two possibilities: leave the spots untouched, having a higher number of candidates to check and risk more false detections, or try to remove these spots, which costs an extra frame based processing. Here, the choice is made to remove the spots with the same image erosion and dilation action as in the previous noise removal block. The kernel must again be small and is chosen the same as in Eq. 4.9. The final step for candidate selection is an 8 point connected component analysis (see Subsection 2.1.2) to switch from pixel based processing to object based processing.

The gradient and luminance statistics of this algorithm were verified with three dry weather sequences, not used for earlier statistics. The results were almost similar to those described in this section. However, as mentioned earlier, the available video sequences are all of almost the same temperature. The gradient statistics may be different in other sequences. In the available sequences, the pedestrians appear in the upper part of the luminance histogram. Block 5 and 6 in the block diagram of Fig. 4.9 (global luminance thresholding and temporal filtering) are no longer correct if this is not the case for other conditions/cameras. Connected component analysis can then be done after the local thresholding step.

4.2.3 Gradient matching

The previous section showed that gradient thresholding can be useful for pedestrians detection. In the gradient based local thresholding algorithm, no real edge detection algorithm was used, neither edge thinning, nor closing. Although low luminance objects had the possibility to pass the thresholds, the algorithm still depended on luminance thresholding and a missing line could break up the connected component analysis and pedestrian detection. In this section, we want to present a second gradient based thresholding and candidate selection method. In this algorithm, it is not necessary to use luminance thresholding. The global structure of the algorithm is shown in the block diagram in Fig. 4.14. All blocks will be
explained in the next paragraphs.

The algorithm starts with an edge detection on the original grayscale night vision image. For the edge detection, the Canny edge detection algorithm [27] is used. This detector takes edge orientation into account, performs edge thinning and gap closing. The gap closing is performed by using two thresholds, a low threshold $T_L$ and a high threshold $T_H$. If an edge with pixel luminances higher than $T_L$ and lower than $T_H$, is connected to an edge with pixel luminances higher than $T_H$, than the edge is passed to the output edge image. The thresholds are fixed numbers for the rainy weather sequence and the dry weather sequences for the same reason as explained in the previous section. The threshold $T_L$ is set to 13 for dry weather sequences, and to 6 for the rainy weather sequence, just like the thresholds in the gradient-based local thresholding algorithm. The high threshold, $T_H$, is chosen as $2.5 \times T_L$. If $T_H$ becomes higher than the highest gradient in the frame, $T_H$ will be lowered to 13 (6 for rain) and $T_L$ to $0.4 \times T_H$. The output of the edge detector is a binary image $Y_{canny}(x, y)$ with a one for each edge pixel and zero for all other pixels (see Fig. 4.15).

The edge detector block is followed by an 8-point connected component analysis as described in Subsection 2.1.2. This block is the start of two candidate selection chains, a blue and a red chain. First the red chain will be explained. Pedestrians at a large distance from the camera appear mostly as a solid spot in the night vision images, pedestrians nearby on the other hand show more detail. Because of this, the edge detector finds complicated structures for pedestrians nearby (see Fig. 4.15b) and often complete contours for pedestrians at a large distance, with sometimes one or more connected edges of the horizon line (see Fig. 4.15a). We want to pick up this last type of pedestrians in the ROI list in the red chain. Requirement for these detection blocks is that a pedestrian has a complete contour, with at most one gap in one of the horizontal edges (for example, one side of the body is missing). Pedestrians who do not fulfill this requirement, have to be detected in the blue chain. All actions in the chain are performed on the ROIs coming from the connected component analysis individually. The first red block makes a vertical projection on the $x$-axis of the number of edges available in each column of an ROI. Next, the intervals where 2 or more edges in a column can be found.

![Block diagram of gradient matching algorithm for candidate selection.](image)
4.2. GRADIENT BASED SEGMENTATION

Figure 4.15: Edge images $Y_{canny}$ from Canny edge detector with pedestrians far away (a) and nearby (b). Pedestrians are marked with green boxes.

in the vertical projection, are stored. In Fig. 4.16, an example is shown of an edge image of an ROI with at the left side a pedestrian, connected to the horizon line (top image). In the bottom image, the corresponding vertical projection is shown. The red lines show the intervals that will be selected by the first red block of the algorithm, in this case [1 5] and [27 36].

The second red block makes for each interval a horizontal projection on the y-axis of the number of edges in all scan lines of the ROI. The vertical intervals are searched, where one or more edges are available. In our example ROI of Fig. 4.16, the chosen vertical intervals would be [1 8] and [4 9] (for the x-intervals [0 5] and [27 36], respectively). Each valid combination of x- and y-intervals is stored in the ROI list for candidate classification. The second red block can be extended with the requirement that more than $p$ percent of the vertical interval needs at least two edges. With the red chain, all more or less trivial pedestrians can be detected. A disadvantage is that, for example, also all rear lights will be perfectly detected in this part.

In the blue chain, all other pedestrians like partly visible ones have to be detected. Because of insulating clothes, some body parts become less visible. However, trousers are normally less insulating and the legs and the head are clearly visible in the night vision data. By joining these parts, we want to detect these pedestrians. The basic idea of the blue chain is the presence of relatively many high horizontal gradient pixels on top of each other (caused by the side of the body and the legs) in an ROI containing a pedestrian. For each ROI, the two columns are searched with the highest total positive and negative gradient. If a pedestrian is split into two or more parts, there should be more ROIs with a peak in the total positive and negative gradient in the same column of the image. If such other ROIs are found, then the edges in these columns can be merged. This is done for the positive and negative gradients separately. If the combined ROI contains a pedestrian, then the merged positive gradient edges should have more or less the same height and y-coordinates as the merged negative gradient edges. If this is indeed the case, then the combined ROI will be classified as possible pedestrian.

In the first blue block "Positive/Negative Maximum search", a gradient image $Y_{Gx}(x, y)$ is needed, with all pixel gradients $V_x f(x, y)$ in x-direction (see Eq. 4.6). $Y_{Gx}(x, y)$ is an intermediate product of the Canny edge detector and therefore already available. An overview of this first blue block is given in Fig. 4.17. For each ROI, two vertical projections are made
on the x-axis. One projection is from the positive gradients larger than threshold $T_G$ and one projection of the negative gradients smaller than $-T_G$. $T_G$ is again the gradient threshold, 13 for dry weather sequences and 6 for the rainy weather sequence. The vertical projections will be used for finding the x-coordinates of the maxima and minima in the positive and negative gradients, respectively. For the vertical projection of the negative gradient, the absolute values will be used in order to re-use the maximum search algorithm. The horizontal gradient of a pedestrian is considered to give a large peak in the projection data. To prevent smaller peaks for appearing in the maxima list, the average magnitude of the projection data is subtracted from the projection data itself. In the upper part of Figure 4.18, an example is given of an ROI containing the legs of a pedestrian. In the lower part, the corresponding vertical projections of positive gradients (blue line) and negative gradients (red line) are plot. The bounding boxes can be very small, sometimes 3 or 4 pixels, and flat peaks are not uncommon. This makes it hard to find the exact locations of the maxima, which are very important in this algorithm. This is solved by considering all locations of a flat peak as location of a maximum. Most false locations will be rejected further on in the algorithm or in the candidate classification. The output of this block are two lists, $\text{MaxPos}$ and $\text{MaxNeg}$, of x-coordinates of the maxima for positive and negative gradients in each ROI.

The second blue block, "Begin/End Positive/Negative Edge Listing", searches the vertical intervals belonging to each x-coordinate of $\text{MaxPos}$ and $\text{MaxNeg}$. The boundaries of an interval are selected as the lowest and highest y-coordinate of the edge pixels in a column, exceeding threshold $(-)T_G$. The output of this block are two lists (one for positive and one for negative gradients) with the x-locations and corresponding y-intervals of possible vertical pedestrian edges. In Figure 4.19, an example is shown. The output of this block are indicated with red boxes.

The third blue block, "Vertical Edge matching", tries to combine entries from a list with other entries in its own list, to a new entry. For each ROI is searched to other ROIs on the own list where the x-coordinate is within a radius of $N$ pixels of its own x-coordinate. For each two ROIs that satisfy this constraint, a new entry is created in the list with a new combined y-interval. This is performed for the positive and negative gradient ROIs separately. The
variable \( N \) is chosen to be 4 in our implementation, but must be chosen differently for other resolution images or can be made proportional to the height of an \( y \)-interval. An example of an output of this block is shown with blue boxes in Figure 4.19.

The last blue block, "Horizontal Edge Matching", finally tries to combine bounding boxes of the positive gradient list with corresponding boxes of the negative gradient list. Before a positive gradient list entry is combined with a negative gradient entry, the following conditions have to be met:

- The \( x \)-coordinate of a negative gradient has to be larger than the \( x \)-coordinate of a positive gradient.
- The heights of the \( y \)-intervals may at most differ \( H\% \) of each other.
- The minimum and maximum \( y \)-coordinates may at most differ \( V \) pixels of each other.
- If the positive or the negative gradient is an entry consisting of two vertically combined parts of the third blue block, than the mean luminance of the upper part and lower part may differ at most 60% (see Subsection 5.4.2). With the upper and lower part is meant the pixels between the positive and negative gradient of the original entries, before the vertical merging was performed.

The variable \( H \) is chosen in our implementation as 20\% and \( V \) is set to 10 pixels. A positive-negative gradient pair is discarded, if one of the conditions is not met. All other pairs are passed to the ROI list and added to the entries of the red detection chain. The result of this last block in the detection chain is visualised with a green box in Figure 4.19. The ROI is somewhat too small, the head and feet are not completely detected, but this should not be a problem for the candidate classification. If necessary, the algorithm can be extended by taking also vertical gradients into account. Head and feet produce high vertical gradients and are clearly visible in a gradient image. Drawback is that the algorithm becomes much more
complicated, computationally almost twice as expensive and because of the small width of a pedestrian, a there exists a high risk for errors.

The algorithm is able to detect pedestrians that appear as two or more hot areas due to insulating clothes. Furthermore, since there is only processing based on gradients (absolute pixel differences), low luminance objects can be selected as candidate pedestrian.

4.3 Candidate selection performance

In the last two sections, four candidate selection methods were described. Two based on luminance thresholding (in fact one method but with two different sets of parameters), and two based on gradient thresholding. In this section, we want to benchmark their performance and more specifically, the detection rate. The purpose of the detection rate will in this benchmark be used slightly different from its normally used application. Normally, the detection rate indicates the percentage of positive detections of a complete pedestrian detection algorithm. Here, the detection rate will be used to indicate the percentage of the pedestrians that are available in the ground-truth data as well as in the set of selected pedestrian candidate ROIs. The higher the detection rate, the better the candidate selection method, no matter the number of false detections. In case of equal detection rates, the algorithm with the least false rate is better. The detection rates of this test are not necessarily the maximum achievable detection rates. With help of post processing, like splitting combined objects and merging separate legs and heads, the detection rate can still be increased. However, the found detection rates give a good indication of the performance.

For the benchmark, the metric presented in Section 2.3 is used. The detection and false rates are calculated for different values of SSA (see Eq. 2.7). SSE is in all calculations chosen equal to SSA. If a candidate ROI has a larger overlap with the pedestrian, it may also not be "too large". If SSA approaches a value of 1, the candidate must match the pedestrian almost
perfectly. In practise, this situation will rarely occur due to the square root in Eq. 2.7. The interesting domain in SSA and SSE will be between 0.2 and 0.6 (44.7% and 77.5% overlap).

The test is performed for the first and the second batch of BMW videos and for the dry weather and rainy weather sequences separately. For the dry weather sequences, the detection rate is calculated as the average detection rate of all separate videos. The detection and false rates for the first batch are plot in Fig. 4.20 (rainy weather (a) and dry weather (b)). In the rainy weather sequence, the LM parameters perform much better than the ViPs parameters. However, this sequence was 50% of the training set for the LM parameters, which explains the large difference. In the dry weather sequences, the results are almost equal, just like the \( FR \), which is almost continuously 100%. The gradient based local thresholding algorithm performs much worse. If the \( DR \) and \( FR \) are calculated before the "bimodal/Luminance thresholding" step (see Fig. 4.9), similar results as with the luminance thresholding are achieved in dry weather and better results in rainy weather.

The gradient matching algorithm performs much better than all other algorithms in the rainy weather sequence. In dry weather, it scores for almost all SSA values better (up to 20%). The \( DR \) drops very fast after an SSA value 0.67, what can be easily explained. The gradient matching algorithm detects mainly the sides of the body and will miss detect the arms, most ROIs will therefore be too small. Also the top of the head and feet are mostly not detected (see Fig. 4.19). The \( FR \) of the gradient matching method is smaller than all the others.

The \( DR \) and \( FR \) are also calculated for the second batch of BMW videos (see Fig. 4.21). Differences become here even more clear. None of these videos were in the training set of the LM optimisation. With their totally different characteristics, they floor the LM optimised parameters. As mentioned in Section 4.1, a much larger training set is required. The gradient based local thresholding suffers from many connected objects. Too much pixels pass the thresholds and objects become connected to each other. The luminance thresholding algorithm with ViPs parameters performs rather good, but suffers from ROIs falling apart in
multiple ROIs if pedestrians come closer to the camera. The gradient matching algorithm outperforms the other algorithms in these sequences, a very high DR and the lowest FR. At ±0.6, the DR drops again very fast.

Not only the DR and FR are important, also the number of selected ROIs for candidate classification. More ROIs means a higher chance on false positives and a larger computation time, less ROIs increases the chance on a low DR. For the ViPs, LM, gradient local threshold and gradient matching algorithm, the average number of ROIs per frame were calculated for all sequences together and are 108, 147, 26 and 117, respectively. The gradient local threshold algorithm gives by far the least number of ROIs, but performs also worse. The LM optimised algorithm has by far the most ROIs per frame, this can mainly be ascribed to the rainy weather sequence, where it generates as much as 516000 bounding boxes in only 1300 frames. The gradient matching and the ViPs algorithm have a moderate average, but the number of ROIs of the gradient matching algorithm heavily depends on the scene content. Urban sequences will rapidly increase the number of candidates.
Chapter 5

Candidate classification

Candidate classification is performed in the ViPs system with a cascade of feature filters, which eliminates non-pedestrians in the candidate list. The result typically still contains false positives due to limited reliability of these filters. An overview and explanation of the filters were given in Chapter 3. In this chapter, new filters and modifications to existing filters are proposed. In the first section, two methods to calculate the distance between a pedestrian and the camera will be described. In the second section, a new horizon reliability measure will be presented. In the third section, an object splitting algorithm will be proposed to divide pedestrians connected to other objects into separate objects. The fourth section describes four new proposed symmetry filters. In the last section, a modification to the aspect ratio filter will be presented.

5.1 Pedestrian distance

The distance of a pedestrian to a car is interesting information for both the detection system and the driver. We appear to react faster to audio-visual signals, than to visual signals alone. An application of distance calculation can therefore be an audio signal, in addition to the visually highlighted pedestrian, when the pedestrian comes within the critical distance of an emergency stop. Pedestrian ROIs at a large distance of the camera (e.g. 200 m) have typically a small size, for example three by five pixels. Detection of these ROIs results in an increase of the number of false detections. If a car is driven with a speed of 50 kilometres per hour, the critical distance for an emergency stop will be approximately 30 m (depending on the reaction time). The detection of pedestrian ROIs on 200 m distance is then less useful than when driving for example 100 kilometres per hour. The false rate can be decreased by using a speed dependent minimum detection distance. An additional advantage is that the driver will not be distracted unnecessarily by highlighted ROIs on the display which are less important. A last, less useful, reason for distance calculation, is a marketing reason. An option to display the distances on the cars display, can be a nice feature for advertising. For the first two applications, the driving speed and the reaction time of the driver have to be known. The driving speed is digitally available in the car and can easily be used. The reaction...
time of a driver has been a research topic for many years and is well described and discussed in the literature. A detailed discussion can be found in the article of Green [30]. The reaction time is composed of a mental and physical reaction time and is approximately 1.5 seconds for objects, which move suddenly into the driver's field of view. The reaction time can vary a little with other factors like age, gender, weather type and complexity of the traffic situation.

The distance can be measured in a few different ways. In our detection system, some (intrinsic) camera parameters are unknown. The calculations will therefore be limited to distance estimations, based on object size and object motion. In the next section, known camera parameters will be mentioned and some general problems will be explained. In the second section, the distance will be calculated with help of the object size and in the last section with object motion.

5.1.1 Camera parameters

The camera used by BMW for the night vision sequences was a Raytheon 2000B. Specified were the pixel size on the sensor die (50 $\mu$m) and the field of view (FOV) of the camera (36°×24°horizontally and vertically, respectively). The velocity and steering angle of the test car were available in a text file for each frame of a night vision video. The last specified value was the position of the camera mounted on the car. For the distance calculation, only the sequences of the "lower" camera were used. The position of the camera is schematically drawn in Fig. 5.1. The camera C is positioned 0.35 m left from the middle axis of the car and 0.33 m above the road. The width of the car used for the night vision sequences is unknown. According to technical information of various BMW cars [5], the width is set to 2.0 m.

The most important unknown camera parameter is the focal length $f$. Given the camera type and FOV, the product specification mentions a 25 mm lens with an adjustable focal length. $f$ can be calculated by assuming a pinhole camera model and measuring the height of a pedestrian in the image (in pixels) on a known distance and comparing that to the actual (a-priori known) height. In principle, this is a reverse calculation of the method that will be described in next section. All calculated focal lengths lie between 24.7 and 25 mm. It is very likely that the focal length was set to 25 mm, since pedestrians can then be detected on the largest distance. In the rest of the calculations, $f$ is assumed to be 25 mm.

There is one important unknown parameter left: the height of the detected pedestrian. This variable will always be unknown. The height can be calculated from the detected ROIs, this possibility will be mentioned in next section. In our calculations, the height is set to 1.79 m, which is a reasonable average height in NW Europe for adults, but not applicable for children and other regions.

To validate the calculated distance, the actual distance between the pedestrian and the car must be known. Unfortunately, this distance is unknown for the available video material. However, since the frame rate (30 fps) and the speed are known, the distance can in theory be estimated very well. Only sequences where pedestrians walk along the road (not on the pavement) and where the road is straight (see for example Fig. 3.1) were used. The distance can in theory be calculated by counting the number of frames between the moment of first sighting and the moment of passing of a pedestrian. The only unknown part remains the distance between the pedestrian and the car, when the pedestrian leaves the field of view. Here, again an assumption has to be made. It is assumed that a car is driving along a
pedestrian with a horizontal distance of 1 m between the side of the car and the side of the pedestrian (see Fig. 5.2). The distance \( y \) can then be calculated by simple geometry resulting in 7.23 m. To be precise, the diagonal (real) distance should be calculated, but for sake of simplicity \( y \) is calculated as approximation. The angle of the FOV is small enough to justify this simplification.

In theory the actual distance can be calculated very well with these (camera) parameters. However, due to all assumptions the calculated actual distance will be rather imprecise. For the estimated distance, which will be calculated in next sections, an additional error will be produced by the uncertainty of the position of the detected ROI.

### 5.1.2 Size-based object distance

Knowing the camera parameters mentioned in the previous section, the distance from the camera to a pedestrian can be calculated with help of the size of the detected ROIs (in the image plane). The distance calculation will be described in the first part of this section. In the second part, the actual height of a pedestrian will be estimated by combining the detected ROI sizes in two different frames and the driven distance of the car. In the third part, the calculated distance will be refined by applying a median filter to the detected ROI heights. In the last part conclusions will be drawn.

For the distance calculation, the camera system is modelled by a pinhole camera model (see Fig. 5.3). An object \( X \) is positioned on a distance \( z_{w, obj} > f \), where \( f \) is the focal length of the camera. The object with height \( H_{obj} \) (in metres) is then represented by an object with height \( H_{im} \) (in pixels) in the image plane. The same applies to the width \( W_{obj} \). The distance \( z_{w, obj} \) can now be calculated by simple geometry:

\[
z_{w, obj} = -f \frac{H_{obj} - H_{im} S_p}{H_{im} S_p}, \tag{5.1}
\]
where $s_p$ is the pixel size on sensor die. Or written more generally:

$$z_w = -f \frac{y_w - y}{y},$$

(5.2)

where $y$ can also be replaced by an $x$. The correctness of the calculated distance depends on the accuracy of the detected ROIs and the estimated pedestrian height (1.79 m). The error in the detected ROI height (in the image plane) will vary from frame to frame. An error in the estimated pedestrian height will cause a constant error in the computed distance. The sensitivity of $z_{w, obj}$ to the ROI height can be calculated by taking the first derivative of $z_{w, obj}$ and is quadratic inversely proportional to the ROI height:

$$\frac{dz_{w, obj}}{dH_{im}} = \frac{H_{obj} f}{H_{im}^2}$$

(5.3)

As an example, the distance is calculated for a sequence of the second batch of BMW videos. The car is driving constantly 50 km/h on a straight road. For this sequence, no text file with meta-data with yaw rate and speed is available. Yaw rate is considered to be zero, because of the straight road. The reference distance is calculated as described in previous section and is determined to be 95 m. ROIs detected by the gradient matching candidate selection method were used for calculation of the distance with the pinhole model. The results are shown in Fig. 5.5, where the green line represents the reference (actual) distance and the blue line the calculated distance with the pinhole model. Two remarkable occurrences can be seen in Fig. 5.5: the sharp peaks in the calculated distance due to inconsistencies in the detected ROI heights and the general offset. The offset can be caused by several factors. The assumptions made in the previous section are not completely correct and cause inaccuracies. The horizontal distance between the pedestrian and the car can be more than 1 metre (for example, if the distance is two metres, the reference distance will get an offset of +3 m) or the focal length is slightly smaller than 25 mm, which will give a negative offset of the calculated distance. The height of the pedestrian was a guess and may cause an additional offset. A last shortcoming in the distance calculation is the ignored movement of the pedestrian. If the pedestrian is moves in the same direction as the car, the calculated distance will be too high.
in comparison with the reference distance, where the pedestrian is assumed to stand still (in 185 frames, the difference can amount up to 10 metres).

The height of a pedestrian can be estimated by measuring the height of the corresponding ROIs in two different frames (see Fig. 5.4). The distance $\Delta z_w$ between the two ROIs can be calculated from the car speed, the frame rate and the number of frames between the measurements. The pedestrian height can now be determined by calculating the distance difference with help of Eq. 5.1:

$$H_{obj} = \frac{H_{im1} H_{im2} z_{w,1} - z_{w,2}}{f} = \frac{H_{im1} H_{im2}}{f} \frac{\Delta z_w}{\Delta H_{im}}$$  \hspace{1cm} (5.4)

$$\frac{\partial H_{obj}}{\partial H_{im1} \partial H_{im2}} = -\frac{2 \Delta z_w H_{im1} H_{im2}}{f (H_{im1} - H_{im2})^3}$$  \hspace{1cm} (5.5)

where $H_{im1}$ and $z_{w,1}$ are properties of the pedestrian $M$ frames before the current frame and $H_{im2}$ and $z_{w,2}$ are pedestrian properties of the current frame (see Fig. 5.4). The accuracy of the calculated pedestrian height depends on the accuracy of both detected ROI heights. The sensitivity is given in Eq. 5.5. For the same sequence where the distance was calculated (50 km/h, straight road), the pedestrian height is calculated. The delay between the two ROIs, $M$, used for the computation was set to 5 frames. A larger delay will improve the accuracy of the detected ROI height, but the inaccuracy of $\Delta z_w$ increases also due to pedestrian movement. The results are plot in Fig. 5.6. As we can see, the height calculation of pedestrians on a large distance is very inaccurate. When the pedestrian approaches the car, the calculated height starts fluctuating very heavily. The calculation of the height turns out to be useless. Even if the height was accurate the last 20 frames, this would be too late to use for distance calculations. The estimated pedestrian height of 1.79 m can therefore not be refined during the distance calculation and will always cause a constant offset in the calculated distance.

The inaccuracies in the detected ROI heights by the candidate selection algorithm, caused peaks in the calculated distance (see Fig. 5.5). The accuracy can be increased by estimating the ROI height in the image plane by using the height of the ROI $M$ frames in the past and

---

**Figure 5.5:** Calculated distance between pedestrian and car. Green line: reference distance. Blue line: calculated distance with pinhole model. Red line: calculated distance with pinhole model and median filter.

**Figure 5.6:** Calculated height of a detected pedestrian.
the driven distance \( \Delta z_w \). The height can be estimated by rewriting Eq. 5.4:

\[
H_{im,\text{est}}^{n-M}(n) = \frac{H_{obj} f H_{im}(n-M)}{H_{im}(n-M) \Delta z_w(n-M) + H_{obj} f},
\]

(5.6)

where \( n \) is the current frame number. To extend the number of estimations, an array is created with all height estimates of the \( M \) past frames:

\[
H_{im,\text{est}} = \begin{pmatrix}
H_{im,\text{est}}^{n-1}(n) \\
H_{im,\text{est}}^{n-2}(n) \\
\vdots \\
H_{im,\text{est}}^{n-M}(n)
\end{pmatrix}
\]

(5.7)

A single ROI height will be selected for distance calculation with Eq. 5.1. The selection was performed with a median filter [28], with \( H_{im,\text{est}} \) and \( H_{im}(n) \) (the ROI height in the current frame) as input:

\[
H_{im,\text{med}}(n) = \text{med}(H_{im,\text{est}}, H_{im}(n))
\]

(5.8)

In this way, outliers can be excluded from selection. Because of pedestrian movement, causing inaccuracies in \( \Delta z_w \), \( M \) must be kept small. For validation of this method, the distance was again calculated for the test sequence with a car driving 50 km/h. The result is plot in Fig. 5.5 with a red line. The calculated distance is now much more stable, almost without abrupt peaks.

Distance calculation based on the pinhole model turned out to be too inaccurate to display on a driver information system if a small step size is desired. For distance indications, a step size of approximately 20 m can be used fairly safe. For applications like a speed-distance adaptive detection system, the calculated distance can be used, taking into account some error region. The calculation can be refined by increasing the number of past frames, \( M \), if the walking direction of a pedestrian can be obtained by motion detection. In our example, a straight road was used, which simplified the calculation of \( \Delta z_w \). In other cases, the yaw rate must be taken into account.

### 5.1.3 Motion-based object distance

Our second method to calculate the distance between a pedestrian and the camera is based on motion. The motion based method does not need the actual pedestrian height as input, which eliminates an addition error source. Furthermore, the ViPs group has considerably expertise in the area of motion estimation and motion based algorithms. The camera will again be modelled by a pinhole camera model (see Fig. 5.3). In this section, we will follow the manuscript of Braspenninck [31] about perspective flow. The transformation of real world coordinates to image coordinates using the pinhole camera model can be written as:

\[
x = -\frac{fx_w}{z_w - f},
\]

(5.9)

where \( x \) can also be replaced by an \( y \).

A displacement of a point \( X \) to the point \( X' \) in a time \( \delta t \) can generally be written as [22]:

\[
X' = RX + T,
\]

(5.10)
5.1. PEDESTRIAN DISTANCE

where \( R \) is the rotation matrix and \( T \) the translation matrix. The rotation matrix will denote rotation angles around one of the coordinate axes \( \theta \). If the angles in the rotation matrix are assumed to be small (\( \sin \theta \approx \theta \) and \( \cos \theta \approx 1 \)) and if second and higher order cross terms are neglected \(^{22} \), the matrix becomes:

\[
R = \begin{pmatrix}
1 & -\theta_z & \theta_y \\
\theta_z & 1 & -\theta_x \\
-\theta_y & \theta_x & 1
\end{pmatrix} = \begin{pmatrix}
0 & -\theta_z & \theta_y \\
\theta_z & 0 & -\theta_x \\
-\theta_y & \theta_x & 0
\end{pmatrix} + I = \Theta + I \tag{5.11}
\]

Substituting Eq. 5.11 in Eq. 5.10 and rewriting this equation, gives:

\[
X' - X = \Theta X + T \tag{5.12}
\]

Dividing by \( \delta t \) and taking the limit of \( \delta t \) to zero gives the angular and linear velocities \( \Phi \) and \( V \):

\[
\dot{X} = \Phi X + V \tag{5.13}
\]

The velocities in the 2D image can be obtained by differentiating Eq. 5.9 to time:

\[
v_i = \frac{\partial x_i}{\partial t} = \frac{f \dot{x}_i}{f - x_z} + \frac{x_i \dot{x}_z}{f - x_z}, \tag{5.14}
\]

with \( i = x, y \) the index of the used axis. Substituting Eq. 5.13 in Eq. 5.14 gives us:

\[
v_x = \frac{f \dot{x}_x - \dot{\phi}_x \phi_y + \dot{\phi}_y \phi_x + v_x}{f - z_w} + \frac{x \dot{\phi}_x - \dot{\phi}_x x \phi_y + \phi_x v_x}{f - z_w} \tag{5.15}
\]

It is assumed that a car does not have rotations around the \( x \) and \( z \) axis, so \( \phi_x = \phi_z = 0 \). Occasionally, these variables are non zero in case of an obstacle such as a speed bump or pot­-hole, but these cases are not considered here. Substituting Eq. 5.9 in Eq. 5.15 and applying the assumptions about the rotations, results in:

\[
v_x = \frac{f v_{x.w} + x \phi_y - (x \phi_y - \phi_x x \phi_y + \phi_x v_x)}{f - z_w} \tag{5.16}
\]

The equations for the velocity in the image, \( v_x \) and \( v_y \), are now described by a so-called pan­zoom camera model \(^{28} \), where \( v_{x.w} \) and \( v_{y.w} \) are the translations \( t_x \) and \( t_y \) of the camera in \( x \) and \( y \) direction, \( v_{z.w} \) is the zoom \( s \) of the camera (here the forward or backward movement of the car) and \( \phi_y \) is the rotation \( r \) of the camera:

\[
v_x = at_x + b_1 s + c_1 r_y \]
\[
v_y = at_y + b_1 s + c_1 r_y \tag{5.17}
\]

From both velocity equations \( v_x \) and \( v_y \), the distance \( z_w \) between the camera and pedestrian can be solved. The vertical movement of the pedestrian in the image plane is very limited, especially in the sequences shot by the "lower" camera. Therefore, the horizontal movement will be used to calculate the distance (an extra index will be used to denote that \( z_w \) is calculated from \( v_x \)):

\[
z_{w,x} = \frac{-f^2 v_{x.w} - f x^2 \phi_y + f x z_{z.w} - f^2 v_x}{f^2 \phi_y + x^2 \phi_y + f v_x} \tag{5.18}
\]
A car cannot have a translation in x or y direction during natural driving conditions (with the exception of a sideslip), thus \( v_{x, y} = v_{y, x} = 0 \). The velocity \( v_x \) can be calculated with help of the displacement, or motion, vector in x direction for the object between two frames:

\[
v_x = \frac{(x(n) - x(n - M)) \cdot S_p \cdot P_r}{M}, 
\]

(5.19)

where \( n \) is the current frame number, \( M \) the number of frames delay used for the displacement calculation, \( S_p \) is the pixel size on die and \( P_r \) is the frame rate of the camera (30). \( x \) is the x-coordinate of the object in the image plane in frame \( n - M \), where the origin of the coordinate system is in the centre point of the image ((160, 120) in our case). \( z_{w,x} \) represents the distance of the object in frame \( n - M \). We need the distance in frame \( n \), therefore the driven distance \( \Delta z_w \) (see Fig. 5.4) has to be subtracted:

\[
z_{w,x}(n) = -\frac{-f x^2(n - M) \phi_y + f x(n - M) v_{z,w} - f^2 v_x}{f^2 \phi_y + x^2(n - M) \phi_y + f v_x} - \Delta z_w
\]

(5.20)

All variables in the equation of \( z_{w,x}(n) \) are available for our night vision system. \( v_{z,w} \) is the speed of the car, \( \phi_y \) is the yaw rate, the frame rate \( P_r \) and the pixel size on die \( S_p \) are given and the position \( x \) is available from the detected bounding box. The focal length \( f \) was calculated in Subsection 5.1.1 and the object speed \( v_x \) can be calculated with motion tracking or motion estimation.

For evaluation, the distance is calculated for the same sequence as used in the previous section (driving 50 km/h, straight road). The meta data with the speed and yaw rate is not available for this sequence. The yaw rate is considered to be zero. The initial pedestrian distance in frame 1 was calculated to be 95 metres (see Subsection 5.1.1). The used ROIs were detected by the gradient matching candidate selection method. The delay between the two ROIs used for the distance calculation, \( M \), was set to 10 frames. The centre point of the ROI was defined as position \( x \) of the pedestrian. The pedestrian velocity \( v_x \) was calculated from the centre point of the ROI in frame \( n - M \) to the centre point of the same (displaced) ROI in frame \( n \) (see for example the black arrow in Fig. 5.4). The result is shown in Fig. 5.7. The green line is the reference (actual) distance, the blue line the calculated distance \( z_{w,x} \). Inconsistencies in the detected width of the ROI by the candidate selection algorithm causes peaks in the calculated distance. The error of the calculated distance is higher than the error of the calculated distance with the pinhole model when the pedestrian is at a large distance of the camera. This can mainly be ascribed to the high error in terms of percentage in the detected pedestrian width (which is only a few pixels). When the pedestrian is near the camera, the motion model becomes more accurate than the pinhole model (7 metres error compared to 15 metres error at frame 180). Just like the pinhole model based distance calculation, a general offset can be seen between the reference distance and the calculated distance. This is most likely caused by inaccuracies in the calculated focal length and the assumed horizontal distance between the car and the pedestrian.

In order to remove the peaks in the estimated distance, a median filter will again be applied to make the calculated distances more robust. The distance can be estimated by subtracting the driven distance from the calculated distance \( M \) frames earlier:

\[
z_{w,x \text{-est}}^{n-M}(n) = z_{w,x}(n - M) - \Delta z_{w,[n,n-M]} 
\]

(5.21)

To make the estimation more accurate, an array with estimations of the last \( M \) frames is created, analogue to the height estimate array of the previous section. The distance will now
5.2. RELIABILITY OF THE HORIZON DETECTION

Figure 5.7: Calculated distance between pedestrian and car using motion. Green line: reference distance. Blue line: calculated distance with motion model. Red line: calculated distance with motion model and median filter. Note the decreasing error when the pedestrian approaches the car and the smoothing effect of the median filter.

be selected by a median filter:

\[
z_{w,x,med}(n) = \text{med}[z_{w,x,est}, z_{w,x}(n)]
\]  

Outliers can be avoided in this way. The result of the median filter is plot in Fig. 5.7 with a red line. The number of frames \( M \) used for the estimation was set to 10 frames. The result is a much smoother distance plot.

The distance calculation with the motion model is just like the pinhole based model too inaccurate for a driver information system. Especially, if a pedestrian is at a large distance of the camera, the calculated distance is very inaccurate, with errors up to 200% and even a peak to 300%. Up to a distance of 50 metre, the calculated distance is more accurate than the pinhole based calculation. The calculations can be used for a distance-speed adaptive detection system if an error is taken into account of ±20-30 m. With motion estimation, it is possible to calculate a more reliable velocity \( v_t \). Yet, the calculation is dependent on the detected width of the pedestrian, which can be very inaccurate. In our example, a straight road was used. For other sequences, the yaw rate must be used for \( \phi_y \) and for the calculation of \( \Delta z_w \). The motion based model has the advantage that the height of the pedestrian is not required, which saves a possible additional error source.

5.2 Reliability of the horizon detection

In Subsection 3.2.1, the horizon detection was described. The detection was based on two assumptions. First, the road and sky have to be a significant large part of the scene, and second the road and sky must be colder than the other objects. Based on these assumptions, the road-sky mask \( M_{RS} \) (see Eq. 3.1) will have the shape of an hourglass. Furthermore, it is assumed that the road forms a larger area than the sky. Unfortunately, these assumptions are not always correct (see Subsection 3.2.1). Sometimes, no or little sky is visible in \( M_{RS} \) and the road can be warmer than other objects as in the second batch of night vision videos. A side effect is that the sky becomes a larger area than the road in the \( M_{RS} \) mask. This results
in inaccurate horizon estimations and since all ROIs higher than the horizon will be rejected, the detection rate can be affected substantially. With the help of a reliability measure, we want to indicate when the horizon can be used for the feature filters and two limits are calculated where the horizon has to be in between. The reliability is based on the validity of the hourglass assumption. The horizon can be detected, if the horizontal projection $P_h(y)$ (see Eq. 3.2) consists of the combination maximum-minimum-maximum. This is a property of a polynomial with an even degree. For the horizon reliability calculation, the maximum degree is limited to degree four. Observation of the data showed that the variance of the data is too high for a polynomial of degree 2. The projection can then be modelled by:

$$f(y) = a_4y^4 + a_3y^3 + a_2y^2 + a_1y + a_0,$$

(5.23)

where $a_0 \ldots a_4$ are the polynomial coefficients of function $f(y)$. The coefficients must be determined in such a way that $f(y)$ gives the best fit for the projection $P_h(y)$ or in formula form, we are searching for:

$$f(y_i) = P_h(y_i) \quad \text{for } i = 1 \ldots H,$$

where $i$ is the sample number in the horizontal projection and $H$ is the height of the image. This is a standard polynomial curve fitting problem, which can be solved with least square optimisation [32].

With the least square optimisation, the optimal parameters are calculated for the best curve fit of degree four of the projection $P_h(y)$. The curve fit will be used to determine a reliability measure for the horizon estimation. The projection $P_h(y)$ is modelled with the function $f(y)$ of degree four (see Eq. 5.23). The properties of the function are defined by its derivatives, in particular the first and second derivative:

$$f'(y) = 4a_4y^3 + 3a_3y^2 + 2a_2y + a_1,$$

(5.24)

$$f''(y) = 12a_4y^2 + 6a_3y + 2a_2,$$

(5.25)

The roots of $f'$ can be found by setting Eq. 5.24 equal to zero and solving this explicitly (ABCD-formula (fastest) or by calculation of the eigenvalues of the Companion matrix) or by using a numerical approach like the Newton-Raphson iteration [33]. This gives at least 1 and at most 3 maxima/minima for $f(y)$. The points of inflection (poi) and discriminant can be found by solving $f''(y) = 0$ and can be calculated directly from the found coefficients:

$$poi_{1,2} = \frac{-a_3}{4a_4} \pm \frac{1}{12a_4} \sqrt{D}$$

(5.26)

$$D = 9a_3^2 - 24a_2a_4,$$

(5.27)

where $poi_{1,2}$ are the points of inflection and $D$ is the discriminant. Because of the low degree of $f(y)$, there is a limited number of graph shapes possible (see Table 5.1). Number 4 and 5 are no valid graphs of a polynomial of degree 4. To be precise, these are graphs with 2 maxima/minima, but 1 maximum/minimum is outside the domain of $P_h(y)$ ([0 H]).
Table 5.1: Possible graph shapes of a function of degree four with corresponding properties and reliability of the horizon detection algorithm.

<table>
<thead>
<tr>
<th>Number</th>
<th>Graph shape</th>
<th># max</th>
<th># min</th>
<th># pois</th>
<th>reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1" alt="Graph 1" /></td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td><img src="image2" alt="Graph 2" /></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td><img src="image3" alt="Graph 3" /></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td><img src="image4" alt="Graph 4" /></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td><img src="image5" alt="Graph 5" /></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td><img src="image6" alt="Graph 6" /></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td><img src="image7" alt="Graph 7" /></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td><img src="image8" alt="Graph 8" /></td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td><img src="image9" alt="Graph 9" /></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td><img src="image10" alt="Graph 10" /></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td><img src="image11" alt="Graph 11" /></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td><img src="image12" alt="Graph 12" /></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td><img src="image13" alt="Graph 13" /></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td><img src="image14" alt="Graph 14" /></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
For the reliability $R$, four classes $R = 1 \ldots 4$ were made, where $R = 1$ is the lowest reliability and $R = 4$ the highest. A block diagram of the selection process is given in Fig. 5.8.

The curve fit $f(y)$ will be a parabola (number 9 and 10 in Table 5.1) or an asymptote (number 11-14), if the discriminant $D$ is smaller than two. In these cases, a reliability of 1 is assigned. If $D$ is larger than or equal to two, four cases can be distinguished:

1. $f(y)$ has two maxima and one minimum (number 1)
2. $f(y)$ has two minima and one maximum (number 8)
3. $f(y)$ has one maximum and one minimum (number 4 and 5)
4. $f(y)$ has one minimum or maximum (number 3, 6 and 7)
5.2. RELIABILITY OF THE HORIZON DETECTION

Table 5.2: The limits where the horizon has to lie in between for different reliabilities.

<table>
<thead>
<tr>
<th>Reliability</th>
<th>Limit</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( V_{\text{start}} )</td>
<td>( V_{\text{high}} )</td>
</tr>
<tr>
<td></td>
<td>( V_{\text{end}} )</td>
<td>( V_{\text{low}} )</td>
</tr>
<tr>
<td>2</td>
<td>( V_{\text{start}} )</td>
<td>( V_{\text{high}} )</td>
</tr>
<tr>
<td></td>
<td>( V_{\text{end}} )</td>
<td>( V_{\text{low}} )</td>
</tr>
<tr>
<td>2</td>
<td>( V_{\text{start}} )</td>
<td>maximum</td>
</tr>
<tr>
<td></td>
<td>( V_{\text{end}} )</td>
<td>minimum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ( \bar{\gamma}<em>\text{max} = 1 ) and ( \bar{\gamma}</em>\text{min} = 1 )</td>
</tr>
<tr>
<td>3</td>
<td>( V_{\text{start}} )</td>
<td>( \text{minimum} - 0.5 \times</td>
</tr>
<tr>
<td></td>
<td>( V_{\text{end}} )</td>
<td>( \text{minimum} + 0.5 \times</td>
</tr>
<tr>
<td>3</td>
<td>( V_{\text{start}} )</td>
<td>\text{minimum}</td>
</tr>
<tr>
<td></td>
<td>( V_{\text{end}} )</td>
<td>point of inflection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If ( \bar{\gamma}<em>\text{max} = 1 ) and ( \bar{\gamma}</em>\text{min} = 1 )</td>
</tr>
<tr>
<td>4</td>
<td>( V_{\text{start}} )</td>
<td>( \frac{1}{6} \times \text{maximum}_1 + \frac{5}{6} \times \text{minimum} )</td>
</tr>
<tr>
<td></td>
<td>( V_{\text{end}} )</td>
<td>( \frac{2}{3} \times \text{maximum}_2 + \frac{1}{3} \times \text{minimum} )</td>
</tr>
</tbody>
</table>

Each case will be described separately:

- In case 1 (two maxima and one minimum), the values of the derivatives of the points of inflection are calculated. The minimum and maximum have almost the same y-coordinate if the derivative is smaller than one. In this case, the minimum and one maximum are replaced by the point of inflection and treated as case three. If the derivatives of the points of inflection are higher than one, the function is an ideal fit of an hourglass and reliability 4 is assigned.

- For case 2 (two minima and one maximum), the same procedure is followed. If the derivative is smaller than one, the maximum and one minimum are replaced by the poi and the fit will be treated as case three. If the derivative is higher than one, the projection does not represent the initial assumptions and a reliability of 1 is assigned.

- In the third case, one minimum and one maximum, a reliability of 3 is assigned if the maximum is at the right side (maximum is caused by the road pixels, see Fig. 5.9a) of the minimum else the reliability is set to 2. It is assumed that a part of the road will always be visible. Sky on the other hand can be covered by trees, rain etc.

- The fourth case, only one maximum or minimum and a point of inflection, is almost the same as the third. A reliability of 3 is assigned if the maximum is at the right (or at the left in case of a minimum) of the point of inflection, else \( R \) is set to 2. The road is again assumed to be at least partly visible, while the sky can be covered.

Based on the reliability, an upper limit \( V_{\text{start}} \) and a lower limit \( V_{\text{end}} \) can be defined for the horizon (see Fig. 5.9a). For each reliability, two limits were defined (see Table 5.2). For the
CHAPTER 5. CANDIDATE CLASSIFICATION

reliabilities $R = 2$ and $R = 3$, separate limits were chosen for curve shapes of the type of case 3 (one maximum and one minimum). Two extremes, $V_{\text{high}}$ and $V_{\text{low}}$, are defined as 50 and $H - 50$ respectively, where $H$ is defined as the number of lines in the image (height). The selected values for $V_{\text{high}}$ and $V_{\text{low}}$ are somewhat arbitrary. The horizon is assumed to be always between these extremes. Two limits are chosen rather remarkably. For $R = 4$, the limits are more biased toward the road than toward the sky. The horizon was defined as the end of the road. However, the minimum of the curve fit will be situated between the end of the road and the beginning of the sky. A biasing toward the road will increase the accuracy of the limits. For $R = 3$, the limits are chosen wider by including more road and sky. Because of the lower reliability, the limits have to be chosen more carefully.

In Fig. 5.9, an example is given for a frame from the "lower" camera. The horizontal projection $P_{h}$ is plot in blue, the corresponding curve fit in red. The curve fit gives a reliability of 4. The corresponding limits $V_{\text{start}}$ and $V_{\text{end}}$ are indicated with red vertical lines. The actual horizon is selected manually and is indicated with a blue bar. The first horizon estimation $V_1$ (see Eq. 3.4) from the horizon detection algorithm (see Subsection 3.2.1) is plot with a green bar. This part of the detection algorithm mainly relies on the hourglass assumption. As can be seen, the detection got stuck in a local maximum of $P_{h}$. The reliability measure tells us that the projection has the right shape and that the horizon should lie between the red lines. If the horizon search in the detection algorithm (see Subsection 3.2.1) was started from $V_{\text{start}}$, the horizon was found on almost the right place (123 instead of 135).

The horizon reliability measure is evaluated by visual inspection. Evaluation of the chosen horizon limits $V_{\text{start}}$ and $V_{\text{end}}$ can be automated if ground-truth data about the y-coordinate of the actual horizon is available in the meta data. However, the reliability $R$ of the validity of the hourglass assumption can only be checked visually.
5.3 Object splitting

Some candidate selection methods are sensitive for merging different objects. Merged objects can be caused for example by warm objects behind a pedestrian or by pedestrians walking side by side. All candidate selection methods in this report are vulnerable for this problem. Merged objects are very likely to be rejected by the aspect ratio feature filter (see Subsection 3.3.1). In this section, a simple object splitting algorithm will be presented. It is assumed that a pedestrian is fully visible in an ROI. Only the ROIs are considered that do not pass the aspect ratio test. Furthermore, only horizontally connected objects are considered, which is also the most occurring situation according to observed night vision data. The object splitting algorithm of this section requires luminance thresholded images as input. Only the candidate selection method of the ViPs system, the LM optimised thresholding and the gradient based local thresholding method have luminance images as output. The gradient matching candidate selection algorithm (see Subsection 4.2.3) uses only edge images and will therefore not be taken into account in the object splitting algorithm. This is no problem, since this algorithm by itself hardly suffers from connected objects. Candidates are selected on many large horizontal gradient pixels in a column. If a pedestrian is connected with a few pixels to another object, it will still find the pedestrian. The candidate selection will only fail, if a complete side of a body is connected to another object, which rarely occurs.

A basic assumption in the algorithm is that pedestrians are relatively large vertically-oriented objects in an ROI. The algorithm starts with one of the described candidate selection methods and requires a binary image as output. This is the case for the ViPs hot spot detection and the Levenberg-Marquardt (LM) optimised hot spot detection. The gradient based local thresholding algorithm produces a gray-scale image, which is converted to a binary image using a threshold of zero (non-important parts of the image were already set to zero by the candidate selection algorithm, whereas other parts maintained their original luminance (see Subsection 4.2.2)). The ROIs of these algorithms are used as input for the aspect ratio filter as defined in Eq. 3.8 with aspect ratio boundaries of 1.25 and 7 (the upper bound was set to 7 in order to be able to compare the results with the gradient matching algorithm, see Section 5.5). The ROIs that did not pass this test were used as input for the object splitting algorithm. For each ROI in the binary image \( Y_b(x, y) \), a vertical projection is made:

\[
P_{v,ROI}(l) = \sum_{m=1}^{H_{ROI}} Y_b(l, m),
\]

(5.28)

where \( l \) and \( m \) are the \( x \) and \( y \)-coordinates in an ROI and \( H_{ROI} \) is the height of the ROI in pixels. The basic assumption of the object splitting algorithm was that a pedestrian is a large object in an ROI. A pedestrian will therefore cause a large peak in the vertical projection curve. Other warm objects in the ROI will cause smaller peaks. The mean, \( \mu(P_{v,ROI}) \), is subtracted from the projection in order to isolate the large peaks in the curve. The resulting negative values are clipped to zero:

\[
P_{v,ROI,m} = \begin{cases} P_{v,ROI} - \mu(P_{v,ROI}) & \text{if } P_{v,ROI} - \mu(P_{v,ROI}) \geq 0 \\ 0 & \text{otherwise} \end{cases}
\]

(5.29)

For each peak in \( P_{v,ROI,m} \), the start and end point are determined and these intervals are marked as possible new ROIs. Within each interval in the ROI, the highest and lowest
Figure 5.10: An example of object splitting for two pedestrians walking side by side. The blue boxes are ROIs, which passed the aspect ratio test successfully. Red boxes were rejected, green boxes were re-added by the object splitting algorithm.

Figure 5.11: Detection rates of the rain sequence of the first batch of BMW videos for different values of SSA and SSE. Dashed lines indicate the detection rate after aspect ratio filter, whereas solid lines mark the detection rate after the aspect ratio filter followed by the object splitting algorithm. Blue lines: Luminance thresholding with ViPs parameters. Black lines: Luminance thresholding with Levenberg-Marquardt optimised parameters. Green lines: Gradient based local thresholding. Red line: Gradient matching algorithm.

Figure 5.12: Detection rates of the dry weather sequences of the first batch of BMW videos for different values of SSA and SSE. For line style and colour explanation, see Fig. 5.11.

Figure 5.13: Detection rates of the dry weather sequences of the second batch of BMW videos for different values of SSA and SSE. For line style and colour explanation, see Fig. 5.11.

y-coordinate is searched where \( Y_b(l, m) \) has a binary 1. With the detected intervals and y-coordinates, new candidate ROIs, \( ROI_{cand} \), are created. The validity of the candidates is tested with the aspect ratio \( (R_{asp}) \) and filling ratio \( (R_{fill} \), see Eq. 3.9) feature filter. For the aspect ratio, the same parameter values are chosen as in Subsection 3.3.1. For the filling ratio, we require the ratio to lie between 0.55 and 0.9. This avoids creation of unnecessary new ROIs. Each candidate ROI, which passes the two feature filters, is added to the ROI list for the frame and passed to the other feature filters.

Fig. 5.10 shows an example of the result of object splitting. The blue boxes are ROIs which passed the aspect ratio filter, the red boxes were rejected. All rejected boxes were taken into consideration again by the object splitting algorithm and as a result, the green
boxes were re-added to the ROI list. The two pedestrians, walking side by side did not pass the aspect ratio test, but by splitting them into two separate objects, they can be detected.

The re-adding of ROIs influences the detection rates and the false detection rates of the candidate selection methods (see Section 4.3). The false detection rate will definitely increase, since most large bounding boxes can be divided into smaller ones, while still satisfying the aspect ratio and filling ratio rule. For the calculation of the detection rate ($DR$), the same procedure is followed as in Section 4.3. $DR$ is calculated for different SSA values (SSE is always chosen the same as SSA). The rain sequence and dry sequences of the first and second batch of BMW videos are dealt with separately. For each candidate selection method, the ROI list is passed through the aspect ratio filter. After this filter, $DR$ is calculated (dashed lines in Fig. 5.11, 5.12 and 5.13). The rejected ROIs were passed to the object splitting and after this filter, $DR$ was calculated again (solid lines). As can be seen, the detection rates increase in most cases enormously, producing sometimes almost twice as many positive detections. The ViPs and LM optimised algorithm profit, as expected, equally from this method, because they are based on the same thresholding principle. The gradient based local thresholding benefits least, because it is based on edges that prevent pedestrians to become connected to other objects in the ROI. For comparison, the results of the gradient matching candidate selection method are also shown with red lines (after application of the aspect ratio filter). For the first batch of dry sequences, the luminance threshold based candidate selection methods now perform almost equal. For the other sequences, the gradient matching algorithm is still better.

For the three used candidate selection methods, the average number of ROIs per frame is calculated again. In Section 4.3, the average number of generated ROIs by the ViPs, LM optimised, gradient local threshold and gradient matching algorithm was calculated to be 108, 147, 26 and 117 ROIs per frame, respectively. If we add the extra ROIs of the object splitting method, these values change to 117, 154, 29 and 117, respectively. The value of the gradient matching algorithm remains constant, because the object splitting is not applied to this algorithm. The other algorithms show an almost equally limited increase (5-12%) in the number of ROIs.

The object splitting algorithm is able to increase the detection rate of all tested candidate selection algorithms. In the second batch of night vision videos, the algorithm is even able to realise an increase of more than 40% (at SSA = 0.25). The number of ROIs added to the candidate pedestrian ROIs is limited. The computational complexity of the algorithm is limited, for each ROI rejected by the aspect ratio filter a vertical projection and a peak hunting algorithm.

5.4 Symmetry filters

A prominent feature of pedestrians is that they show symmetry in the horizontal direction. This feature is also used in the ViPs system, but mainly to remove tires from the ROI list. Here, filters will be introduced to remove various objects that do not contain some symmetry. Four symmetry filters will be presented: two edge based and two luminance based filters. They differ in performance and computational efficiency. The filters based on edges are only tested in combination with the gradient matching candidate selection method, since the edge map is there already available. The two other filters can be used in combination with all methods, but will only be tested here in combination with the gradient matching algorithm, because
it turned out to have the highest detection rate in the candidate selection\(^1\) (see Section 4.3). All filters are designed to be shape independent, so it does not matter if a pedestrian is facing the camera or not or if he is walking or standing still.

Four sequences, one with rain and three with dry weather conditions of the first batch, were used for each filter to derive statistics for parameter settings. For these statistics, the manually selected pedestrian ROIs were used (see Section 4.1).

5.4.1 Edge based symmetry

Edge strength

A vertical edge strength filter (VES filter) is already introduced in the description of the ViPs system in Subsection 3.3.2. For the VES filter it is assumed that a pedestrian always has two vertical edges and that these edges are parallel to each other. The number of scan lines in an ROI with more than one edge therefore has to exceed a certain threshold. In the edge strength filter of this section, we want to be less strict. A pedestrian is assumed to be symmetrical around the vertical middle line of an ROI. Each side of the ROI should now contain more or less the same number of edge pixels\(^2\). The ratio \(R_{ES}\) between the number of edge pixels in the left half and right half of an ROI is now defined as:

\[
R_{ES} = \frac{\#E_L}{\#E_R},
\]

where \(E_L\) are the edge pixels in the left half, \(E_R\) in the right half and \(\#\) denotes the number of elements. \(R_{ES}\) is expected to be around 1. An ROI is classified as a pedestrian if:

\[
ROI_i = \begin{cases} 
\text{pedestrian} & \text{if } \alpha_1 < R_{ES} < \alpha_2 \\
\text{non - pedestrian} & \text{otherwise}
\end{cases}
\]

where \(\alpha_1\) and \(\alpha_2\) are constants. In order to find appropriate values for the constants, the ratios of the manually selected pedestrian ROIs were calculated. The ratios are plot in the histogram of Fig. 5.14. The ratios are mainly situated between 0.5 and 1.5. The "symmetry" axis of the histogram lies at a value below 1. This can be explained by the manually selected pedestrian bounding boxes. For each pedestrian, a rectangle is drawn from the top left to the right bottom corner. For the top left corner, it is difficult to set the exact location of the left pedestrian edge and some background can be included. For the right side, the rectangle can exactly be matched with the right pedestrian edge. The result is a pedestrian, positioned toward the right of the bounding box. The edge strength will therefore contain more values smaller than 1. The same applies to the Edge Distance filter of Subsection 5.4.1 and the Luminance Inertia filter of Subsection 5.4.2. The histogram of Fig. 5.18 is therefore more biased toward the negative values and the symmetry axis of the histogram of Fig. 5.31 is lower than 1.

---

\(^1\)Due to a programming error, only candidates selected by the blue chain of the gradient matching algorithm (see Subsection 4.2.3) were used in this section. The average number of selected candidates per frames of the blue chain is 103. The gradients matching algorithm still has the best detection rate, even without the red chain.

\(^2\)In this case it is not necessary for edges to come in pairs, they may each scan line be on an other ROI half.
5.4. SYMMETRY FILTERS

The quality and effectiveness of the filter can be examined by calculating the detection and false rate before and after application of the filter. Nine sequences were used, one rain sequence and six dry weather sequences of the first batch and two dry weather sequences of the second batch. All sequences were from the "upper" camera. The histogram of $R_{ES}$ for all ROIs from the candidate selection is much wider than the histogram of the pedestrian ROIs, but the maximum lies also at $R_{ES} = 1$. The constants $\alpha_1$ and $\alpha_2$ define the decrease in detection and false rate. In our evaluation, the constants $\alpha_1$ and $\alpha_2$ were chosen 0.2 and 1.8, respectively, thereby accepting some loss in detection rate. The filter is applied directly after the candidate selection of the gradient matching algorithm. Fig. 5.15, 5.16 and 5.17 show the detection and false rates for the rain sequence, for the dry weather sequences of the first batch and for the dry weather sequences of the second batch for different values of SSA and SSE. As expected, a loss in detection rate can be seen, which is highest in the dry sequences of the first batch ($\pm 3-4\%$ or $\pm 1$ pedestrian per 46 frames). The false rate stays almost constant ($\leq 1\%$ lower, which is $\pm 1$ ROI per frame). In the rain sequence and the sequences of the second batch, it is contrary. Almost no detection rate loss and $\pm 5\%$ and $\pm 7\%$ (measured at SSA = 0.15) decrease in false rate, respectively (which corresponds to $\pm 3$ and $\pm 11$ ROIs per frame).
CHAPTER 5. CANDIDATE CLASSIFICATION

From a computational point of view, this is an inexpensive filter. The binary edge image is already available, only a summation has to be performed. The impact of the filter is rather low, which can probably be ascribed to the used candidate selection method. This method already preserves that the number of vertical edges is almost equal in both halves of an ROI. This filter may perform better on a luminance based candidate selection method. A disadvantage is that for these methods then also an appropriate gradient threshold must be chosen.

Edge distance

In the Edge Strength filter, the only assumption was that the number of edge pixels in the left and right half was more or less the same. The shape of the object in the ROI did not really matter. In the Edge Distance filter, we want to take the shape into account. It is again assumed that a pedestrian is symmetrical to the vertical middle line of an ROI. The distance of an edge pixel to the middle line should then be equal for the left and the right half and the difference between the two should be zero. Of course, this is not always the case, but on the average over all scan lines of an ROI, the difference should be around zero.

A problem arises, if a pedestrian is not exactly symmetrically in the ROI, but shifted one or two pixels to one side. For a small bounding box, the distance differences will maybe still be within tolerable values, but for larger bounding boxes, the difference can easily exceed all thresholds. Therefore, the average difference will be calculated. A second problem concerns large pedestrian ROIs. Pedestrians near the camera show much more detail than pedestrians at a large distance from the camera. It is possible that some details will be visible in the left ROI half (for example an arm), which are not visible in the right ROI half. The result of the edge distance filter will then become worse. To avoid this problem, the average distance difference will be normalised to the number of used scan lines. The average distance will become small for large ROIs and large ROIs will almost always pass this filter.

The edge distances for a scan line are now defined as:

\[
\begin{align*}
    d_l &= \frac{1}{N} \sum_{i} |l_{EIi} - l_c | \\ 
    d_r &= \frac{1}{M} \sum_{j} |l_{ERj} - l_c |, \\
\end{align*}
\]

(5.32)

where \(d_l\) and \(d_r\) are the average distances between the edge pixels on a line and the middle line for the left and right half of an ROI, respectively. \(N\) is the number of edge pixels on a line in the left half and \(M\) the number of edge pixels on a line in the right half. \(l_c\) is the \(x\)-position of the vertical middle line of the ROI, \(l_{EIi}\) and \(l_{ERi}\) are the \(x\)-coordinates of the horizontal position of the edge pixels in the left and right half, respectively. The distance difference \(\Delta d\) is calculated as:

\[
\Delta d = d_l - d_r \quad \text{if} \quad (d_l \neq 0) \land (d_r \neq 0)
\]

(5.33)

The distance differences are now known and the average distance can be calculated. If no line satisfies the condition of Eq. 5.33, the average distance is undefined and the ROI will always
5.4. SYMMETRY FILTERS

Figure 5.18: Histogram of the ratio between the distance of edge pixels in the left and right half of a pedestrian ROI to the middle line, normalised to the height. The chosen values of $\beta_1$ and $\beta_2$ are denoted with red lines.

Figure 5.19: Detection and false rate for the rain sequence before (dashed line) and after (solid line) the edge distance filter. The blue line denotes the detection rate, the red line the false rate.

Figure 5.20: See Fig. 5.19, but for dry weather, first batch.

Figure 5.21: See Fig. 5.19, but for dry weather, second batch.

fail for this filter:

$$d_{ROI} = \begin{cases} \frac{1}{H_{ROI}} \sum_{m=1}^{H_{ROI}} \Delta d_m & \text{if } \{\Delta d_m\} \neq \{0\} \\ \text{undefined} & \text{otherwise,} \end{cases}$$

where $H_{ROI}$ is the number of scan lines in an ROI that satisfy the condition of Eq. 5.33 and $m$ is the y-coordinate in the ROI. The last step is to calculate the ratio $R_{ED}$ between $d_{ROI}$ and $H_{ROI}$:

$$R_{ED} = \frac{d_{ROI}}{H_{ROI}}$$

$R_{ED}$ is expected to be around zero. An ROI is classified as pedestrian if:

$$ROI_i = \begin{cases} \text{pedestrian} & \text{if } \beta_1 < R_{ES} < \beta_2 \\ \text{non-pedestrian} & \text{otherwise,} \end{cases}$$

where $\beta_1$ and $\beta_2$ are two constants and since $R_{ED}$ is symmetrical, we can define $\beta_1 = -\beta_2$. To obtain an appropriate value for $\beta_2$, a histogram is made of the ratios of manually selected pedestrian ROIs (see Fig. 5.18). The ratios are mainly situated between $-0.5$ and $+0.5$. The
histogram of ROIs coming from the candidate selection is again much wider than the pedestrian ROI histogram.

The detection and false rate are determined for the same sequences as used in the Edge Strength filter section, again directly after the candidate selection step of the gradient matching algorithm. The constants $\beta_1$ and $\beta_2$ were chosen as $-0.45$ and $+0.45$, respectively (see red lines in Fig. 5.18). Fig. 5.19, 5.20 and 5.21 show the detection and false rates for different values of SSA and SSE, for the rain sequence, the dry weather sequences of the first batch and the second batch sequences, respectively. A loss in detection rate, comparable with the edge strength symmetry filter, can be seen in the rain and dry weather sequence of the first batch. However, the false rate shows an enormous decrease. The rain sequence has a decrease of $\pm 16\%$ or 11 ROIs per frame, the dry sequence of the first batch has a decrease of $\pm 2\%$ or 2 ROIs per frame and the second batch $\pm 23\%$ or 35 ROIs per frame (all measured at SSA = 0.15).

The computational complexity of this filter is limited, because the binary edge image is already available in the candidate selection method. No pixel based processing is used, which speeds up the algorithm. If this filter is applied to the luminance based candidate selection methods, an extra edge detection has to be performed which makes the filter less attractive (from computational point of view). The impact of this filter on the false rate is rather high, while the detection rate remains approximately constant.

### 5.4.2 Luminance based symmetry

**Luminance inertia**

In the previous section, we used horizontal symmetry of edges to determine if an ROI is a possible pedestrian or not. In this section, the absolute luminance value will be used. A pedestrian ROI typically has the high luminance pedestrian pixels in the centre of the bounding box and some background in the corners (and sometimes between the legs). As already mentioned by Fang et al. [12], each pedestrian has a certain inertia. The inertia is defined as the multiplication of the luminance value of a pixel in an ROI with the distance of the pixel to the centre point of the ROI. Since all pedestrians are almost equally shaped, the position of high luminance pixels will also be roughly the same. For candidate classification, Fang et al. compares the inertia of an ROI with the inertia of a template pedestrian (see Subsection 2.2.1). This has a few disadvantages, the candidate ROI must be normalised to the template size, which consumes additional computation time. If the template is a pedestrian during summer, the inertia will be quite different from a pedestrian during winter wearing insulating clothes (high luminance pixels are not necessarily in the centre of the ROI anymore). Although, it is called a shape independent method, not all pedestrians will have the same pose as the template. Most pedestrians are standing (or walking) straight, but a person running or just starting to walk has a more diagonal shape, which results in a higher inertia (distance between high luminance pixel and centre point increases). A single template pedestrian cannot cover these pedestrian shapes. Therefore, a modification is proposed to the luminance inertia filter. Instead of using the inertia ratio between an ROI and a template pedestrian, the ratio between the left and right half of the ROI is used. Pedestrians are roughly ellipse shaped. The inertias of the left and right half must therefore be more or
less equal. The same applies to the top and bottom half of the ROI, since an ellipse has a symmetry in the centre point of the ROI, which makes the method shape independent. It is assumed, that a bounding box fits almost precisely around the pedestrian and that the head and legs can "compensate" each other when the pedestrian wears insulating clothes. In this section, only the inertia ratio between the left and right half of an ROI will be considered, for the top and bottom half, the same algorithm can be used by replacing the used ROI parts.

The inertias of the left and right half of an ROI are defined as:

\[
I_l = \sum_{l=1}^{L} \sum_{m=1}^{M} F(l, m) \cdot d(l, m)
\]
\[
I_r = \sum_{l=L}^{W} \sum_{m=1}^{M} F(l, m) \cdot d(l, m)
\]
\[
d(l, m) = \sqrt{(l - l_c)^2 + (m - m_c)^2},
\]

where \( I_l \) and \( I_r \) are the inertias of the left and the right half of an ROI, respectively. \( F(l, m) \) is the luminance of the pixel at location \((l, m)\), and \( d(l, m) \) is the distance of a pixel to the centre point \((l_c, m_c)\) of the ROI. \( W_{BB} \) and \( H_{BB} \) are the width and height of the bounding box, respectively. The ratio \( R_{LL}^{lr} \) of luminance inertias between the left and right half of the ROI is then defined as:

\[
R_{LL}^{lr} = \frac{I_l}{I_r}
\]

\( R_{LL}^{lr} \) is expected to be around one. An ROI is classified as pedestrian if:

\[
ROI_l = \begin{cases} 
  \text{pedestrian} & \text{if } \gamma_1 < R_{LL}^{lr} < \gamma_2 \\
  \text{non - pedestrian} & \text{otherwise,}
\end{cases}
\]

where \( \gamma_1 \) and \( \gamma_2 \) are constants around 1. For our test set consisting of four sequences, the inertia ratios of the pedestrians were calculated and plotted in the histogram of Fig. 5.22.

The detection and false rate are again calculated with the same nine sequences as in the Edge Strength filter section, directly after the candidate selection. The constants \( \gamma_1 \) and \( \gamma_2 \) were set to 0.25 and 1.75, respectively (see red lines in Fig. 5.22). The detection and false rates for the different values of SSA and SSE are plotted in Fig. 5.23, 5.24 and 5.25. The detection rate shows sometimes a small decrease, while the false rate remains almost unaffected. The filter seems to have no filtering effect at all. In the next section, we will see that the filter is effective, but the effect is too small to be visible in the graphs.

The filter needs pixel based processing in each ROI, which makes this filter computationally expensive and not suitable for processing on large amounts of ROIs. The performance of the filter differs considerably from the results of Fang et al. The histogram of inertia ratios is almost the same, but the comparison with a pedestrian template gives better results for false rate reduction. However, the detection rate decreases also considerably. Our luminance inertia filter can best be described as a uniformness filter. An ROI containing only background (for example sky), will just as a pedestrian pass this filter.
CHAPTER 5. CANDIDATE CLASSIFICATION

Vertical mean

The last symmetry filter based on luminance, uses the statistics between different parts of an ROI. It is assumed again that a pedestrian can be modelled by an ellipse. The centre point of the ROI is then the symmetry point. The left and the right half as well as the top and the bottom half should now have an almost constant ratio between their mean luminances and their standard deviations. Based on this property, four filters can be defined. In this section, we will first describe the filter where the ratio between the mean luminance of the top and bottom part of an ROI is calculated.

The average luminances of the top and bottom half of an ROI are defined in a straightforward way:

\[
\mu_t = \frac{1}{N \cdot W_{BB}} \sum_{l=1}^{m_c} \sum_{m=1}^{m} F(l, m)
\]

\[
\mu_b = \frac{1}{N \cdot W_{BB}} \sum_{l=1}^{m_c} \sum_{m=m_c}^{m} F(l, m),
\]  

(5.42)
5.4. SYMMETRY FILTERS

Figure 5.26: Histogram of the ratio between the average luminance of the top and bottom half of an ROI. The chosen values of $\delta_1$ and $\delta_2$ are denoted with red lines.

Figure 5.27: Detection and false rate for the rain sequence before (dashed line) and after (solid line) the vertical mean luminance filter. The blue line denotes the detection rate, the red line the false rate.

Figure 5.28: See Fig. 5.27, but for dry weather, first batch.

Figure 5.29: See Fig. 5.27, but for dry weather, second batch.

where $\mu_t$ and $\mu_b$ are the luminance averages of the top and bottom half of the ROI, respectively. The ratio of the Vertical Mean (VM) filter is then defined as:

$$R_{VM} = \frac{\mu_t}{\mu_b}$$  \hspace{1cm} (5.43)

An ROI is classified as pedestrian, if:

$$ROI_i = \begin{cases} 
  \text{pedestrian} & \text{if } \delta_1 < R_{VM} < \delta_2 \\
  \text{non - pedestrian} & \text{otherwise},
\end{cases}$$  \hspace{1cm} (5.44)

where $\delta_1$ and $\delta_2$ are constants. To determine these constants, again a histogram is made (see Fig. 5.26). Most ratios in the histogram are higher than 1, which can be explained by the fact that most pixels of the body of a pedestrian lie in the upper part of an ROI. Only when wearing insulating clothes, the ratio will become smaller than 1. The Vertical Mean filter is already implicitly used in the gradient matching candidate selection algorithm (see Subsection 4.2.3). For the gradient matching algorithm, the constants $\delta_1$ and $\delta_2$ were set very wide (0.4 and 1.6, respectively) to let through as many as possible plausible candidates. The filter is repeated here for two reasons. The gradient matching algorithm consists of two
detection chains, the red and the blue chain (see Subsection 4.2.3). The VM filter was only applied to candidates of the blue chain, which were composed of two vertically matched edges and the ratio was calculated only between the intervals of the two vertically matched edges and not over the total ROI. The second reason is that the filter is not used at all in the other three available candidate selection methods.

For candidate classification, the constants $\delta_1$ and $\delta_2$ were set more strictly to 0.6 and 1.6, respectively, accepting some loss in detection rate (see red lines in Fig. 5.26). The detection and false rate were calculated for different values of SSA and SSE for the nine selected test sequences (see Fig. 5.27, 5.28 and 5.29). The detection rates remained almost unaffected. The false rate shows a decrease of ±2% in the rain sequence and dry weather sequences of the first batch (this corresponds to 1 and 2 ROIs per frame, respectively). The second batch of videos show a larger decrease of ±7% or 11 ROIs per frame.

The computational complexity of this filter is moderate. Pixel based processing is needed, but calculating an average is a standard operation available in hardware. The effect of this filter is positive, even after having applied this filter already one time, the filter is able to reduce the false rate with a few percent. Therefore, if applied to other candidate selection algorithms, higher percentages can be expected. In the beginning of this section, we mentioned that also three other luminance filters, based on average luminance and standard deviation, can be defined. These filters were implemented, but the effect was not noticeable in the false rates. The results of these filters are therefore not shown here. A horizontal mean filter can eventually be applied, since the required data is available in the Vertical Mean filter and only the ratio needs to be calculated.

5.4.3 Combined symmetry filters

In the previous sections, four symmetry filters were described with different impact on the detection and false rates. For calculating the effect on the false rate, each filter was applied separately to the ROI list of the candidate selection. However, some filters can be combined to have more effect. We can distinguish two types of filters, separable and non-separable filters. Filters of the separable category can be applied to the ROI list after each other, without losing effectiveness (the order of application of the filters is only relevant for the computation time). This type of filters has a big computational advantage, since subsequent filters can be applied to smaller ROI lists. A requirement for separable filters is that they operate on mutually independent properties. An example is the Edge Distance and Vertical Mean filter, the fact that an object is symmetrical does not imply anything about the luminance distribution in the ROI. The edge distance and vertical mean property can be plot in a two dimensional feature space, see Fig. 5.30. The same ROIs were used as used for the histograms of previous sections. As can be seen, the pedestrians can be filtered well by a rectangular filter, what is typical for separable filters.

Non-separable filters can be applied individually, but the performance will then be lower than when applied together. An example of a non-separable filter is the Luminance Inertia (LI) filter. As mentioned, a pedestrian can be modelled as an ellipse, which gives the left and right half of an ROI the same inertia and a ratio of ±1. The same applies to the top and the bottom half. Each quarter of an ROI is present in both ratios, top-bottom and left-right. The two ratios are mutually dependent and in this case a circular form is expected in the 2D
5.4. SYMMETRY FILTERS

Figure 5.30: Scatter plot of ED and VM property. The blue circles are ROIs coming from the candidate selection, the red circles are from manually selected pedestrians.

Figure 5.31: Scatter plot of the luminance inertia property for top-bottom and left-right ratio. The black lines denote the chosen thresholds for the separate filters, the green ellipse shows a possible criterion if a 2D filter was used.

Figure 5.32: Scatter plot of the vertical and horizontal mean property. The black lines denote the chosen thresholds for the separate filters, the green ellipse shows a possible boundary as could be chosen in the 2D filter. The false rate can be further reduced than is possible with the two separate filters. A second example of a non-separable filter is the Vertical Mean filter in combination with a Horizontal Mean filter. These are mutually dependent for the same reason as the Luminance Inertia filters. Again a circular filter can be expected in the 2D feature space. Fig. 5.32 shows the result of this filter. The green ellipse denotes again a possible filter, preserving the detection rate, while reducing the false rate. A last example is given in Fig. 5.33, where the Edge Distance of ROIs is plot versus the ROI height. The Edge Distance was divided by the number of used scan lines \( H_{ROI} \) in the calculation of \( R_{ED} \) (see Eq. 5.35). The larger an ROI is, the smaller the ratio \( R_{ED} \) will become. Therefore, the boundaries can be made adaptive to the ROI height as shown with green lines in the figure.
The 2D feature space can be extended to an $n$-dimensional feature space, with $n$ mutually dependent filters. In this way, thresholds can be better shaped to the properties of the pedestrians. A disadvantage is the rapidly increasing complexity and the computation time.

### 5.5 Modified aspect ratio filter

In most pedestrian detection systems available in literature, the aspect ratio ($R_{asp}$) of a pedestrian is a very important feature. An aspect ratio filter is computationally cheap and still effective, since pedestrians have a limited range in aspect ratio. This filter is also part of the cascade of feature filters of the ViPs system (see Section 3.1). A definition and description of the $R_{asp}$ filter is already given in Subsection 3.3.1. For the boundaries of the $R_{asp}$ filter, values can be found in the literature between 2 and 5. In the ViPs systems, the minimum and maximum ratio were set to 1.25 and 5. The minimum ratio was lowered in order to detect pedestrians at a large distance. Probably due to the camera gain or limited camera resolution, pedestrians at a large distance from the camera are more square shaped. Lowering the minimum $R_{asp}$ condition improves the detection rate and distance, but at the cost of an increased false rate. Here, we modify the $R_{asp}$ filter by making it dependent to the height of the ROI.

For four sequences (1 rain and 3 dry weather) of the first BMW batch, the aspect ratios of the ground-truth data were calculated and plot versus the height in Fig. 5.34. A funnel-shaped form can be distinguished, starting from the smallest ROIs. The lower boundary can best be described by a logarithmic shaped function. To keep the filter computationally inexpensive, the lower boundary will be approximated by a piecewise linear line. The chosen boundaries are depicted with the green line in the scatter plot. For small ROIs the $R_{asp}$ is lowered to 1 in order to increase the detection rate. To compensate the increased false rate, the $R_{asp}$ will be linearly increased to 2 for larger ROIs. The upper boundary is raised to 7, because of the behaviour of the gradient matching candidate selection algorithm. Mostly the side of a body is detected without the arms, which increases the $R_{asp}$. For other candidate selection algorithms, the upper boundary can be lowered to the in the literature more used ratio 5. The black lines in the scatter plot denote the ViPs boundaries, the upper boundary is here also increased to 7 for benchmarks.

To test the effect of the $R_{asp}$ and modified $R_{asp}$ filter, the detection and false rates of nine test sequences were calculated for different values of SSA and SSE. The test sequences were the same as in the symmetry filter tests, 1 rainy weather sequence, six dry weather sequences of the first batch and two dry weather sequences of the second batch. The gradient matching algorithm is used for candidate selection. The results are shown in Fig. 5.35, 5.36 and 5.37 for the rain sequence, the dry weather sequence of the first batch and the sequences of the second batch, respectively. As can be seen, the modified $R_{asp}$ filter preserves for all sequences the detection rate better than the $R_{asp}$ filter and the false rate is further decreased. The modified $R_{asp}$ filter gives a decrease in false rate of $\pm 18\%$, $\pm 3\%$ and $\pm 52\%$, respectively, which corresponds to a decrease of 21, 4 and 61 ROIs per frame (measured at SSA = 0.15). The false rate can be decreased further if a higher loss in detection rate is accepted by increasing the lower boundary of the modified $R_{asp}$ to 1.25.
5.5. MODIFIED ASPECT RATIO FILTER

Figure 5.34: Scatter plot of aspect ratio versus ROI height. A small random value is added to each data point, to make the distribution more clear. The black lines denote the boundaries of the $R_{asp}$ filter, the green lines of the modified $R_{asp}$ filter.

Figure 5.35: Detection and false rate for different values of SSA and SSE for the rain sequence of the first batch. Red line: false rate. Blue line: detection rate. Dashed line: Candidate selection result. Dashed-dotted line: $R_{asp}$ filter result. Solid line: modified $R_{asp}$ filter result.

Figure 5.36: See Fig. 5.35, but for dry weather, first batch.

Figure 5.37: See Fig. 5.35, but for dry weather, second batch.
Chapter 6

Performance of pedestrian detection systems

In the previous chapters, four candidate selection methods and fourteen candidate classification filters were described. Advantages and disadvantages were mentioned and the effect on the detection and false rate has been calculated. The remaining question is, can we improve the current available pedestrian detection system of the ViPs group by increasing the detection rate or decreasing the false rate? In this chapter, a benchmark will be made of three different detection systems, the available ViPs detection system, the ViPs system extended with the object splitting algorithm and a gradient based system with the symmetry filters introduced in the previous chapter with two different sets of parameters. In the first section, an overview will be given of the detection systems. In the second section, the detection results will be presented.

6.1 Overview detection systems

Three pedestrian detection system will be used for the performance benchmark:

1. The first detection system is the non-motion based part of the system of the ViPs group, hereafter called the ViPs system. An overview of this system is already given in Chapter 3 (see also Fig. 3.2). For the performance evaluation, an exact implementation was made of the described detection system, including the mentioned parameter values. An exception is made for the horizon detection. The y-coordinate of the horizon is chosen to be a fixed parameter, which will be determined for each sequence individually.

2. The second detection system is again the ViPs detection system, but extended with the object splitting algorithm, hereafter denoted as the modified ViPs system. The object splitting algorithm, described in Section 5.3, was inserted in the ViPs system between the aspect ratio and filling ratio filter (see Fig. 3.2). The rest of the detection system is exactly the same as the ViPs implementation, including the parameter values and the
6.2. PERFORMANCE EVALUATION

fixed horizon.

3. The last detection system used for the performance evaluation is a system based on gradient thresholding. An overview of this system has not been described yet and will be presented in the remainder of this section. Two sets of used parameters will be given, one set with very strict settings for the filter parameters in order to reduce the false rate and a second set with less strict parameter settings in order to preserve the detection rate as much as possible. The gradient based detection system with false rate optimised parameter settings will hereafter be denoted as the GBFRopt system. The detection system with detection rate optimised parameters will be denoted as the GBDRopt system.

The gradient based detection system is based on the same principle as the ViPs system: a candidate selection method followed by a cascade of feature filters. In Fig. 6.1, a block diagram is shown with a schematic outline of the system. The detection algorithms (green blocks) are adopted from the ViPs system (see Section 3.2), just as the feature filters shown as red blocks (see Section 3.3). As candidate selection algorithm (orange block), the gradient matching algorithm is used, which is described in Subsection 4.2.3 (also the same parameter settings will be used). The purple blocks denote feature filters that are not used in the ViPs system or which use different parameter settings as the ViPs system.

Most feature filters shown in purple blocks are 2 dimensionally combined feature filters (see Subsection 5.4.3). The Luminance Thresholding block is the first filter where a luminance thresholding is performed. The basic idea of the gradient based system was to avoid luminance thresholds with accurately chosen parameters like \( k_1 \) and \( k_2 \) in the ViPs system (see Section 4.1). However, the false rate remains unacceptably high if only edge based filters are used. The chosen luminance threshold is very low, to prevent eliminating too many positive pedestrian detections, but high enough to remove ROIs that mostly contain background.

The gradient matching candidate selection algorithm generates many ROIs, which are of almost the same size and position, but shifted with a few pixels with respect to each other. In the purple Merge ROI block, two or more overlapping ROIs will be combined to one new ROI, defined as the union of the separate ROIs.

In Appendix B, the parameter settings for the purple blocks will be presented for the false rate optimised and detection rate optimised gradient based detection system, respectively. The Relative Size filter (red block) was not used in the GBDRopt system, as it decreases the detection rate too much.

6.2 Performance evaluation

In the candidate selection and candidate classification sections, the detection and false rate of different methods and filters were calculated for the individual algorithms. In this section, the performance of the complete detection systems will be calculated. The detection rate \( DR \) will again be used as performance measure (see Section 2.3), the false rate will be replaced by the average number of false detections per frame \( F_{fr} \). The false rate cannot be used any more for comparison, because the number of candidate ROIs depends on the used candidate
CHAPTER 6. PERFORMANCE OF PEDESTRIAN DETECTION SYSTEMS

Figure 6.1: Block diagram of detection algorithms and feature filters of the gradient based detection system.
6.2. PERFORMANCE EVALUATION

The $F_{fr}$ is defined as:

$$F_{fr} = \frac{\#ROI s \cdot FR_{[SSA,SSE]}}{\#frames}, \quad (6.1)$$

where $FR_{[SSA,SSE]}$ is the false rate at a predefined overlap value of SSA and SSE (see Section 2.3). The values of SSA and SSE will be variable in our tests. For the performance evaluation holds: the lower the number of false positives per frame at a constant detection rate, the better the detection system.

All nine sequences with available ground-truth data were used for the performance tests. The sequences are divided in three groups. The first group contains the rain sequence of the first batch, the second group the six dry weather sequences of the first batch and the third group contains two dry weather sequences of the second batch. All used sequences are from the "upper" camera, which was situated on the roof of the car. The parameter settings of the ViPs system were optimised for the first batch of sequences (both rainy weather and dry weather). The parameters for the gradient based system GBDRopt and GBFRopt were optimised for four sequences of the first batch, one rain sequence and three dry weather sequences. The $DR$ and $F_{fr}$ results will be determined per group, by calculating the average results over all sequences within the group.

The detection rate and number of false detections per frame were calculated for SSA and SSE values between 0.15 and 0.8 (39%-89% overlap). The $DR$ results are shown in Fig. 6.2, 6.4 and 6.6. The $Fr$ results are plot in Fig. 6.3, 6.5 and 6.7. The following conclusions can be drawn:

- The $F_{fr}$ of the ViPs system and the GBFRopt system is almost equal for all sequences. The detection rate is almost equal for the dry sequences of the first batch. The GBFRopt system performs much better than the ViPs system in the rain sequence and the dry sequences of the second batch, ±20% and ±45%, respectively, better.

- The modified ViPs system is approaching the detection result of the GBFRopt system in the rain sequence and the dry weather sequences of the seconds batch and even outperforms the GBFRopt system in the dry weather sequences of the first batch. However, this is at the cost of a higher $F_{fr}$.

- The GBDRopt system is overall the best detection system, with a $DR \geq 70\%$ at SSA = 0.25 for all sequences. However, the high $DR$ is here also at the cost of a much higher $F_{fr}$.

We can conclude that the GBDRopt system is overall the best detection system. Additional filters must be designed in order to reduce the $F_{fr}$. If the $F_{fr}$ is a critical issue, the GBFRopt system can best be used. If a luminance thresholding based algorithm is preferred, then the modified ViPs system can be used. For practical usage, $F_{fr}$ is still too high for all detection systems. Even the best $F_{fr}$ (dry sequences of the second batch), generates ±18 false positives per minute, what is unacceptably high in an actual driving situation.
Figure 6.2: Detection rate for the different pedestrian detection systems for a range of SSA and SSE values for the rain sequence of the first batch. Blue line: ViPs system. Black line: modified ViPs system. Red line: GBFRopt system. Green line: GBDRopt system.

Figure 6.3: Number of false detections per frame for the different pedestrian detection systems for a range of SSA and SSE values for the rain sequence of the first batch. Blue line: ViPs system. Black line: modified ViPs system. Red line: GBFRopt system. Green line: GBDRopt system.

Figure 6.4: See Fig. 6.2, but for dry weather, first batch.

Figure 6.5: See Fig. 6.3, but for dry weather, first batch.

Figure 6.6: See Fig. 6.2, but for dry weather, second batch.

Figure 6.7: See Fig. 6.3, but for dry weather, second batch.
Chapter 7

Conclusion and discussion

7.1 Conclusion

In this study, pedestrian detection methods from the literature for far-infrared night vision data were examined and new detection algorithms were proposed. Generally, the non-training based detection systems can be divided in two sub-systems, candidate selection and candidate classification. The challenge of our research was to improve the detection rate after candidate selection and to decrease the number of false-positives per frame left after candidate classification of the available detection system of the ViPs group. A limited number of nine test sequences was available for algorithm design and performance evaluation.

Candidate selection

Warm objects, like pedestrians, appear as white areas in far-infrared night vision images. Almost all candidate selection algorithms available in literature use luminance thresholding to separate these objects from the background. The ViPs system also uses a luminance threshold, which is based on $k_1$ times the mean luminance of a frame and $k_2$ times the standard deviation of the frame. The values for $k_1$ and $k_2$ were very accurately chosen by inspection. To find more optimal values, a cost function was defined, which depends on the mean luminance and the standard deviation of a frame and the average pedestrian ROI luminance. With the Levenberg-Marquardt least square optimisation algorithm and three sequences as training set, optimal parameters were calculated. In the performance evaluation, the luminance threshold with optimised parameters turned out to perform better than the ViPs parameters on sequences which were used in the training ($\pm 20\%$ better) and worse in other sequences. For a better optimisation, a much larger test set is required. The average number of selected candidates was 147 candidates per frame, which is almost 50% higher than the number of candidates selected by the ViPs system (108 candidates per frame).

Pedestrians wearing insulating clothes appear typically only with their head and legs as white area in the night vision images. Luminance thresholding will divide the pedestrian in two or more parts. Analysis of the gradients showed that 87% of the pedestrian edge
gradients belonged to the 3% highest frame gradients. Therefore, two edge based candidate selection algorithms were proposed. The first algorithm, the gradient based local thresholding algorithm, performed a local luminance thresholding based on the luminance of pixels with a high gradient, in order to remove background pixels. In this algorithm, both high and low luminance objects can pass the threshold. It was assumed that after the local thresholding, the pedestrians would be a significantly large part of the thresholded image. A frame based threshold was applied to further reduce the number of non-pedestrian pixels. In the performance evaluation, this algorithm turned out to perform better in the sequence with rainy weather, but it performed slightly worse than the ViPs candidate selection method in other sequences. Advantage of this method is the small number of selected candidates (26 candidates per frame), which is only one fifth of the number of candidates of the ViPs method (108 candidates per frame). The second edge based candidate selection method, is the gradient matching algorithm. This algorithm merges vertical line segments of high gradient pixels if the line segments are in approximately the same column and if the gradients have the same sign. The vertically merged line segments of positive and negative gradients are horizontally merged with each other if they are on the same scan lines of the image. In this way ROIs can be selected based on edge segments. This algorithm is able to select pedestrians wearing insulating clothes as candidate. The performance evaluation showed that this algorithm was able to obtain a detection rate which was higher than 93% in all sequences, with 50% overlap between the candidate ROI and the pedestrian. The number of selected candidates was with 117 ROIs per frame slightly higher than the ViPs system.

Candidate classification

In the candidate classification phase, the number of false-positives per frame must be minimised. The first proposed method is to estimate the distance between the car and the pedestrian. When driving 50 km/h, it is less useful to detect pedestrians at a distance of 200 metres than when driving 100 km/h. In order to reduce the false rate, the detection distance can be made speed adaptive and for example be limited to at most 50 metres at lower speeds. Two methods for distance calculation were considered, one based on the pinhole model and one motion based method. Some camera parameters, like the focal length, were unknown and were estimated by using the available video material and by making some assumptions. The pinhole based method was able to estimate the distance with an inaccuracy of approximately 20 metres. The motion based method performed slightly worse for pedestrians at a large distance of the camera. If a pedestrian is within a range of 50 metres, the motion based method is more accurate, with a detection error of ± 7 m. The distance calculation with the motion model can be refined by using motion vectors.

The "above-the-horizon" filter is a very powerful filter in the ViPs system. Unfortunately, the horizon estimation is sometimes inaccurate, causing a decrease in detection rate or an increase in false rate. A horizon reliability measure is proposed to avoid these problems. The reliability algorithm examines if the night vision image satisfies the assumptions made in the horizon detection algorithm. A curve fit of degree four will be made for each frame to validate the presence of an hourglass shape in the road-sky mask. Depending on the shape of the curve fit, a reliability measure between 1 and 4 is assigned to the detected horizon and two limits are calculated, where the horizon has to lie in between. The results were only evaluated visually.
Pedestrian ROIs coming from the selection algorithms, are sometimes connected to other warm objects. This is mainly the case with the luminance based candidate selection methods and the gradient based local thresholding algorithm due to incorrectly (too low) chosen luminance thresholds. Connected pedestrians no longer satisfy properties of a pedestrian, such as aspect ratio. This complicates the detection of these pedestrians. An object splitting algorithm is presented to divide the connected objects into separate objects. The algorithm is based on the property that pedestrians are relatively large warm objects in an ROI. The object splitting algorithm is only applied to ROIs that did not pass the aspect ratio test. The algorithm realises an increase in detection rate of ±15% in the rainy weather sequence, ±10% in the dry weather sequences of the first batch and ±45% in the dry weather sequences of the second batch.

Pedestrians that are walking or standing still have rather symmetrical shapes. We modelled a pedestrian with an ellipsoid and defined two edge based and two luminance based symmetry filters. The most promising filter was the edge distance filter, where the ratio of average distances of the edges of the left and right half of an ROI to the vertical middle line of the ROI were calculated. With a minimal decrease in detection rate, this filter was able to decrease the false rate up to ±20% in the dry sequences of the second batch. The other three symmetry filters decreased the false rate with very small percentages.

Pedestrians at a large distance of the camera have a smaller aspect ratio than pedestrians near the camera, probably due to limited camera resolution or flare. It is proposed to make the tolerated aspect ratios dependent on the height of the ROI. Small ROIs may have smaller aspect ratios than large ROIs. The result of this modification of the aspect ratio filter is a better preserved detection rate (±5%-10%, depending on the type of sequence) and the false rate is further decreased.

Detection system performance

The proposed modifications to filters and (new) detection algorithms were used for a benchmark of the complete detection system. Four systems were implemented with two different candidate selection methods. The luminance thresholding based candidate selection of the ViPs system is used in combination with the original non-motion based part of the ViPs system and in combination with the same original system extended with the object splitting algorithm. The edge based gradient matching candidate selection algorithm is used in combination with the new proposed modified aspect ratio and symmetry filters. Two sets of parameters were used for the performance evaluation, one set optimised for reducing the number of false detections per frame and one set for preserving the detection rate as much as possible. If a detection system is desired with a very high detection rate, the gradient matching algorithm with detection rate optimised filter parameters can best be used. This system gives a detection rate of at least 75% in all sequences (at an overlap percentage of 50%). However, this is at the cost of the number of false detections per frame, which can be up to 2.2 false-positives per frame. Additional filters have to be designed, which can further reduce the number of false detections. If the number of false detections per frame is the key issue in the desired system, the gradient matching algorithm with false rate optimised filter parameters can best be chosen. The number of false detections per frame is together with the original ViPs system the lowest (at most ±0.6 false detections per frame at 50% overlap).
But, the detection rate in the rain sequence is 20% higher and in the dry sequences of the second batch even 45% higher than with the original ViPs system. The ViPs system extended with the object splitting algorithm has a performance which is predominantly between the results of the original ViPs system and the false rate optimised gradient matching system. However, the false rate is higher than in these systems.

7.2 Discussion

In this report, two thresholds were used for the gradient thresholding. All available sequences were recorded at more or less the same outside temperature. Only the weather type, rainy or dry weather, defined which threshold had to be used. It can be assumed that the gradient threshold will be dependent on the standard deviation of a frame. However, more sequences in different weather types are needed to validate this assumption. Also sequences at different outside temperatures must be considered to examine the gradient threshold dependencies with respect to the temperature.

The gradient matching candidate selection method, performed better than the other considered candidate selection algorithms. However, for higher overlap percentages the detection rate decreases. This is mainly caused by miss detected arms and feet and sometimes partly miss detected heads in the selection process. The algorithm only uses horizontal gradients. By also taking into account the vertical gradients, the position of the selected candidates can be made more accurate. This will be at the cost of a higher complexity of the algorithm and a much larger computation time.

The performance of the described candidate classification filters were considered for each filter individually. It is not taken into account that two or more filters can affect the same subset of candidate ROIs. Application of one filter can make other filters useless. These redundancy relations need to be considered in future work in order to reduce computational complexity of the complete detection system. In Section 5.4, the effect of the feature filters on the detection and false rate was evaluated for arbitrary chosen values for the filter limits, while the amount of overlap between the detected pedestrian and the pedestrian ROI from ground-truth data was made variable. For future work, it is interesting to make the amount of overlap constant and the filter limits variable. The filter limits can then be better tuned to desired false or detection rates. In Subsection 5.4.3, we mentioned that some feature filters can be combined to non-separable filters to increase the effectiveness. Examples were given for combinations of two filters in a two dimensional feature space. This can be extended to an \( n \)-dimensional feature space with \( n \) different non-separable filters. This will result in an increased complexity of the system, but the filter boundaries can be calculated before implementation in hardware. The only disadvantage for a final system will be that all used filters have to be applied to the same (eventually large) candidate ROI list.

In all evaluated detection systems, the feature filters were applied sequentially, each filtering out some non-pedestrian ROIs. However, each filter has its limitations according to the parameter settings. At some point, a further reduction of the false rate will also decrease the detection rate. By applying the filters sequentially, a choice needs to be made to have a large decrease in false rate or to preserve the detection rate. However, it is possible that some pedestrian candidates are classified as non-pedestrian by one filter, but classified as pedestrian by all other filters. In the current system, these candidates will then be classified
as non-pedestrian since each filter has a "veto". This can be avoided by using fuzzy logic and to assign a reliability to each candidate to be pedestrian or not. The reliability of a candidate will be increased if more filters classify it as pedestrian. The parameters of the filters individually can then be set more strictly. Disadvantage of this method is the need to use the complete candidate ROI list for each filter. The computational complexity will therefore increase.

The performance of a filter was determined by visual inspection of the effect of the filter on the detection and false rate. Evaluation of the filter and determination of parameter settings can be speed up by defining a performance rating for a filter. The decrease percentage of the false ratio influences the performance positively, the decrease percentage of the detection ratio will affect the performance rating negatively. The computational complexity of a filter should also be considered. The more ROIs a filter can process per second, the better the filter is. A possible performance rating could look like this:

\[
PR = \frac{\Delta FR}{FR} - \frac{\Delta DR}{DR} \log \left( \frac{n_{ROI}}{s} + 1 \right),
\]

where \( PR \) is the performance rating, \( FR \) the false rate, \( DR \) the detection rate and \( s \) the number of seconds required for processing the ROIs. The measure for the number of ROIs per frame that can be processed will however be dependent on the platform that will be used (FPGA, ASIC, PC).

Because of the limited number of available night vision sequences, no training based algorithms were used. However, if more video material becomes available, this may be an option to be considered. In the literature, already a few training based algorithms are proposed, but the results do not seem to be better than non-training based approaches. Also combinations of feature based and training based algorithms can be considered. In the case of connected objects, this may be very helpful. A luminance thresholding followed by a connected component analysis defines the ROIs in a frame. A trained filter can then be applied to these ROIs, making a frame based filtering unnecessary.
References


REFERENCES


Appendix A

Levenberg-Marquardt least square optimisation

The Levenberg-Marquardt (LM) algorithm is a combination of the Vanilla gradient descent and Gauss-Newton iteration and is known as a very fast converging minimal error search algorithm. It is more robust than the Gauss-Newton method and therefore usually preferred. LM optimisation minimises the following function:

\[ f(x) = \frac{1}{2} \sum_{i=1}^{m} r_i^2(x), \]  

where \( f(x) \) is the function to be minimised, \( x = (x_1, x_2, \ldots, x_n) \) is a vector with the parameters to be estimated, \( i \) is the sample number in the data set, \( m \) the number of samples and \( r_i \) are the residuals of the difference between the estimated data and the real data set. The residuals \( r_i \) can be written as a residual vector \( r(x) \):

\[ r(x) = (r_1(x), r_2(x), \ldots, r_m(x)) \]

The first and second gradient of \( f(x) \) to \( x \) are than defined as:

\[
\nabla f(x) = \sum_{i=1}^{m} r_i(x) \nabla r_i(x) = J(x) r(x) \]  

\[
\nabla^2 f(x) \approx J(x)^T J(x) \]  

where \( J(x) \) is the Jacobian matrix. The approximation in Eq. A.3 is valid as long as the residuals \( r_j(x) \) are small, or if the error function is almost linear in \( x \) in the neighbourhood.
of the solution, which makes the $\nabla^2 r_i(x)$ term small.

The Vanilla gradient descent algorithm is the most straightforward and intuitive implementation of a minimum search algorithm:

$$x_{j+1} = x_j - \lambda \nabla f(x), \quad (A.4)$$

where $\lambda$ is a constant which scales the rate of convergence and $j$ is the iteration number in the optimisation process. If the gradient is large, the step should be small and if the gradient is small, the step size should be large to speed up the convergence. However, the Vanilla gradient descent algorithm does exactly the opposite, which can lead to convergence errors. A second drawback of the "Vanilla" method is the low sensitivity for the direction of the gradients. For example if the error function consists of a slow descending, small channel, the "Vanilla" method will give preference to the steep walls of the channel, which leads to slow convergence. The effectiveness of the algorithm can be improved by taking the curvature (second derivative) of the error surface into account. Newton applied this idea by using a Taylor serie for calculating the gradient of $f(x)$:

$$\nabla f(x) = \nabla f(x_0) + (x - x_0)^T \nabla^2 f(x_0) + O, \quad (A.5)$$

where $O$ are the higher order terms. If $f$ is assumed to be quadratic around $x_0$, the higher order terms $O$ can be ignored and the minimum is determined by calculating $\nabla f(x) = 0$. This results in the update step of the Gauss-Newton iteration:

$$x_{j+1} = x_j - (\nabla^2 f(x_j))^{-1} \nabla f(x_j) \quad (A.6)$$

The Hessian (matrix of second partial derivatives, or $\nabla^2 f(x)$) is approximated by Eq. A.3. The Gauss-Newton method provides a quick converging minimisation algorithm which takes the curvature into account, but requires almost linear starting locations $x_j$ (assumed to be at most quadratic).

Levenberg proposed an algorithm that combines the advantages of the Vanilla gradient descent and Gauss-Newton method. The update step of the Levenberg iteration is defined as:

$$x_{j+1} = x_j - (H_j + \lambda I)^{-1} \nabla f(x_j), \quad (A.7)$$

where $H$ is the Hessian matrix. The constant $\lambda$ will be changed during the iterations of the optimisation. If the quadratic assumption of $f(x)$ turns out to be valid, it is biased to the Gauss-Newton algorithm by decreasing $\lambda$ in the next update step. If the quadratic assumption is invalid, $\lambda$ will be increased. However, if $\lambda$ becomes too large, the Hessian will not be used at all and the Levenberg algorithm fades into the Vanilla gradient descent algorithm. Marquardt improved the update step, which resulted in the final Levenberg-Marquardt update step:

$$x_{j+1} = x_j - (H_j + \lambda \text{diag}(H_j))^{-1} \nabla f(x_j), \quad (A.8)$$

where $\text{diag}(H)$ is the diagonal of the Hessian matrix. The result is a large step size in the direction with a low gradient and a small step size in other directions.
Appendix B

Parameter settings

In Section 6.1, an overview was given of the gradient based detection system (see Fig. 6.1). The purple blocks in the block diagram, denote filters where the used parameters are not mentioned in the report before, or where different parameters were used. In this chapter, the parameter settings of all purple blocks will be given for a false rate and a detection rate optimised system. In the first section, the parameter settings for the false rate optimised system will be given, together with references to the related sections and figures. In the second section, only the parameter settings will be mentioned for the detection rate optimised system.

B.1 False rate optimised parameter settings

The modified aspect ratio filter uses the height of the bounding box to define the accepted aspect ratios for pedestrian ROIs (see Section 5.5). In the false rate optimised system the following settings were used:

\[
ROI_i = \begin{cases} 
non - pedestrian & \text{if } (2.5 < H_{ROI} < 15) \land (R_{asp} > \frac{2}{3} H_{ROI}) \\
non - pedestrian & \text{if } (H_{ROI} \leq 16\frac{1}{4}) \land (R_{asp} < 1.25) \\
non - pedestrian & \text{if } (16\frac{1}{4} < H_{ROI} < 50) \land (R_{asp} < \frac{1}{45} H_{ROI} + \frac{2}{3}) \\
non - pedestrian & \text{if } (H_{ROI} \geq 50) \land (R_{asp} < 2) \\
non - pedestrian & \text{if } R_{asp} > 7 \\
pedestrian & \text{otherwise,}
\end{cases}
\] (B.1)

where \(H_{ROI}\) is the height of an ROI and \(R_{asp}\) is the aspect ratio. See Fig. 5.34 for an example of possible settings (other than here specified).

The Edge Distance filter (see Subsection 5.4.1) in the detection system is made adaptive to the ROI height (see Subsection 5.4.3 and Fig. 5.33). The boundaries are defined as:

\[
ROI_i = \begin{cases} 
non - pedestrian & \text{if } (H_{ROI} \leq 25) \land (0.4 < R_{ED} < -0.4) \\
non - pedestrian & \text{if } (25 < H_{ROI} \leq 50) \land \\
& ((-\frac{7}{500} H_{ROI} + 0.85) < R_{ED} < (\frac{7}{500} H_{ROI} - 0.85)) \\
non - pedestrian & \text{if } (H_{ROI} > 50) \land (0.12 < R_{ED} < -0.12) \\
pedestrian & \text{otherwise,}
\end{cases}
\] (B.2)
where $R_{ED}$ is the edge distance ratio. See Fig. 5.33 for an example of possible boundaries.

The Vertical Mean filter is combined with the Horizontal Mean filter (see Subsection 5.4.2). The boundaries for this filter are defined with an ellipse with centre coordinate $(0.8, 1)$ and a horizontal axis of length 0.5 and a vertical axis of length 0.6:

$$ROI_i = \begin{cases} 
\text{non - pedestrian} & \text{if } ((R_{HM}^{0.8})^2 + (R_{VM}^{0.6})^2) > 1 \\
\text{pedestrian} & \text{otherwise,}
\end{cases}$$

(B.3)

where $R_{HM}$ is the ratio between the mean luminance of the left and right half of an ROI. $R_{VM}$ is the ratio between the mean luminance of the top and bottom half of an ROI. An example of the boundaries can be found in Fig. 5.32.

The Vertical Standard Deviation versus Horizontal Standard Deviation filter combines the ratio of standard deviations between the top and bottom half of an ROI ($R_{VS}$) with the ratio of the standard deviations between the left and right half of the ROI ($R_{HS}$). A rectangle shaped boundary is used:

$$ROI_i = \begin{cases} 
\text{non - pedestrian} & \text{if } (1.3 < R_{HS} < 0.5) \lor (4 < R_{VS} < 0.5) \\
\text{pedestrian} & \text{otherwise,}
\end{cases}$$

(B.4)

The boundaries of the Edge Strength filter (see Subsection 5.4.1) are defined as:

$$ROI_i = \begin{cases} 
\text{non - pedestrian} & \text{if } 1.5 < R_{ES} < 0.5 \\
\text{pedestrian} & \text{otherwise,}
\end{cases}$$

(B.5)

where $R_{ES}$ is the ratio of the number of edges in the left and right half of an ROI.

The Luminance Inertia Filter (see Subsection 5.4.2) is the last 2D feature filter in the filter cascade. The ratio $R^{0.75}_{IR}$ of the luminance inertia between the left and right half of an ROI is combined with the ratio $R^{0.95}_{IB}$ between the top and the bottom half of the ROI. The boundaries of the filter are defined by an ellipse with centre coordinate $(0.75, 0.95)$ and a horizontal axis of length 0.55 and a vertical axis of length 0.7:

$$ROI_i = \begin{cases} 
\text{non - pedestrian} & \text{if } ((R^{0.75}_{IR})^2 + (R^{0.95}_{IB})^2) > 1 \\
\text{pedestrian} & \text{otherwise}
\end{cases}$$

(B.6)

In Fig. 5.31, an example is shown of a possible boundary for this filter.

Till now, no luminance thresholding was performed. The idea of the gradient based detection system was to be less dependent on a carefully chosen luminance threshold. However, with only edge based filters it is at this moment impossible to reduce the false rate to acceptable values. Therefore some luminance based filters will be used. The Luminance Thresholding block classifies an ROI as non-pedestrian, if:

$$ROI_i = \begin{cases} 
\text{non - pedestrian} & \text{if } F_{ROI} < T \\
\text{pedestrian} & \text{otherwise}
\end{cases}$$

(B.8)

where $F_{mean}$ is the average frame luminance and $F_{std}$ the standard deviation of a frame. The parameters $k_1$ and $k_2$ were both set to 1. The threshold will by this be chosen lower than the ViPs threshold, to be sure that the threshold will be lower than the average pedestrian luminance. The Luminance Thresholding block classifies an ROI as non-pedestrian, if:

$$ROI_i = \begin{cases} 
\text{non - pedestrian} & \text{if } F_{ROI} < T \\
\text{pedestrian} & \text{otherwise}
\end{cases}$$

(B.8)
where $F_{ROI_{mean}}$ is the average luminance of the ROI under consideration.

The Filling Ratio filter is defined as in Eq. 3.9 in Subsection 3.3.1. The only difference with the Filling Ratio filter of the ViPs system is the used threshold:

$$ROI_i = \begin{cases} 
  \text{non - pedestrian} & \text{if } R_{fill} < T_{fill} \\
  \text{pedestrian} & \text{otherwise,}
\end{cases}$$

where $R_{fill}$ is the filling ratio and $T_{fill}$ is the threshold, which is set to 0.5.

The gradient matching candidate selection algorithm generates a lot of ROIs, which are of almost the same size and position, but shifted with a few pixels with respect to each other. The Merge ROI filter combines these ROIs to one new ROI if the amount of overlap between the ROIs exceeds some thresholds. For the thresholds, the SSA and SSE are used (see Section 2.3). Two or more ROIs are combined, if the SSA and SSE are both higher than $\sqrt{0.7}$ (70% overlap).

## B.2 Detection rate optimised parameter settings

The boundaries of the modified aspect ratio filter are defined as:

$$ROI_i = \begin{cases} 
  \text{non - pedestrian} & \text{if } (2.5 < H_{ROI} < 15) \land (R_{asp} > \frac{7}{5} H_{ROI}) \\
  \text{non - pedestrian} & \text{if } (H_{ROI} \leq 10) \land (R_{asp} < 1) \\
  \text{non - pedestrian} & \text{if } (10 < H_{ROI} < 60) \land (R_{asp} < \frac{7}{50} H_{ROI} + \frac{1}{5}) \\
  \text{non - pedestrian} & \text{if } (H_{ROI} \geq 60) \land (R_{asp} < 2) \\
  \text{non - pedestrian} & \text{if } R_{asp} > 7 \\
  \text{pedestrian} & \text{otherwise,}
\end{cases}$$

The boundaries of the Edge Distance versus ROI height filter are defined as:

$$ROI_i = \begin{cases} 
  \text{non - pedestrian} & \text{if } (H_{ROI} \leq 25) \land (0.5 < R_{ED} < -0.5) \\
  \text{non - pedestrian} & \text{if } (25 < H_{ROI} < 50) \land \\
  & ((-\frac{7}{500} H_{ROI} + 0.85) < R_{ED} < (\frac{7}{500} H_{ROI} - 0.85)) \\
  \text{non - pedestrian} & \text{if } (H_{ROI} > 50) \land (0.15 < R_{ED} < -0.15) \\
  \text{pedestrian} & \text{otherwise,}
\end{cases}$$

The boundaries of the Vertical Mean versus Horizontal Mean filter are defined as:

$$ROI_i = \begin{cases} 
  \text{non - pedestrian} & \text{if } ((R_{HM} - 0.95)^2 + (R_{VM} - 1.05)^2) > 0.65^2 \\
  \text{pedestrian} & \text{otherwise}
\end{cases}$$

The boundaries of the Vertical Standard Deviation versus Horizontal Standard Deviation filter are defined as:

$$ROI_i = \begin{cases} 
  \text{non - pedestrian} & \text{if } (1.5 < R_{HS} < 0.5) \lor (4 < R_{VS} < 0.5) \\
  \text{pedestrian} & \text{otherwise}
\end{cases}$$

The boundaries of the Edge Strength filter are defined as:

$$ROI_i = \begin{cases} 
  \text{non - pedestrian} & \text{if } 1.8 < R_{ES} < 0.2 \\
  \text{pedestrian} & \text{otherwise}
\end{cases}$$
The boundaries of the Luminance Inertia filter are defined as:

\[
ROI_i = \begin{cases} 
\text{non-pedestrian} & \text{if } ((R_{Li}^r - 0.9)^2 + (R_{Li}^b - 0.95)^2) > 0.7^2 \\
\text{pedestrian} & \text{otherwise}
\end{cases}
\]  \hspace{1cm} (B.15)

The threshold of the Luminance Thresholding filter is defined the same as in the false rate optimised system:

\[
T = k_1 F_{\text{mean}} + k_2 F_{\text{std}},
\]  \hspace{1cm} (B.16)

where \( k_1 = 1 \) and \( k_2 = 1 \). An ROI is classified as pedestrian if:

\[
ROI_i = \begin{cases} 
\text{non-pedestrian} & \text{if } F_{ROI\text{mean}} < T \\
\text{pedestrian} & \text{otherwise}
\end{cases}
\]  \hspace{1cm} (B.17)

The threshold for the Filling Ratio filter is defined as:

\[
ROI_i = \begin{cases} 
\text{non-pedestrian} & \text{if } R_{fill} < T_{fill} \\
\text{pedestrian} & \text{otherwise},
\end{cases}
\]  \hspace{1cm} (B.18)

where \( T_{fill} \) is set to 0.3.

The thresholds of the Merge ROI filter are defined the same as in the false rate optimised system. SSA and SSE must both be at least \( \sqrt{0.7} \).
Appendix C

Available night vision videos

First batch

Rainy weather

<table>
<thead>
<tr>
<th>U11</th>
<th>Filename</th>
<th>Camera type</th>
<th>#frames</th>
<th>#pedestrian ROIs</th>
<th>Meta-data available</th>
<th>Ground-truth data available</th>
<th>Driving speed</th>
<th>Type of sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nfn01</td>
<td>&quot;upper&quot; and &quot;lower&quot;</td>
<td>1322</td>
<td>977</td>
<td>Yes</td>
<td>Yes, &quot;upper&quot; only</td>
<td>Variable</td>
<td>Urban</td>
</tr>
</tbody>
</table>

Dry weather

<table>
<thead>
<tr>
<th>U12</th>
<th>Filename</th>
<th>Camera type</th>
<th>#frames</th>
<th>#pedestrian ROIs</th>
<th>Meta-data available</th>
<th>Ground-truth data available</th>
<th>Driving speed</th>
<th>Type of sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nft05</td>
<td>&quot;upper&quot; and &quot;lower&quot;</td>
<td>315</td>
<td>394</td>
<td>Yes</td>
<td>Yes, &quot;upper&quot; only</td>
<td>Variable</td>
<td>Urban</td>
</tr>
</tbody>
</table>
## APPENDIX C. AVAILABLE NIGHT VISION VIDEOS

<table>
<thead>
<tr>
<th>Video</th>
<th>Filename</th>
<th>Camera type</th>
<th>#frames</th>
<th>#pedestrian ROIs</th>
<th>Meta-data available</th>
<th>Ground-truth data available</th>
<th>Driving speed</th>
<th>Type of sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>U13</td>
<td>nft08</td>
<td>&quot;upper&quot; and &quot;lower&quot;</td>
<td>86</td>
<td>41</td>
<td>Yes</td>
<td>Yes, &quot;upper&quot; and &quot;lower&quot;</td>
<td>Variable</td>
<td>Urban</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U14</td>
<td>nft09</td>
<td>&quot;upper&quot; and &quot;lower&quot;</td>
<td>927</td>
<td>Unknown</td>
<td>Yes</td>
<td>No</td>
<td>Variable</td>
<td>Urban</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U15</td>
<td>nft18</td>
<td>&quot;upper&quot; and &quot;lower&quot;</td>
<td>136</td>
<td>65</td>
<td>Yes</td>
<td>Yes, &quot;upper&quot; and &quot;lower&quot;</td>
<td>Variable</td>
<td>Rural</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U16</td>
<td>nft26</td>
<td>&quot;upper&quot; and &quot;lower&quot;</td>
<td>1631</td>
<td>0</td>
<td>Yes</td>
<td>Yes, &quot;upper&quot; and &quot;lower&quot;</td>
<td>Variable</td>
<td>Rural</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### U17

<table>
<thead>
<tr>
<th>Filename</th>
<th>nft36</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera type</td>
<td>&quot;upper&quot; and &quot;lower&quot;</td>
</tr>
<tr>
<td>#frames</td>
<td>919</td>
</tr>
<tr>
<td>#pedestrian ROIs</td>
<td>1018</td>
</tr>
<tr>
<td>Meta-data available</td>
<td>Yes</td>
</tr>
<tr>
<td>Ground-truth data available</td>
<td>Yes, &quot;upper&quot; only</td>
</tr>
<tr>
<td>Driving speed</td>
<td>Variable</td>
</tr>
<tr>
<td>Type of sequence</td>
<td>Urban</td>
</tr>
</tbody>
</table>

### U18

<table>
<thead>
<tr>
<th>Filename</th>
<th>nft37</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera type</td>
<td>&quot;upper&quot; and &quot;lower&quot;</td>
</tr>
<tr>
<td>#frames</td>
<td>341</td>
</tr>
<tr>
<td>#pedestrian ROIs</td>
<td>339</td>
</tr>
<tr>
<td>Meta-data available</td>
<td>Yes</td>
</tr>
<tr>
<td>Ground-truth data available</td>
<td>Yes, &quot;upper&quot; only</td>
</tr>
<tr>
<td>Driving speed</td>
<td>Variable</td>
</tr>
<tr>
<td>Type of sequence</td>
<td>Urban</td>
</tr>
</tbody>
</table>

### Second batch

**Dry weather**

### U21

<table>
<thead>
<tr>
<th>Filename</th>
<th>Philips.30.R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera type</td>
<td>&quot;lower&quot;</td>
</tr>
<tr>
<td>#frames</td>
<td>1197</td>
</tr>
<tr>
<td>#pedestrian ROIs</td>
<td>Unknown</td>
</tr>
<tr>
<td>Meta-data available</td>
<td>No</td>
</tr>
<tr>
<td>Ground-truth data available</td>
<td>No</td>
</tr>
<tr>
<td>Driving speed</td>
<td>30 km/h</td>
</tr>
<tr>
<td>Type of sequence</td>
<td>Rural</td>
</tr>
</tbody>
</table>
### U22

<table>
<thead>
<tr>
<th>Filename</th>
<th>Philips_50_R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera type</td>
<td>&quot;lower&quot;</td>
</tr>
<tr>
<td>#frames</td>
<td>406</td>
</tr>
<tr>
<td>#pedestrian ROIs</td>
<td>188</td>
</tr>
<tr>
<td>Meta-data available</td>
<td>No</td>
</tr>
<tr>
<td>Ground-truth data available</td>
<td>Yes</td>
</tr>
<tr>
<td>Driving speed</td>
<td>50 km/h</td>
</tr>
<tr>
<td>Type of sequence</td>
<td>Rural</td>
</tr>
</tbody>
</table>

### U23

<table>
<thead>
<tr>
<th>Filename</th>
<th>Philips_80_L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera type</td>
<td>&quot;lower&quot;</td>
</tr>
<tr>
<td>#frames</td>
<td>196</td>
</tr>
<tr>
<td>#pedestrian ROIs</td>
<td>171</td>
</tr>
<tr>
<td>Meta-data available</td>
<td>No</td>
</tr>
<tr>
<td>Ground-truth data available</td>
<td>Yes</td>
</tr>
<tr>
<td>Driving speed</td>
<td>80 km/h</td>
</tr>
<tr>
<td>Type of sequence</td>
<td>Rural</td>
</tr>
</tbody>
</table>