Delft University of Technology
Department of Computer Vision

Joint probabilistic pedestrian head and body orientation estimation

Madalin Dumitru-Guzu

Supervisors: Prof. Dr. Dariu M. Gavrila,
Assistant Prof. Dr. Laurens van der Maaten

Daily supervisor: Ph.D Student Fabian Flohr

Submitted in part fulfillment of the requirements for the degree of
Master of Science in Media Knowledge and Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science, July 2014

Abstract

This work presents an approach for joint estimation of the pedestrian head and body orientation in the context of active pedestrian safety systems. It involves a probabilistic framework, where a set of orientation-specific detectors are used for each body part for both localization and orientation estimation, their responses being converted to a continuous probability density function.

To improve the localization, spatial anatomical constraints between the head and body are used, in a *Pictorial Structure* approach, to balance the part-based detector responses. The single-frame head and body orientations are integrated over time by particle filtering and estimated jointly to account for orientation restrictions and to obtain anatomical possible orientation configurations.

The experimental evaluation is done over 65 pedestrian tracks in realistic traffic settings, obtained from an external stereo-vision-based pedestrian detection system. The results show that the proposed joint probabilistic orientation estimation framework decreases the absolute mean head and body orientation error by approximately 15 degrees. Also, the system runs in near-real-time (8–9 Hz), which allows the use in the car.

Acknowledgements

I would like to express my gratitude to my thesis supervisors, Prof. Dr. Dariu Gavrila and Assistant Prof. Dr. Laurens van der Maaten, for introducing me to computer vision and to the field of active pedestrian safety systems. I am grateful to them for giving me the opportunity to work on this research project within Daimler Research and Development, for guiding my work and for their scientific reviews, which had a major impact on improving my work. I have learned a lot on how to conduct research, perform experiments and write scientific documents.

I would also like to thank Fabian Flohr for being an outstanding daily supervisor. I appreciate that he always made time for me to discuss any encountered problems, despite his busy schedule, and his invaluable support and feedback during the realization of this thesis.

Furthermore, I am very grateful to Julian Kooij and Nicolas Schneider for their advices on different aspects during my internship at Daimler AG.

Finally, I thank to my family, especially my parents, for everything they done for me throughout the years. Without their moral and, especially, financial support, doing this master's programme would have been impossible.

Contents

List of Figures

Chapter 1

Introduction

1.1 Context

Road traffic injuries are a leading cause of death, killing nearly 1.2 million people every year worldwide [1]. Many organizations, automobile manufacturers, or research institutions have focused their attention and resources on studying the automobile traffic environment.

More and more car manufacturers equip their cars with sensors which permit the automobiles to guide the drivers to change the driving line safely, to sense speed limit signs, to reduce the speed or keep the distance to the surrounding objects. There are significant advances in developing systems that can track the driver's attention, keep his focus on the road or warn him or even brake automatically to avoid collisions or reduce the severity of the injuries.

Pedestrians play a dominant role in this context since they are the most vulnerable road users. Active pedestrian safety systems have also become very popular over the last few years, not only in the research community, but also in the industrial community. Many important companies started to allocate their resources to build intelligent devices to prevent unwanted events on public roads that involve pedestrians.

Pedestrian detection is an interesting and challenging task in computer vision, with a great

[1] http://www.who.int/violence_injury_prevention/road_safety_status/2013/en/

number of application domains (surveillance, media analysis, robotics or automotive safety). In the domain of intelligent vehicles, knowing auxiliary information about the environment and the behavior of the road users (drivers, bicyclists, pedestrians etc.) could help to improve the traffic safety.

This work has be done in collaboration with Daimler's research and development department from Ulm, Germany. In the area of video-based pedestrian detection, Daimler already introduced to the market the first commercial active pedestrian systems in the 2013-2014 Merdeces-Benz S-, E- and C-class models.

1.2 Motivation and objectives

A sophisticated situation analysis module is a necessity of any modern active pedestrian safety system. One very important requirement of such a module is the ability to make accurate path predictions. The prediction of the next move of the pedestrians is a very difficult task as they are highly maneuverable, being capable of changing their walking direction or accelerating / decelerating very fast.

For this reason all the useful pedestrian information which is available should be used to reduce the uncertainty about that the pedestrian will do next. Several studies were conducted and their conclusion is that the head orientation can be considered as a good indicator of the pedestrian next move. Moreover, the body orientation plays an important role in deciding on the movement direction of the pedestrian, making the reaction to dynamic changes faster.

For example, in the study of Schmidt and Färber [4] several participants were tested. They had to watch several videos of pedestrians walking towards the curbside and decide whether the pedestrians would stop or cross, at various time instants. Each participant was given a different amount of visual information (different parts of the scene were hidden from the observer) and the effect on their classification performance was examined. The conclusion of the study was that the head motion is one of the most important indicators of future actions of a pedestrian.

Hamaoka et al. [5] studied the head turning behaviors at pedestrians crosswalks to determine the best point of warning for inattentive pedestrians. The pedestrians were equipped with gyro sensors to record the head turning motion and their task was to press a button when they recognized an approaching vehicle.

Motivated by the above findings, this thesis will focus on pedestrian orientation estimation in the context of intelligent vehicles. The objective is to use as much as possible all the information that is available in this context and to obtain a robust, near real-time orientation estimate of a pedestrian (in real urban traffic this would mean around 10 fps).

To this end, the following question is of interest for the task: *"How can we efficiently and accurately estimate the pedestrian head and body orientations over time using a commercial stereo-vision-based setup on board of a moving vehicle?"*

1.3 Contributions

The pedestrian orientation is divided into the orientation of head and of body because empirical evidence shows that these body parts do not always point in the same direction. This work deals with estimating the head and body orientation of a pedestrian over time. Using a standard stereo-vision-based sensor setup (Figure 1.1), which is nowadays already available on the market, the task can be made more accessible.

The head and body orientation estimation module can be built on top of an existing system which delivers a 3D pedestrian track. Here, this pedestrian tracker is composed from a HOG / linSVM pedestrian detector [6] and Kalman Filtering, but is not part of paper contributions.

The main contribution is a joint probabilistic head and body orientation estimation framework that handles faulty part detections, continuous orientation estimation, coupling between the body- and head- localization and orientation, the latter one being tracked over time.

The work is different from other papers in several ways. First, compared to the related work, which is mainly based on surveillance applications and uses mono-vision-based systems, here

Figure 1.1: Stereo sensor setup already available on the market

the intelligent vehicle context and a stereo-vision-based setup are considered. Second, the full continuous distribution of the head and body orientations is modeled, even though only a small set of detectors for canonical body part orientations is used. Third, the detectors are also used to jointly localize the head and body (additionally exploiting disparity information and knowledge about body parts configuration).

Chapter 3 describes an overview of the proposed system. Figure 3.1 illustrates a general picture of the modules involved by the approach.

1.4 Outline of the report

This report is structured as follows. Chapter 2 covers previous work on orientation estimation, while Chapter 3 presents an overview of the approach and sets some basic mathematical notations. Chapter 4 discusses the localization and single-frame orientation estimation phase. It starts with a general description of the used detector and then different schemes for training it in a multi-class setting are briefly presented. Then it continues with the description of how the body parts are localized. First, it presents how the regions that have to be classified are generated and, then, how the correct locations of the head and body are obtained and constrained on each other. Chapter 5 discusses the tracking of the head and body orientations for improving the estimation results. This chapter also presents an easy way to include anatomical

orientation constrains in the framework. In Chapter 6 follows a discussion on how to deal with orientation estimation for multiple people and brings possible practical solutions to make the system run in near real-time in a car setup. Chapter 7 presents the experimental evaluation of the system, while Chapter 8 concludes the report and proposes future work directions.

1.5 Publications

This work is included in one conference article (Flohr et al. [7]). Also, it is part of a submitted journal article (Flohr et al. [1]).

Chapter 2

Related Work

There is a very extensive literature on person pose estimation. This work focuses on the head and full body orientation estimation. First, the research on single-frame orientation estimation is investigated and, then, possible solutions for tracking this information over time are searched. A summary of some of the related work can be found in Table 2.1.

2.1 Related work on orientation estimation

The particular application which is considered has a major impact on the techniques used for body parts orientation estimation and localization. [8] and [9] are surveys that investigate the problems of head orientation estimation.

Human-Machine Interaction (HMI) [10, 11] or entertainment applications [12] often have at their disposal high resolution images. In this kind of applications the orientation estimation has to be more precise than in other applications, but the subjects usually cooperate with the system and the environment is highly controlled (for example the background and lighting are fixed).

On the other hand, surveillance [13, 14, 15, 16, 17] and intelligent vehicles [18, 19] domains need to deal with low resolution images. Moreover, the subjects do not cooperate with the application

and the environment is uncontrolled. The background is more complex and dynamical, the lighting can change fast.

Because the application domain of this work is intelligent vehicles, the focus is more on techniques applied in this domain or related ones. To overcome some of the domain challenges, lower-level image features are commonly used as they are more robust then using the image directly. Some popular ones are SIFT/HOG features, being used in [20, 13, 14, 21, 19]. [18, 22] use Haar features, while [19] extracts Local Rreceptive Features (LRF) . Features based on distance metrics are also used by [20, 11, 15].

These features are combined with different classification schemes. One very popular classifier is Support Vector Machines (SVMs), being used by [14, 21, 15, 22, 19]. Other popular classification schemes are Neural Networks (NNs) [19], Random Regression/Decision Trees or Ferns [20, 13, 11] or Boosting cascades [18].

The approaches above can be used for both, head and body orientation estimation. Enzweiler and Gavrila [19] train four orientation specific classifiers to obtain a pedestrian detection. The classifiers are then reused, in a weighting scheme, to infer a continuous orientation for the corresponding pedestrian detection.

Schulz et al. [18] use a boosting cascade in an one-versus-all manner on Haar features to learn eight head orientations. The maximum classifier response over all possible hypotheses in different scales and locations and orientation classes is then selected as the final estimation for the head location and orientation.

Benfold and Reid [20] train a random fern architecture on HOG and color based features to estimate a head orientation. The head is localized by a HOG-based detector. Again eight orientation classes are used.

Most of the methods use manually labeled data for training. In contrast to this, Benfold and Reid [13] learn the head orientation in an unsupervised manner, using the output of a tracking system [23]. They make the assumption that the head orientation is dependent on the walking direction. Robertson and Reid [16] assume that the walking direction can be an indicator for

the body orientation, if people are only moving forward.

An alternative to these approaches is to estimate the orientation directly from an applied shape model. Models like Active Shape Models (ASMs) and Active Appearance Models (AAMs), which were introduced by Cootes [24, 25], are popular for inferring orientation information. Both, ASMs and AAMs, are based on feature correspondence and fitting them to an image can give sub-optimal solutions, by getting stuck in local maxima.

Gavrila et al. [26] represent the pedestrian shape as a set of multiple Statistical Shape Models (SSMs), accounting in this way for different shape aspects, as feet apart or feet closed. This idea of multiple linear subspaces can also be found in Lee and Kriegman [27], where they use the method of Hall et al. [28, 29] and apply an incremental on-line update of multiple linear-subspace models, each representing a face orientation.

Another interesting approach is the one of Zhu and Ramanan [30]. There a face is detected and its orientation is estimated by a mixture of trees. These trees share the same pool of facial landmarks and use a global mixture, similar with AAM, to capture topological changes due to viewpoint.

Low resolution images make this type of strategies to be unsuitable for head orientation estimation due to the inaccurate shape information with decreasing resolution. Body orientation estimation can exploit, even in lower resolution images, information about the shape, using it as prior knowledge (e.g. [31]).

One of the shortcomings of some of the work above is that they do not model both the head and body orientation or they do not model an orientation relationship between these two. Head and body orientation estimation can be improved by introducing anatomical constrains. This idea can be found in the work of [13, 32, 14, 16, 33, 17]. Smith et al. [17] only constrain the head location with respect to the body location to get a physically possible configuration. Zhao et al. [33] use the body orientation only to differentiate between difficult head orientations (e.g. opposite directions).

Benfold and Reid [13] apply a Conditional Random Field (CRF) to model the interaction be-

tween the head orientation, walking direction and appearance and to recover the gaze direction. While Robertson and Reid [16] constrain the head orientation on the velocity direction, Chen et al. [32] couple the head orientation with the body orientation and the body orientation with the velocity direction. In the latter, the couplings are modeled with *von Mises* distributions.

In contrast with the strategies mentioned above, Chen and Odobez [14] use constrains between the head and the body orientation directly during classifier training.

2.2 Related work on time integration of the estimated orientations

Another drawback of some of the work presented until now is that it only offers single-frame orientation estimates, which are potentially noisy. A way to further improve the orientation estimation is the filtering of the single-frame results, whenever the application domain permits it. This smooths out the orientation signal by eliminating part of the noise.

One of the simplest approaches for integrating over time is to choose the most frequent direction over a fixed number of frames [22]. Other, more sophisticated, models use for example Hidden Markov Models (HMMs) (e.g. [21]).

One popular tracking algorithm in this area is the Particle Filtering (PF) framework. [16, 34, 17, 10, 32] use it to keep track of a body part information distribution over time. An advantage of PF frameworks is that they allow for easily coupling of the body parts through the dynamical model. This is done in the work of Robertson and Reid [16] or Chen et al. [32].

Smith et al. [17] uses a Reversible-Jump Markov Chain Monte Carlo (RJMCMC) sampling scheme for particle filtering to handle a large state space consisting of inter-person (multi person tracking) and intra-person (localization between head and body) interactions.

For completeness, there is also an extensive work on articulated 3D body pose recovery, e.g. see surveys [35, 36]. These typically require multiple cameras, are computationally intensive

and still have issues with robustness.

2.3 Approach to answer the question of interest

After reviewing the related work, the motivation of this thesis comes from the desire of building a system where the head and body locations and orientations are accurately estimated, in the context of intelligent vehicles, avoiding some of the drawbacks of the previous work (by using a coupling between body parts location and orientation and tracking over time). The following approach is used answer the proposed question of interest.

First, a set of orientation detectors are trained for head and body, separately. Their responses are used together with spatial constraints to localize the head and body and to give a discrete estimate of the part orientation. Then the discrete orientations are transformed into a continuous distribution over the entire orientation domain. Secondly, the continuous distribution is filtered over time with a PF to remove some of the detection noise. This phase also models restrictions between the body parts orientations.

The proposed approach, along with considering the intelligent vehicle context, also differentiate this work from [16, 32, 14, 20]. A more detailed overview description can be found in Chapter 3.

Table 2.1: Related work summary

	[17]	[16]	[14]	[13]	[10]	[20]	[15]	[21]	[22]
Joint head detection and orientation estimation	yes	no	no	no	yes	no	no	no	no
Appearance/Feature based approach for head-pose tracking	appearance	appearance	appearance	appearance	appearance	feature based	appearance	appearance	appearance
Orientation estimation per component	head (no body orientation)	body and head	body and head	head and movement direction	only head	only head	only head	body	body
Single/Multiple people jointly	Multiple people jointly	single people	single people	single people	single people	single people	single people	single people	single people
State per person	body (location,scale,aspect ratio) and head components ((location,scale,aspect ratio,in-plane rotation),(head pose))	body and head orientations	minimization of an objective function which takes care also of the coupling between the head and the body	head direction and ground-plane velocity	location, scale, in-plane rotation angle, pose	location and velocity	no tracking; just classification (using SVM)	no tracking (classification using SVM which provides probabilities of each discrete orientation)	no tracking, just classification
Dynamic Model	for body location, head location and head pose (independent of each other)	head model dependent on the body model	-	a CRF is used to model the interaction between the head motion, walking direction and appearance to recover gaze direction	separate for location and scale, pose, in-plane rotation	replaced with velocity estimates from feature tracking	- no tracking; just classification (using SVM)	-	no tracking (16 classifiers; the direction is given by a decision function that considers a classifier and the 2 classifiers corresponding to the neighboring directions)
Measurement Model	body model (based on binary and color features which are compared with learned head model (based on texture, skin color and silhouette features compared with learned models) (independent of each other)	body orientation measured based on velocity; head orientation measured by matching a precomputed database	multi-level HoG features	-	head texture and color models (conditionally independent given the state)	HoG and CTCs features	descriptor based on similarity distance maps that index eac pixel of the head image to the mean appearance templates of head images at different poses	HoG features	Haar wavelets
Coupling between head&body	no coupling	coupled, by constraining the dynamic model of the head such that there is aa balance between the fact that people tend to look in the same direction as their body is pointing and temporal consistency	coupled (using a factor defined as the discrepancy between the body pose and head pose classifier outputs)	coupled, using a matrix that keeps prior knowledge about the head orientation given the curent movement direction	no coupling	no coupling	no coupling	no coupling	no coupling
Temporal Integration (batch mode / incremental filtering)	incremental filtering	incremental	Non-iterative global optimization scheme	incremental	incremental filtering	no temporal; no batch processing; classification of each head image (using random ferns)	classification of each head image	using a HMM, where the transition probabilities between classes are learned from training data	individual classifications are combined over a walking sequence by choosing the most frequently direction

Chapter 3

Overview

Figure 3.1 presents an overview of the proposed approach. The availability and the efficiency of previous modules motivates the use of a decoupled pedestrian tracker At each time step t it provides estimates for the pedestrian's position $\boldsymbol{x}_t = [x_t, y_t]$, pedestrian's height h_t, defined in world coordinates on the ground plane, and the velocity $\dot{\boldsymbol{x}}_t = [\dot{x}_t, \dot{y}_t]$. The pedestrian tracks represent the input of the orientation estimation framework, which, in the end, tracks the head ω_t^H and the body ω_t^B orientations jointly as $\boldsymbol{\omega}_t = [\omega_t^H, \omega_t^B]$. Therefore it is assumed that all $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ and h are already known up to time t, when orientation framework starts working, and the focus is only on estimating $\boldsymbol{\omega}_t$.

The system is composed from two submodules. The first one is a head / body localization module, which also provides a single-frame continuous estimation of the head / body orientation (through the pan angle). It is described in Section 4.6. The second submodule takes the single-frame orientation estimates of the first one and tracks them, while also adding constraints between the two orientations of interest to make the estimation more robust. It is described in Chapter 5.

Because the framework is only provided with an estimate of the pedestrian's full bounding box and not with the exact location of the body parts, these need to be searched in the image. To do so multiple candidate regions are taken into account for both parts. Let $\boldsymbol{z}_t = [\boldsymbol{z}_t^H, \boldsymbol{z}_t^B]$ be the observed image data at time t, which can be decomposed into head $\boldsymbol{z}_t^H$ and body $\boldsymbol{z}_t^B$

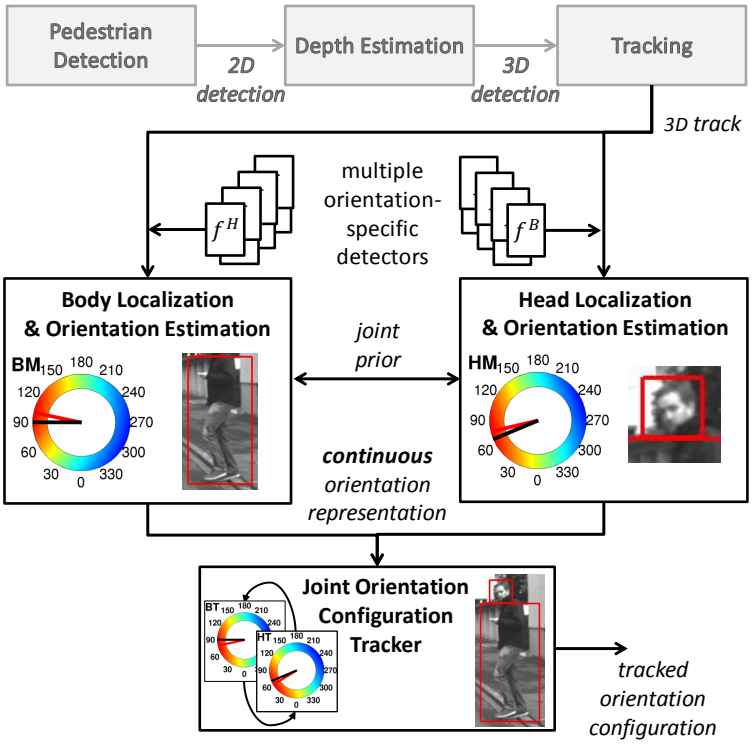


Figure 3.1: Proposed joint probabilistic orientation estimation approach – re-used from Flohr et al. [1]

observations. For example, if there are N candidate regions generated for the head at time t, the output of the corresponding observation can be written out as $\boldsymbol{z}_t^H = [z_t^{H(1)}, z_t^{H(2)}, \ldots, z_t^{H(N)}]$.

The angular domain $[0°, 360°)$ is discretized into a fixed set of orientation classes Ω, e.g. centered around angles of $0, 45, \ldots, 315$ degrees. Then a detector is assigned to each class, e.g. $f_0, f_{45}, \ldots f_{315}$, for both head and body, separately, to evaluate how well an image region corresponds to a specific body party in a certain orientation. The detector response $f_o(z)$ is the strength for evidence that the image region z contains the body part in orientation class o. Here a trade-off has to be made, as having more classes and detectors requires more training data and computational effort, but also yields a more precise estimation of the true angle (up to some extent). An extra non-target or background detector $f_-(z)$ is used to assign a probability to the case that z does not contain the body part. The training stage of all these detectors is presented in Chapter 4.

The output of all detectors $f_o(z)$ and $f_-(z)$ are then used to determine if and where the body part is located in the image, also relying on spatial constraints, based on a disparity pedestrian

segmentation and a *Pictorial Structure* (PS) [37] on the head and body configuration as a spatial priors.

The last phase of this submodule is to transform the orientation of the selected body parts from a discrete space to a continuous one, by combining the orientation detector responses with a set of *von Mises* distributions, centered on the corresponding class value, to obtain a probability distribution over the entire orientation domain.

The output of the first module (the head and body mixtures of *von Mises* distributions) represents the input of the orientation tracking module. The latter one comes in form of a PF [3], whose measurement model is given by the input mixtures. The dynamical model incorporates anatomical and dynamical constraints between the head, body and the direction of movement, to further improve the estimations (as not all the head and body orientation configurations are likely).

Chapter 4

Localization and single-frame orientation estimation

This chapter describes the chosen detector architecture used for body part localization and orientation estimation and the training procedure, discussing the motivation behind this choice. It also presents the motivation of the one-versus-all against multi-class training choice and the head / body training setups. Then it continues to explain how the head and body are localized and how the single-frame orientation estimates for both parts are obtained. First, the process of region generation is presented, then the mathematical model for obtaining the location and the single-frame continuous orientation estimates is discussed and, in the end, the prior used in the model is described.

4.1 Detector description

For detecting the body parts and decide on their orientation a multilayer-feed forward neural network, which uses adaptive local receptive fields (LRF) as features, was chosen. This detection architecture will be referred to as NN/LRF in the remainder of this work.

The used NN/LRF represents a variant of an adaptable Time Delay Neural Network (TDNN) (a

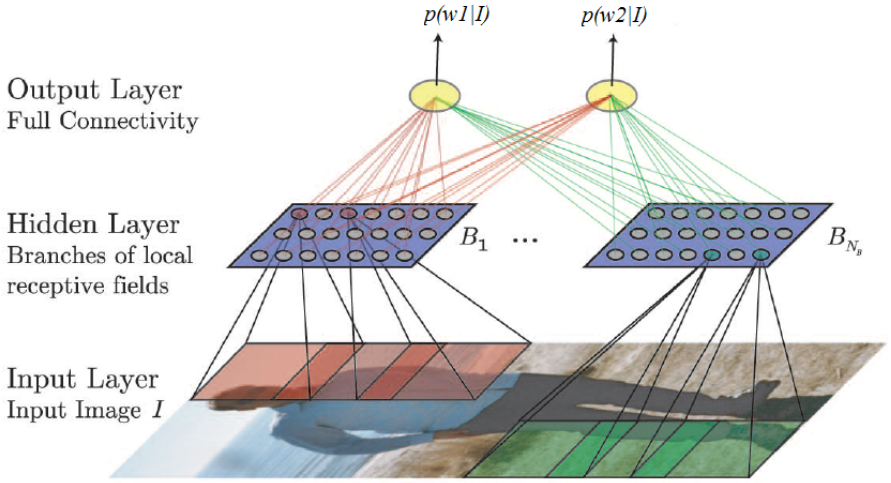


Figure 4.1: Overview of the NN/LRF architecture – re-used from [2]

type of convolutional networks), described in [38]. There TDNN was applied to the task of image sequence analysis for detecting human walking patterns, but here the temporal component was discarded and the network is used to classify single 2D images.

The NN/LRF architecture is presented in Figure 4.1 and it consists of three layers.

The input layer (L1) is a two-dimensional one, referring to an input image patch. Each pixel value of a gray-scale image is fed into the network by a distinct neuron. This means that the number of neurons of L1 equals the dimension $S_x \times S_y$ of the input pattern.

The second layer (L2–hidden layer) receives input from a limited local region of the input layer, referred to as the receptive field of a second layer neuron. Adjacent second layer neurons have adjacent receptive fields on the input layer. This idea is inspired from the human visual system and contrasts the Multilayer Perceptron Network (MLP), which is fully connected. L2 is composed from N_{RF} branches B_i ($i = 1, \ldots, N_{RF}$), each containing a set of sigmoidal neurons receiving input only from the input region defined by their receptive fields. Inside each branch, the set of neurons share the same synaptical weights (also called shared weights principle). Because of this, each branch can be interpreted as a spatial feature detector, on the whole input pattern.

The advantages of the shared weight principle are twofold. First, the number of synaptical

weights is reduced and this enables training with a limited amount of training data. Even if there would be enough training data, the computation complexity and memory requirements would be very large. Secondly, the use of receptive fields incorporates local spatial relations between neighboring input pixels, so both spatial feature extraction and pattern classification is achieved at the same time in the NN/LRF architecture.

The output layer of the NN/LRF (L3) is composed from a number of sigmoidal output neurons, depending on the number of classes to be classified. This layer is fully connected with layer L2. The activation of the output neurons is scaled between 0 and 1 and can be regarded as a scaled estimates of the posterior probabilities of the target classes of interest. This translates into assigning a test input to a class according to a threshold on the activation of the output neuron.

4.2 Training procedure

An instance of the presented architecture is determined using training examples.

Wöhler and Anlauf [38] use a backpropagation on-line gradient descent rule for learning the synaptic weights of the NN/LRF. Denoting with τ_k and γ_k the desired and the actual output of output neuron k, the error ε for an arbitrary input of class k_i is:

$$\varepsilon = \frac{1}{2} \sum_{k=0}^{N_K-1} (\gamma_k - \tau_k)^2 \text{ with } \tau_k = \begin{cases} A, & \text{if } k = k_i \\ -A, & \text{if } k \neq k_i \end{cases} \tag{4.1}$$

The synaptic weights are updated by computing the derivative of the error measure in Eq. 4.1 with respect to the synaptic weights. The choice of the neural activation function $g(x)$, with x being the input potential, influences the desired neural activation $A/-A$. In the backpropagation algorithm, the weight update is proportional to the synaptic input (which is the weighted output of the layer before), the postsynaptic error and the derivative of the activation function.

The chosen activation function here is the hyperbolic tangent function: $g : \mathbb{R} \rightarrow [-1, +1]$, $g(x) = tanh(x)$.

4.3 Motivation for the chosen detection architecture

The motivation for using the NN/LRF comes from two directions.

Firstly, previous work in pedestrian detection shows that LRF features give better performance compared to global features (e.g. PCA) or local non-adaptive features (Haar wavelets) [39]. Global features are inferior to LRF features because sometimes very small details (e.g. hands or feet position, the shape of the head) make the difference between pedestrians and non-pedestrians, details that are lost by the dimensionality reduction of the PCA. The LRF features are superior to the non-adaptive local features because they are tuned to the object of interest during training.

Secondly, even though studies showed that nonlinear SVM classifiers give the best absolute performance in combination with LRF features, the neural network architecture needs lower computational costs (especially in terms of memory requirements), when trained on a large data set and only slightly decrease the performance [39].

4.4 Binary class versus Multi-class detectors

This section motivates the choice between using a single neural network trained on multiple classes and training multiple neural networks in an one-versus-all manner.

One neural network trained on multiple classes involves more weights that need to be adapted to the training samples than a neural network trained in one-versus-all manner. Learning more weights translates into two problems. First, more training data is needed, which might not always be available. Second, having more weights increases the computational costs, especially the memory requirements.

Because the training data which was available and computational resources were limited, the decision to use multiple NN/LRF trained in a modified one-versus-all manner was taken. Section 4.5 presents different one-versus-all training architectures that were investigated during the training process.

4.5 Training schemes

This section presents some of the different one-versus-all architectures that were investigated for training the head and the body orientation detectors. The experimental evaluation of these trials can be found in 7.3. First, the setups for body training are presented, followed by the ones for the head.

The available orientation training data was labeled with 16 orientation labels, but the decision was to use only 8 aggregated orientation classes to have enough data for each class and to limit the computational effort. This decision raised the problem of handling samples with in-between orientations.

In following sections "sample label" refers to one of the 16 orientation labels {0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5, 180, 202.5, 225, 257.5, 270, 292.5, 315, 337.5}. "Orientation class" refers to one of the 8 aggregated orientation classes in which the samples have to be separated {0, 45, 90, 135, 180, 225, 270, 315}. "Detector class" refers to one of the two classes (positive / negative) of orientation detectors.

To obtain more training data, the head / body samples were mirrored and jittered around the original position. The jittering adds also some robustness towards the body part localization noise to the orientation detectors. A trade-off has to be made, as more jittering means less searching at test time, while less jittering translates into a better representation of the training data, but needs more searching time during testing. Moreover, a border was applied to the samples to prevent the detector from learning some border effects. The size of the added border was one of the parameters that was varied during the detectors training.

Another varied parameter at this stage was the dimensions to which all samples were resized before they were fed into the detectors. As the heads / bodies come in different sizes, depending on the scale, all the head / body samples had to be resized to a common head / body dimension before giving them as input to the detectors. This parameter also has a direct influence on the recognition performance, as using larger sizes means learning more weights, but also on the speed at test time.

4.5.1 One-versus-all training setups for the body

Here, several one-versus-all architectures for training body detectors are presented.

4.5.1.1 Setup 1: Orientation class O_i versus Non-body samples

In this setup, the positive class for each orientation detector is given by the corresponding aggregated orientation class. The negative class only contains non-body samples. There are two possibilities for obtaining the aggregated orientation classes used as positive class.

First, an aggregated orientation class is only composed from samples having the same label as the orientation of interest. For example, a 0 degree class is only formed from 0 degree labeled samples.

The second option is to combine the samples labeled with the orientation of interest with samples having neighboring labels. For example, in this case of a 0 degree class samples with labels $\{337.5, 0, 22.5\}$ would be merged together.

For the extra, non-body detector the positive class is formed from non-body samples and the negative class contains all orientation classes.

This setup, with the two options, is illustrated in Figure 4.2.

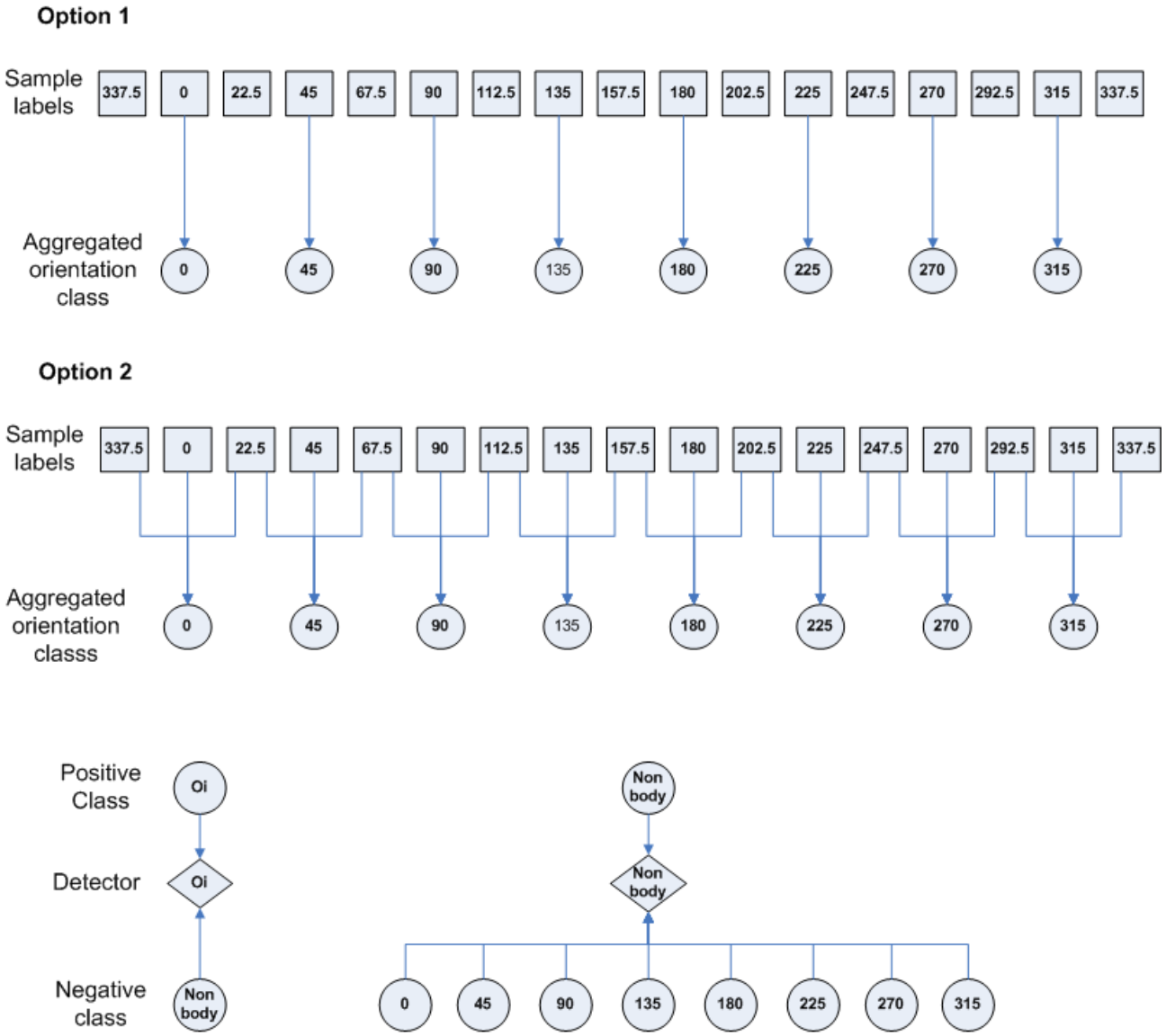


Figure 4.2: Body orientation training setup 1: Orientation class O_i versus Non-body samples

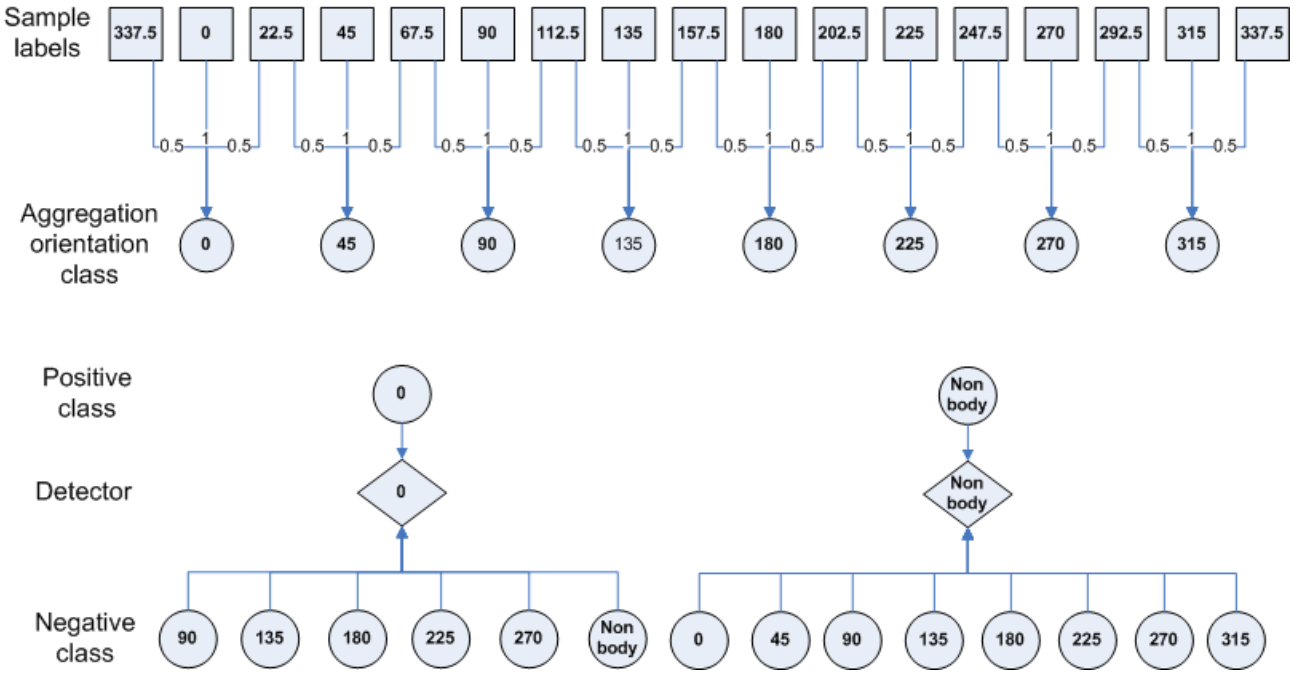


Figure 4.3: Body orientation training setup 2: Weighted label neighbors versus All others, except neighboring classes, & non-body samples

4.5.1.2 Setup 2: Weighted label neighbors versus All others, except neighboring classes, & non-body samples

In this setup, for an orientation detector the positive class is formed by merging samples with the corresponding label with the samples with neighboring labels into a single class. So, for example, the orientation detector for 45 degrees would have in positive class samples with labels {22.5, 45, 67.5}. In this way, the samples with labels {22.5, 67.5, 112.5, 157.5, 202.5, 247.5, 292.5, 337.5} appear multiple times into the positive classes. To reduce this effect these samples are weighted with a factor of 0.5 when they are passed as input to the detector.

The negative class for an orientation detector is given by the all other orientation classes, expect the neighboring ones, and the non-body samples.

For the extra, non-body detector the positive class is formed from non-body samples, while the negative class is given by all the body aggregated orientation classes.

This setup is presented in Figure 4.3.

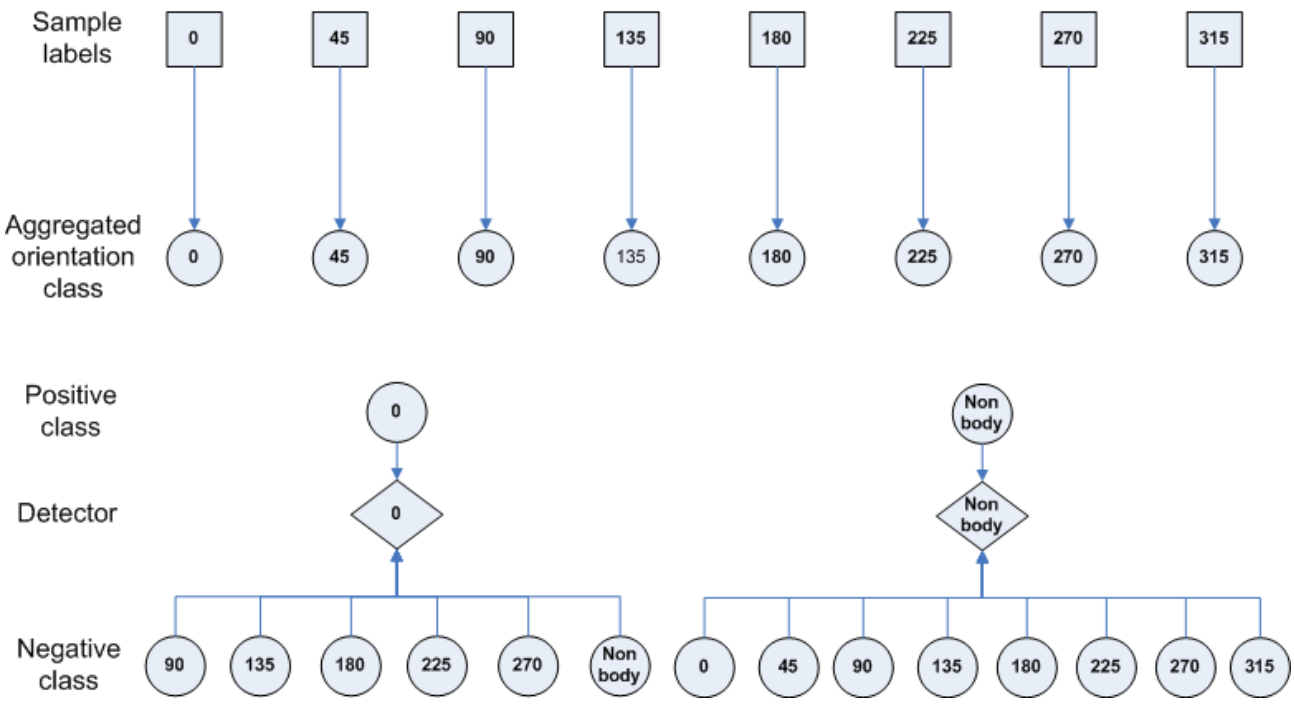


Figure 4.4: Body orientation training setup 3: No label neighbors versus All others, except neighboring classes, & non-body samples

4.5.1.3 Setup 3: No label neighbors versus All others, except neighboring classes, & non-body samples

In this setup, for an orientation detector the positive class is composed only by samples with the corresponding label. No merging with direct label neighbors is done. The negative class is given by all other orientation classes, except the direct orientation neighbors, and the non-body samples.

For the extra, non-body detector the positive class is formed from non-body samples, while the negative class is given by all the aggregated orientation classes.

This setup is presented in Figure 4.4.

4.5.1.4 Setup 4: MLP on top of LRF features

In this setup, a MLP network is trained on top of LRF features. An aggregated orientation class is formed by merging the samples with the corresponding labels with samples with neighboring

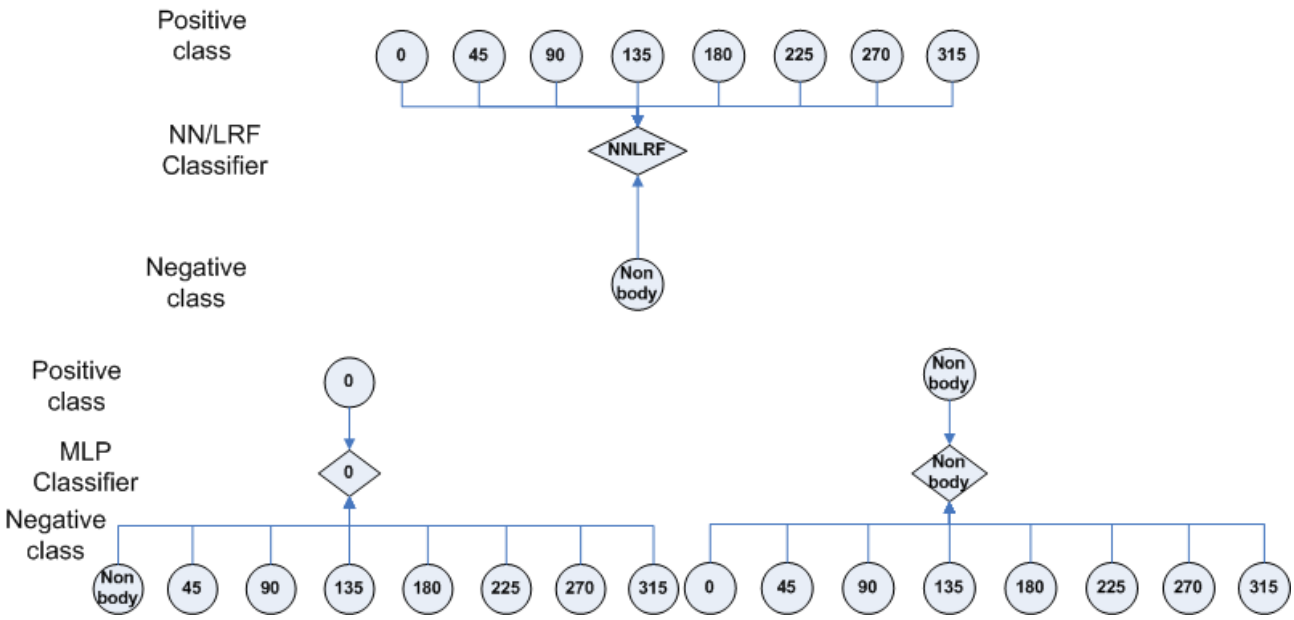


Figure 4.5: Body orientation training setup 4: MLP on top of LRF features

labels (these samples are not weighted anymore; all samples are treated as equal).

The output of the second layer of a NN/LRF (L2) is interpreted as image features. These features are obtained by training a single NN/LRF which has in the positive class all the body aggregated orientation classes, while the negative class is formed from the non-body samples.

Then, an orientation MLP is trained using the features described above. This MLP has as positive class the features extracted from the corresponding aggregated orientation class and as negative class the features of all other orientation classes and the non-body samples.

The extra, non-body detector is trained on non-body samples versus all body aggregated orientation classes.

Before passing the samples to the MLP netowork, they are first passed through the NN/LRF to obtain the LRF features. Only the output of L2 is given as input to the MLP.

This setup is presented in Figure 4.5.

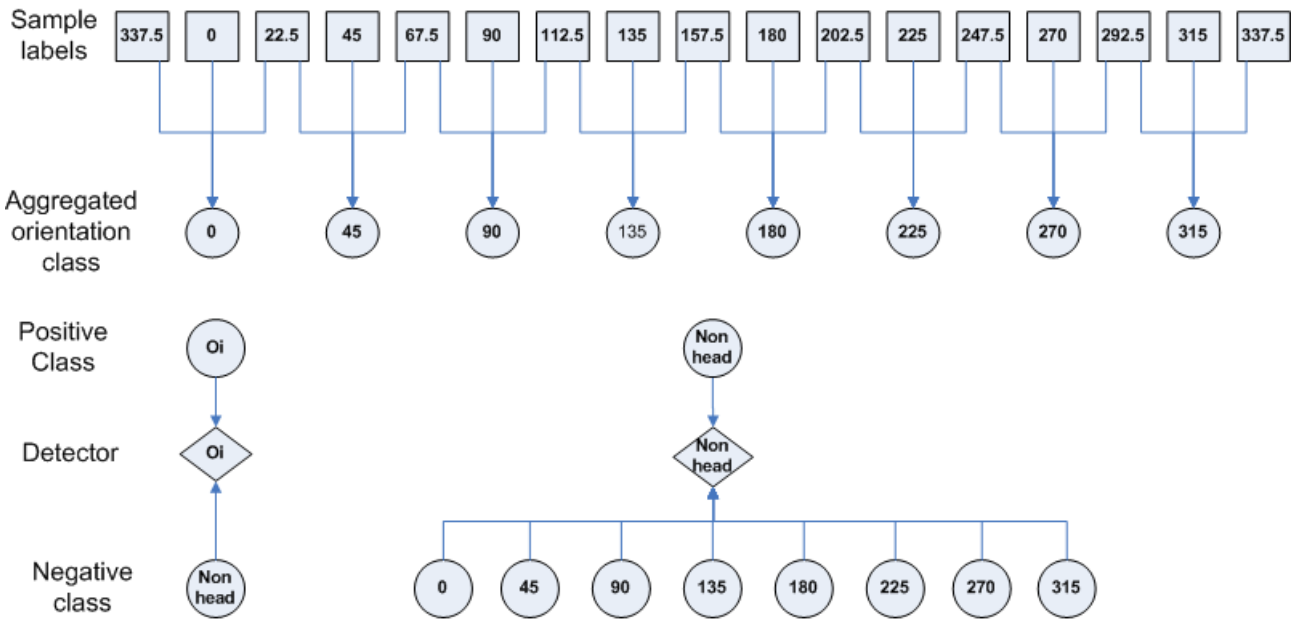


Figure 4.6: Head orientation training setup 1: Orientation class O_i versus Non-head samples

4.5.2 One-versus-all training setups for the head

Here, several one-versus-all architectures for training head detectors are proposed.

4.5.2.1 Setup 1: Orientation class O_i versus Non-head samples

In this architecture, each orientation detector has as the positive class the corresponding aggregated orientation class and as a negative class the non-head samples. An aggregated orientation class is composed by merging the samples with the corresponding orientation label with samples with neighboring labels (this time they are not weighted and all samples are treated as being equal).

The extra, non-head detector is trained on non-head samples versus all the aggregated orientation classes.

The architecture is illustrated in Figure 4.6.

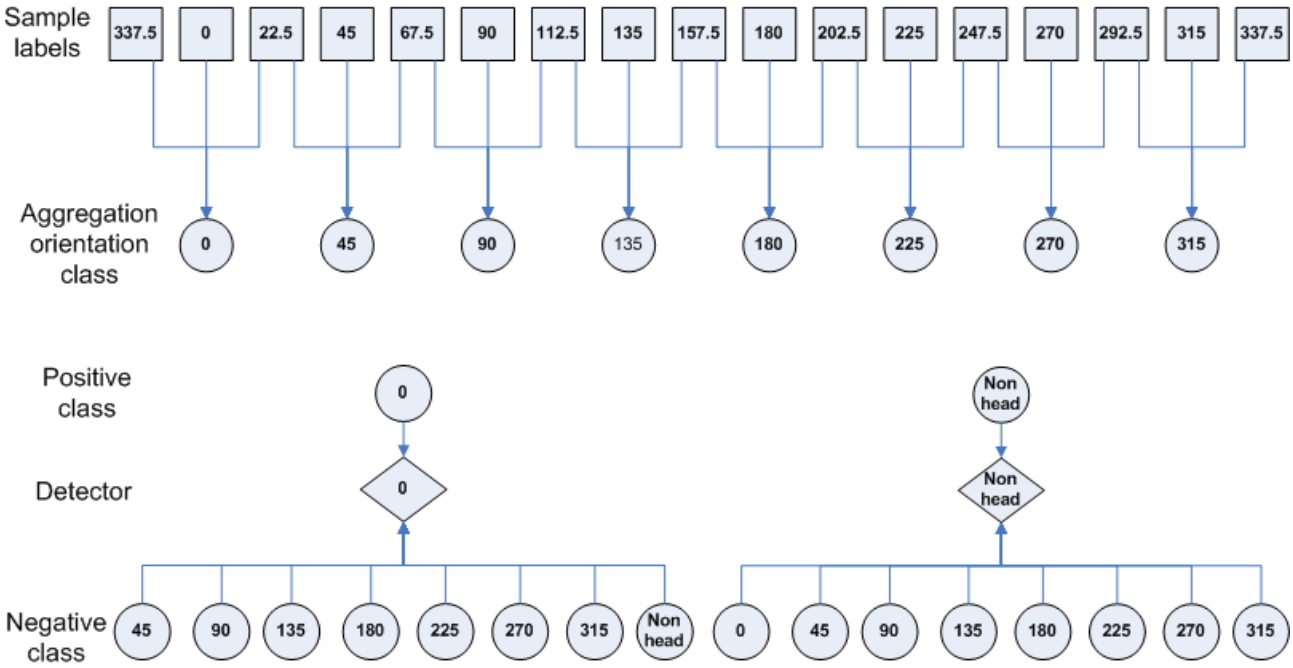


Figure 4.7: Head orientation training setup 2: Merged label neighbors versus All others & non-head samples

4.5.2.2 Setup 2: Merged label neighbors versus All others & non-head samples

For this case, the positive class of an orientation detector is given by the aggregated orientation class. The latter one is composed from samples with the corresponding orientation label and from samples with neighboring labels (all samples are equally important in the orientation class).

The negative class of the orientation detector is formed from all other aggregated orientation classes and the non-head samples. This architecture uses in the negative class the direct orientation class neighbors.

The non-head detector is again trained on non-head samples versus all other orientation classes.

Figure 4.7 presents this setup.

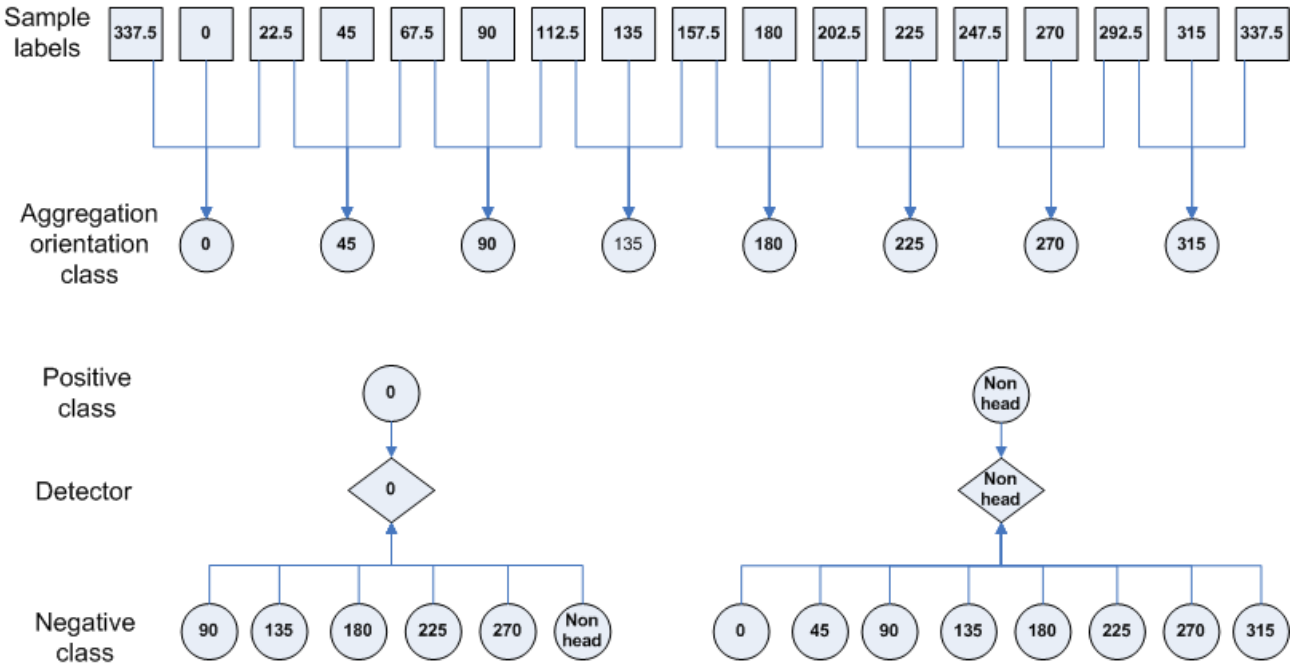


Figure 4.8: Head orientation training setup 3: Merged label neighbors versus All others, except neighboring classes, & non-head samples

4.5.2.3 Setup 3: Merged label neighbors versus All others, except neighboring classes, & non-head samples

This setup is similar with the previous one. The difference here is that, for the orientation detector, the negative classes does not contain the direct orientation neighbors. This architecture is presented in Figure 4.8.

4.5.2.4 Setup 4: Merged label neighbors versus All others, except neighboring classes

The architecture which is presented here is similar with the previous one. The difference is again in the negative class of the orientation detector. The negative class contains all other orientation classes, except the direct neighbors. This time the non-head samples are not included anymore in the negative class.

Figure 4.9 presents this architecture.

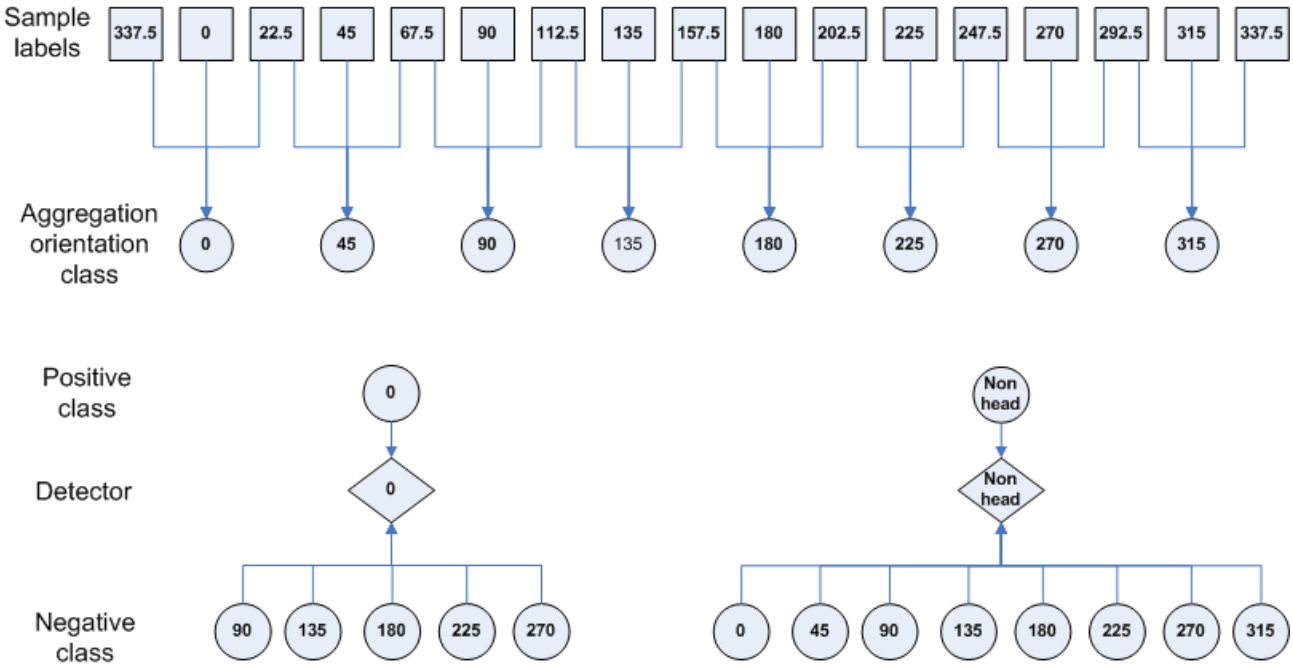


Figure 4.9: Head orientation training setup 4: Merged label neighbors vs. All others, except neighboring classes

4.5.2.5 Setup 5: No label neighbors versus All others, except neighboring classes, & non-head samples

For training an orientation detector with this architecture the positive class is given by the aggregated orientation class of interest. The negative class is formed from all other orientation classes, except the direct neighbors, and the non-head samples.

The aggregated orientation class is composed only from samples which have the orientation label of interest. No merging with samples with neighboring labels is done.

The additional, non-head detector is trained, as usual, on non-head samples versus all the aggregated orientation classes.

The architecture is presented int Figure 4.10.

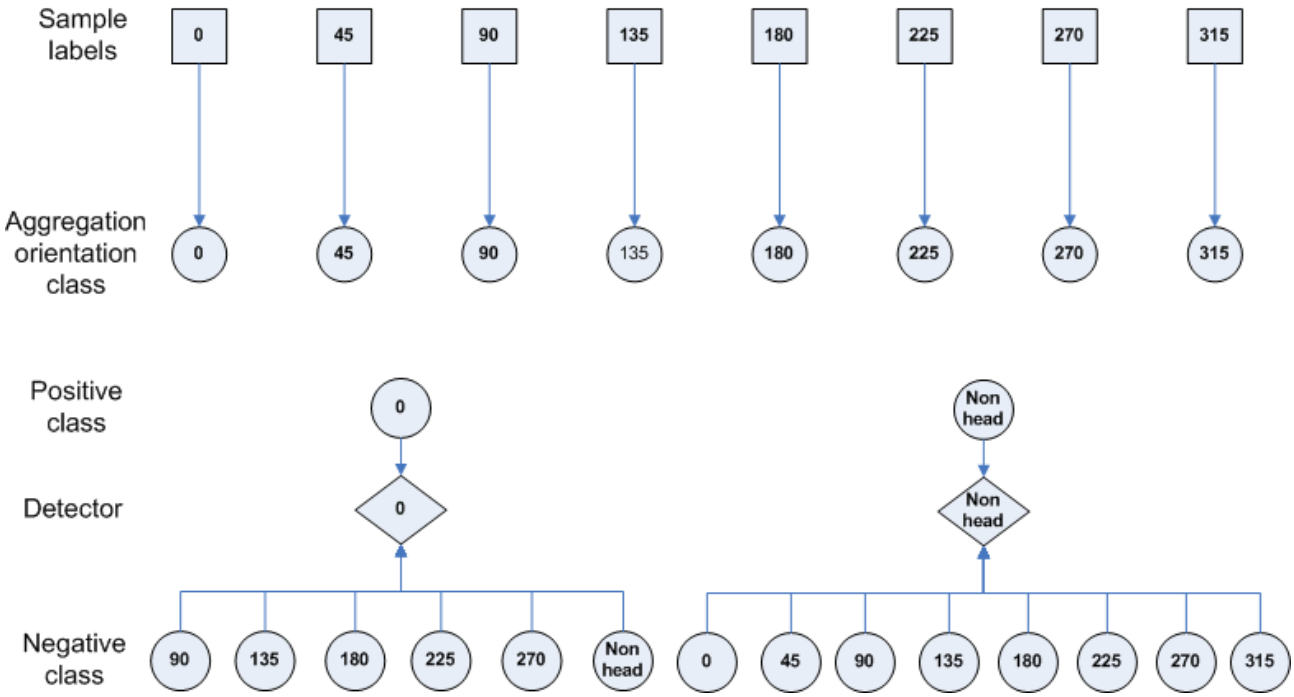


Figure 4.10: Head orientation training setup 5: No label neighbors versus All others, except neighboring classes, & non-head samples

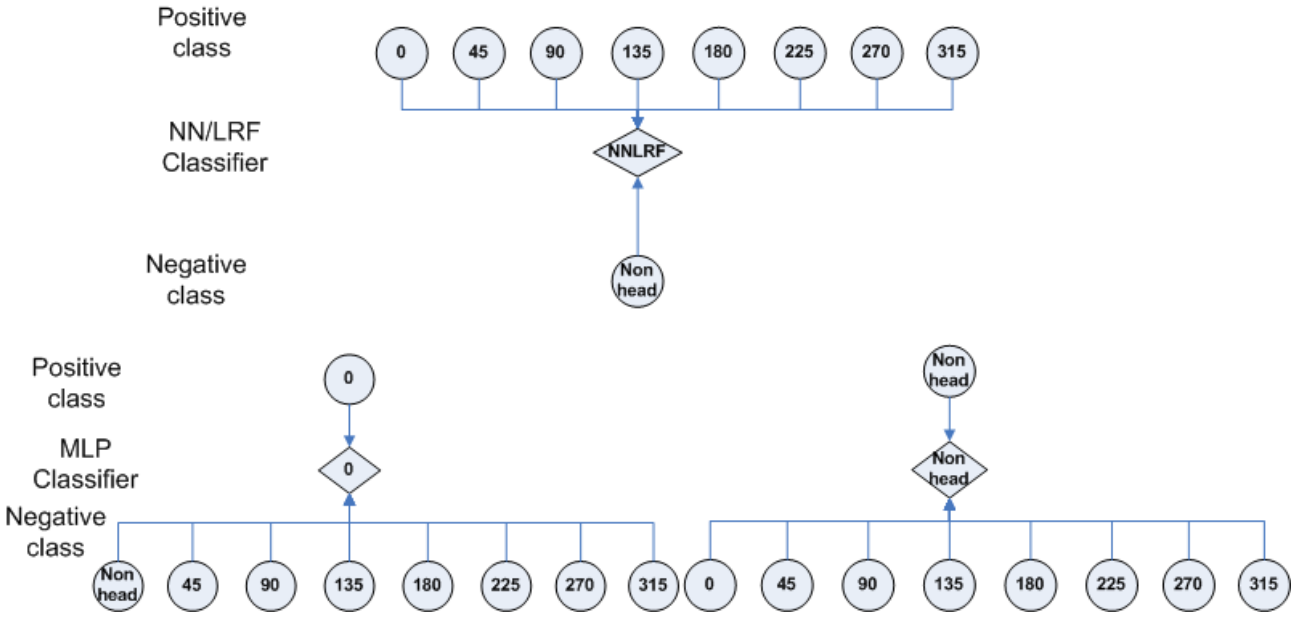


Figure 4.11: Head Orientation Setup 6: MLP on top of LRF features

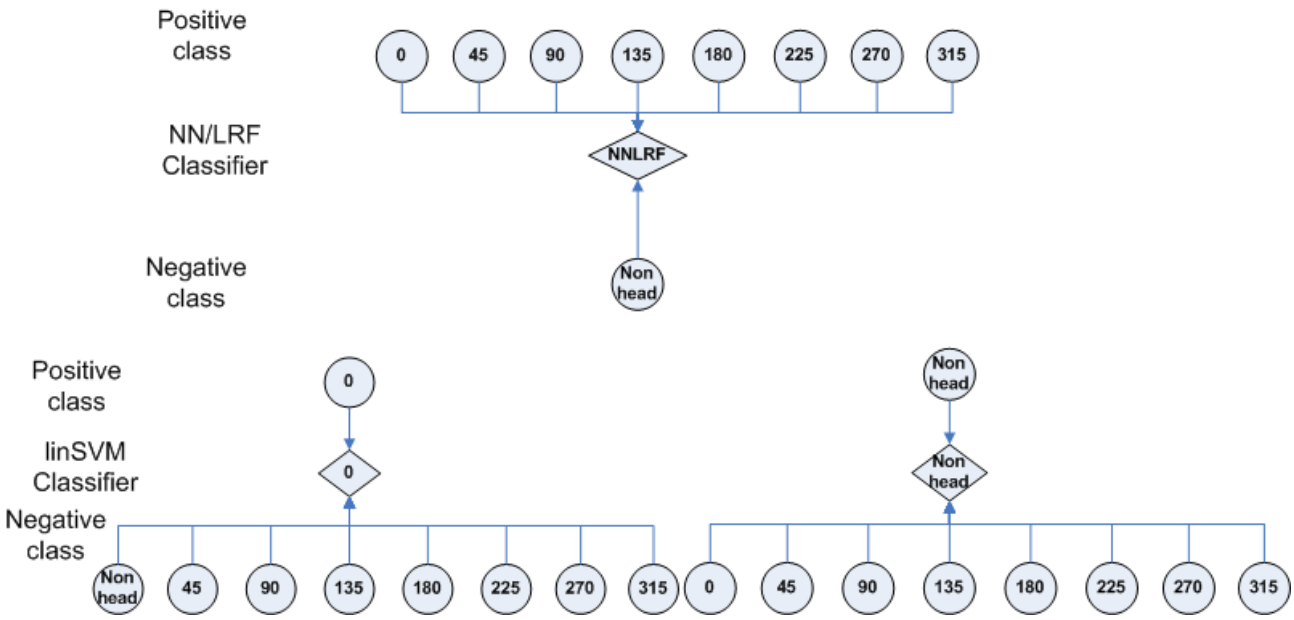


Figure 4.12: Head Orientation Setup 7: linSVM on top of LRF features

4.5.2.6 Setup 6: MLP on top of LRF features

This setup is similar with the corresponding setup for the body. The only difference here is that the body samples are replace with head samples. This architecture is presented in Figure 4.11.

4.5.2.7 Setup 7: Linear SVM on top of LRF features

This setup is similar with the previous one. The difference is that the MLP neural network is replace in this case with a linear SVM. The architecture is presented in Figure 4.12.

4.6 Body parts localization and single-frame orientation estimation

4.6.1 Region generation

As discussed in Chapter 3, the input to the orientation estimation module is a set of tracks coming from an external pedestrian tracker. The latter one only provides estimates of the

pedestrian's bounding box and the 3D world coordinates. The exact locations of head and body are not known and they have to be searched inside or in the proximity of the given pedestrian box.

To achieve this, multiple body part regions are generated and evaluated. In the end a decision on the head and body location is taken. Even though the pedestrian bounding box gives some information about where the body parts could be located, a brute force approach would still be too expensive for a system that is desired to run in near real-time, as the regions have to be generated at different scales and locations.

A better approach would be to use information coming from estimations of the pedestrian height and the horizontal gravity line. Knowing the height solves the problem of generating body part regions in different scales because the scale can be approximated from the height. If the height of the pedestrian is known, then the head can be approximated as being 15% of whole body, with a 1×1 aspect ratio. The body represents the remaining 85% of the pedestrian height, with an aspect ratio of 2×1. The height estimation is described in Section 4.6.1.1.

The horizontal gravity line represents the medial axis of the pedestrian and it can be used to generate body part regions around it. Moreover, estimates for the head and body centers can be used as additional information about the head / body locations. These are described in Section 4.6.1.2.

The height, horizontal gravity line, head and body centers are all estimated based on stereo data.

4.6.1.1 Height estimation

The estimation of the pedestrian height is based on stereo data. Using the stereo sensor setup, a disparity map is computed using the Semi Global Matching (SGM) [40]. An example of such disparity map is presented in Figure 4.13.

A median disparity value $\tilde{d}$ is calculated over all disparity values inside the bounding box provided by the external pedestrian tracker. Based on this and on a learned parameter, $\epsilon = 1.5$,

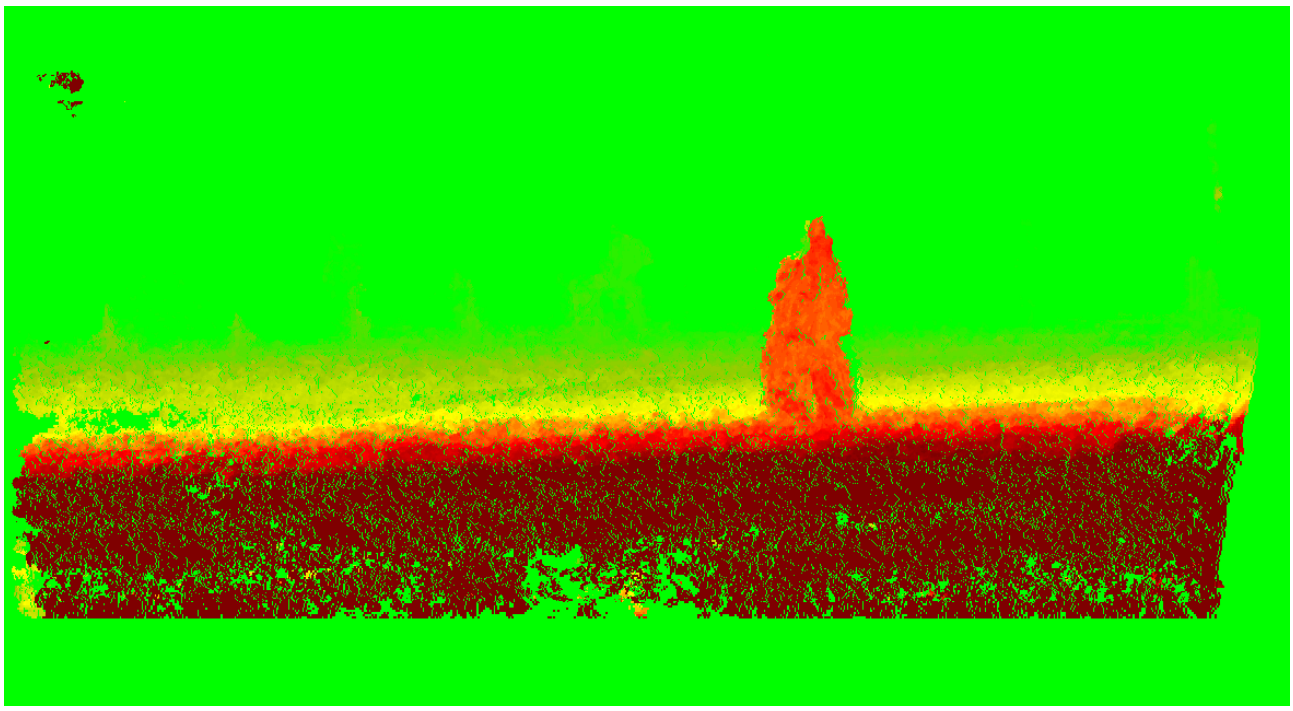


Figure 4.13: Example of disparity map

that accounts for disparity estimation errors, a disparity segmentation is computed inside the bounding box. The values which lay in the range $\boldsymbol{D} < \tilde{d} - \epsilon$ and $\boldsymbol{D} > \tilde{d} + \epsilon$ are considered to belong to the pedestrian. The rest is considered to be background. Such a segmentation can be observed in the left image from Figure 4.14.

Based on this pedestrian segmentation, a vertical / height histogram can be computed. The histogram contains as many bins as the number of rows in the tracked pedestrian bounding box. Each histogram bin counts the number of foreground / pedestrian pixels in the corresponding segmentation row. The histogram obtained in this way is then smoothed using a uniform filter, that averages the values which lay in a certain window. Such a smoothed histogram can be observed in the right image from Figure 4.14. In the smoothed histogram the first and the last value higher than a threshold are considered the highest and the lowest pedestrian points. These points, expressed in image coordinates, are then converted into $3D$ points by using camera projections from image to world coordinates. The $3D$ distance between them is considered to be the pedestrian height.

These height measurements are highly dependent on the disparity map and the pedestrian

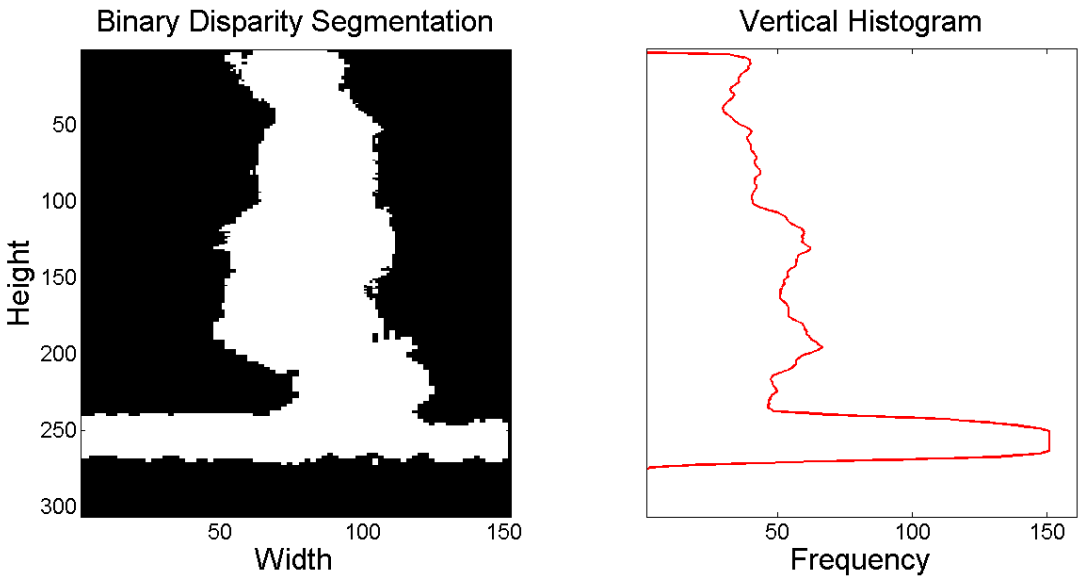


Figure 4.14: Left: Pedestrian segmentation; Right: Corresponding height histogram

disparity segmentation quality. As there are cases where these two do not have a very high accuracy, the height measurements are tracked over time using a particle filter. This tracker lays outside the orientation estimation framework, but still it is described in the following paragraph.

The PF for tracking the height accepts measurements expressed in world coordinates. This decision was taken to account easily for errors in the segmentation computation. The disparity computation is influenced by the distance, being more blurry far away. This happens because it is harder for the SGM algorithm to match the two images when there are few features (edges) extracted. In these cases also the pedestrian segmentation, so the height measurements, are noisier. The dynamical model of the particle filter has a very low system noise, as the height is a fixed quantity for a pedestrian, and it is modeled with a Gaussian with a small variance. Some small variations can come from bad segmentations or because the pedestrian is walking (case when the distance between the highest and lowest point increases). The measurement model (likelihood) is given by a Gaussian distribution with the mean given by the measured height.

The output of this external height tracker (h_t) is used for regions of interest generation.

4.6.1.2 Horizontal gravity line, head and body centers estimation

The disparity map can also be used to determine the pedestrian medial axis / horizontal gravity line and the head and body centers.

For example, the horizontal gravity line is obtained in the following way. The pedestrian stereo segmentation is used to compute a horizontal histogram. The histogram has as many bins as there are columns in the stereo segmentation. Each bin counts the number of foreground pixels in the corresponding column. This noisy histogram is smoothed in the same way as it was done for the height histogram. In the smoothed version the dominant peak is searched. This one is considered as the horizontal gravity line of the pedestrian (g_c^x).

The horizontal gravity line is used to generate body part regions around it, as it represents the horizontal center of the pedestrian.

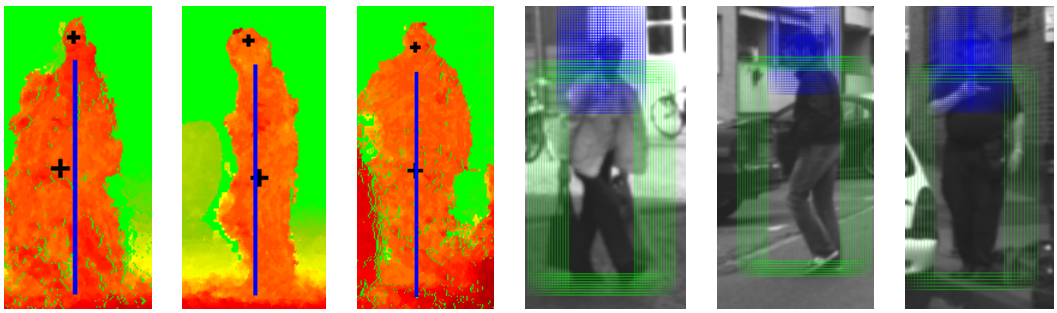
The disparity map can also be used to compute estimates of the head and body centers. These centers can be used as extra information about the location of the body parts. For example, a region has a higher probability to be the correct region of interest if it is closer to the head center.

Examples of these estimates can be find in Figure 4.15 a).

4.6.1.3 Other head / body regions generation considerations

Due to efficiency reasons, the possible head and body regions are generated based on the tracked pedestrian height (h_t) and the estimated horizontal gravity line (g_c^x), inside the tracked pedestrian bounding box. As defined in Chapter 3, let $\boldsymbol{z}^H$ and $\boldsymbol{z}^B$ denote the set of generated head and body regions.

Since the tracked bounding box should already give an appropriate estimation of the body location, the number of used body hypotheses can be much less than the one used for the head. It is important to note that the number of generated hypotheses affects the speed of the system, as each region has to be classified by the corresponding 9 detectors.



a) b)

Figure 4.15: Head and body region generation – adapted from Flohr et al. [1]

The size of the generated head and body regions is set according to the estimated pedestrian height h_t. The step size between regions is set to be dependent on the pedestrian distance to the cars. If the distance is bigger, then the step size is smaller to obtained a finer search. If the pedestrian is close to the car, the step size is bigger because the body parts are bigger and more area has to be covered. In the end, the variation of the step size should produce the same amount of body part regions.

Figure 4.15 shows an example of region generation process. Figure 4.15 a) presents the disparity map with the estimated head and body centers ($\boldsymbol{h}_c$, $\boldsymbol{b}_c$) (black crosses) and the gravity line (the blue line), which guides the process of region generation, in three situations. Figure 4.15 b) presents the corresponding generated regions for head (represented with blue) and body (represented with green).

4.6.2 Location and single-frame continuous orientation estimation from multiple regions

Once the regions are generated, it is desirable to find out what is the continuous orientation $\omega = [\omega^H, \omega^B]$ given the observed image $\boldsymbol{z} = [\boldsymbol{z}^H, \boldsymbol{z}^B]$ at each moment of time t: $p(\omega_t|\boldsymbol{z}_t)$. Assuming conditional independence between the head and body observations, two terms are obtained:

$$p(\omega_t|\boldsymbol{z}_t) = p(\omega_t^H|\boldsymbol{z}_t^H)p(\omega_t^B|\boldsymbol{z}_t^B) \tag{4.2}$$

The superscripts H and B refer to the head and to the body, respectively. Because both terms in Eq. 4.2 are computed in the same way, the superscripts H, B and the time index t will be dropped when referring to either term.

4.6.2.1 From discrete to continuous orientations

The desired orientation $\omega \in \mathbb{R}$ is a continuous value in the domain $[0°, 360°)$, but only responses of orientation detectors for discrete orientation classes Ω (which are approximations of cluster-specific posterior probabilities $p(\Omega|z)$) are available. Therefore, the continuous orientation ω is defined in terms of the class Ω of the current $\boldsymbol{z}$ in the following way:

$$p(\omega|\boldsymbol{z}) = \sum_{\Omega} p(\Omega|\boldsymbol{z})p(\omega|\Omega) \tag{4.3}$$

For each discrete class o, $p(\omega|\Omega = o)$ expresses the probabilistic relationship between the continuous orientation angle ω and the discrete class Ω and it is modeled using a *von Mises* distribution,

$$p(\omega|\Omega = o) = \mathcal{V}(\omega; c_o, k_o), \tag{4.4}$$

with c_o and k_o the mean and concentration of the distribution for orientation class o. This relation is independent of the image $\boldsymbol{z}$. The *von Mises* $\mathcal{V}(\cdot; \omega, \kappa)$ is an analogue of the normal distribution for the circular domain, with mean angle ω and concentration κ. A higher concentration means more mass around the mean. It reduces to a circular uniform distribution when $\kappa = 0$. Now the only term that needs to be defined is $p(\Omega|\boldsymbol{z})$, which is the probability of having an orientation class (instead of a continuous angle), given the observed image.

4.6.2.2 Posterior with auxiliary variables

To obtain the location and orientation estimates two auxiliary variables, R and V, are first introduced to express $p(\Omega|\boldsymbol{z}, R, V)$. In Section 4.6.2.3 $p(\Omega|\boldsymbol{z})$ will be expressed in terms of this extended posterior. The variable $R = r$, $r = \{1 \dots N\}$, indicates which region $z^{(r)}$ of the possible regions in $\boldsymbol{z}$ fits the sought head / body. As a consequence, it also specifies that

all other regions do not fit the head / body. Additionally, the Boolean variable $V = v$, with $v \in \{0, 1\}$, indicates whether there exist a head / body in any of the N regions at all ($V = 1$), or whether none of the regions contain it ($V = 0$).

Using the auxiliary variables, the orientation posterior can be expressed as being proportional to the detector responses as

$$p(\Omega = o|z^{(s)}, R = r, V) \propto \begin{cases} f_o(z^{(s)}), & \text{if } s = r \wedge V = 1 \\ f_-(z^{(s)}), & \text{otherwise.} \end{cases} \tag{4.5}$$

As mention in Section 4.1, $f_o(z^{(s)})$ and $f_-(z^{(s)})$ are values in $[0, 1]$. It is assumed that all candidate regions $z^{(s)}$ are conditionally independent given the orientation class Ω, the correct region R and the fact that there is a head / body in one of the regions V or only given the last two, so the complete data posterior over $\boldsymbol{z}$ can be approximated as in Eq. 4.6.

The region independence assumption is a strong one since regions overlap and the head / body might be well represented in more than one generated region.

$$\begin{aligned} p(\Omega|\boldsymbol{z}, R, V) &= p(\Omega|z^{(1)}, z^{(2)}, \ldots, z^{(N)}, R, V) = \frac{p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}|\Omega, R, V)p(\Omega, R, V)}{p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}, R, V)} \\ &= \frac{\prod_{s\in\overline{1,N}} p(z^{(s)}|\Omega, R, V)p(\Omega, R, V)}{p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}, R, V)} = \frac{\prod_{s\in\overline{1,N}} \frac{p(\Omega|z^{(s)},R,V)p(z^{(s)},R,V)}{p(\Omega,R,V)} p(\Omega, R, V)}{p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}, R, V)} \\ &= \frac{\prod_{s\in\overline{1,N}} [p(\Omega|z^{(s)}, R, V)p(z^{(s)}, R, V)]}{p(\Omega, R, V)^{N-1} p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}, R, V)} = \frac{\prod_{s\in\overline{1,N}} p(\Omega|z^{(s)}, R, V) \prod_{s\in\overline{1,N}} p(z^{(s)}, R, V)}{p(\Omega, R, V)^{N-1} p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}, R, V)} \\ &= \frac{\prod_{s\in\overline{1,N}} p(\Omega|z^{(s)}, R, V) \prod_{s\in\overline{1,N}} [p(z^{(s)}|R, V)p(R, V)]}{p(\Omega, R, V)^{N-1} p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}|R, V)p(R, V)} \\ &= \Big[\frac{p(R, V)}{P(\Omega, R, V)}\Big]^{N-1} \frac{\prod_{s\in\overline{1,N}} p(\Omega|z^{(s)}, R, V) \prod_{s\in\overline{1,N}} p(z^{(s)}|R, V)}{p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}|R, V)} \\ &= \Big[\frac{1}{P(\Omega|R, V)}\Big]^{N-1} \frac{\prod_{s\in\overline{1,N}} p(\Omega|z^{(s)}, R, V)p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}|R, V)}{p(z^{(1)}, z^{(2)}, \ldots, z^{(N)}|R, V)} \\ &= \Big[\frac{1}{P(\Omega|R, V)}\Big]^{N-1} \prod_{s\in\overline{1,N}} p(\Omega|z^{(s)}, R, V) \propto \prod_{z^{(s)}\in\boldsymbol{z}} p(\Omega|z^{(s)}, R, V) \end{aligned} \tag{4.6}$$

Intuitively, one would expect that all orientation classes are equally likely when the head / body is not contained in any region and therefore unobserved. This property indeed follows from Eq. 4.5 and 4.6 since the orientation is independent on the selected region R when $V = 0$,

$$p(\Omega|\boldsymbol{z}, R, V = 0) = p(\Omega|\boldsymbol{z}, V = 0) \propto \prod_{z^{(s)} \in \boldsymbol{z}} f_{-}(z^{(s)}). \tag{4.7}$$

4.6.2.3 Removing the auxiliary variables

To remove the introduced variables, first, an optimal value $\hat{r}$ for the region indicator R has to be selected. Assuming that there is a head ($V^H = 1$) and a body ($V^B = 1$) in one of the head and body regions, the most probable head and body region configuration $\hat{r} = [\hat{r}^H, \hat{r}^B]$ is selected by:

$$\begin{aligned}
\hat{r} &= \underset{R^H, R^B}{\operatorname{argmax}}\, p(R^{H,B}|\boldsymbol{z}^{H,B}, V^{H,B} = 1) \\
&= \underset{R^H, R^B}{\operatorname{argmax}} \Big[\sum_{\Omega^{H,B}} p(\Omega^{H,B}, R^{H,B}|\boldsymbol{z}^{H,B}, V^{H,B} = 1) \Big] \\
&= \underset{R^H, R^B}{\operatorname{argmax}} \Big[\sum_{\Omega^{H,B}} p(\Omega^{H,B}|R^{H,B}, \boldsymbol{z}^{H,B}, V^{H,B} = 1) p(R^{H,B}|\Omega^B, \boldsymbol{D}, V^{H,B} = 1) \Big].
\end{aligned} \tag{4.8}$$

The head orientation is considered to be independent of the body orientation, the generated body regions ($\boldsymbol{z}^B$), the correct body region ($R^B = r^B$) and even the fact that there is a body or not (V^B), accounting in this way for occlusions. These assumptions are also valid for the body with respect to the head and the following equation is obtained:

$$\hat{r} = \underset{R^H, R^B}{\operatorname{argmax}} \Big[\sum_{\Omega^H} p(\Omega^H|\boldsymbol{z}^H, R^H, V^H = 1) \sum_{\Omega^B} p(\Omega^B|\boldsymbol{z}^B, R^B, V^B = 1) p(R^{H,B}|\Omega^B, \boldsymbol{D}, V^{H,B} = 1) \Big] \tag{4.9}$$

$p(R^H, R^B|\Omega^B, \boldsymbol{D}, V^{H,B} = 1)$ will be described in detail in Section 4.6.3. It introduces prior knowledge about the joint region configuration of head and body from disparity data $\boldsymbol{D}$ and from a PS model [37] dependent on the body orientation. A graphical illustration of this process is presented in Figure 4.16. Inferred hidden state variables are unshaded and observation

variables are shaded.

The term $p(\Omega|\boldsymbol{z})$, without the auxiliary variables, is now obtained by fixing R to $\hat{r}$ and integrating out the variable V:

$$\begin{aligned} p(\Omega|\boldsymbol{z}) &= \sum_{v\in\{0,1\}} p(\Omega|\boldsymbol{z}, V=v, R=\hat{r})p(V=v) \\ &= p(\Omega|\boldsymbol{z}, V=1, \hat{r})p(V=1) + p(\Omega|\boldsymbol{z}, V=0)p(V=0). \end{aligned} \tag{4.10}$$

The fixed Bernoulli distribution $p(V)$ can be used to incorporate more scene prior knowledge on the occurrences of false positives (e.g. set high probability for $p(V=0)$ if false positives are common), or just set to uniform to rely on the observation only.

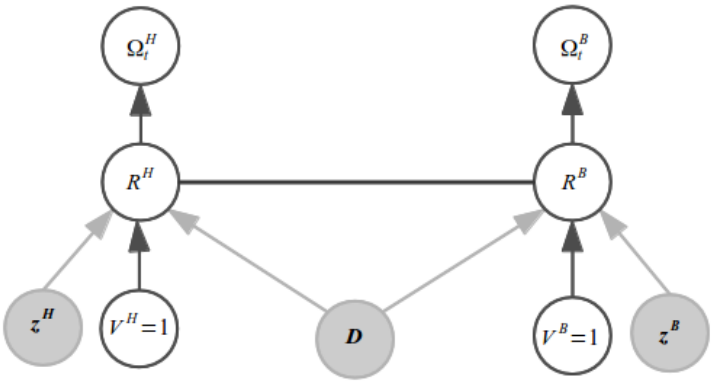


Figure 4.16: Graphical model showing the process of region selection and single-frame orientation estimation

Using Eq. 4.5 and 4.6 to expand 4.10 further, it can be seen that the term can be efficiently evaluated up to a constant factor:

$$p(\Omega = o|\boldsymbol{z}) \propto f_o(z^{(\hat{r})})p(V=1) + f_-(z^{(\hat{r})})p(V=0). \tag{4.11}$$

Eq. 4.11 shows that the stronger the background detector response f_- is (relative to the orientation detector f_o), the higher the weight of the second term is, and therefore the smaller the relative differences between the probabilities of the different orientation classes is. In the extreme case where only f_- gives a very strong response, the term is the same for all orientations

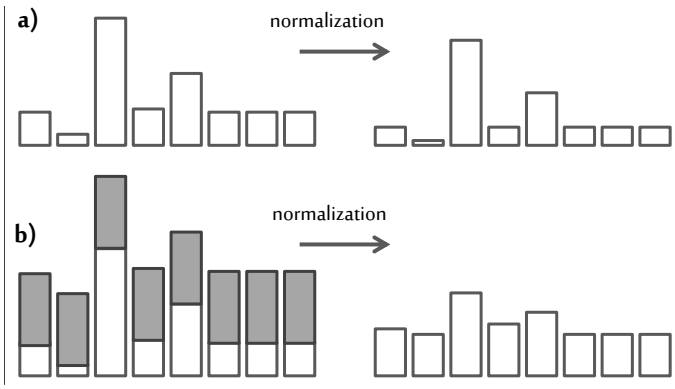


Figure 4.17: Normalization with and without the addition of the background detector

(no information on the true orientation was gained at this time step).

In the case of the discrete space, Figure 4.17 illustrates this property. The addition of the background term (dark gray part) in b) results in a more uniform distribution after normalization, i.e there is more uncertainty to what class the observation belongs.

4.6.3 Spatial prior over the body parts regions

This section explains how the prior from Eq. 4.9 can be obtained. This prior can be factored into:

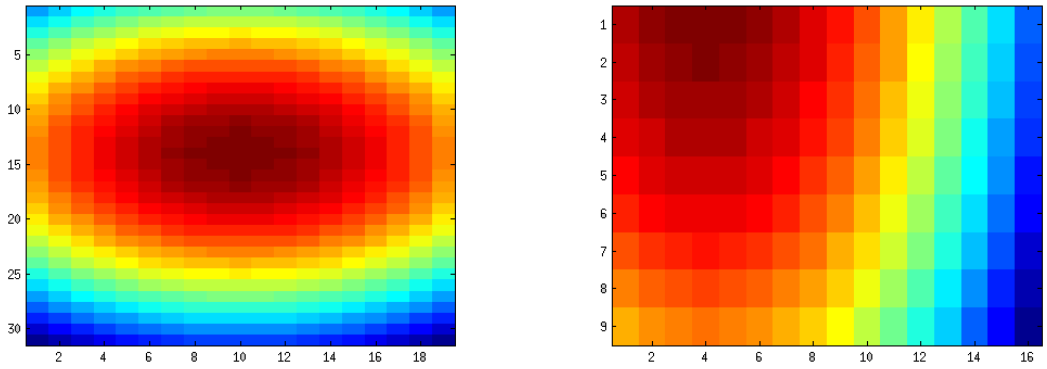
$$p(R^H, R^B|\Omega^B, \boldsymbol{D}, V^{H,B} = 1) \propto$$
$$p(\boldsymbol{h}_c(\boldsymbol{D})|R^H, V^H = 1) \times p(\boldsymbol{b}_c(\boldsymbol{D})|R^B, V^B = 1) \times p(R^H, R^B|\Omega^B, V^{H,B} = 1), \quad (4.12)$$

where $\boldsymbol{h}_c(\boldsymbol{D})$ and $\boldsymbol{b}_c(\boldsymbol{D})$ represent the head and body mean pixel locations as functions of the disparity $\boldsymbol{D}$, giving an estimate of the head and body positions.

The probability of the head region is then modeled with

$$p(\boldsymbol{h}_c(\boldsymbol{D})|R^H = r^H, V^H = 1) = \mathcal{N}(\boldsymbol{h}_c(\boldsymbol{D}); \boldsymbol{\mu}(r^H), \boldsymbol{C}^H). \quad (4.13)$$

$\boldsymbol{\mu}(r^H)$ denotes the center (in image coordinates) of a given head region r^H, while $\boldsymbol{C}^H$ denotes the corresponding covariance. The probability $p(\boldsymbol{b}_c(\boldsymbol{D})|R^B = r^B, V^B = 1)$ of a body region is



a) b)

Figure 4.18: Region probability: a) head and b) body for the first set images in Figure 4.19

.

a) b)

Figure 4.19: Examples of region probabilities and selected head & body configuration – adapted from Flohr et al. [1]

.

modeled in the same way. An example such region probabilities can be found in Figure 4.18.

Up to this point there is no relationship between the head and body locations. Experimental evidence (see Figure 4.19) showed that not all the time the highest detector responses also give the correct location of the head. Because a better localization would also result into a better orientation estimation, modeling such constrains would bring many benefits.

These constrains come in the form of a joint spatial prior $p(R^H, R^B|\Omega^B, V^{H,B} = 1)$, dependent on the body orientation, similar with a *Pictorial Structure* model [37]:

$$p(R^H = r^H, R^B = r^B|\Omega^B = o^B, V^{H,B} = 1) = \mathcal{N}(\boldsymbol{l}^D(r^H, r^B); \boldsymbol{\mu}^D_{o^B}, \boldsymbol{C}^D_{o^B}). \tag{4.14}$$

$\boldsymbol{l}^D(r^H, r^B)$ denotes the distance between head and body region centers relative to the width of the body region. The parameters $\boldsymbol{\mu}^D_{o^B}$ and $\boldsymbol{C}^D_{o^B}$ for the PS between head and body region for each discrete orientation were learned from training data. These learned parameters are illustrated in Figure 4.20, which shows for each orientation in the training set the histogram of the normalized distances between the head and body centers. Only the parameters for $\{0, 45, 90, 135, 180, 225, 270, 315\}$ were used (because only for those orientation there are trained detectors).

Figure 4.19 a) shows the region probabilities. It can be observed that the ambiguities are efficiently resolved. By combining head and body localization with the PS model (Figure 4.19 b)) the correct head and body configuration is chosen.

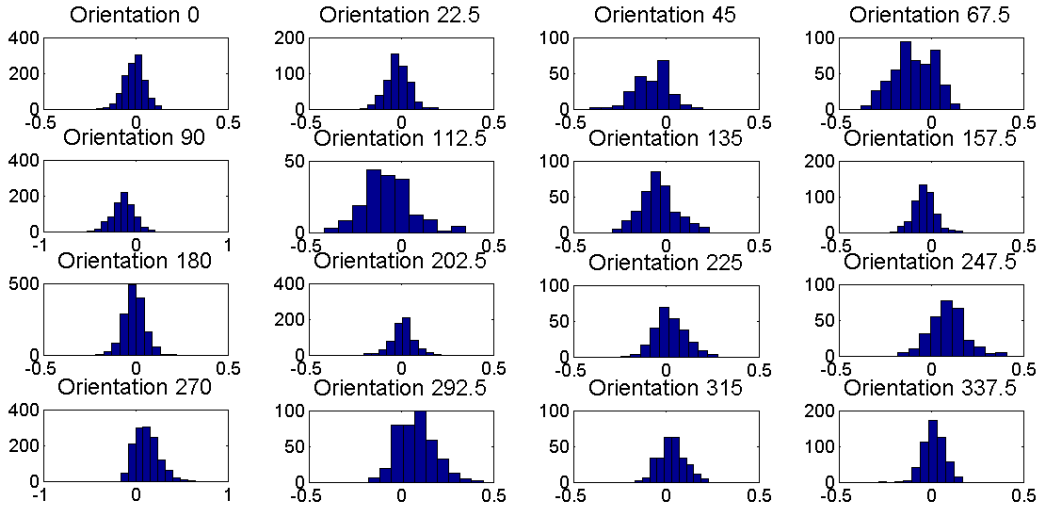


Figure 4.20: Deformation model: learned parameters

Chapter 5

Orientation tracking

5.1 Motivation

The orientation estimation that comes from the detection phase can be noisy, changing from one time step to another with a significant amount. Empirical evidence, obtained by observing multiple pedestrians in real urban traffic videos, shows that these sudden orientation modifications are not natural and the pedestrian changes the head and body orientation more smoothly.

Orientation tracking is used to model this smoothness into the orientation estimation. Keeping information about the head / body orientation over time and using this prior belief regarding former time steps can bring a number of benefits.

One benefit of tracking is the filtering of the noisy estimations. The challenge here is to obtain a smooth estimation, but still keep the capability of sensing orientation changes fast enough. For example, in the case of reasonably certain measurements, where in the previous time step there was an orientation pointing to the left, the current estimate cannot point to the right. This has to be smoothed out, but cases where smaller transitions are made should still be possible in the model. Of course, cases with big transitions should be allowed if at the previous time steps only uncertain measurements were available.

Another benefit tracking the orientation is the capability of choosing the best estimation when

the evidence is equally strong for two directions. For example, if the current detection estimate gives a multi-modal distribution (e.g. there is a confusion between front and back), using the prior belief regarding previous time steps, the mode closer to the previous orientation should get a bigger importance (but information about the other mode should also be kept).

5.2 Particle Filtering motivation and background

To integrate the head / body orientation over time the particle filter algorithm was chosen. Particle filters (PF) [3] are algorithms that estimate in an on-line manner the posterior density of a state space by using the Bayesian recursion equations. To do this they make use of a set of particles. Their main advantage is that they do not make any restrictive assumptions on the dynamic model of the state space. The state space can be non-linear and can take any desired form. The disadvantage of a standard PF is that it does not scale well when they are applied to high-dimensional systems, but there are methods to reduce this drawback.

Denoting with x the state and with z the observation process, the objective of the particle filter is to estimate the sequence of $\boldsymbol{X}_t = \{x_1 \dots x_t\}$, given the sequence of measurements / observations $\boldsymbol{Z}_t = \{z_1 \dots z_t\}$.

In particle filtering, the following two assumptions are made:

1. $x_1, \dots, x_t$ is a first order Markov process: $p(x_t|x_{t-1}, x_{t-2}, \dots, x_1) = p(x_t|x_{t-1})$;

2. The observations $z_1, \dots, z_t$ are conditionally independent if $x_1, \dots, x_t$ are known:

$$p(\boldsymbol{Z}_t|\boldsymbol{X}_t) = \prod_{i=1}^{t} p(z_i|x_i) \tag{5.1}$$

   The observation process is therefore defined by specifying the conditional probability $p(z_t|x_t)$ at each time t. In practical examples this can be time independent.

An example of this is the following:

$$x_t = g(x_{t-1}) + w_t \tag{5.2}$$

$$z_t = h(x_t) + v_t \tag{5.3}$$

where w_t and v_t are mutually independent with known probability density functions and $g(\cdot)$ and $h(\cdot)$ are known functions. $g(\cdot)$ gives the drift of the particles, while w_t gives their diffusion. Figure 5.1 presents the three phases: drift due to the deterministic component of the object dynamics; diffusion due to the random component; reactive reinforcement due to observations. If $g(\cdot)$ and $h(\cdot)$ are linear and w_t and v_t are Gaussian, the Kalman filter is approximated.

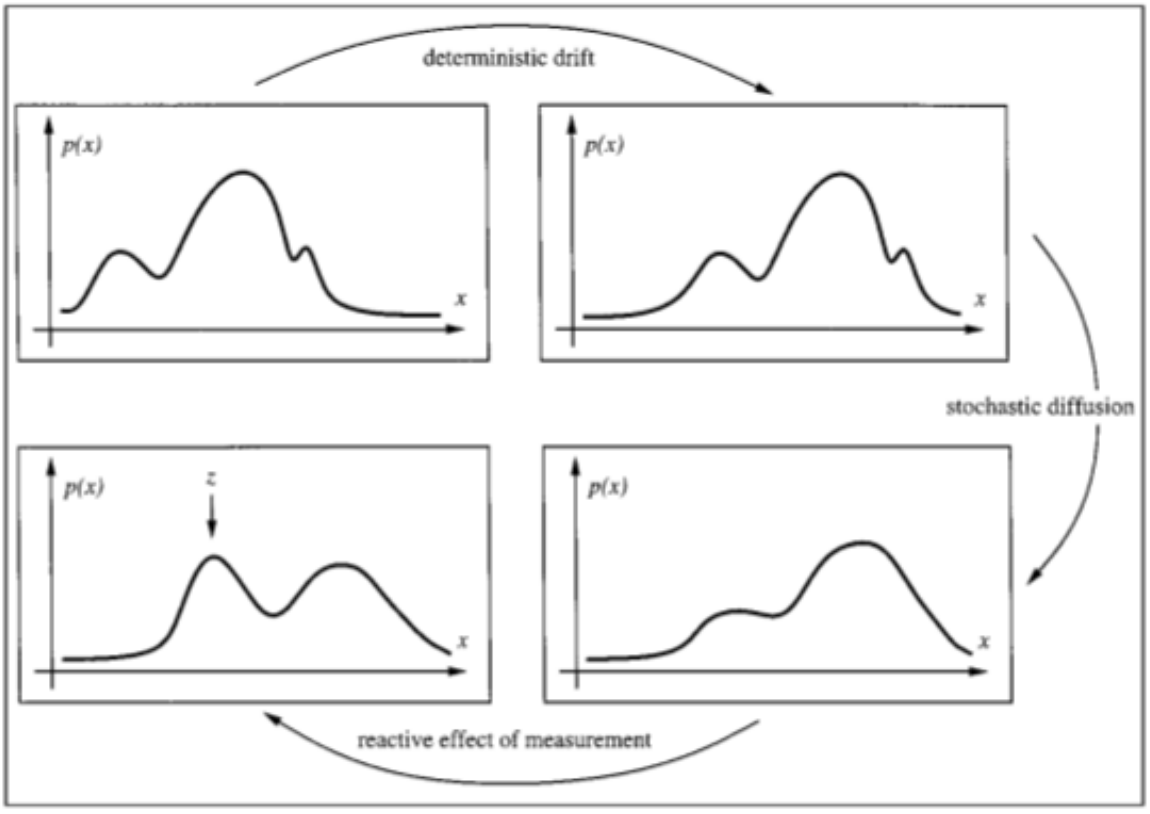


Figure 5.1: Probability density propagation as it occurs over a discrete time-step – re-used from [3]

All the Bayesian estimates of x_t come from the posterior distribution $p(x_t|\boldsymbol{Z}_t)$. This keeps all the information about the state at time t that is available from the whole observation sequence.

Propagation of the state density over time is given by the following formula:

$$p(x_t|\boldsymbol{Z}_t) = k_t p(z_t|x_t) p(x_t|\boldsymbol{Z}_{t-1}) \tag{5.4}$$

where k_t is a normalization constant that does not depend on x_t and

$$p(x_t|\boldsymbol{Z}_{t-1}) = \int_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|\boldsymbol{Z}_{t-1}) \tag{5.5}$$

is a prediction taken from posterior $p(x_{t-1}|\boldsymbol{Z}_{t-1})$ of the previous time-step. The multiplication in 5.4 by the observation density $p(z_t|x_t)$ applies the reactive effect expected from observations (Figure 5.1).

To be able to deal with non-Gaussian observations, PF uses a version of factored sampling extended to deal with temporal sequences, called condensation algorithm.

The factored sampling algorithm generates a random variate x from a distribution that approximates the posterior $p(x|z)$. First a sample-set $\{s^{(1)} \dots s^{(N)}\}$ is generated and then an index $n \in \{1, \dots, N\}$ is chosen with probability π_n, where:

$$\pi_n = \frac{p(z|s^{(n)})}{\sum_{j=1}^{N} p(z|s^{(j)})} \tag{5.6}$$

In the condensation algorithm each time-step is an iteration of factored sampling. The output of an iteration is a weighted, time-stamped sample-set $\{s_t^{(n)}, n = 1, \dots, N\}$ with weights $\pi_t^{(n)}$, that represent an approximation of the posterior density at time t, $p(x_t|\boldsymbol{Z}_t)$.

The process must begin with a prior density from which the particles are sampled first time. The effective prior for time-step t is $p(x_t|Z_{t-1})$. This is derived from the sample set $\{(s_{t-1}^{(n)}, \pi_{t-1}^{(n)}), n = 1, \dots, N\}$ that represents the output from the previous time-step $p(x_{t-1}|\boldsymbol{Z}_{t-1})$, to which the prediction is applied.

The number of particles N has a major influence on the performance of the algorithm. Having a large N gives a good approximation of the posterior density, but also reduces the speed of

prediction.

In particle filtering algorithms, weight disparity, that leads to weight collapse, is a common issue. However it can be mitigated by including a resampling step before the weights become too uneven. In the resampling step, the particles with negligible weights are replaced by new particles in the proximity of the particles with higher weights. This technique is called Sequential Importance Resampling (SIR).

5.3 Independent tracking

The first idea was to integrate the head and body orientations separately from each other. Both were done in exactly the same way, so in the following paragraphs no distinction between the two body parts will be made.

Let $\boldsymbol{z}_{1:t}$ denote all observations up to and including time t. The state space ω is given by the body part orientation. The belief about the state ω at time t, after observing $\boldsymbol{z}_{1:t}$, is obtained through filtering, being represented by represented by $p(\omega_t|\boldsymbol{z}_{1:t})$.

The filter performs the following two steps every time instance. First, a prediction is made based on the earlier observations,

$$p(\omega_t|\boldsymbol{z}_{1:t-1}) = \int p(\omega_t|\omega_{t-1})p(\omega_{t-1}|\boldsymbol{z}_{1:t-1})d\omega_{t-1} \tag{5.7}$$

where $p(\omega_{t-1}|\boldsymbol{z}_{1:t-1})$ keeps the information available at the previous time step and $p(\omega_t|\omega_{t-1})$ is the dynamical model.

Second, an update has to be made to incorporate the new evidence $\boldsymbol{z}_t$ in the prediction,

$$p(\omega_t|\boldsymbol{z}_{1:t}) \propto p(\omega_t|\boldsymbol{z}_t)p(\omega_t|\boldsymbol{z}_{1:t-1}) \tag{5.8}$$

where $p(\omega_t|\boldsymbol{z}_t)$ is the observation model.

Since exact inference is intractable, a PF is used for approximate inference. The two elements, the dynamical model and the observation model, have to be defined. First, the dynamical model of the PF is given by the following equation:

$$p(\omega_t|\omega_{t-1}) = \mathcal{V}(\omega_t; \omega_{t-1}, \kappa) \tag{5.9}$$

where κ is the concentration parameter for the *von Mises* distribution. Equation 5.9 models the assumption that the current orientation is distributed around the previous orientation, assuring temporal consistency. Second, the observation model is the probability given by the corresponding body part detection step $p(\omega_t|\boldsymbol{z}_t)$ [41].

For a new pedestrian track, a filter is initialized by sampling orientations ω_1^H and ω_1^B from an uniform circular distribution, performing the update with $\boldsymbol{z}_1$.

5.4 Joint tracking

5.4.1 Motivation

The independent tracking presented in the previous section solves the problem of filtering the noise and smoothing the orientation estimates. But the results can still be improved. The anatomical laws constrain the head and body orientations. For example, the head orientation should be around the body orientation (e.g. differences bigger than 90° are very improbable). By incorporating such constrains the head / body orientation estimates can become more robust, by eliminating cases where these two orientations point into directions that are anatomically impossible.

Moreover, if it is assumed that the body orientation is similar with the velocity direction (which is true if the pedestrian is moving and he is not walking backwards), an extra source of information can be used to improve the body orientation estimation.

5.4.2 Procedure

Again, let $\boldsymbol{z}_{1:t}$ denote all the observations up to and including time t. $\dot{\boldsymbol{x}}_{1:t}$ represents the corresponding pedestrian velocities provided by the external tracker. The state space, this time, is composed from two variables $\boldsymbol{\omega}_t = [\omega_t^H, \omega_t^B]$. Again, the belief about the state $\boldsymbol{\omega}$ at time t, after observing $\boldsymbol{z}_{1:t}$ and $\dot{\boldsymbol{x}}_{1:t}$, is obtained through filtering, being represented by $p(\boldsymbol{\omega}_t|\boldsymbol{z}_{1:t}, \dot{\boldsymbol{x}}_{1:t})$.

For each time instance the filter performs the following two steps. First, a prediction is made given the earlier observations,

$$p(\boldsymbol{\omega}_t|\boldsymbol{z}_{1:t-1}, \dot{\boldsymbol{x}}_{1:t}) = \int p(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1}, \dot{\boldsymbol{x}}_t)p(\boldsymbol{\omega}_{t-1}|\boldsymbol{z}_{1:t-1}, \dot{\boldsymbol{x}}_{1:t-1})d\boldsymbol{\omega}_{t-1} \tag{5.10}$$

where $p(\boldsymbol{\omega}_{t-1}|\boldsymbol{z}_{1:t-1}, \dot{\boldsymbol{x}}_{1:t-1})$ is the posterior from the previous time step, keeping the information available at the previous time steps, and $p(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1}, \dot{\boldsymbol{x}}_t)$ is the dynamical model.

Second, an update has to made to incorporate new evidence $\boldsymbol{z}_t$ in the prediction,

$$p(\boldsymbol{\omega}_t|\boldsymbol{z}_{1:t}, \dot{\boldsymbol{x}}_{1:t}) \propto p(\boldsymbol{\omega}_t|\boldsymbol{z}_t)p(\boldsymbol{\omega}_t|\boldsymbol{z}_{1:t-1}, \dot{\boldsymbol{x}}_{1:t}) \tag{5.11}$$

where $p(\boldsymbol{\omega}_t|\boldsymbol{z}_t)$ is the observation model, similar with [41].

Figure 5.2 presents the directed graphical model. Inferred hidden state variables are unshaded and observation variables are shaded. Because again exact inference is not tractable, a PF is used for approximation. The PF represents the posterior distribution by a set of particles in the state space, which facilitates the use of non-linear and multi-modal dynamic model. Furthermore, one only needs to evaluate the unnormalized probability density of the observation model. For a new pedestrian track, a filter is initialized by sampling orientations ω_1^H and ω_1^B from an uniform circular distribution, performing the update step with $\boldsymbol{z}_1$.

The dynamic model for the head and body orientations is

$$p(\boldsymbol{\omega}_t|\boldsymbol{\omega}_{t-1}, \dot{\boldsymbol{x}}_t) = p(\omega_t^H|\omega_{t-1}^H, \omega_t^B)p(\omega_t^B|\omega_{t-1}^B, \omega_{t-1}^H, \dot{\boldsymbol{x}}_t). \tag{5.12}$$

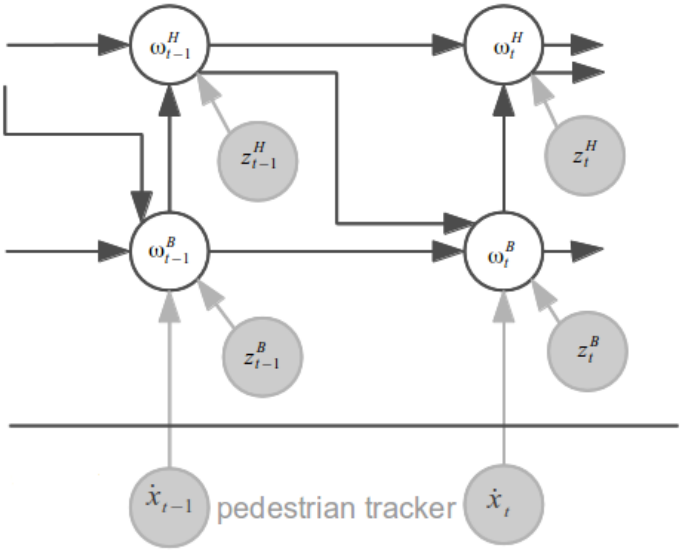


Figure 5.2: Dynamic network model, showing used constrains between head (ω_t^H) and body orientation ω_t^B – adapted from Flohr et al. [1].

The dynamic model is illustrated in Figure 5.2 and it is composed from two terms: one that models the dynamics of the head orientation and one that models the dynamics of the body orientation.

Similar to [16], the head orientation at the current time step is constrained on the head orientation of the previous time step and on the current body orientation with

$$p(\omega_t^H|\omega_{t-1}^H, \omega_t^B) = \alpha_{hh}\mathcal{V}(\omega_t^H; \omega_{t-1}^H, \kappa_{hh}) + (1 - \alpha_{hh})\mathcal{V}(\omega_t^H; \omega_t^B, \kappa_{hb}), \tag{5.13}$$

where κ_{hh} and κ_{hb} are concentration parameters for the *von Mises* distribution.

The first term in Eq. 5.13 models the case that the current head orientation is distributed around the previous head orientation (temporal consistency). The second term covers the (possible alternative) case where the head has moved to a similar orientation as the body. The balance between temporal consistency and the assumption that the head orientation is around the body orientation is given by the weight α_{hh}.

The body orientation is conditioned on the body and head orientation of the previous time step

and on the current pedestrian velocity:

$$p(\omega_t^B|\omega_{t-1}^B, \omega_{t-1}^H, \dot{\boldsymbol{x}}_t) = \tag{5.14}$$
$$\alpha_{bb}\mathcal{V}(\omega_t^B; \omega_{t-1}^B, \kappa_{bb}) + \alpha_{bh}\mathcal{V}(\omega_t^B; \omega_{t-1}^H, \kappa_{bh})$$
$$+ (1 - \alpha_{bb} - \alpha_{bh})\mathcal{V}(\omega_t^B; ang(\dot{\boldsymbol{x}}_t), \kappa_{bv})$$

The angle of the velocity vector is represented with $ang()$. $\alpha_{bb}, \alpha_{bh} \in [0, 1]$ (with $\alpha_{bb} + \alpha_{bh} \leq 1$) denote the weighting factors for the terms.

The first term in Eq. (5.14) expresses that the body orientation is typically around its previous orientation (temporal consistency). Furthermore, the cases when the body orientation changes to where the pedestrian is looking are captured by the second term. The last term expresses that the body orientation might also be aligned with the direction of motion. κ_{bb}, κ_{bh} and κ_{bv} denote concentration parameters. Concentration κ_{bv} however depends, similar to [32], on the velocity magnitude $||\dot{\boldsymbol{x}}_t||$, but also on the external track state (T_S), with $T_S \in \{0$ (initialized), 1 (preliminary), 2 (confirmed)$\}$, and on the external track probability (T_P):

$$\kappa_{bv} = \begin{cases} \kappa_v \cdot (||\dot{\boldsymbol{x}}_t|| - t_v)^2 T_P T_S & \text{if } ||\dot{\boldsymbol{x}}_t|| > t_v \;\&\; T_P > t_p, \\ 0 & \text{otherwise.} \end{cases} \tag{5.15}$$

Here t_v denotes a threshold for the velocity magnitude and t_p is a threshold for the track probability. κ_v is an initial concentration parameter.

Similar with [41], the observation models for the head and body orientations are assumed to be independent:

$$p(\boldsymbol{\omega}_t|\boldsymbol{z}_t) = p(\omega_t^H|\boldsymbol{z}_t^H)p(\omega_t^B|\boldsymbol{z}_t^B|). \tag{5.16}$$

$p(\omega_t^H|\boldsymbol{z}_t^H)$ and $p(\omega_t^B|\boldsymbol{z}_t^B)$ are the continuous distributions that come from the orientation detection step, as explained in Chapter 4.6.

Chapter 6

Orientation estimation for multiple people and optimizations for the use in the car

For using the framework in the car in true urban traffic it has to work in near real-time and handle multiple pedestrians.

Near real-time performance is obtained through module design and implementation. The resized dimensions of the head / body regions and the number of PF particles were carefully chosen to assure satisfactory recognition and time performance. Modules like head and body detection or the PS constrains are parallelized using a multi-core architecture. The head / body pools were split in different chunks, which were fed to different detection threads. The computation of the probability for different body parts configuration is tackled in the same way. A first step to obtain better performance could be to move the implementation to CUDA and benefit from modern graphic adapters with cores, but this was not done in the current work.

Tracking the orientations of multiple persons is handled by starting a new orientation estimation instance for each detected and tracked pedestrian. By doing so, the speed performance would deteriorate significantly with the number of tracked pedestrians, so the solution is to keep

estimating only for the closest pedestrian to the car, as it is the most dangerous one. More sophisticated methods for pedestrian selection could also be employed, for example based on the movement direction or street priors.

The time evaluation for the chosen solution can be found in Section 7.7.

Chapter 7

Experiments

7.1 Training and testing datasets

Two datasets were used to evaluate the system: one for training and validation and one for performance evaluation.

The first one consists of 9300 manually contour labeled pedestrian samples, with a minimum / maximum / mean height of 69 / 344 / 122 pixels, obtained from 6389 images. These images also provided background samples, used to obtain a more robust localization of the head / body. Half of the background samples were extracted from false positive pedestrian detections in the area of the sought head / body. The other half was sampled around the head / body of true positive detection with a maximum overlap of 25% to the true head / body. The total amount of background samples was set based on the number of samples in the other aggregated orientation classes. The dataset obtained in this way was split into two parts. The fist part, counting for 90% of the samples, was used to train different head / body detection architectures. The second part, counting for the rest 10% of the samples, was used to validate these detection architectures before choosing the best one to include into the framework. Examples of training images can be found in Figure 7.1: row a) shows head samples for each one of the eight orientation classes; row b) shows the same thing for the body; rows c) and d) show non-head / body.

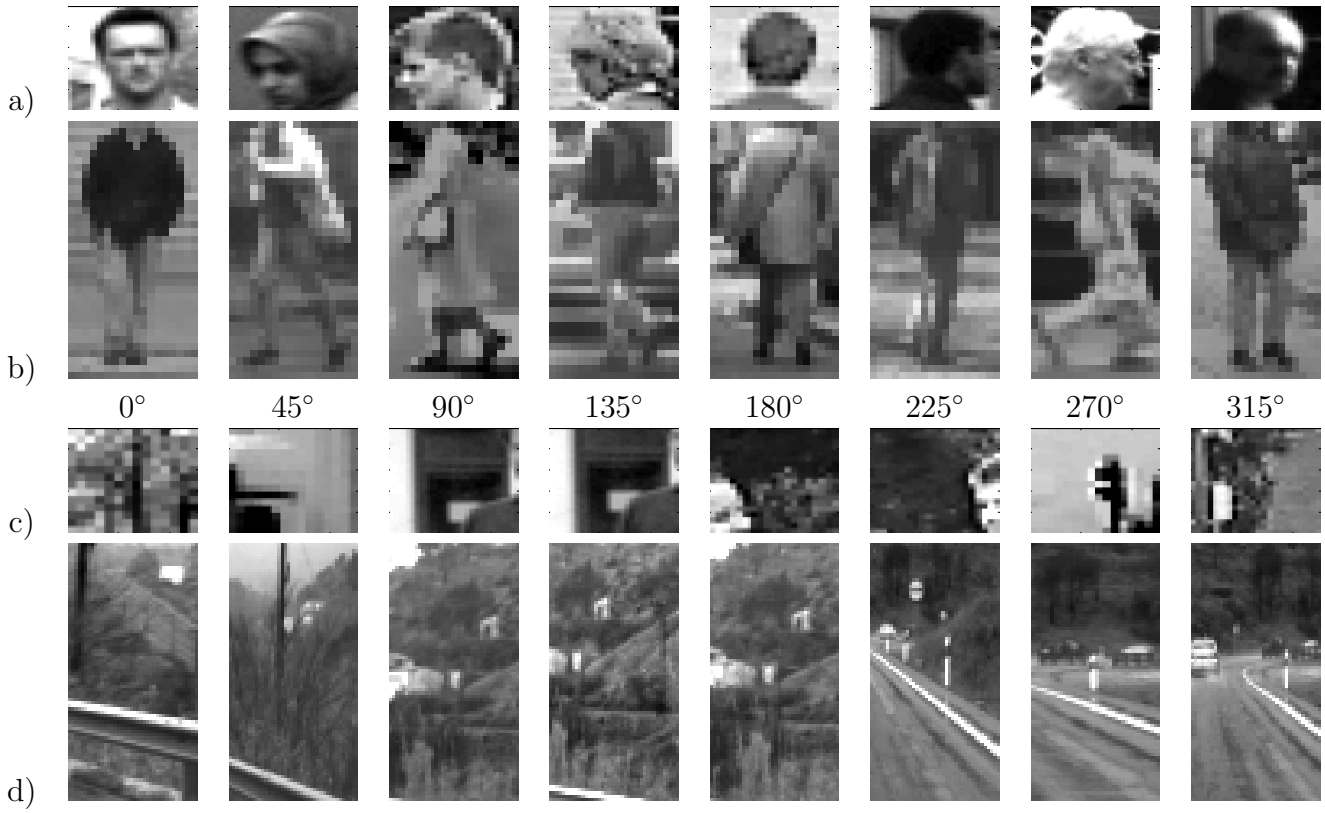


Figure 7.1: Examples of a) head and b) body training images in 8 aggregated orientation classes; c) and d) present non-head / body samples

The second dataset (the test set) consists of 60 image sequences, containing multiple pedestrians in different traffic situations (waiting / stopping, crossing or walking longitudinally with respect to the vehicle). Ground truth was obtained by manual labeling (bounding box location and head / body orientation labels per frame). The input to the framework (according to the shaded modules of Figure 3.1) were the bounding boxes provided by a state-of-art HOG/linSVM pedestrian detector [6] and a Kalman filter.

In each frame, an estimated pedestrian location is associated with a ground truth label when the distance between them is smaller than a threshold. This threshold is set according to a percentage of the Euclidean distance of the ground truth label to the camera. A different percentage of 8% and 12% for lateral and longitudinal directions is used, since uncertainty in lateral direction is in general smaller. For the evaluation of the orientation estimation performance, only estimated tracks that follow more than 80% of their duration a particular ground truth track (several estimated tracks can correspond to a single ground truth track)

are considered. All other estimated tracks are regarded as false positives and they are not used in the evaluation. Furthermore, only track segments with a maximum lateral / longitudinal distance of 5 m / 35 m to the camera are included. Doing so, 65 "valid" estimated tracks with 3133 samples are obtained.

7.2 Parameters settings

The architectures used for the body and the head localization and orientation estimation were presented in Sections 4.5.1 and 4.5.2. Here only the used parameters are presented.

All the detectors were trained using $N_{RF} = 48$ branches. The size of the receptive fields was fixed to 5×5 and they were shifted at a step size of two neurons.

The head was extracted at a fixed aspect ratio of 15% of the whole body from top of the contour labeled shape. Two dimensions to which the head samples were scaled were investigated: 16×16 pixels and 24×24 pixels.

The body detectors use the lower 85% of the whole body to make sure that the head part is ignored and does not affect the body orientation estimate. Also two dimensions to which the body samples were scaled were investigated: 18×36 pixels and 48×96 pixels. The second dimension makes the detection at test time very slow, making it unpractical for real-time use in the car. Using bigger patches means that more weights have to be learned with the same amount of training data, which might lead to a degradation of the performance.

A border of 2 / 4 pixels was added to head / body samples to avoid border effects. Also eight additional samples from each original sample are generated by shifting the corresponding bounding box 1-2 pixels.

The prior $p(V)$ is modeled with an uniform distribution for both the head and body. Parameters $\alpha_{hh} = \alpha_{bb} = 0.7$, $\alpha_{bh} = 0.2$, $\kappa_{hh} = \kappa_{bb} = 4$, $\kappa_{hb} = \kappa_{bh} = 1$, $\kappa_v = 2$, $t_p = 0.8$ and $t_v = 1.4$ (all concentration parameters κ are expressed in *radian* units) were manually tuned on artificial generated data or set based on human studies [42] such that each component of the model

has a correct amount of influence on the final result. An average of $c_o^H = 0.78$ and $c_o^B = 0.68$ from Eq. 4.4 were learned for each class from training data. The PF used 1000 particles for approximation.

7.3 Evaluation of training architectures on validation sets

This section presents the results of the detection architectures investigated in Section 4.5. First the results for the body setups are presented, followed by the ones for the head. The results are based on perfect localization, meaning that no search for head / body is done. The evaluation's purpose is to investigate the performance of the detection alone, when the correct body parts location would be already known. The evaluation is given in terms of confidence matrices, confusion matrices, angular non-absolute error, angular absolute mean error and angular absolute median error.

The confidence matrix presents the responses of the orientation detectors to samples with 16 different orientation labels. It gives an impression of the discrete orientation estimation. Because of the discrete estimation, the evaluation is done on 8 discrete classes, through binning. Bright values represent a higher confidence, while darker values a lower one. Ideally this matrix should be diagonal.

The confusion matrix presents the results obtained from the continuous orientation estimation when samples with 16 different orientation labels are given as input. It gives an impression on the performance for continuous orientation estimation. Because now the orientation estimation is continuous, the evaluation can be done on 16 estimated orientations through binning. Again bright values represent a higher confidence, while darker values a lower one. Ideally this matrix should also be diagonal. The values in the boxes represent the percentage of samples from a class that were estimated to have a certain label.

The angular mean non-absolute error plot presents the non-absolute mean error made for each one of the 16 discrete sample classes.

The angular mean / median absolute error plots present the absolute mean and the absolute median angular errors depending on the distance. The samples are binned based on the distance between the pedestrian and the car (the number of samples for each distance bin is written in the image, above the corresponding bin). Ideally there should be a smaller error at closer distances and a bigger one further away. The error should increase with the distance, especially in the case of the head, as the resolution becomes smaller, making the body parts harder to classify (the results are also influenced by the number of samples in each bin, as they are not distributed uniform).

7.3.1 Results of body one-versus-all training setups

Here the results for the one-versus-all training setups used for the body are presented (see Section 4.5.1).

In the case of the body training architectures the results, judging all the evaluation methods, show that the worst performance is obtained with the first training architecture (see Section 4.5.1.1), with option 1 (Figure 7.2). This means that the negative classes are not general enough and the properties of a specific orientation are not learned correctly. The third architecture (see Section 4.5.1.3) gives the best performance judging all four evaluation methods (Figure 7.5). This architecture is selected for later use within the framework for detecting the body. The other body training setups give in-between results.

7.3.2 Results of head one-versus-all training setups

Here the results for the one-versus-all training setups used for the head are presented (see Section 4.5.2).

The third 7.9 and sixth 7.12 architectures (see Sections 4.5.2.3, 4.5.2.6) give the best results when judging all the evaluation methods. Because the sixth one involves using an extra MLP network on top of the LRF features extracted from the base NN/LRF, which involves extra

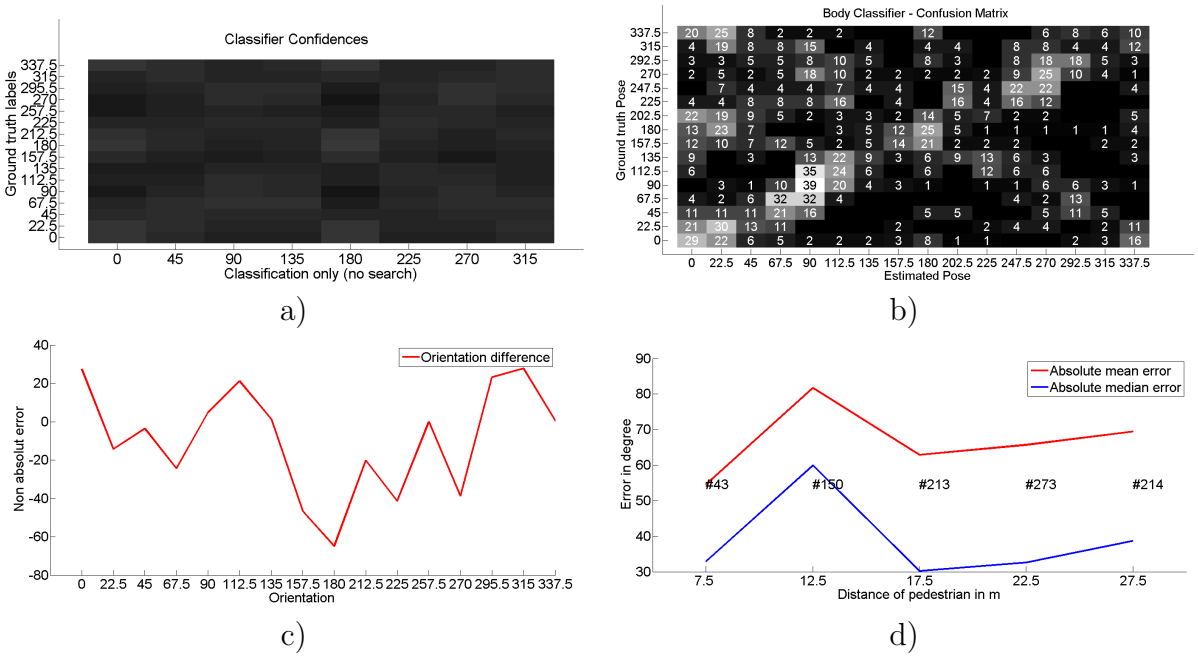


Figure 7.2: Results for body training setup 1: Orientation class O_i versus Non-body samples – Option 1, 4.5.1.1

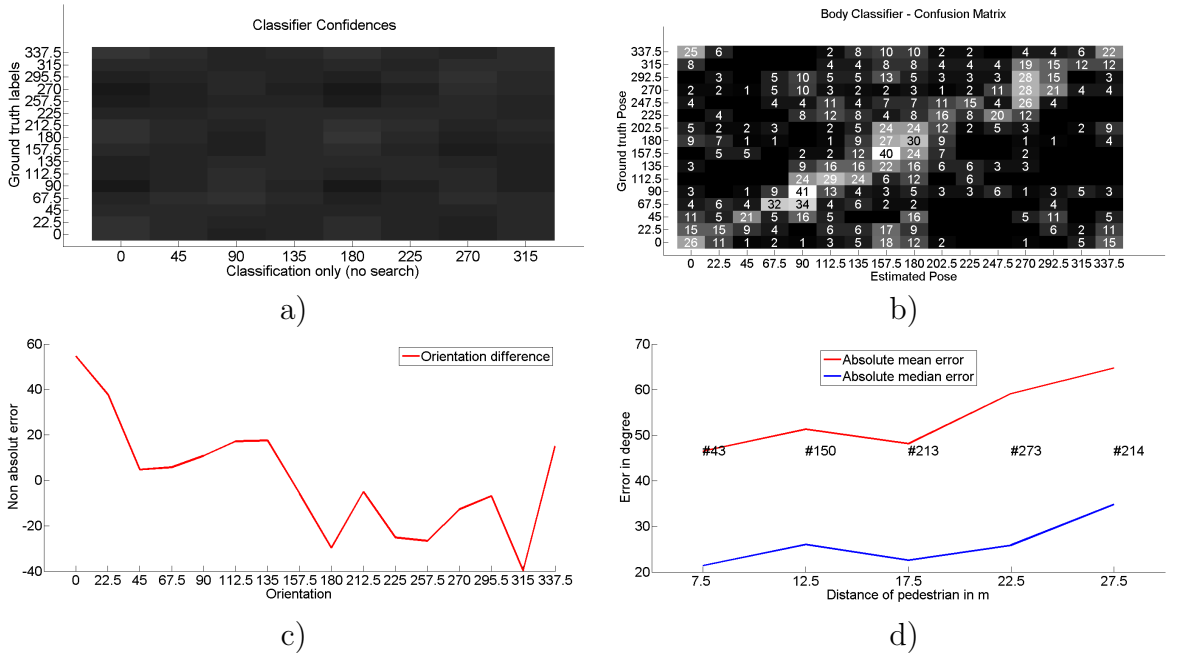


Figure 7.3: Results for body training setup 1: Orientation class O_i versus Non-body samples – Option 2,4.5.1.1

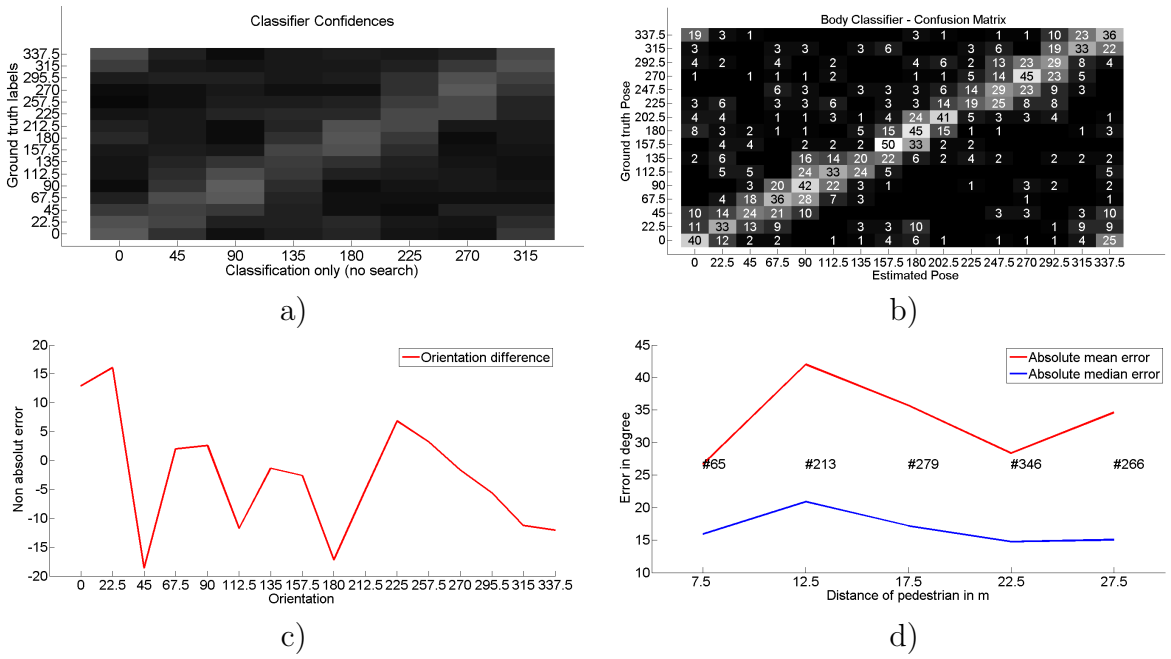


Figure 7.4: Results for body training setup 2: Weighted label neighbors versus All others, except neighboring classes, & non-body samples, 4.5.1.2

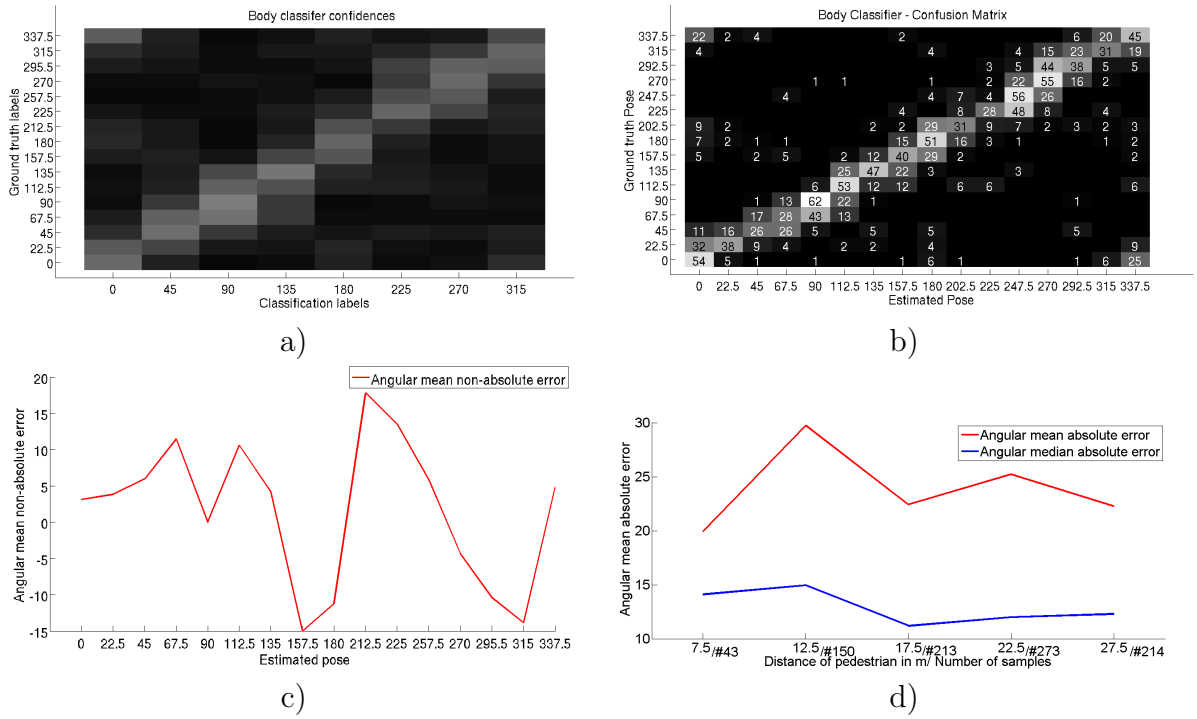


Figure 7.5: Results for body training setup 3: No label neighbors versus All others, except neighboring classes, & non-body samples,4.5.1.3

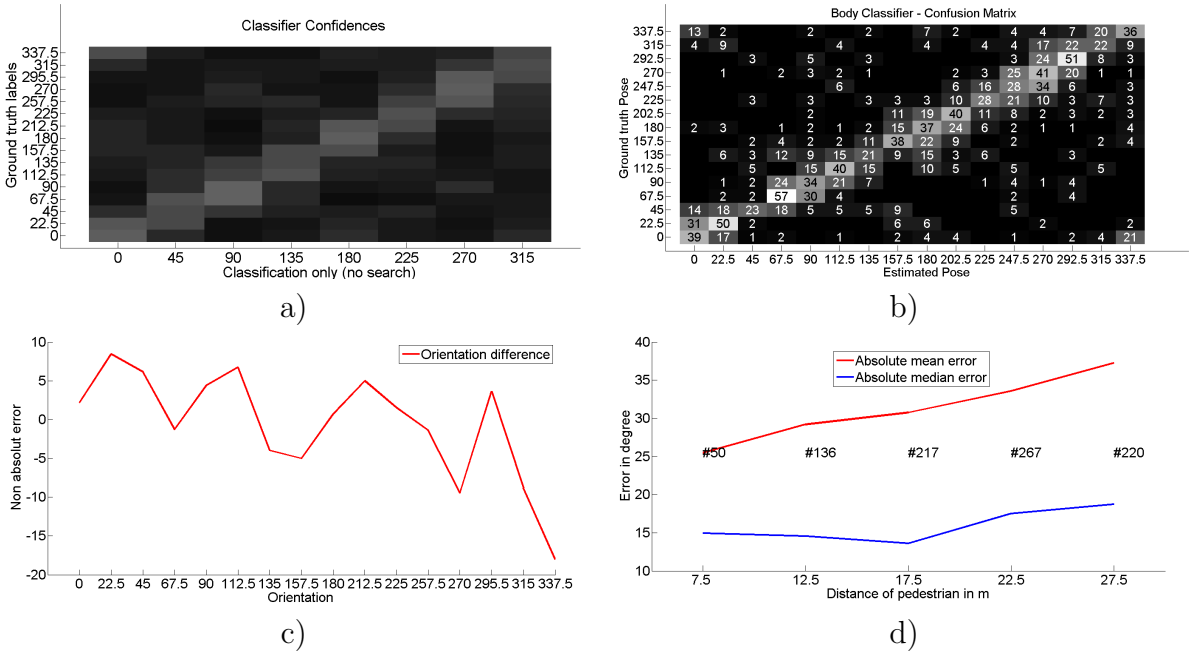


Figure 7.6: Results for body training setup 4: MLP on top of LRF features,4.5.1.4

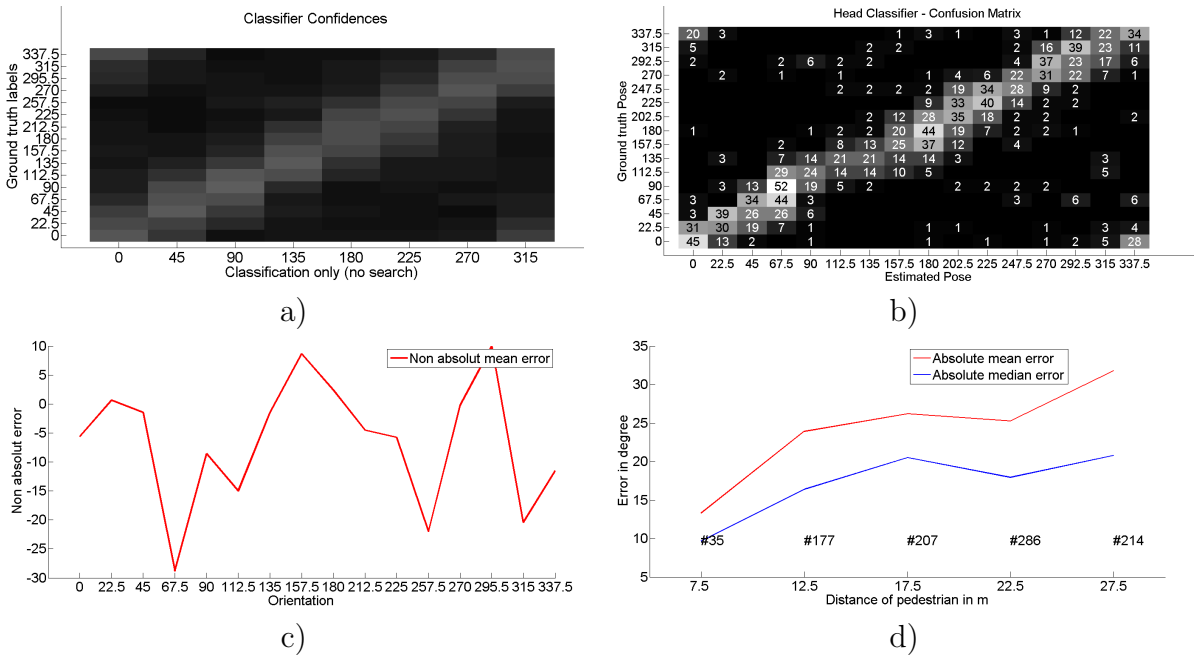


Figure 7.7: Results for head training setup 1: Orientation class O_i versus Non-head samples, 4.5.2.1

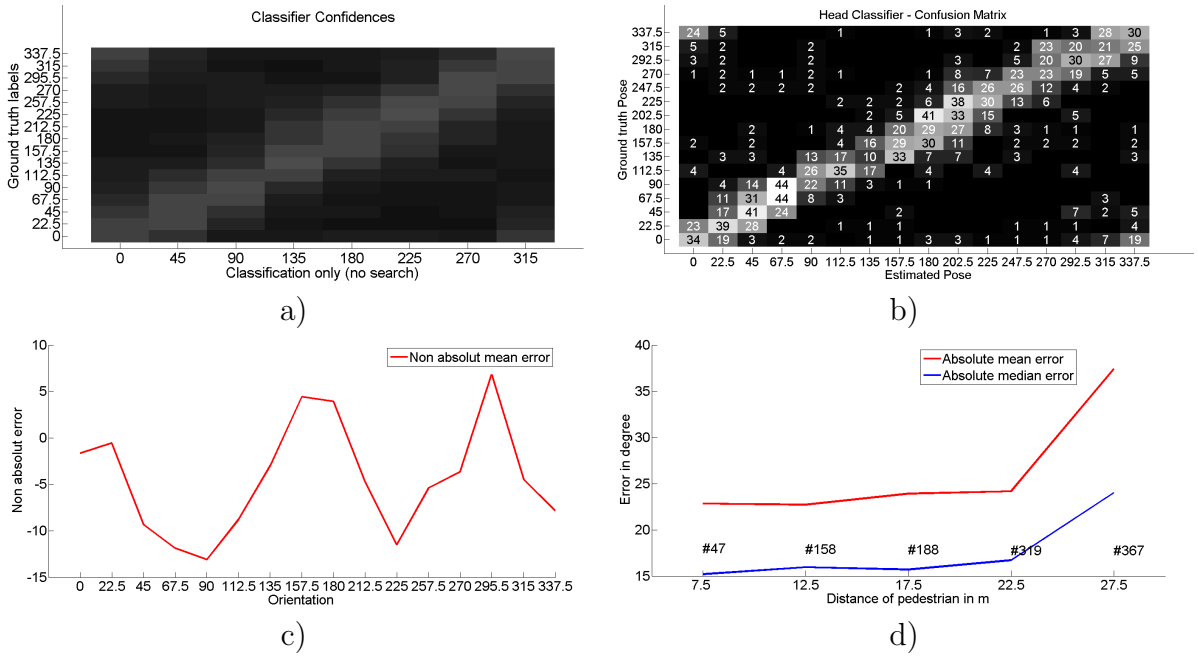


Figure 7.8: Results for head training setup 2: Merged label neighbors versus All others & non-head samples,4.5.2.2

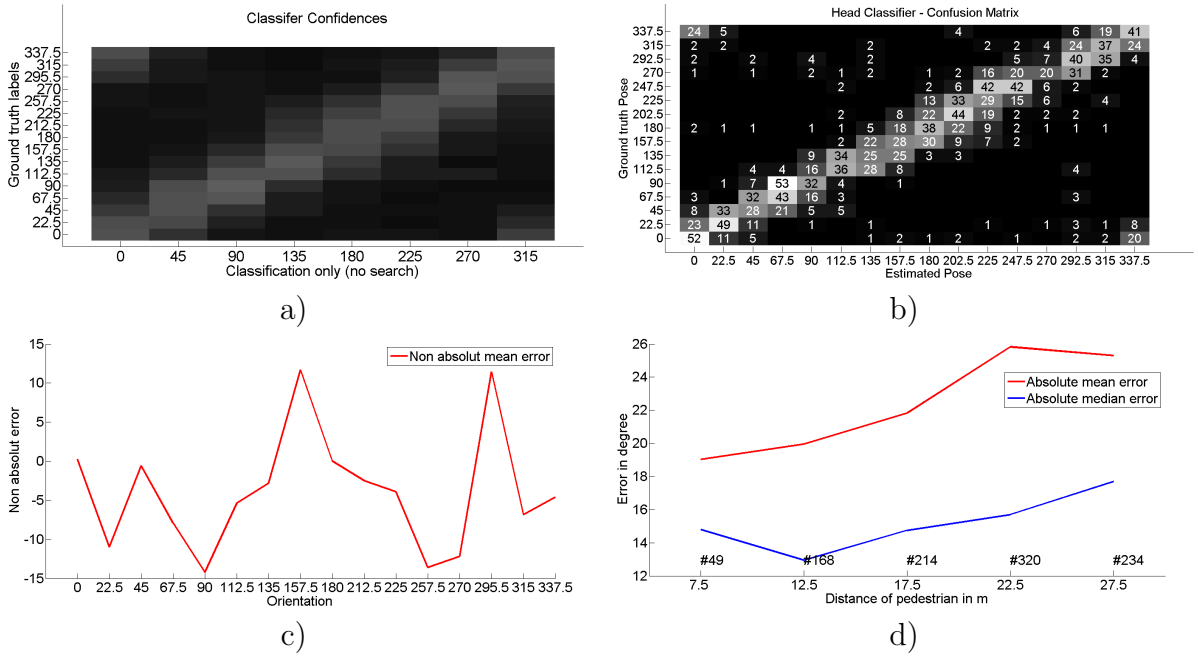


Figure 7.9: Results for head training setup 3: Merged label neighbors versus All others, except neighboring classes, & non-head samples, 4.5.2.3

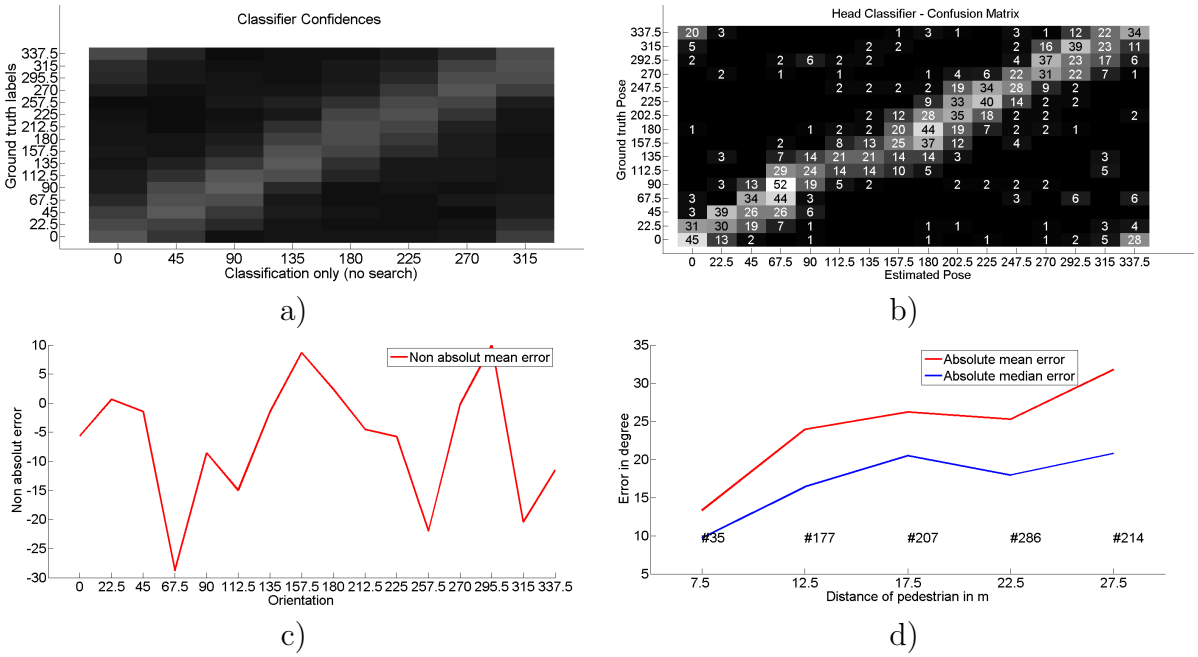


Figure 7.10: Results for head training setup 4: Merged label neighbors versus All others, except neighboring classes, 4.5.2.4

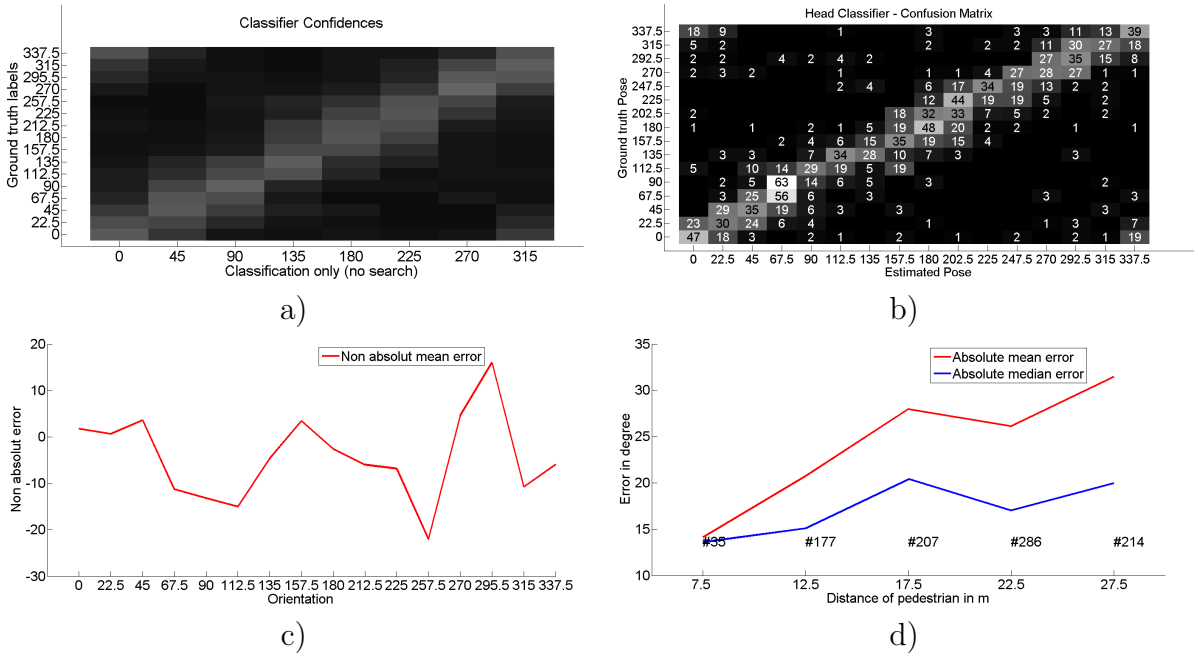


Figure 7.11: Results for head training setup 5: No label neighbors versus All others, except neighboring classes, & non-head samples, 4.5.2.5

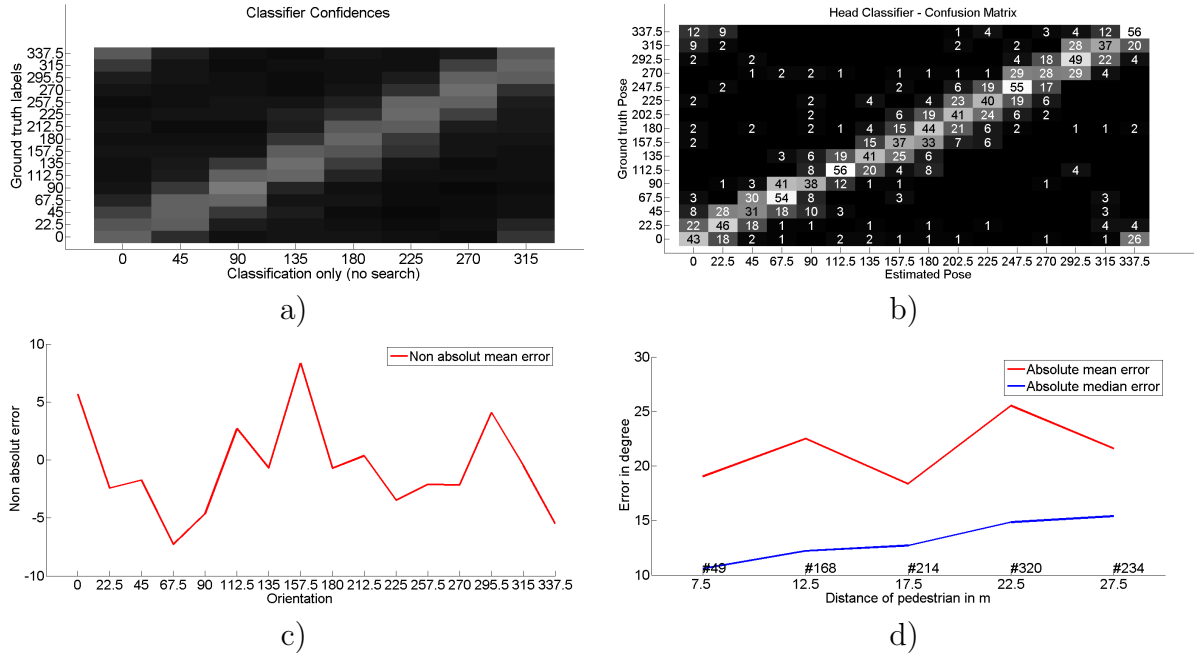


Figure 7.12: Results for head training setup 6: MLP on top of LRF features, 4.5.2.6

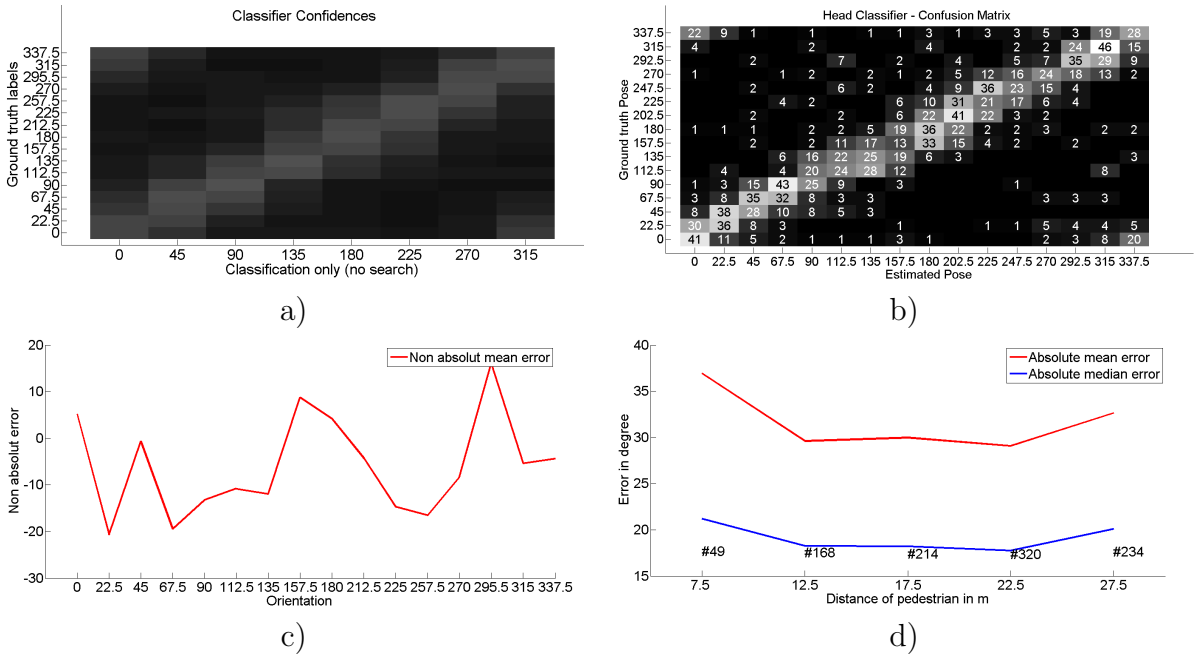


Figure 7.13: Results for head training setup 7: Linear SVM on top of LRF features, 4.5.2.7

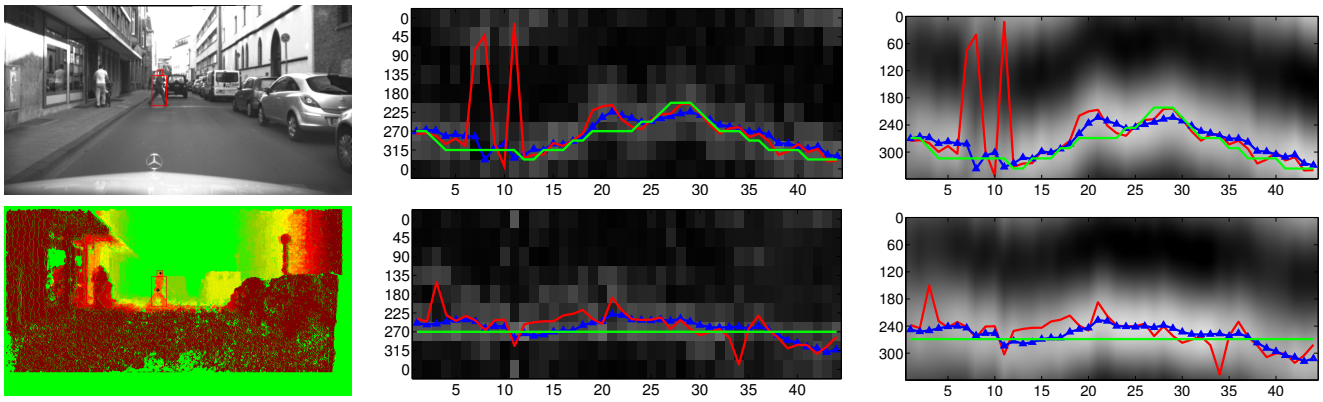


Figure 7.14: Orientation estimation over an entire pedestrian track – Example 1 – re-used from Flohr et al. [1]

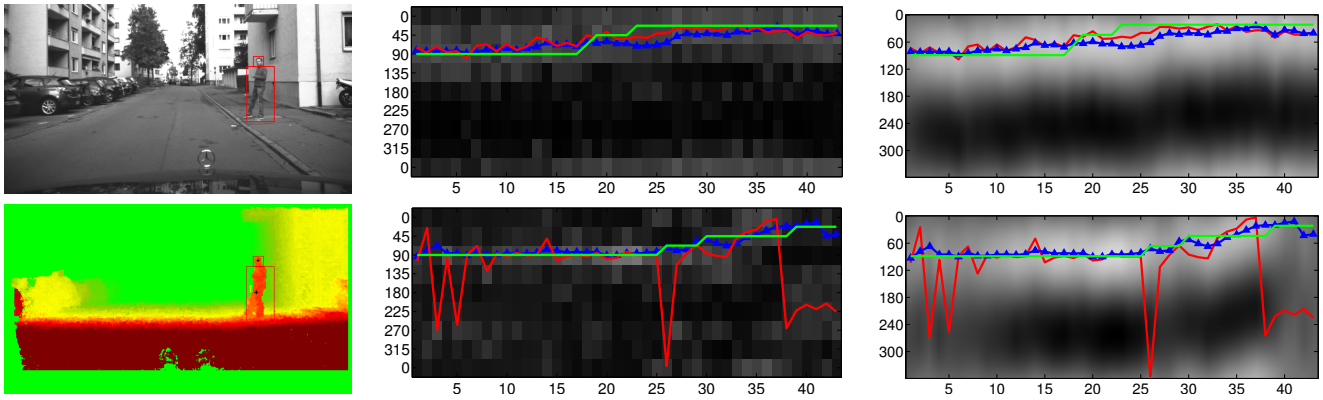


Figure 7.15: Orientation estimation over an entire pedestrian track – Example 2 – adapted from Flohr et al. [1]

time required at test-time, and does not bring significant performance improvement, the third architecture is selected for later use within the framework for detecting the head.

7.4 Framework qualitative evaluation

The framework is first subjected to a qualitative evaluation. Figures 7.14 and 7.15 give an impression on how the estimation looks like over one complete track. Figures 7.14 and 7.15 a) show one of the gray value images (top) and the corresponding disparity image (bottom) from the track. The red boxes show the selected head and body regions. Figures 7.14 and

7.15 b) present the orientation specific output of head orientation detectors (top) and body orientation detectors (bottom) over time (frames). Figures 7.14 and 7.15 c) show the posterior distributions estimated by joint tracking of head (top) and body (bottom) over time. Brighter values indicate a higher belief of the estimation. In b) and c) the ground truth is represented with green lines, the single-frame estimation with PS with red lines and the joint tracking result with blue lines. It can be observed that the tracker manages to smooth out outliers and still be able to react to small changes in the orientation. In Figure 7.15 c), between frames 37–44, it is shown that improbable head and body orientation configurations given by the single-frame estimation phase are rejected (given the strongly confident head orientation, the single-frame body orientation (red) is improbable; a more probable orientation for the body is chosen by the tracker – blue line).

In Figure 7.16 a) a sample that gives a multi-modal estimate of the body orientation (BM) caused by confusing opposite directions (back and front) is presented (red encodes more mass in the corresponding area, while blue encodes less mass). The ambiguity is solved successfully by the joint tracking approach (BT). Also the maximum posterior estimate (red line) and ground truth orientation (black line) are shown. It can be observed that in the case of BM even though the maximum posterior estimate is wrong, there is mass at the correct orientation. Figure 7.16 b) shows the benefits of integrating the PS spatial constraint. In the right image the localization of the head and body is improved over left.

Figure 7.17 shows the disparity (left) and the gray value image (right) of every sixth frame of five estimated tracks with the continuous estimation results of the proposed method. The red boxes show again the selected head and body region. Below the images it is shown the posterior distributions of the approach for the head (HT) and body (BT) orientations, maximum posterior estimate (red line) and ground truth orientation (black line). Black crosses in disparity images denote estimated head and body centers.

It can be seen from Figure 7.17 that the framework offers good localization and a robust continuous orientation estimate of head and body. Even in cases with limited stereo support for the head (e.g. first row, fourth and fifth image), it is still localized thanks to the detectors

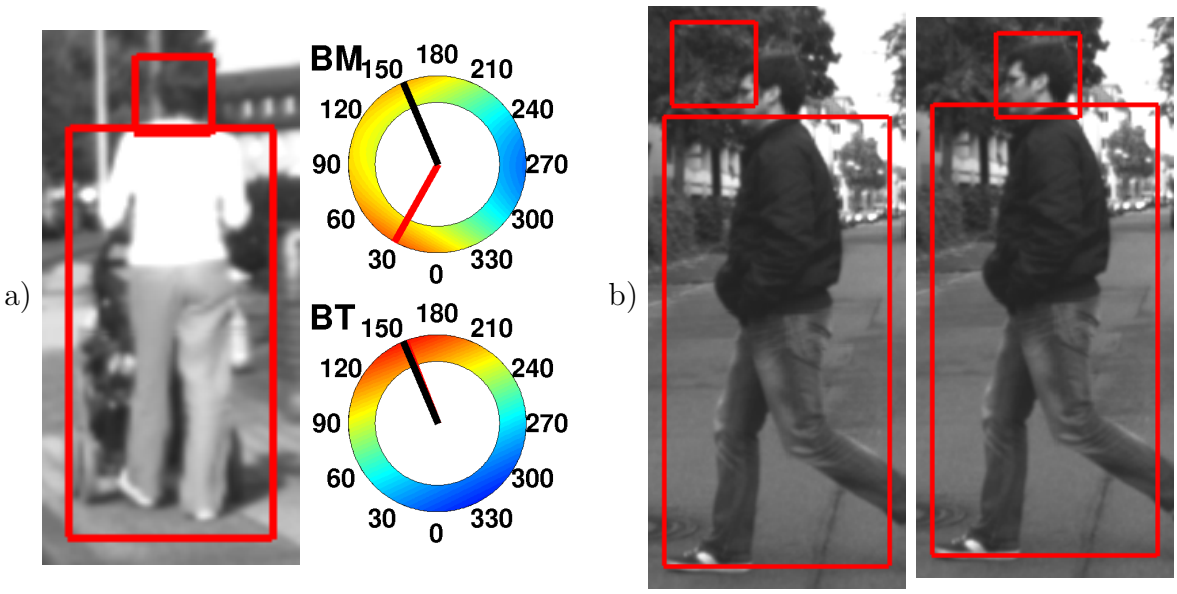


Figure 7.16: a) Multi-modality solving and b) the benefit of a PS localization constraint – adapted from Flohr et al. [1].

outputs. The last track (last row) shows various problem cases of the current approach causing a wrong localization of the head and body and therefore a bad orientation estimation. Reasons for that are stronger deviations of mean head position and rotation (second image), pedestrian groups (fifth image), or contrast and lighting issues.

7.5 Framework quantitative evaluation

The framework is also subject to a quantitative evaluation on the complete test set using all 65 valid, estimated tracks. Figure 7.18 shows the obtained confusion matrices for the single-frame with PS (a) and c)) and the joint tracking (b) and d)) for the head (a) and b)) and body (c) and d)). Now, the evaluation includes also the searching phase of the body parts. It can be observed that the joint tracking estimation matrices have less noise than the single-frame ones. Also, when the single-frame with PS confusion matrices are compared with the validation matrices, it can be seen that the localization is still an important issue.

Figure 7.19 shows the angular mean absolute error for head and body orientation estimation with increasing distance. The average error over all distances is shown in the legend. The results of the joint tracking (purple) are compared to with results of independent tracking (cyan)

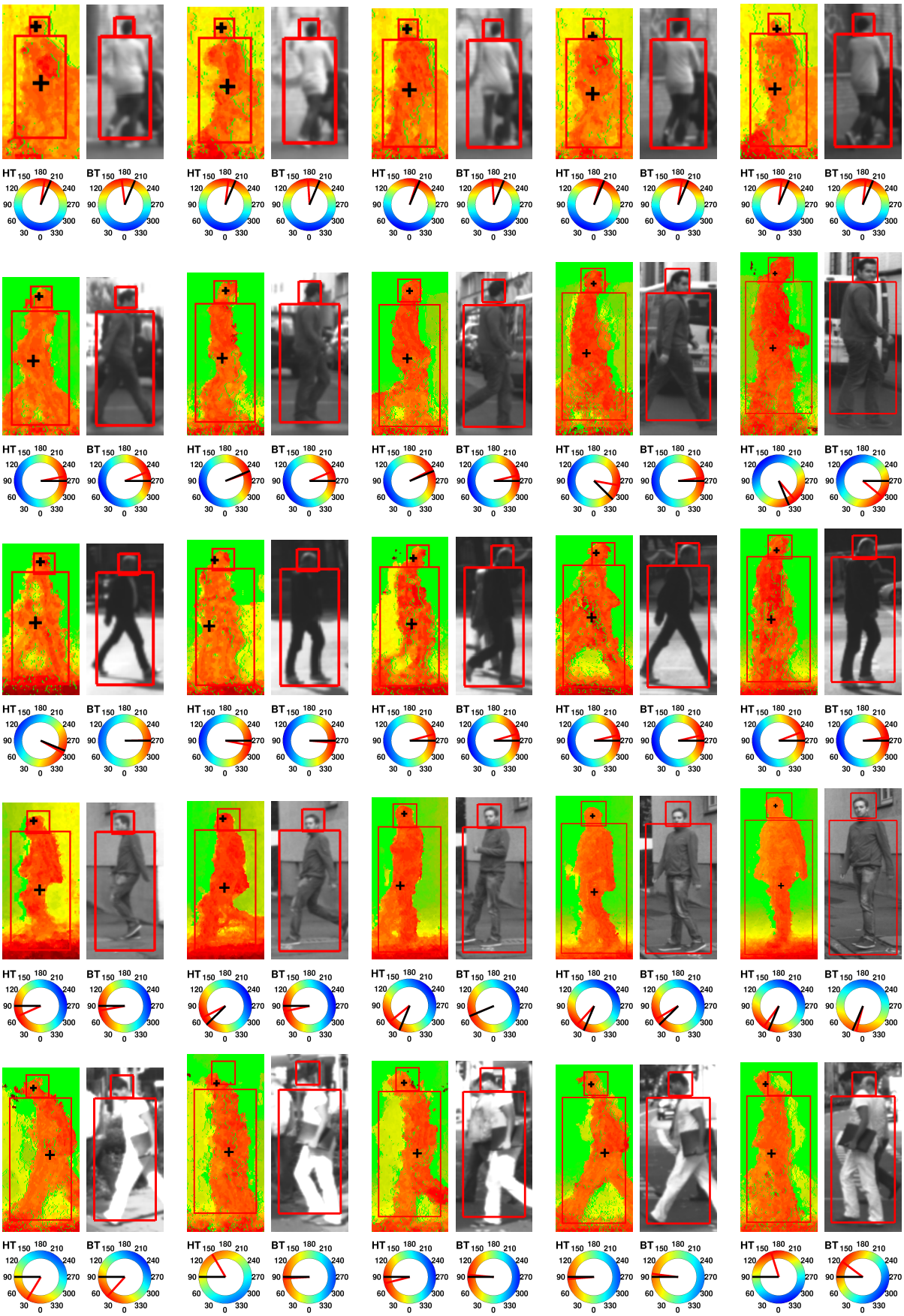


Figure 7.17: Every sixth frame of five estimated tracks – adapted from Flohr et al. [1]

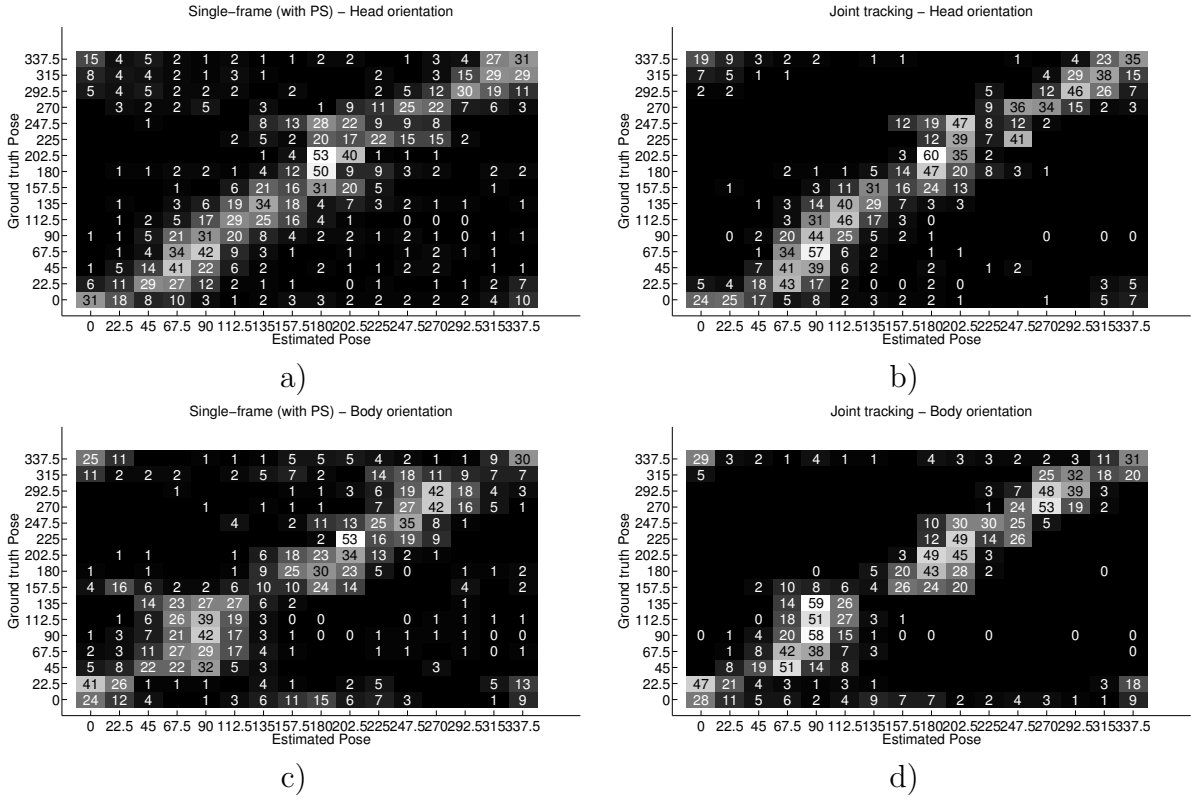


Figure 7.18: Single-frame (with PS) and joint tracking confusion matrices

and single-frame orientation estimation with PS (green) and without PS (red). For both independent and joint tracking the spatial PS constraint is used. It can be seen that the tracking significantly decreases the mean error. Joint tracking decreases the head / body orientation error over all samples by 16° / 14° compared with single frame estimation without PS. This benefit is mainly caused by the removal of outliers compared to single-frame estimation (e.g. confusion between opposite body directions, which visually can look very similar). Furthermore in comparison to independent tracking, the error decreases by 4° / 5° for head / body orientation. Anatomical and movement constrains within tracking as defined in Section 5.4.2 help here to reject impossible configurations between head and body orientation.

One should expect the error to increase with distance, as the resolution decreases and the orientation detection becomes really difficult, expectation that can be validated by Figure 7.19.

Figure 7.20 shows additional boxplots to get a better impression of the estimation uncertainty and the error distribution. The median error is represented with the red line, while the outliers are illustrated with red crosses. Boxes contain 50 % of samples. Used whiskers define 99.3 %

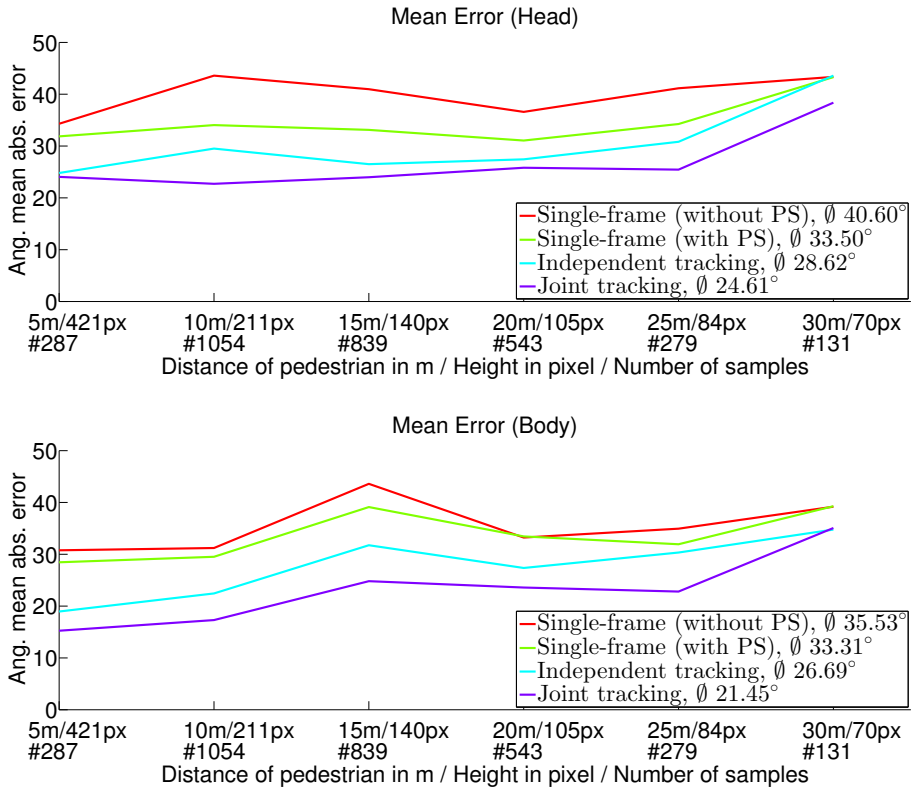
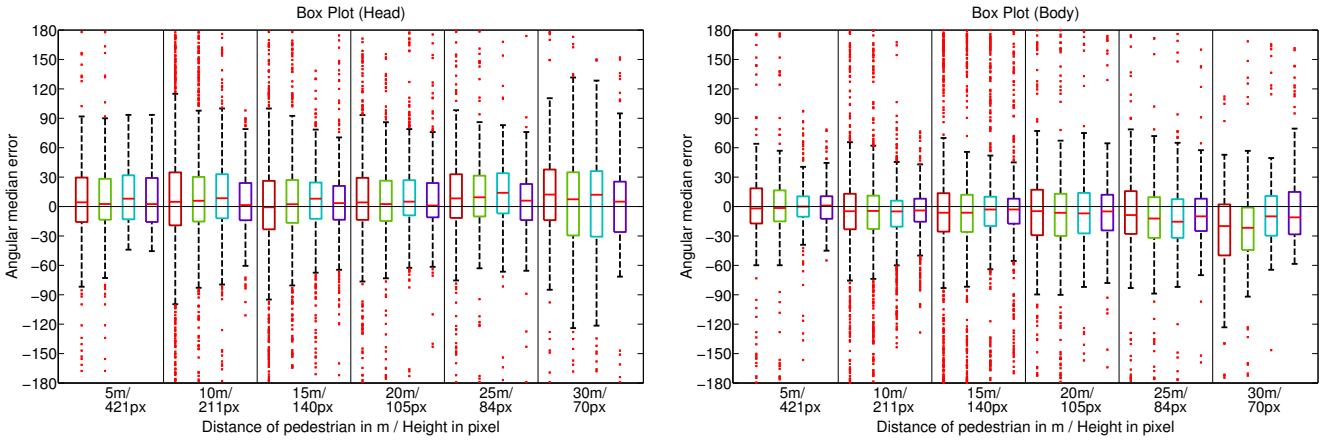


Figure 7.19: Absolute angular mean error over increasing distance for head (top) and body (bottom) orientation estimation – re-used from Flohr et al. [1]

data coverage. The same color coding as before is also kept here: red boxes – single-frame estimation without PS, green boxes – single-frame estimation with PS, cyan boxes – independent tracking and purple boxes – joint tracking. It can be seen that joint tracking gives smaller boxes and whisker lengths for both head and body for most of the distances, meaning that it reduces the uncertainty. Also the number of outliers, especially the ones that give high errors, is reduced.

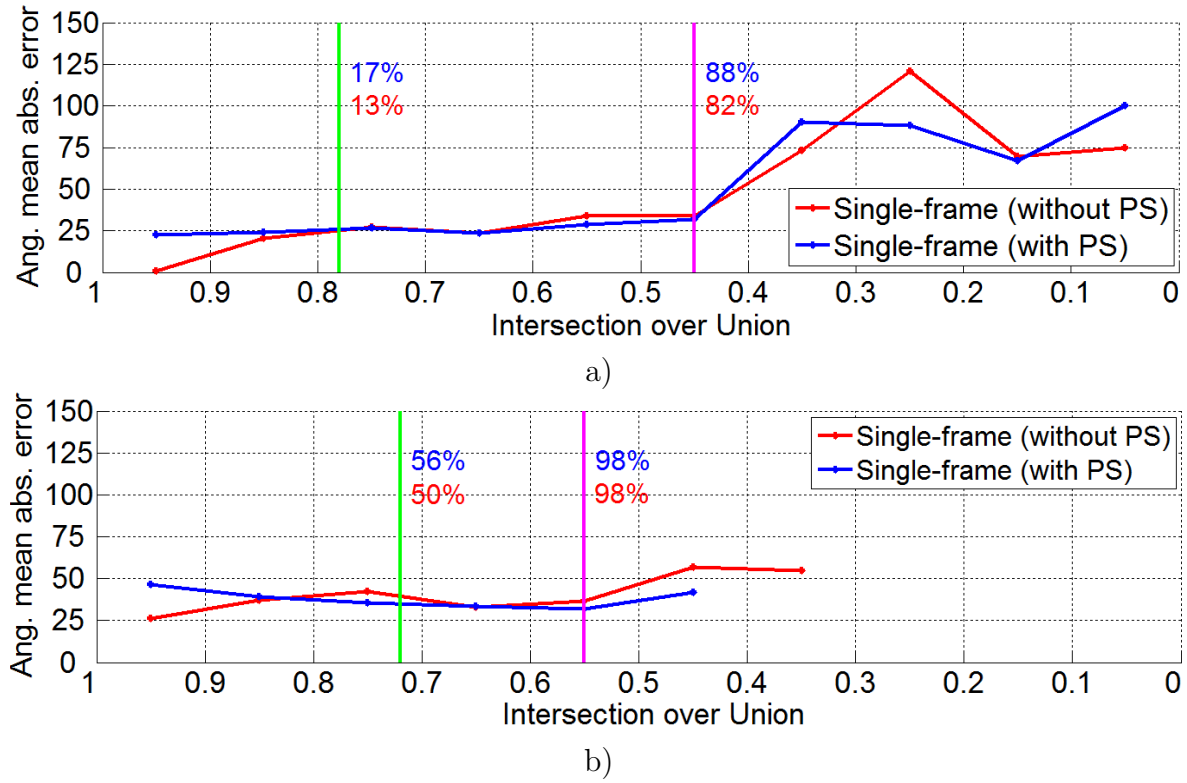
7.6 Framework localization evaluation

The localization performance was also tested. 229 evenly distributed samples from the complete test data were annotated with the ground truth head / body locations (in terms of bounding boxes). The Intersection over Union (IoU) measure was computed between the ground truth and estimated bounding boxes. A value of 1 corresponds to a perfect overlap and a value of 0



a) b)

Figure 7.20: Boxplots for a) head and b) body orientation estimation – re-used from Flohr et al. [1]



a)

b)

Figure 7.21: Absolute angular mean error over decreasing localization accuracy (measured by IoU) in intervals of 0.1 for a) head and b) body – re-used from Flohr et al. [1]

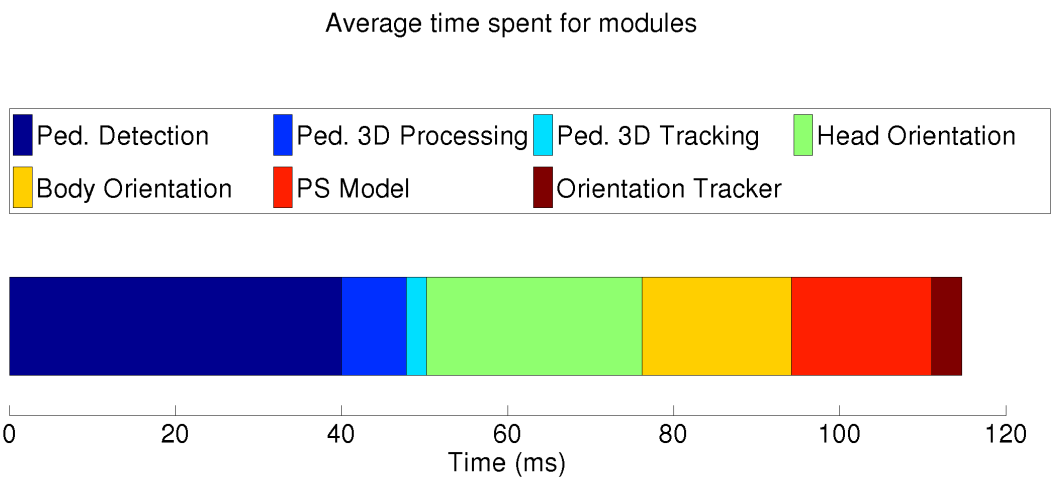


Figure 7.22: Different modules and their running time. All modules need on average approx. – re-used from Flohr et al. [1]

corresponds to no overlap.

Figure 7.21 shows how the angular mean absolute error is affected by the localization performance. a) shows that 88% – blue (82% – red without PS) of the head samples have a localization performance better than 0.45 (magenta line), while still getting an acceptable orientation estimates. The green line shows the computed localization performance (theoretical maximum localization error) threshold regarding a possible shift of 1 pixel for a 16×16 pixels image in each direction, as done in the training to increase the amount of training samples. The performance on the validation sets would be reproduced if one would consider as acceptable only samples with this localization error. Due to big variations in the training set, a much lower localization performance (as showed by the magenta line) can be accepted, keeping good orientation estimates. b) shows the same for the body.

Relying only on measurements regarding head and body localization is not optimal, so in future work the approach has to be extended to integrate the location of the head and body over time.

7.7 Framework time evaluation

The current implementation of the framework runs on a machine with a 3.33 GHz i7-CPU processor and 12GB RAM and needs on average less than 120 ms per frame. Figure 7.22 shows how this time is distributed among different components used in the system.

The state-of-the-art HOG/linSVM pedestrian detector (Ped. Detection) needs 40 ms for computing pedestrian detections, being the only module presented here which has a FPGA implementation. A 3D world estimation is performed in $\sim$ 7.5 ms (Ped. 3D Processing), using stereo-vision. The pedestrian is then tracked by a standard Kalman Filter in $\sim$ 2.5 ms (Ped. 3D Tracking). As described in this work, the single-frame orientation estimation of the head (Head Orientation) and body (Body Orientation) is computed in $\sim$ 26 ms and $\sim$ 18 ms respectively, using the Pictorial Structure Model (PS Model) that adds an extra $\sim$ 17 ms. Then they are jointly tracked by a particle filter (Orientation Tracker), using various constraints, in $\sim$ 3.6 ms .

It is to be noted that the head orientation estimation needs more time, since more region hypotheses are generated on average compared to the body orientation estimation. In future work, an additional time benefit is expected from using feature sharing between the detectors and by further parallelization of the modules or hardware implementation.

Chapter 8

Conclusions

This work presented a probabilistic framework for the joint estimation of pedestrian head and body orientations in the context of vision-based active pedestrian safety systems. The framework deals with faulty part detections, builds a continuous orientation distribution that implicitly incorporates uncertainty and couples both the body- and head-localization and orientation tracking.

The quantitative evaluation showed that the proposed joint tracking of the head and body orientations decreases the mean absolute head / body orientation error by 16° / 14° compared to single frame estimation and further by 4° / 5° compared to independent tracking. In absolute terms, this comes down to mean absolute head / body orientation error which is fairly constant up to a distance of 25 m, namely about 25° / 21°. Being one of the first works on pedestrian head and body orientation estimation in vehicle context it is hard to make comparisons with other work.

As pedestrian orientation estimation is in an early stage, making strong statements regarding the head and body orientation estimation quality that is required to facilitate an advanced situation analysis is hard. Path prediction from a moving vehicle is an emerging problem and an appealing idea is to infer from the relative head orientation whether the pedestrian is aware of the approaching vehicle, in order to warn the driver sooner if the pedestrian is inattentive. Still, the relation between the "is- / is-not-aware-of-vehicle" and the relative head orientation

still needs to be investigated by human factors studies.

Nevertheless the preliminary orientation results that were obtained can provide valuable cues to some advanced situation analysis, especially when the entire probability density function is utilized (rather than just a single value estimate).

Based on the current results, it can be concluded that head / body orientation estimation will play an important role in next-generation, intelligent driver warning and vehicle control strategies.

Future work can further improve the performance by using a larger training dataset, higher resolution images and more accurate pedestrian segmentation methods (i.e including prior knowledge about the shape or texture). Future work can also focus on improving the running time by using feature sharing between the detectors and by further parallelization of different modules. Another solution would be a CUDA implementation or a hardware implementation on FPGAs.

Bibliography

[1] F. Flohr, M. Dumitru-Guzu, J. F. Kooij, and D. M. Gavrila, "A probabilistic framework for joint pedestrian head and body orientation estimation," in *IEEE Trans. on Intelligent Transportation Systems(submitted)*, 2014.

[2] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.

[3] M. Isard and A. Blake, "Condensation–conditional density propagation for visual tracking," *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.

[4] S. Schmidt and B. Färber, "Pedestrians at the kerb–recognising the action intentions of humans," *Transportation research part F: traffic psychology and behaviour*, vol. 12, no. 4, pp. 300–310, 2009.

[5] H. Hamaoka, T. Hagiwara, M. Tada, and K. Munehiro, "A study on the behavior of pedestrians when confirming approach of right/left-turning vehicle while crossing a crosswalk," in *Proc. of the IEEE Intelligent Vehicles Symposium*, June 2013, pp. 106–110.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[7] F. Flohr, M. Dumitru-Guzu, J. F. Kooij, and D. M. Gavrila, "Joint probabilistic pedestrian head and body orientation estimation," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014.

[8] E. Hjelmås and B. K. Low, "Face detection: A survey," *Computer vision and image understanding*, vol. 83, no. 3, pp. 236–274, 2001.

[9] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, 2009.

[10] S. O. Ba and J.-M. Odobez, "A Rao-Blackwellized mixed state particle filter for head pose tracking," in *Proc. of the ACM-ICMI Workshop on MMMP*, 2005, pp. 9–16.

[11] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 617–624.

[12] M. J. Marín-Jiménez, A. Zisserman, and V. Ferrari, ""Here's looking at you, kid". Detecting people looking at each other in videos." in *Proc. of the British Machine Vision Conf. (BMVC)*, 2011, pp. 1–12.

[13] B. Benfold and I. Reid, "Unsupervised learning of a scene-specific coarse gaze estimator," in *Proc. of the International Conf. on Computer Vision (ICCV)*. IEEE, 2011, pp. 2344–2351.

[14] C. Chen and J. Odobez, "We are not contortionists: coupled adaptive learning for head and body orientation estimation in surveillance video," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1544–1551.

[15] J. Orozco, S. Gong, and T. Xiang, "Head pose classification in crowded scenes." in *Proc. of the British Machine Vision Conf. (BMVC)*, vol. 1, 2009, p. 3.

[16] N. Robertson and I. Reid, "Estimating gaze direction from low-resolution faces in video," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006, pp. 402–415.

[17] K. Smith, S. O. Ba, J.-M. Odobez, and D. Gatica-Perez, "Tracking the visual focus of attention for a varying number of wandering people," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1212–1229, 2008.

[18] A. Schulz, N. Damer, M. Fischer, and R. Stiefelhagen, "Combined head localization and head pose estimation for video–based advanced driver assistance systems," in *Proc. of the DAGM Symposium on Pattern Recognition*, 2011, pp. 51–60.

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

[20] B. Benfold and I. Reid, "Guiding visual surveillance by tracking human attention." in *Proc. of the British Machine Vision Conf. (BMVC)*, 2009, pp. 1–11.

[21] T. Gandhi and M. M. Trivedi, "Image based estimation of pedestrian orientation for improving path prediction," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 506–511.

[22] H. Shimizu and T. Poggio, "Direction estimation of pedestrian from multiple still images," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2004, pp. 596–600.

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

[24] T. Cootes, C. Taylor, D. Cooper, J. Graham *et al.*, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[25] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[26] D. M. Gavrila, J. Giebel, and H. Neumann, "Learning shape models from examples," *Pattern Recognition*, pp. 369–376, 2001.

[27] K.-C. Lee and D. Kriegman, "Online learning of probabilistic appearance manifolds for video-based recognition and tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 852–859.

[28] P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 9, pp. 1042–1049, 2000.

[29] P. M. Hall, A. D. Marshall, and R. R. Martin, "Incremental eigenanalysis for classification." in *BMVC*, vol. 98, 1998, pp. 286–295.

[30] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 2879–2886.

[31] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision*, vol. 73, no. 1, pp. 41–59, 2007.

[32] C. Chen, A. Heili, and J.-M. Odobez, "A joint estimation of head and body orientation cues in surveillance video," in *Proc. of the International Conf. on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011, pp. 860–867.

[33] G. Zhao, M. Takafumi, K. Shoji, and M. Kenji, "Video based estimation of pedestrian walking direction for pedestrian protection system," *Journal of Electronics (China)*, vol. 29, no. 1-2, pp. 72–81, 2012.

[34] S. O. Ba and J.-M. Odobez, "Probabilistic head pose tracking evaluation in single and multiple camera setups," in *Proc. of the Workshop on Classification of Events, Activities and Relationships (Multimodal Technologies for Perception of Humans)*, 2008, pp. 276–286.

[35] D. M. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[36] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[37] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[38] C. Wöhler and J. K. Anlauf, "Real-time object recognition on image sequences with the adaptable time delay neural network algorithm – applications for autonomous vehicles," *Image and Vision Computing*, vol. 19, no. 9, pp. 593–618, 2001.

[39] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 11, pp. 1863–1868, 2006.

[40] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[41] T. Chateau, V. Gay-Belille, F. Chausse, and J.-T. Lapreste, "Real-time tracking with classifiers," in *Dynamical Vision*. Springer, 2007, pp. 218–231.

[42] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *Journal of Applied Physiology*, vol. 100, no. 2, pp. 390–398, 2006.