University Politehnica of Bucharest, Romania
Faculty of Automatic Control and Computer Science
Msc Programme Computer Graphics and Virtual Reality

Master Thesis

Human Tracking and Orientation Estimation

Performed at department Game and Media Technology of Utrecht University, The Netherlands, through Erasmus Programme

Scientific Adviser:
Prof. Robby Tan
Dr. Nico van der Aa
Prof. Adina Magda Florea

Author:
Manuela Ichim

2013

Abstract

This thesis presents the research done on human head and body orientation estimation. This problem can be subdivided in two tasks, namely human tracking and orientation estimation. The first task is accomplished using the framework described by Choi et al. which is capable of estimating and tracking the positions of human targets in real worlds coordinates, starting from a video stream captured using a single monoscopic moving camera. The approach of Chen et al. is implemented for solving the second task, namely head and body orientation estimation. My approach for solving this task starts from the main ideas outlined in the original method, such as using HOG descriptors for describing the visual appearance of the targets and additional cues such as the velocity direction and head-body coupling. To address some of the limitations of the original method, as well as to incorporate new elements, a different framework is conceived. Under this new framework, the responses of 3 different classifiers (Gaussian Mixture Model, Neural Network and Support Vector Machine) are combined with information from additional cues. These include the original ones, velocity direction and magnitude and head-body coupling, as well as new ones, face detections and temporal smoothness. The performance of the method is evaluated and the contribution to the final prediction of each classifier and additional cue is assessed. Overall, the performance of the proposed approach is satisfactory, outperforming my own implementation of the original method in an experiment, both in terms of estimation accuracy, as well as computation time. However, a thorough comparison between the proposed and the original approaches was not possible, due to the unavailability of the annotations used by Chen et al.

Acknowledgement

I would like to thank prof. Robby Tan and dr. Nico van der Aa for coordinating my thesis research and project at Utrecht University and providing support, guidance and feedback. I would also like to thank prof. Adina Magda Florea for facilitating my exchange at Utrecht University through the Erasmus Exchange Programme.

Contents

Chapter 1

Introduction

Although Computer Vision emerged as a research field more than 30 years ago, impressive results to notoriously difficult problems began to appear only recently. This can be attributed mainly to advances in the broader field of Artificial Intelligence and Machine Learning, as well as the significant increase in available computing power, supporting progressively more demanding algorithms and techniques.

Among the many problems addressed by Computer Vision, tasks focused on human subjects are increasingly more popular. These include human and face detection and tracking, as well as more detailed analysis of their appearance, such as body pose estimation or emotion recognition.

The estimation of human body and head orientation is a task with many potential use, mainly focused in the area of modeling human activity and interaction. Determining how people move in an environment, either indoor or outdoor, is a key first step in understanding their actions. On top of human tracking, orientation estimation can provide more detailed information regarding a person's activity, such as targets of their attention focus. This information can be useful in building automated systems focused on human interaction, anticipating and detecting abnormal behaviour and commercial applications. One such example is presented in [3], where the authors present a framework capable of assessing how people direct their attention towards outdoor advertising panels. Moreover, other applications such as video surveillance systems can benefit from these techniques, yielding better, more accurate and more informative results.

Moreover, being able to perform this task without specialized equipment, such as stereoscopic video cameras or additional sensors, is particularly important, as such a technique can be applied using existing systems (such as surveillance cameras), or reduce the costs for new systems by only requiring the installation of standard, relatively low-cost equipment.

The task of estimating orientation of the people is a complex one and implies multiple stages including human body and head detection and tracking. Additional computation, such as determining real-world 3D position coordinates of the targets and velocity orientation, can provide useful information for improving the results of the body orientation. Thus, the problem addressed in this thesis is the estimation of the head and body orientation of multiple human targets from a video sequence captured by a single monoscopic moving camera.

Figure 1.1: Orientations for body discretized in 8 directions.

1.1 Challenges

Solving the above mentioned main tasks, namely human tracking and orientation estimation, is not trivial due to various reasons. Identifying these challenges is important for a proper formulation of the task, as well as exploring and deciding the ways in which they can be addressed.

One such challenge is the large variation in human appearance. There are many causes for this, including inherent variations in the individual physiognomy of the human targets, as well as differences in clothing and accessories. Additionally, since the movement of the targets is considered to be unconstrained, meaning they move freely around the scene, their appearance can vary greatly due to their position and orientation relative to the camera.

Furthermore, the environment in which people undertake their activities are usually cluttered, making detection and tracking hard. Also, numerous objects in the environment can generate total or partial occlusions of the people.

Certain limitations of the camera also hinder tracking. Such limitations include narrow fields of view or inability to correctly capture images in difficult lighting conditions. Additionally, the resolution of the images captured by many systems is relatively low, which may restrict the extraction of robust and more detailed characteristics, such as face detection.

1.2 Approach

To mitigate the effects of the above mentioned challenges, several assumptions and simplifications must be added to make the development of a robust system possible.

My approach builds upon the ideas lined out in [2] and thus many assumptions introduced there are also applicable to the methods described in this thesis. The task of determining human orientation is formulated as a multi-class classification task; body and head rotation angles are grouped into 8 distinct classes, as shown in Figure 1.1. It is worth noting that only the rotation in the ground plane (i.e. yaw) is assessed, as it is the only one commonly encountered in most human activities. Complex human motion involving more degrees of freedom defines the subject of human pose estimation, in which individual body parts are tracked and analyzed. However, this task will not be addressed in this thesis.

The appearance of human targets is modeled by a dense grid of HOG descriptors, which are robust to scaling and light conditions, thus increasing the uniformity of appearances within a given class. Since the direct visual information in a video frame provides the main informational content, classification based on these HOG descriptors constitutes an important part of the method as a whole.

Additional cues are also used, such as the velocity orientation of the targets or the coupling between the head and body orientations (which cannot differ greatly because of anatomical restrictions). These cues are also based on the visual information from the frames, but indirectly, as they are the result of additional techniques.

The resemblance between the approach described in [2] and mine is that both make use of HOG descriptors as a basis for classification, as well as incorporating additional cues such as the ones previously mentioned. The differences occur in how these elements are used. The original method is based on computing a linear classifier in a high-dimensional space, where datapoints are projected using a kernel function. The additional cues are included as terms in the objective function of the linear classifier which is then optimized. My approach makes use of several classifiers based on the annotated HOG data which are trained offline. The response of these classifiers is combined online with the additional cues for the final estimation result. The way in which this combination takes place is described using a Bayesian framework, under which the Maximum Likelihood solution is considered.

1.3 Contributions

The contributions made by my approach aim at reducing the effects of the limitations of the method from [2], as well as making better use of the various additional cues.

Although the framework proposed in [2] has an elegant mathematical model, some technical limitations make its use cumbersome in an online setting. This is mainly because their approach requires computing a linear classifier in a high dimensional space for every frame of the video. Furthermore, since the HOG descriptors used for modeling the body appearance have 2268 dimensions, projecting them in an even higher dimensional space and determining a linear classifier implies computations with very large matrices.

To address this limitation, I have decided to separate the classification using HOG descriptors from the additional cues such as velocity direction or head-body coupling. This enabled me to have a separate initial training phase for the HOG-based classifiers.

This decision also allows me more flexibility in choosing the classification methods, my best results being obtained using the combined response from several classifiers (committee).

Another contribution refers to the way velocity information is taken into consideration. In the original method, the classes corresponding to the velocity angle class and adjacent ones were favored over the others, provided the magnitude of the velocity was above a certain threshold. Because of the greater flexibility of my approach, I have modeled the velocity as a pseudo-classifier using a Gaussian distribution centered around the class indicated by the velocity direction, but having a variance inversely proportional to the magnitude of the velocity.

An important cue which allows human individuals to recognize and estimate the orientation of other human targets is the presence of the face. Since face detection can be made relatively fast and is reliable if a minimal set of conditions regarding the image quality are met, another contribution is the consideration of face detection in determining the final result.

The method proposed in [2] considered the features of the targets independently from one frame to the other. However, since the video frames represent successive moments in time, and since human targets cannot abruptly change their orientation in a very short amount of time (such as the one between two consecutive frames), it is also reasonable to include temporal information in the estimation process. Thus, another contribution is the consideration of

temporal smoothness of the orientation change.

1.4 Layout

My thesis is organized as follows. In Chapter 2 I will briefly introduce some relevant related work. Next, in Chapter 3, I will provide the underlying theory of the techniques and methods implemented, as well as a detailed overview of my approach. Chapter 4 describes the experimental setup, datasets used, as well as results obtained. Finally, in Chapter 5 several conclusions are drawn and possible directions for future work are presented.

Chapter 2

Related work

The related work most relevant to this thesis consists of [1] and [2]. The framework of the first paper is used for solving the tracking task and the approach of the second paper represents a starting point for my method. Since they play an important role, both will be described in more detail in Section 3.2 and Section 3.3.1, respectively. This chapter briefly presents other related work addressing similar problems.

Tosato et al.[5] address the problem of human orientation estimation by introducing a novel descriptor, Weighted ARay of COvariances (WARCO). This descriptor is based on the covariance of the features, which has been previously used for pedestrian detection. The improvements implemented through WARCO make the classification of human targets possible according to their orientation, in the context of their appearance being encoded by few, noisy pixels. To be able to apply standard machine learning techniques for classification, the covariance matrices relying in a Riemannian manifold need to be projected onto a unique tangent space. For this task, the authors introduce a new measure to compute distances between projected points, better preserving the original geodesic distance. Furthermore, this novel approach greatly improved the computation times needed for the computation of distances, over previous methods. Once projected, the points are classified into 4 classes (corresponding to the front, back, left and right orientations) using a Support Vector Machine classifier.

An approach more closely related to the method described in this thesis is introduced in Lu et al. [6]. The authors consider a template-based framework for tracking and recognizing athletes' actions using only visual information. The considered targets are encoded with a PCA-HOG descriptor, obtained by applying Principal Component Analysis (PCA) to the Histogram of Oriented Gradients (HOG) descriptor. This ensures a robust representation under variations in illumination and scale, while at the same time keeping computational costs low. The tracking and action recognition are merged into a single task, solved by using a hybrid Hidden Markov Model with two first-order Markov processes. The first is responsible for the estimation of the templates encoding the actions of the targets, while the second is responsible for keeping track of their position.

In their work, Smith et al. [3] address the problem of tracking and estimating the head pose of multiple targets in a video sequence, task defined by the authors as finding the visual focus of attention for a varying number of wandering people (VFOA-W). Their approach is based on a dynamic Bayesian network responsible for simultaneously estimating the number of people in a scene, their body and head locations, as well as their head pose. An efficient exploration of the variable-dimension state space is achieved by using a Reversible Jump

Markov Chain Monte Carlo sampling scheme, in a similar fashion to the one described in [1]. Based on the head pose and location information, the authors propose two models in order to determine if a person is looking or not towards an outdoor advertisement, which constitutes an application of their method. The first is based on a Gaussian Mixture Model, while the second relies on a Hidden Markov Model to take into account the temporal dependencies between focus states.

Munder et al. [7] have run an in-depth experimental study on the task of pedestrian detection, assessing the performance of various features and classifiers. The features considered aimed at highlighting the differences in performance between global and local features, as well as between adaptive and non-adaptive features. Thus, the features tested were PCA coefficients (global, non-adaptive), Haar Wavelets (local, non-adaptive) and Local Receptive Fields (local, adaptive). The classifiers used were the Support Vector Machine, a popular method for a wide range of classification problems, a feed-forward neural network and, as a baseline, k-nearest neighbors. The best results were obtained using the adaptive local features and the SVM classifier.

The final decision of choosing the works of [1] and [2] as a starting point for my thesis is motivated by the more complete and specific approaches addressing the problem presented in Chapter 1. In contrast, [5] addresses only the classification task based on visual features, while [7] is focused more on evaluating the performance of different techniques, rather than addressing a specific problem as a whole. [6] is concerned with both tracking and classification, but focused on a more slightly different problem, namely the classification of athletes' actions for improving tracking performance.

Chapter 3

Theory

In this chapter I will present my approach from a theoretical standpoint. I will start by presenting a general overview of the method, emphasizing the process pipeline. Next, I will present in more detail the two necessary tasks: human tracking and orientation estimation. Human tracking is done using the method proposed in [1] and is described in Section 3.2. The orientation estimation is described in Section 3.3.

3.1 Overview

3.1.1 Problem description

For a better understanding of the following chapters, I will clearly define the goals of the method and the expected circumstances under which it will be applied.

The input data of the system is represented by a video sequence from a single, monoscopic, moving camera, depicting one or more human targets moving unconstrained into, within and away from the scene. The goal of the system is to estimate the orientation angle around the vertical axis (i.e. yaw) of the body and head for each human target at each frame of the video. The output values of the angle are discretized into 8 distinct classes: {0, 45, 90, 135, 180, 225, 270, 315} degrees or, alternatively, {E, NE, N, NW, W, SW, S, SE}, as depicted in Figure 3.1. Additional data for training of the classifiers consists of an annotated dataset with body and head samples.

3.1.2 Pipeline

Two variations of the proposed approach for orientation estimation are presented, which will be referred to as 'Method 1' and 'Method 2'. The classifiers and additional cues used in both of them are the same, but the way in which they interact is different, as indicated in the following pipeline description.

Method 1

The processing pipeline for estimating the body orientation using the first method can be summarized in the diagram presented in Figure 3.2, where the detailed classification block is shown in Figure 3.3.

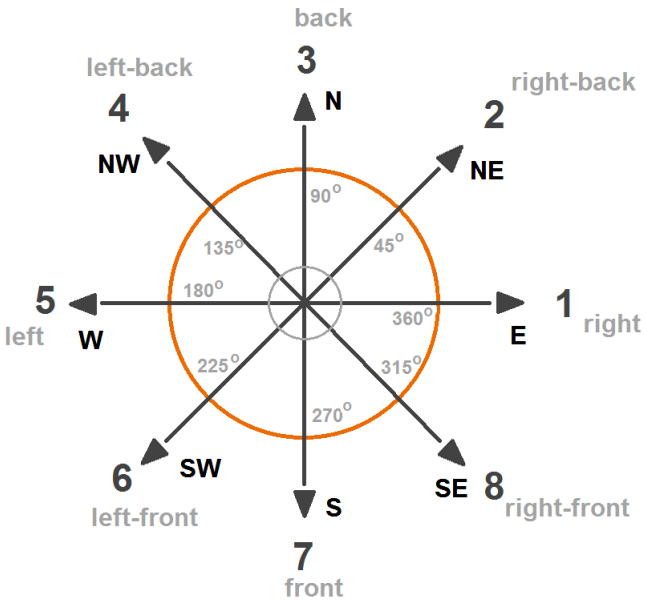

Figure 3.1: Discretization of angles into 8 classes.

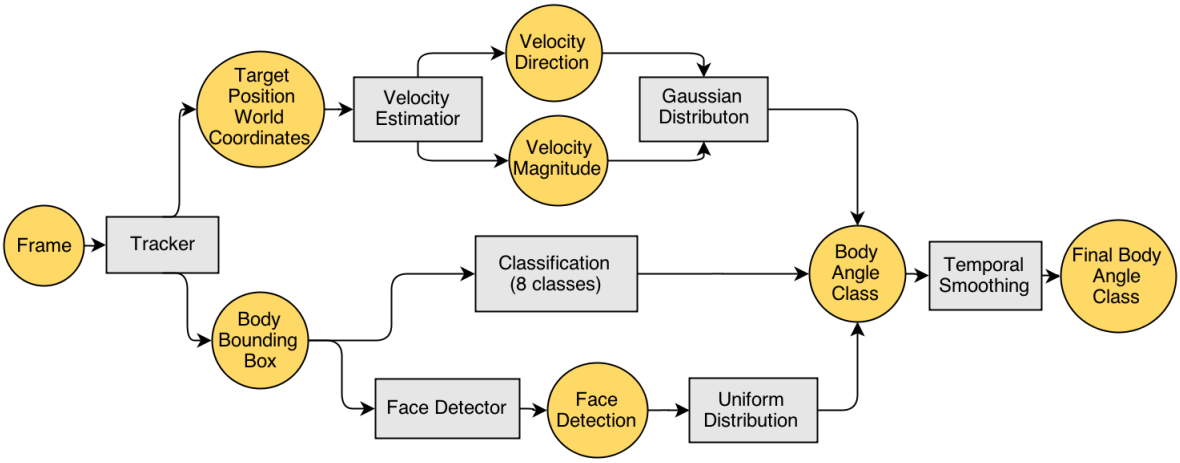

Figure 3.2: Pipeline for body angle estimation using method 1.

The input of the system is represented by video frames upon which human tracking is performed using the method described in [1]. This is preferred over other tracking methods as the video sequences come from a single moving camera and, apart from improved stability and overall performance, it is able to provide estimates of the positions of human targets in the real world coordinate system (not only bounding boxes in the image coordinate system). This additional information is particularly useful in determining the velocity direction and magnitude of the targets, an important cue useful in a later stage.

Apart from the coordinates of the targets, the tracker also returns bounding boxes designating the regions in the frame containing the targets. From these regions HOG descriptors are extracted, encoding the visual appearance of the individuals. These are then supplied to several pre-trained classifiers which output probability estimates for each of the 8 angle classes.

Also performed on the regions specified by the bounding boxes is face detection. This is also an important cue, restricting the plausible angle values, if a face is detected. To maintain

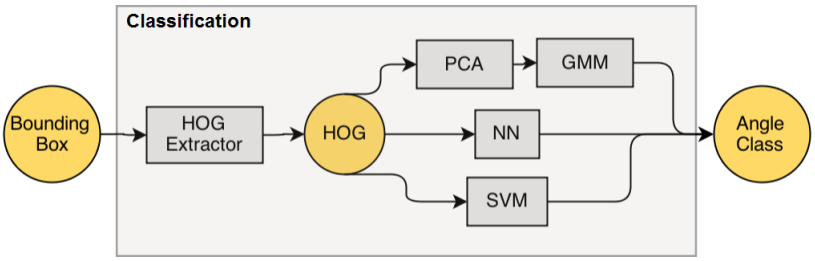

Figure 3.3: Classification block from pipeline.

the consistency of the probabilistic framework, an uniform distribution is generated, whose parameters are determined by the presence or absence of a face detection.

The previously determined velocity is integrated in the framework by fitting a standard Gaussian distribution centered around the velocity direction angle class whose variance is scaled inversely proportional to the velocity magnitude. Thus, relatively high velocity would yield a high probability for the angle class corresponding to the velocity direction and low for the other angles, while a relatively low velocity would yield an almost constant velocity for all angle classes.

The response from all the above classifiers and additional cues are combined and the estimated angle is considered to be the one with the highest probability. However, the final result is filtered using a sliding window. This additional step is performed to insure the temporal smoothness of the change in velocity and to minimize the effect of punctual misclassifications.

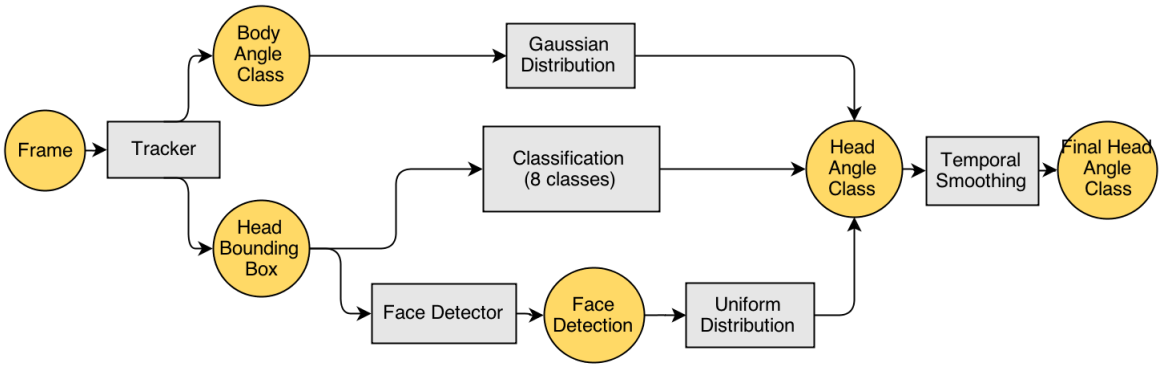

Figure 3.4: Pipeline for head angle estimation using method 1.

The head angle estimation is similar to the body angle estimation, as shown in Figure 3.4, with the exception that the velocity direction is not used. Instead, the body angle estimation is used in a similar fashion, by mapping a standard Gaussian distribution centered around the estimated angle class of the body. This is introduced to model the coupling between the head and body.

Method 2

The formulation of the second method is based on the observation that the appearance of human targets is similar for the diametrically opposed angle classes (i.e. angle classes 180 degrees apart). This can be observed visually, as indicated in Figure 3.5 and can also be confirmed by examining the HOG descriptors. As it will be described in Subsection 3.3.2, each target is described by a 2268-dimensional feature vector. Obviously, this cannot be

visualized directly. By using Principal Component Analysis (PCA), these feature vectors can be reduced to 2 dimensions which can be plotted. Such a plot, for body HOG descriptors, can be observed in Figure 3.6. Although not fully separable (due to the loss of information inherent to the dimensionality reduction process), clear clusters for angle classes are visible. Also noticeable is the fact that most of the points belonging to diametrically opposed classes (plotted in common colors with different symbols) seem to be distributed around the same clusters. This confirms to some degree the visual observation of the opposed angle classes being similar in appearance.

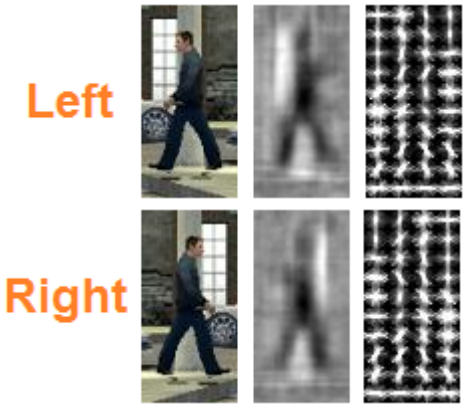

Figure 3.5: Similarity between HOG features and human appearance on diametrically opposed classes.

Following the above described observation, the pipeline of method 2 is described in Figure 3.7.

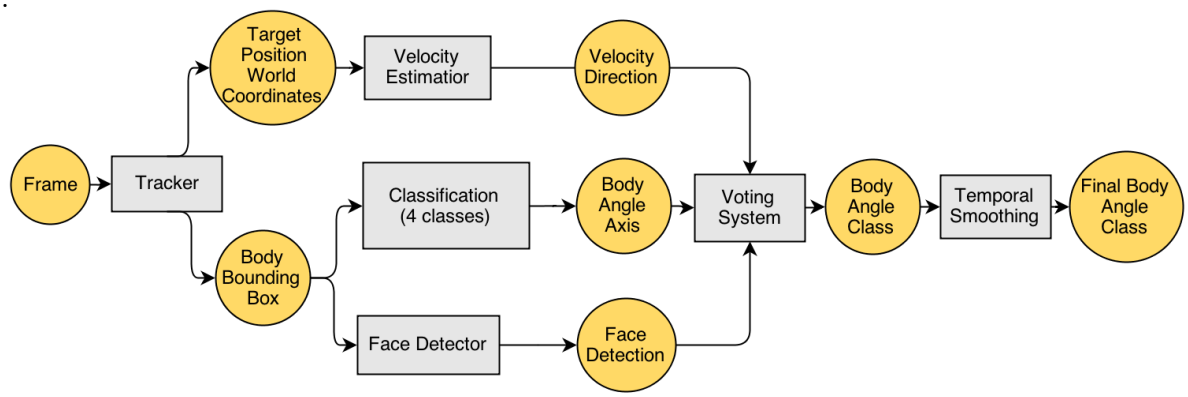

Figure 3.7: Pipeline for body angle estimation using method 2.

In the first stage, human tracking is done in the same way as previously described for method 1.

The main difference occurs in the way the HOG based classifiers are trained and used. Since there is little distinction between the HOG descriptors of diametrically opposed classes, I have decided to relabel the training dataset in a way such that diametrically opposed angle classes receive the same label (basically describing 4 axes: N-S, E-W, NE-SW, NW-SE). This way, the HOG classifiers will output probabilities for each of the 4 axes. The results are combined and the axis with the highest probability is selected.

Determining which angle class of the two described by the previously determined axis

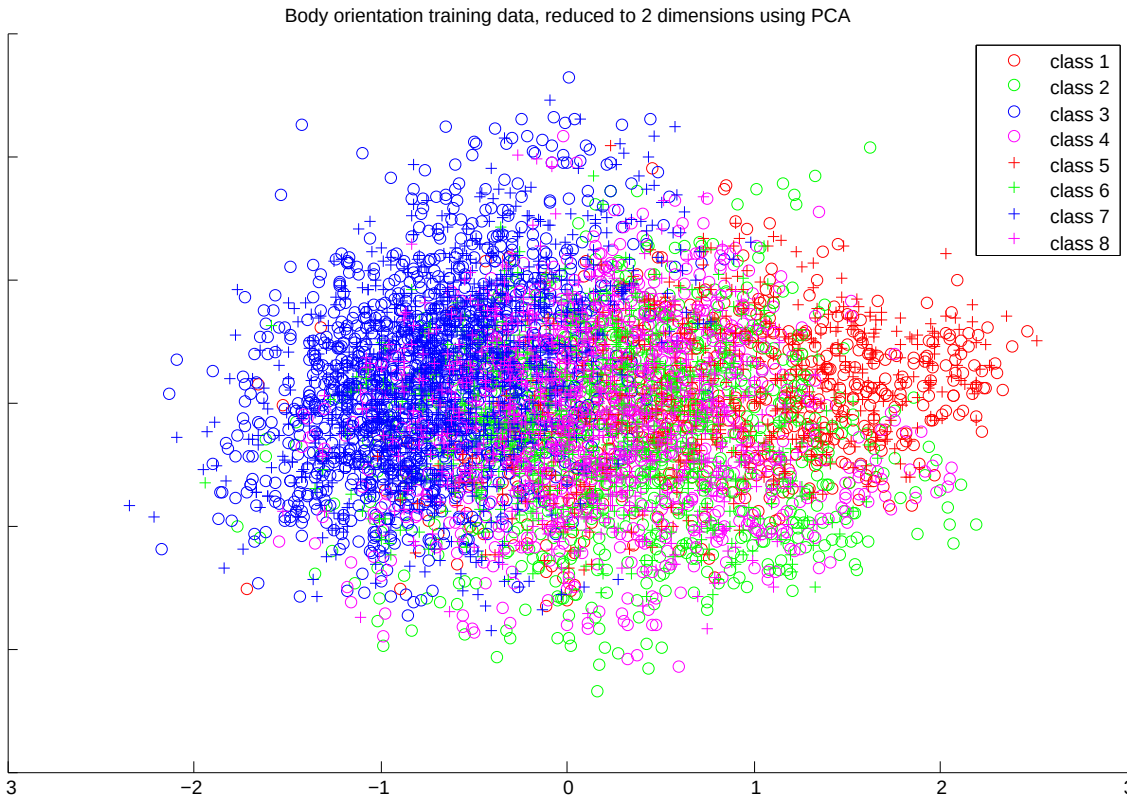

Figure 3.6: Body HOG descriptors reduced to 2 dimensions using PCA.

is the correct one is done by considering the additional cues, namely the velocity direction and face detection, through a simple weighted voting scheme. Thus, the vote of the velocity direction cue is determined by selecting the axis angle closest to the velocity direction. The vote of the face detection (if such a detection is made) is determined by selecting the axis angles under which the presence of a face is plausible. This translates in selecting the S, SE, SW angles if their corresponding axes are determined by the HOG classifiers. In the case of the E-W axis, the type of the face detection is taken into consideration (left or right profile). In this case, if the detection is frontal, none of the angles is preferred (the decision thus being influenced only by the velocity direction). The weighting of the vote is necessary for breaking ties in case both cues are available and their responses are different. Thus, since the face detection is usually more reliable than the velocity direction (which may be inaccurate due to noisy coordinate estimates for the targets' positions), the face detection vote has a higher weight, and it will take precedence in the case in which it is different than the vote based on the velocity.

Lastly, the temporal smoothness of the orientation change is achieved in the same way as for method 1.

It is important to note that the head orientation computation is the same as in method 1, since the assumption of resemblance between diametrically opposed classes does not hold for the appearance of the human head. This can be observed in Figure 3.8, where the clusters

corresponding to diametrically opposed angle classes do not overlap as in the case of the body appearance.

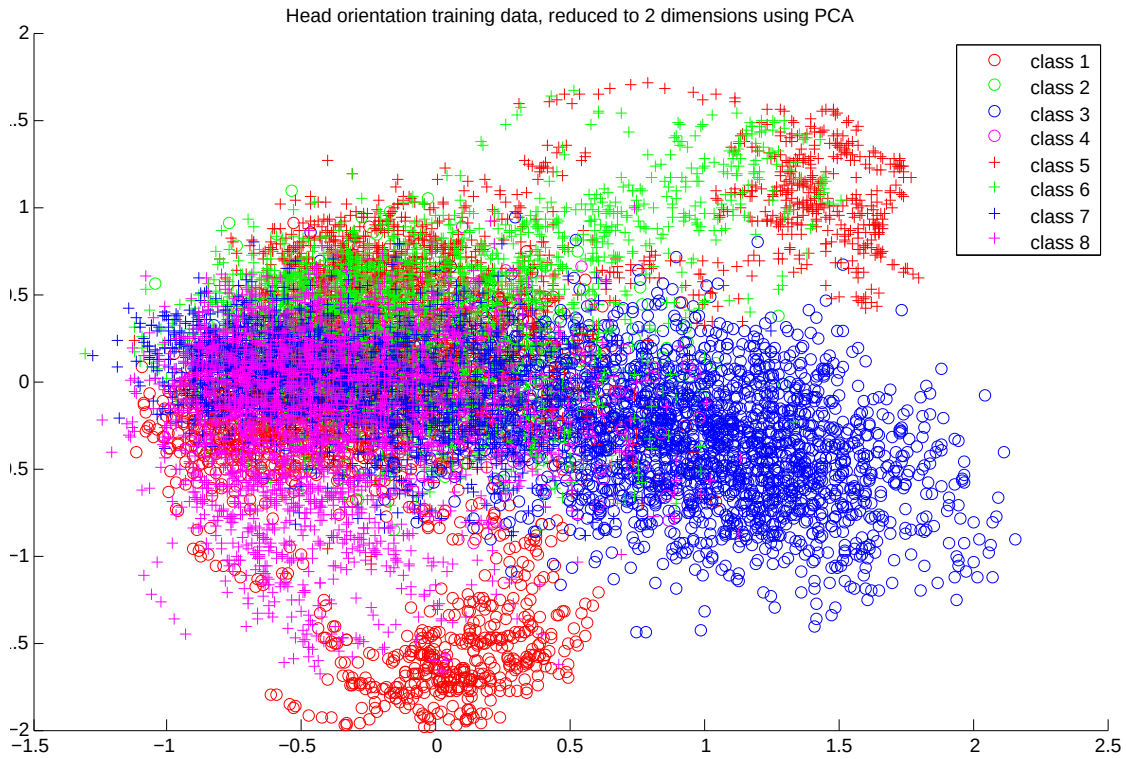

Figure 3.8: Head HOG descriptors reduced to 2 dimensions using PCA.

3.2 Human tracking

As shown in Section 3.1, the first step in the orientation estimation is the tracking of targets. Since my framework takes into account higher level information such as velocity orientation, this task is performed using the technique described in [1]. The goals of their framework are to track the movement of human targets in a video sequence, as well as to determine their position in the 3D coordinates of the world, along with the movement characteristics of the camera.

The assumptions made by the authors include the availability of weak detections hypothesis and initial camera parameters such as focal length, the estimated height in the frame of the horizon line and camera height. Also, all stationary features aiding in the tracking process are assumed to be on the ground (i.e. below the horizon line) and the rotation of the camera is restricted on the vertical axis (i.e. yaw).

The workflow of the method is summarized in the diagram from Figure ??.

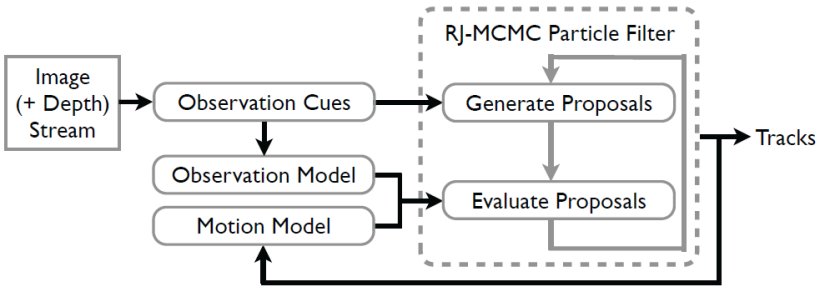

Figure 3.9: Workflow diagram of the approach from [1]

3.2.1 Model representation

Formally, the output of the algorithm for each time step t is a configuration $\Omega_t = \{\Theta_t, Z_t, G_t\}$ consisting of several elements. $Z_t = \{Z_t^i\}$ represents the set of positions for each of the targets, expressed as points in the 3D world coordinate system, $G_t = \{G_t^j\}$ represents a set of stationary features aiding the tracking process and $\Theta_t = \{f, u_c, v_h, \Phi, \mu, h_\Theta, x_\Theta, z_\Theta\}$ represents camera parameters, namely the focal distance f, the image center u_c, the horizon line v_h, the yaw angle Φ, velocity μ and the position in the 3D world coordinates $h_\Theta, x_\Theta, z_\Theta$. This output is defined as the MAP solution:

$$\hat{\Omega}_t = \arg\max_{\Omega_t} P(\Omega_t | I_{1,\dots,t}) \tag{3.1}$$

where $I_{1,\dots,t}$ represents the set of images (frames) from time step 1 to t. The posterior probability $P(\Omega_t | I_{1,\dots,t})$ is formulated using a sequential Bayesian framework:

$$P(\Omega_t | I_{1,\dots,t}) \propto \underbrace{P(I_t|\Omega_t)}_{(a)} \int \underbrace{P(\Omega_t|\Omega_{t-1})}_{(b)} \underbrace{P(\Omega_{t-1}|I_{1,\dots,t-1})}_{(c)} d\Omega_{t-1} \tag{3.2}$$

where the first term (a) represents the observation likelihood, (b) represents the motion prior and (c) the posterior probability at time $t-1$.

3.2.2 Tracking with RJ-MCMC

This distribution is computed using a sampling method, Reversible Jump Markov Chain Monte Carlo (RJ-MCMC), as the exact computation of $P(\Omega_t | I_{1,\dots,t})$ distribution is unfeasible because of the high and changing dimensionality of Ω_t. Because of the special nature of the configuration space, an efficient exploration can be done through Reversible Jump moves.

Thus, the state that maximizes the posterior configuration is approximated using a number of samples (each of them being indexed by r):

$$P(\Omega_t | I_{1,\dots,t}) \approx \{\Omega_t^{(r)}\}_{r=1}^N \tag{3.3}$$

The basic sampling mechanism involves the generation of each new sample upon the previous one by randomly choosing and perturbing one of its elements, namely one of the targets, geometric features or camera parameters. A rejection mechanism insures that the samples get closer to the real distribution.

A visual intuition of how this works is represented in Figure 3.10.

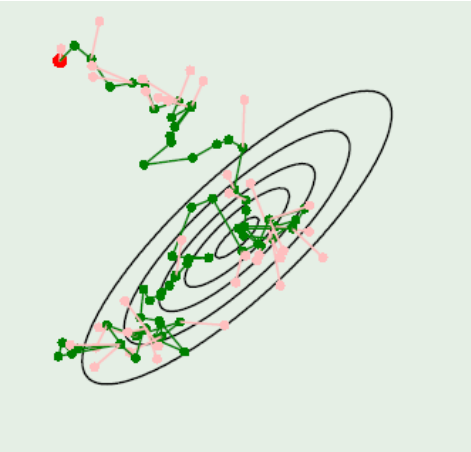

Figure 3.10: Visual representation of the Metropolis Hastings sampling algorithm. The real distribution is supposed to be a 2D Gaussian. The first generated sample is represented with red. It can be observed that each subsequent sample is generated from the last one, the pink being rejected (as it deviates too far from the real distribution) and the green being accepted.

Proposal distributions

The generation of each sample is done according to the proposal distribution $Q(\Omega_t^{'}, \Omega_t)$. This has 3 components corresponding to perturbations of either targets, features or camera parameters.

$$Q(\Omega_t^{'}, \Omega_t) = q_Z Q_Z(\Omega_t^{'}, \Omega_t) + q_G Q_G(\Omega_t^{'}, \Omega_t) + q_\Theta Q_\Theta(\Omega_t^{'}, \Omega_t) \tag{3.4}$$

The proposal distribution for perturbing the targets ($Q_Z(\Omega_t^{'}, \Omega_t)$) and the geometric features ($Q_G(\Omega_t^{'}, \Omega_t)$) are defined using so called reversible jump moves. These are a sort of operators describing how the next sample is different from the previous one. To ensure the convergence to the real distribution each of these moves must be a reversible counterpart of another move. Because of the distinct structure of the camera parameters, the camera proposal $Q_\Theta(\Omega_t^{'}, \Omega_t)$ is just a normal distribution. Furthermore, q_Z, q_G, q_Θ represent the probabilities of perturbing each of the components.

Target proposal

For perturbing the target set, a random target is selected from the set as well as one of the six reversible jump moves (Add, Delete, Stay, Leave, Update, Interaction Flip). A visual intuition on the effects of these moves can be observed in Figure 3.11, which describes how each new target $Z_t^{(r+1)}$ is generated from current sample $Z_t^{(r)}$, using each of the moves.

The *Add* move guarantees that if a target is not present in the sample $Z_t^{(r)}$, it is added in $Z_t^{(r+1)}$. Its counterpart, *Delete*, insures that a random sample is chosen and deleted from the set of targets that has a new detection in the current frame and is also present in the previous sample. The *Stay* move verifies if the target is present in the previous configuration, but not in the previous sample, it is added in the current sample. Analogously, the *Leave* move eliminates a target that was in the previous configuration and also in the previous sample. The *Update* move alters the position of targets according to a normal distribution. Note that the *Update* move is the reversible jump of itself. Lastly, the *Interaction flip* move switches the configurations of two targets.

Each of these moves has a different probability of being chosen, which are fixed and given as parameters of the system.

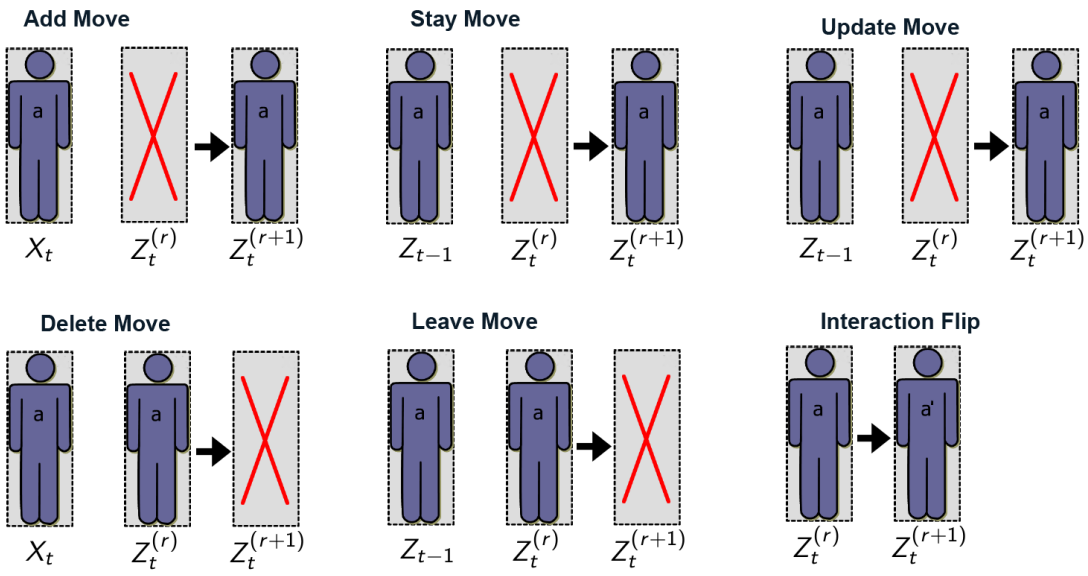

Figure 3.11: Reversible Jump Moves.

Geometric features proposal

Similar jump moves are defined for geometric features: *Stay, Leave* and *Update*, have the same behaviour as before. *Add* and *Delete* moves are not used since features are detected at each frame and their validity is determined only by comparing their successive locations. Each of the moves for geometric features has a different probability of being chosen.

Camera parameters proposal

Since there is a single camera, no reversible jump moves are defined for the camera parameters. These are perturbed using a normal distribution:

$$Q_\Theta(\Theta_t^{(r+1)}; \Theta_t^{(r)}) = \mathcal{N}(\Theta_t^{(r+1)}; \Theta_t^{(r)}) \tag{3.5}$$

Acceptance ratio

According to the Metropolis Hastings algorithm [10], the samples are accepted or rejected according to their acceptance ratio, which conceptually reflects how close a sample is to the real distribution, and it is formally defined as

$$a = \underbrace{\frac{P(I_t|\Omega_t^{(r+1)})}{P(I_t|\Omega_t^{(r)})}}_{(a)} \underbrace{\frac{P(\Omega_t^{(r+1)}|I_{1,2,..t-1})}{P(\Omega_t^{(r)}|I_{1,2,..,t-1})}}_{(b)} \underbrace{\frac{Q(\Omega_t^{(r)}; \Omega_t^{(r+1)})}{Q(\Omega_t^{(r+1)}; \Omega_t^{(r)})}}_{(c)} \tag{3.6}$$

where (a) represents the ratio between image likelihoods and reflects how likely the observed image is under the current configuration, based on projecting the targets and geometric features using the camera parameters and comparing them to the detections, (b) is the ratio between approximated predictions and (c) is the ratio between proposal distributions.

Detectors

For a more robust behaviour, multiple detectors can be used, which serve as weak hypotheses. The authors briefly describe 7 options for the detectors, namely Pedestrian and Upper body location based on HOG features / Deformable Part Model, Face Detector, Skin Color Detector, Depth-based Shape Detector, Motion Detector and Target Specific Appearance-based tracker.

It should be noted that in isolation none of the detectors performs satisfactory, but combining them can generate more reliable results.

To summarize, the framework presented is capable to track the positions of multiple targets from frame to frame, based on weak detection hypothesis, as well as to estimate the movement characteristics of the camera and to model interactions between targets. The feature space is explored in an efficient manner, through the use of specialized operators, reversible jump moves and insuring good real time performance.

3.3 Orientation estimation

As the approach presented in this thesis starts from the core ideas introduced in [2], I will briefly present their original approach in the following subsection.

3.3.1 Original method of Chen et al.

The method presented in [2] assumes that bounding boxes for the bodies and heads of the targets are given and information regarding their velocity direction and velocity magnitude are known. Additionally, an annotated dataset for both body and head descriptors is assumed to be available. The final result of the algorithm consists of orientation estimations for head and body, discretized into 8 classes, as previously mentioned in Section 3.1.1.

The workflow of their method can be summarized in the diagram from Figure 3.12. The image flow from video is provided to a tracker which provides information regarding the bounding boxes of the targets, from which body and head features are extracted, as well as movement characteristics, namely velocity information. These features are provided to a classifier that also makes use of the annotated dataset and outputs the desired results. The classifier is described as a coupled adaptive classifier, considering 2 datasets, one being labeled, while the other one consists of the targets for which the orientation estimation is computed, and thus unlabeled.

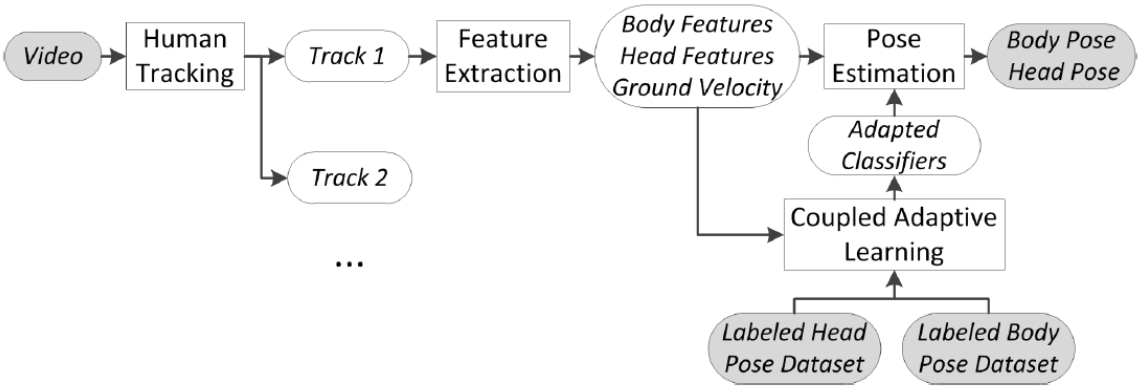

Figure 3.12: Workflow of Cheng et al. approach [2].

The classification problem is formulated within a kernel-based framework. Thus, the features describing the body of the targets are projected in a high dimensional Reproducing Kernel Hilbert Space $\mathcal{F}^b$ through a non-linear mapping $\Phi^b : \mathbb{R}^{d_b} \rightarrow \mathcal{F}^b$, where d_b represents the number of body features. Under this framework, the classifier is defined as a linear function $f^b : \mathbb{R}^{d_b} \rightarrow \mathbb{R}^8$:

$$f^b(\mathbf{x}^b) = \sum_{i=1}^{n_b+n_t} \mathbf{w}_i^b \phi^b(\mathbf{z}_i^b)^\top \phi^b(\mathbf{x}^b) = (\mathbf{W}^b)^\top [\mathbf{\Phi}^b, \tilde{\mathbf{\Phi}}^b]^\top \phi^b(\mathbf{x}^b) \tag{3.7}$$

where n_b represents the number of unlabeled targets, n_t represents the number of labeled data points, $\mathbf{\Phi}^b = [\phi^b(\mathbf{x}_1^b), ..., \phi^b(\mathbf{x}_{n_b}^b)]$, $\tilde{\mathbf{\Phi}}^b = [\phi^b(\tilde{\mathbf{x}}_1^b), ..., \phi^b(\tilde{\mathbf{x}}_{n_b}^b)]$, and $\mathbf{W}^b = [\mathbf{w}_1^b, ..., \mathbf{w}_{n_b+n_t}^b]^\top \in \mathbb{R}^{(n_b+n_t)\times 8}$. Lastly, $\mathbf{z}_i^b$ is a short-hand notation for both the labeled and unlabeled data, $\mathbf{z}_i^b = \mathbf{x}^i$ when $i \leq n_b$ and $\mathbf{z}_i^b = \tilde{\mathbf{x}}_{i-n_b}^b$ when $i > n_b$.

Given the kernel function $k(x_i^b, x_j^b) = \phi^b(x_i^b)^\top \phi^b(x_j^b)$, the training of the classifier translates into determining the optimal weights $\mathbf{W}^b$. This is done by optimizing an objective function which takes into consideration multiple factors:

$$E(\mathbf{W}) = E_l + \alpha E_m + \beta E_c^{bh} + \gamma E_c^{vb} + \lambda E_r \tag{3.8}$$

where $\mathbf{W} = \left[\left(\mathbf{W}^b\right)^\top, \left(\mathbf{W}^h\right)^\top \right]^\top$.

The E_l factor is responsible for insuring that the classifier function respects the labeled information and is defined as the discrepancy between the output of the classifier and the label measured on the labeled dataset. The E_m factor is responsible for modeling the smoothness of the classifier function over the manifold structure, a property more generally described as the fact that similar features should generate similar labels. The E_c^{bh} factor describes the coupling between the body and the head orientation and is defined as the discrepancy between the responses of the head and body classifiers. The E_c^{vb} factor models the coupling between the body orientation and the velocity direction, if the velocity magnitude is large enough. Finally, the E_r is a regularization factor, controlling the complexity of $\mathbf{W}$ for better generality. These factors are weighted with the non-negative parameters α, β, γ and δ.

The authors mention that the objective function $E(\mathbf{W})$ is convex and thus the optimal value for $\mathbf{W}$ is found by imposing:

$$\frac{\partial E(\mathbf{W})}{\partial \mathbf{W}} = 0. \tag{3.9}$$

It is worth noting that although the above equations refer to the body descriptor $\mathbf{x}^b$, they are all defined analogously for the head descriptor $\mathbf{x}^h$.

To be able to improve upon this method, and mainly to address the issue of having to compute a new classifier at each frame, which limits its use in an online setting, I have decided to build a new, different framework. Under my approach the classifiers are trained initially and offline, and are only used for predicting angles online, a task which is performed relatively fast. Although not combined in a single, general objective function, the additional cues such as velocity and body-head coupling are incorporated in the online decision process of estimating the angles.

3.3.2 Histogram of Oriented Gradients (HOG) descriptor

Before I describe the framework under which the classifiers and additional cues interact, as well as more in-depth information on each of the classifiers, I will describe here the features used for characterizing the visual appearance of the targets, as this represents the foundation upon which the orientation estimation is performed. This method is the same as used in [2].

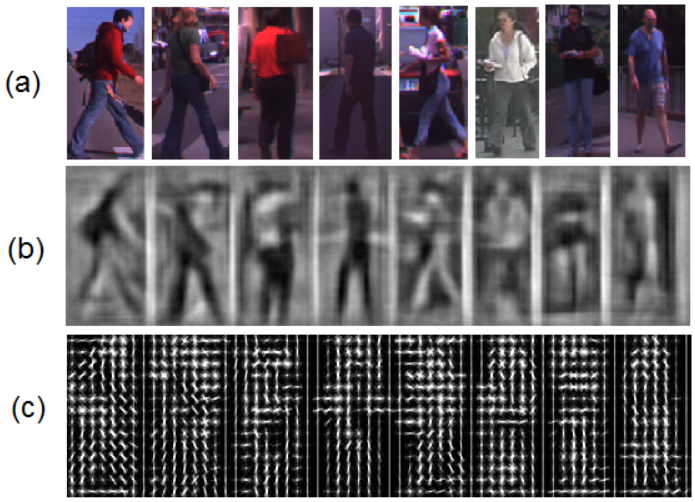

Figure 3.13: Illustration of the HOG descriptor. (a) the original images, (b) the inverted HOG features [4] back to natural images, (c) the HOG features.

The local human appearance and shape is invariant to illumination and can often be characterized rather well by the distribution of local intensity gradients or edge detections, even without precise knowledge of the corresponding gradient or edge positions. One such feature that captures edge and gradient structure is the Histogram of Oriented Gradients, introduced in [20] and illustrated in Figure 3.13.

The first step in obtaining the HOG features is the computation of the gradient values. Given an image I, the gradient magnitude $|G|$, and orientation of the gradient θ are extracted

as follows:

$$|G| = \sqrt{I_X^2 + I_Y^2}, \qquad \theta = \arctan\left(\frac{I_Y}{I_X}\right), \theta \in [-\pi, \pi] \tag{3.10}$$

where $I_X = I * D_X$, $I_Y = I * D_Y$, $D_X = [-1 \quad 0 \quad 1]$, $D_Y = [-1 \quad 0 \quad 1]^T$, symbol '*' representing the convolution operator.

The second step is the orientation binning and consists of creating the cell histograms. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the gradient centred on it, and the votes are accumulated into orientation bins over local spatial regions. The cells are rectangular and the histogram channels are evenly spread over 0 to 180 degrees, the gradient being unsigned. Dalal and Triggs show in [20] that increasing the number of orientation bins improves performance significantly up to about 9 bins. Regarding the vote, it is considered a function of the gradient magnitude at the pixel that gives the best results in practice.

The last step is to locally normalize the gradient strengths, which requires grouping the cells together into larger, spatially-connected blocks. The HOG descriptor is a vector of the components of the normalized cell histograms from all of the block regions. The block geometry used was rectangular and consists of a squared grid.

Before extracting features a histogram equalization was done over the cropped region of the image representing a target person in order to achieve better contrast and better capture the gradients. The equalization process implies mapping histogram of the given image section to a wider and more uniform distribution of intensity values. The intensity values are spread over the whole range. The equalization effect is accomplished through remapping of the cumulative distribution that is normalized such that the maximum value is the maximum value for the intensity of the image. So given the histogram $H(i)$, i representing intensity values, its cumulative distribution $H'(i)$ is $H'(i) = \sum_{0 \leq j < i} H(j)$. The result of the application of histogram equalization on the image can be seen in Figure 3.14.

Due to low resolution of both human body and head images, HOG features are extracted from multiple levels: for the body 3 levels were used ($1 \times 3, 2 \times 6, 4 \times 12$ blocks), while for the head 2 levels were used ($2 \times 2, 4 \times 4$ blocks). Each block is divided into 2×2 cells. Each cell accumulates a local 1D histogram of gradient directions (edge orientations) with 9 bins. The HOG descriptor length is the product between the number of blocks, the number of cells per block and the number of bins for the cells. Thus, the combination of the histograms results in two feature vectors of length $d_b = 2268$ for body and $d_h = 720$ for head.

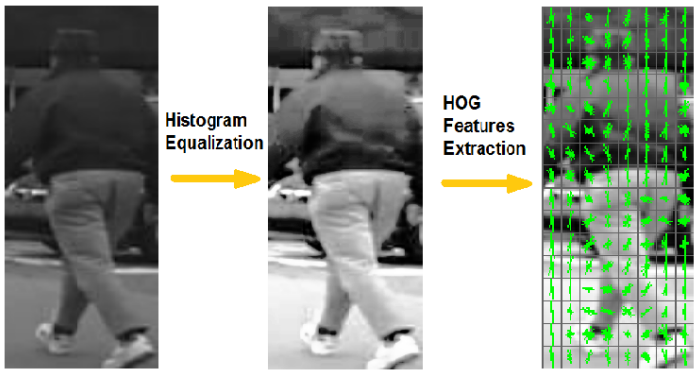

Figure 3.14: Histogram equalization over the region of interest is done before applying HOG extraction. The green star from each cell shows the strength of the edge orientation in the histogram.

3.3.3 Probabilistic framework

To achieve a greater flexibility in improving the method described in [2], I have decided to reformulate the problem under a Bayesian framework. Thus, the task of estimating the orientation of a particular target at a given moment in time (frame) can be expressed as

$$\hat{\alpha} = \arg\max_{\alpha} P(\alpha|\mathbf{x}) \tag{3.11}$$

where the α variable represents the desired angle class, having 8 possible values, while the $\mathbf{x} = (\mathbf{x}^b, \mathbf{x}^h, v_d, v_m, f_d)$ variable encompasses the information known about the target, namely its HOG features for the body ($\mathbf{x}^b \in \mathbb{R}^{2268}$), HOG features for the head ($\mathbf{x}^h \in \mathbb{R}^{720}$), velocity direction $v_d \in \{1, 2, ..., 8\}$, velocity magnitude $v_m \in \mathbb{R}$ and face detection $f_d \in \{0, 1, 2, 3\}$ (0 meaning no face detection, 1 meaning left facing face detection, 2 meaning frontal face detection and 3 meaning right facing face detection).

Furthermore, the a posteriori probability $P(\alpha|\mathbf{x})$ can be expressed under the Bayesian framework as:

$$P(\alpha|\mathbf{x}) = \frac{P(\mathbf{x}|\alpha)P(\alpha)}{P(\mathbf{x})} \tag{3.12}$$

To be noted is the fact that, assuming unconstrained movement of the targets, all orientation values have the same probability $P(\alpha)$. Thus, the Maximum A Posteriori solution described in equation (3.11) is equivalent to the Maximum Likelihood solution :

$$\hat{\alpha} = \arg\max_{\alpha} P(\alpha|\mathbf{x}) = \arg\max_{\alpha} P(\mathbf{x}|\alpha) \tag{3.13}$$

The likelihood $P(\mathbf{x}|\alpha)$ is determined by the combined response of the classifiers and cues mentioned in Section 3.1.2. This can be expressed as

$$P(\mathbf{x}|\alpha) \propto \exp(l_{GMM}(\mathbf{x}|\alpha) + l_{NN}(\mathbf{x}|\alpha) + l_{SVM}(\mathbf{x}|\alpha) + l_{velocity}(\mathbf{x}|\alpha) + l_{face}(\mathbf{x}|\alpha)) \tag{3.14}$$

where $l_{GMM}(\mathbf{x}|\alpha)$, $l_{NN}(\mathbf{x}|\alpha)$, $l_{SVM}(\mathbf{x}|\alpha)$, $l_{velocity}(\mathbf{x}|\alpha)$ and $l_{face}(\mathbf{x}|\alpha)$ denote the log-likelihood given by the Gaussian Mixture Model classifier, Neural Network classifier, Support Vector Machine classifier, velocity cue and face detection cue, respectively. Details on the definitions of each of these likelihoods are given in the following subsections.

My decision for using a combination of classifiers, rather than a single one is based on the argumentation given in [8], where the author points out that the overall error of the committee is at best $E_{\text{COM}} = \frac{1}{M} E_{\text{AV}}$ and at worst $E_{\text{COM}} = E_{\text{AV}}$, where E_{COM} denotes the error of the committee, E_{AV} denotes the average error of the classifiers and M denotes the number of classifiers in the committee. The best-case scenario happens only if the errors have zero mean and are uncorrelated. Obviously, for the proposed approach the errors will be highly correlated, since the same data is used for training. However, the way in which the data is used is different from one method to the other, assuring a certain degree of variability in the response of each classifier. Thus, at least some improvement on the overall error is to be expected.

Probabilistic framework for head orientation estimation

The above described framework applies for both the body and head orientation estimation, with the remark that in the case of the head angle estimation the velocity direction value is replaced by the estimated body angle. Additionally, in the previously mentioned formulae, the body HOG features $\mathbf{x}^b$ are replaced with the head HOG features $\mathbf{x}^h$.

3.3.4 Gaussian Mixture Model (GMM)

The first of the HOG based classifiers makes use of Gaussian Mixture Models to represent the classes in the feature space. Thus, for each of the 8 classes, a Gaussian Mixture is computed based on the data points in the training dataset belonging to that class. Thus, the likelihood associated with the GMM classifier is:

$$l_{GMM}(\mathbf{x}|\alpha) = \log P_{GMM}(\mathbf{x}|\alpha) = \log \sum_{j=1}^{C} \pi_j^{(\alpha)} \mathcal{N}(\mathbf{x}^b | \mu_j^{(\alpha)}, \Sigma_j^{(\alpha)}) \tag{3.15}$$

where C represents the number of components in a Gaussian Mixture, $\mathcal{N}$ denotes the Gaussian distribution, $\pi_j^{(\alpha)}$ are mixing factors and $\mu_j^{(\alpha)}$ and $\Sigma_j^{(\alpha)}$ represent the mean and covariance of each Gaussian distribution. Note that the subscripted indices mark the Gaussian within the Gaussian Mixture of a class, while the superscripted indices indicate the angle class (the corresponding Gaussian Mixture).

The fitting of each Gaussian Mixture onto the training data points of a given class is accomplished using the Expectation-Maximization (EM) algorithm. This is an iterative general optimization algorithm, whose goal in particular for the GMM is to maximize the likelihood of the training data points with respect to the parameters, consisting of means and covariances of each component, as well as the mixing coefficients.

An outline of the algorithm, as presented in [9], is given below:

1. Initialize the means μ_k, covariances $\sum_k$ and mixing coefficients π_k, and evaluate the initial value of the likelihood.

Note: because the EM algorithm takes many more iterations to reach convergence compared with the K-means algorithm, and each cycle requires significantly more computation, we run the K-means algorithm to find a suitable initialization for a Gaussian mixture model that is subsequently adapted using EM.

2. **E step.** Evaluate the responsibilities using the current parameter values. The weighting factor for data point x_n is given by the posterior probability $\gamma(z_{nk})$ that component k^{th} was responsible for generating data point x_n.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

3. **M step.** Re-estimate the parameters using the current responsibilities

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$$

4. Evaluate the log likelihood

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

To illustrate the final result of the EM algorithm, Figure 3.15 depicts the projection in 2 dimensions of some of the training datapoints for the classes 1 and 3, along with the fitted GMM for each of the classes. Please note that Figure 3.15 is presented only for visualization purposes, as for the actual classification task, a projection in more than 2 dimensions will be used.

3.3.5 Principal Component Analysis (PCA)

One of the limitations of the GMM is the fact that the maximum likelihood estimation, the optimization target of the EM algorithm, is prone to yielding singular or near-singular covariance matrices if the data is high-dimensional, but residing on a lower dimensional manifold (which is usually the case in practice, according to [14]). This happens as a Gaussian distribution, part of the mixture, is driven towards modeling a single datapoint.

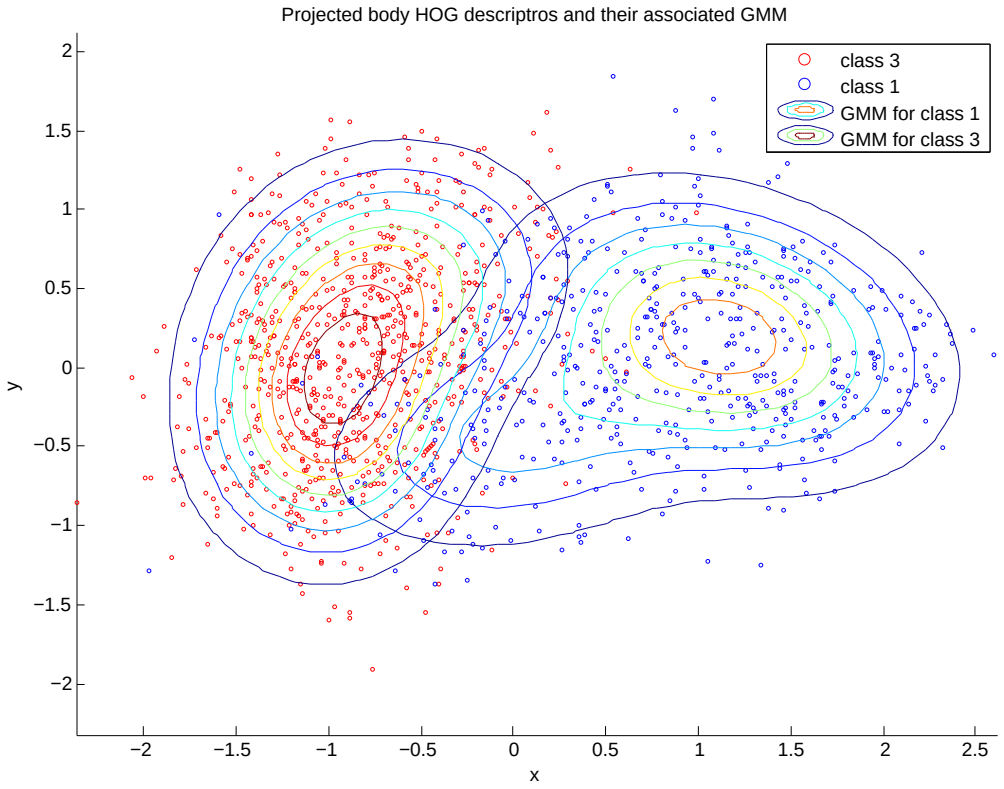

Figure 3.15: 2D projection of training body features for classes 1 and 3, along with their fitted GMM.

Another limitation of the GMM model is that fitting over high-dimensional data is a slow process. This is due to the fact that the EM training algorithm is iterative and at each iteration covariance matrices are computed for each Gaussian distribution in the mixture and these matrices are quadratic in the number of dimensions of the datapoints.

To mitigate the above mentioned limitations, I have decided to reduce the dimensionality of the HOG descriptors before using the GMM model for classification. One relatively simple method, but effective in practice (according to [15]), is Principal Component Analysis. PCA can be defined as the orthogonal projection of the data onto a lower dimensional linear space, such that the variance of the projected data is maximized [16]. Because the variance of the data is maximized, the separation between the points belonging to different classes is preserved as much as possible. Additionally, the PCA can discard the redundant and noisy information, thus improving the classification process.

3.3.6 Neural Network (NN)

The second HOG based classifier considered to be included in the committee is a Neural Network. I have chosen this method to address the high dimensionality of the data in a more natural way than the previously mentioned PCA. According to [11], the feed forward neural network, which I have used, can be regarded as an approach to fix the number of

basis functions (represented here by the individual neurons), but allowing them to be adaptive (represented by the connection weights between the neurons, which can be regarded as parameters adapted during training). Furthermore, it can be considered that the extraction of relevant features in the data and the classification process are merged together. The disadvantage raised by having this flexibility in automatically adapting the parameters of the method (weights) to the training data is the fact that the objective function optimized during training is no longer a convex function of the model parameters ([11]). This translates in a more lengthy training process, but the model, given its architecture, is fast at processing new data.

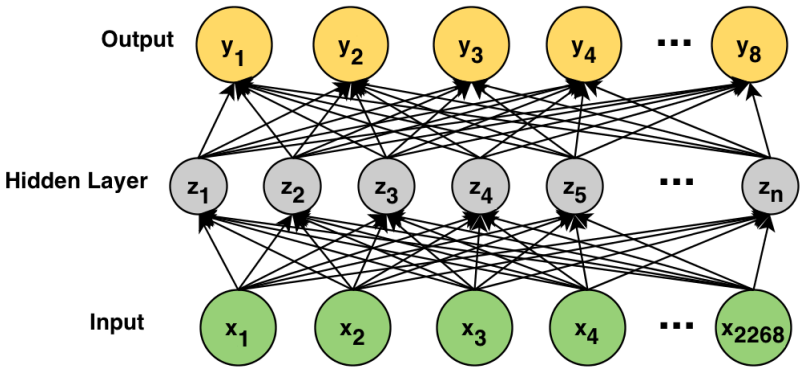

Figure 3.16: Structure of the Neural Netowk used.

Since the high dimensionality of the data does not represent an obstacle in implementing the Neural Network classifier, as it was the case for the GMM, in this case I have decided to use all the HOG features. Thus, the considered structure of the network, as depicted in Figure 3.16, has 2268 input nodes for the body classifier and 720 input nodes for the head classifier. The output of the network is given through 8 output nodes of the head classifier and the body classifier of method 1, and through 4 output nodes in the case of the body classifier of method 2. Each of these output nodes correspond to one of the possible angle classes. Please note that the response of each of the output nodes is not binary, but rather reflects the probability of the input data point belonging to a certain class (thus relying in the $[0 \dots 1]$ interval). Thus, denoting the response of the i-th output neuron (corresponding to the i-th angle class) of the network when presented with the HOG features of a target, as $P_{NN}(\mathbf{x}^b|\alpha)$, the corresponding term from equation (3.14) becomes:

$$l_{NN}(\mathbf{x}|\alpha) = \log P_{NN}(\mathbf{x}^b|\alpha) \tag{3.16}$$

The neurons in the hidden layer, as well as the output neurons have a Sigmoid activation function $f(t) = \frac{1}{1+e^{-t}}$, where t represents the input of the neuron, namely the weighted sum of the HOG descriptor in the case of the hidden neurons and the weighted sum of the responses of the hidden neurons in the case of the output nodes. The weights of the links between all neurons are computed during training using back-propagation. Because of the above described network structure, the expected response of the network is defined using the 1-of-8 encoding (or 1-of-4 encoding in the case of the body classifier for the second method) of the training labels.

3.3.7 Support Vector Machine (SVM)

The last of the HOG based classifiers considered is the Support Vector Machine. The motivation for this choice is the good performance obtained in various classification tasks, particularly in object recognition, where features such as the HOG descriptors are used.

The basic working principle of the SVM, as defined for a 2-class classification problem, consists of projecting the data points $\mathbf{x}$ into a high dimensional space, through the use of a mapping function $\phi(\mathbf{x})$, and then computing the weights of a linear classifier $y(\mathbf{x})$:

$$y(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b \tag{3.17}$$

such that

$$\begin{cases} y(\mathbf{x}_n) > 0 & \text{for all } t_n = +1 \\ y(\mathbf{x}_n) < 0 & \text{for all } t_n = -1 \end{cases} \tag{3.18}$$

where t_n represents the label of the n-th data point.

An additional constraint imposed to the linear classifier is the maximization of the margin between the two classes, i.e. the orthogonal distance between the decision boundary and the closest data points. As shown in [12], this translates into minimizing $\|w\|^2$ such that

$$t_n(\mathbf{w}^\top \phi(\mathbf{x}_n) + b) \geq 1 \tag{3.19}$$

which can be solved using Lagrangian multipliers:

$$\mathcal{L}(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{n=1}^{N} a_n \{t_n(\mathbf{w}^\top \phi(\mathbf{x}_n + b) - 1\} \tag{3.20}$$

Setting the derivative of $\mathcal{L}(\mathbf{w}, b, \mathbf{a})$ to zero we obtain the conditions

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \tag{3.21}$$

$$0 = \sum_{n=1}^{N} a_n t_n \tag{3.22}$$

Reintroducing these in equation (3.20), yields

$$\tilde{\mathcal{L}}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x_n}, \mathbf{x_m}) \tag{3.23}$$

where $\mathbf{a}$ is subject to the conditions

$$a_n \geq 0, n = 1, ..., N \tag{3.24}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{3.25}$$

and $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$ is a kernel function, thus eliminating the need of explicitly considering the mapping function $\phi(\mathbf{x})$.

Although at its core the SVM is a 2-class, hard assignment classifier, various techniques and algorithms have been developed for multi-class classification and probability estimates for each of the classes. For my approach, I am using the variant described in [17], which allows multi-class classification with soft assignments (probability estimates), as it can be integrated seamlessly in the probabilistic framework previously described. Thus, as the SVM is able to provide a probability estimate for a certain angle, given the HOG descriptors, denoted $P_{SVM}(\mathbf{x}^b|\alpha)$, the associated likelihood mentioned in equation (3.14) becomes:

$$l_{SVM}(\mathbf{x}|\alpha) = \log P_{SVM}(\mathbf{x}^b|\alpha) \tag{3.26}$$

3.3.8 Velocity

As also mentioned in article [2], the velocity direction often represents a cue for the body orientation. However, two factors affect the precision of this cue: the inaccuracy of the estimation of 3D position for the targets and the dependency on the speed of the target. The first disadvantage represents a limitation of the tracker and not much can be done to mitigate this effect. The second observation relies on the fact that a target with a high velocity has a lower chance of changing its orientation than one with a low velocity. In the framework of article [2], this is taken into consideration by only using velocity directions for the targets whose speed is above a certain threshold.

To make better use of both velocity direction and speed, as well as to incorporate this information seamlessly into the previously described framework, I have decided to build a pseudo-classifier by defining a Gaussian probability distribution centered in the angle class corresponding to the velocity direction and with a variance inversely proportional to the speed of the target. This way, in the case of a target moving with high speed, the probability of it facing the direction of the movement is relatively high, while a near-stationary target (whose speed is very low) will have a near equal probability for all angle classes, as the Gaussian with a high variance will be close to an uniform distribution across all angles.

$$l_{velocity}(\mathbf{x}|\alpha) = \log P_{velocity}(\mathbf{x}|\alpha) = \log \mathcal{N}(\alpha|v_d, 1/v_m) \tag{3.27}$$

where the v_d represents the velocity direction, v_m represents the velocity magnitude and $\mathcal{N}$ denotes the normal distribution.

3.3.9 Face detection

One inherent limitation of the classifiers based on HOG descriptors is the fact that, given the relatively low resolution of individual targets, the HOG descriptor is only able to represent the rough outline of the human body. This represents a problem, as usually the appearance

of the human body outline is very similar for diametrically opposed angles, as suggested in Figure 3.6. In such cases, a strong cue differentiating the two orientations is the presence of the face.

Face detection can be performed relatively fast, using for example a cascade Local Binary Pattern classifier [18], searching only upper section of bounding box. Furthermore, this classifier is able to provide information regarding the type of face detection, i.e. frontal, left-lateral or right-lateral, further aiding the orientation estimation process (as mentioned in the description of method 2 in Section 3.1).

Given the probabilistic framework described so far, a reasonable approach to model this information is by using an uniform probability distribution over the values of the angle corresponding to the body orientations in which the presence of a face is plausible. Thus, the associated likelihood becomes:

$$l_{face}(\mathbf{x}|\alpha) = \log P_{face}(\mathbf{x}|\alpha) \tag{3.28}$$

$$P_{face}(\mathbf{x}|\alpha) = \begin{cases} 1/5 & \text{if } f_d \neq 0 \text{ and } \alpha \in \{1,5,6,7,8\} \\ 0 & \text{if } f_d \neq 0 \text{ and } \alpha \in \{2,3,4\} \\ 1/8 & \text{if } f_d = 0 \end{cases} \tag{3.29}$$

Please note that the numerical values from the above equation correspond to the values of the uniform distribution. Thus, the first two lines correspond to the situation in which a face is detected ($f_d \neq 0$) and the probability is uniformly distributed over the 5 angles in which the face can be visible (first line), all other angles have a zero probability (second line). Lastly, if no face is detected ($f_d = 0$), the probability is evenly distributed among all angles (as the lack of a face detection does not necessarily imply the absence of a face in the image).

3.3.10 Temporal smoothness

Another cue for the orientation estimation is based on the fact that human targets do not usually change their orientation suddenly from frame to frame, especially considering the fact that frames succeed themselves at least at 1/24 seconds in most video sequences. This can be regarded as a temporal smoothness of the change in orientation angle. Thus, to restrict the abrupt changes in estimated orientation angles, I have decided to implement a sliding window approach in which the final estimated angle is determined by a majority vote from the angle estimations of the current frame and the past 5 frames (window size was determined empirically). If there is a tie between the angle class estimated for the current frame and another value, the former takes precedence.

3.3.11 Head coupling

As also mentioned in [2], the estimated head angle cannot be very different from the estimated body angle, because of anatomical constraints of the human body. This is modeled in the original approach using an additional term in the objective function, similar to the term used for taking into account the velocity direction.

Given the different framework formulation, I have also decided to include this additional cue in the similar fashion as the velocity, by fitting a standard Gaussian distribution centered

around the estimated angle of the body. To be noted is the fact that, unlike in the case of the velocity cue, the variance is not scaled anymore, as there is no additional cue equivalent to the velocity magnitude which formed the basis of the scaling in the former case.

3.4 Final remarks

To verify the validity of my approach, I have run several experiments, whose results are presented in Chapter 4. Given the multiple classifiers and cues used, the goal of the experiments is to assess the contribution of each of these elements towards improving the overall angle estimation.

Chapter 4

Experimentation

4.1 Meta-parameter estimation

As previously shown, the proposed method for orientation estimation has several meta-parameters which influence the quality of the classification. These meta-parameters are the number of dimensions to which the PCA reduces the HOG descriptors for the GMM, the number of components in each GMM, the number of neurons in the hidden layer of the Neural Network and the kernel type used for the SVM.

To determine suitable values for these meta-parameters, I have employed a k-fold cross-validation procedure using the available training dataset. Thus, for each parameter configuration of a given classifier, its classification accuracy was computed as an average over the values obtained by training the classifier with a fraction of $(k-1)/k$ of the dataset and estimating the accuracy on the remaining $1/k$ fraction of the dataset. The results of the cross-validation for each of the classifiers are given in the following subsections.

4.1.1 GMM validation

For determining the meta-parameters of the GMM classifier, namely the number of dimensions to which the PCA reduces the HOG descriptors to and the number of components in each mixture, I have employed 4-fold cross validation. The reason of choosing $k = 4$ instead of the more common $k = 10$ is the lengthy computation time needed for the entire procedure. However, despite the relatively low value for k, I still consider the results to be representative due to the relatively high number of annotated data points used in training. Thus, even if only 3/4 of the available data was used for training and 1/4 for estimating the error, the overall error values suggest that the classifier was able to exploit the patterns in the data set sufficiently well.

The results of the cross-validation are presented in Figure 4.1. It can be observed that for relatively low numbers of dimensions, the performance of the classifier is poor, as too much information is lost in the dimensionality reduction process, making robust classification difficult. The performance improves significantly after 20 dimensions and it stabilizes between 30 and 40 dimensions, suggesting that the high-dimensional HOG features relay in a lower, 40 dimensional, manifold.

The number of components in each mixture has less impact on the performance, when compared to the number of dimensions. However, the higher error obtained for a single component indicates that the data has a more complex structure than a simple Gaussian

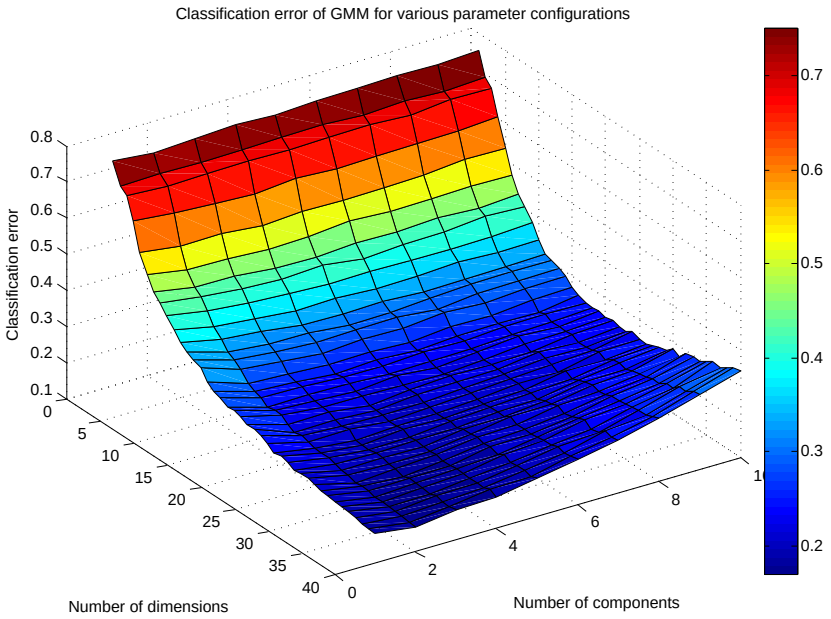

Figure 4.1: Classification error of the GMM classifier for various parameter configurations (namely the number of dimensions and the number of components), during validation stage.

distribution, while a high number suggests overfitting taking place, as the performance drops. The best values are obtained for 2-3 components per mixture, these providing the best approximation of the real structure of the data.

4.1.2 NN validation

The Neural Network classifier has a single parameter to be set, namely the number of neurons in the hidden layer. This made it possible to do a 10-fold cross-validation for more accurate results, instead of the previously employed 4-fold, as the computation times were reasonable.

The evolution of the classification error is shown in Figure 4.2. The decreasing evolution of the classification error is obvious, stabilizing after a value approximately 60 nodes. Although it is impossible to assess the role of each neuron and thus to provide a solid explanation for the correlation between the number of neurons and performance of the network, one can argue that this size of the hidden layer is influenced by the number of relevant features in the data, similarly to the minimum number of dimensions that yield reasonable good results (as shown in the previous section). Should that be the case, the activation of each neuron is more heavily influenced by one of these implicit relevant features.

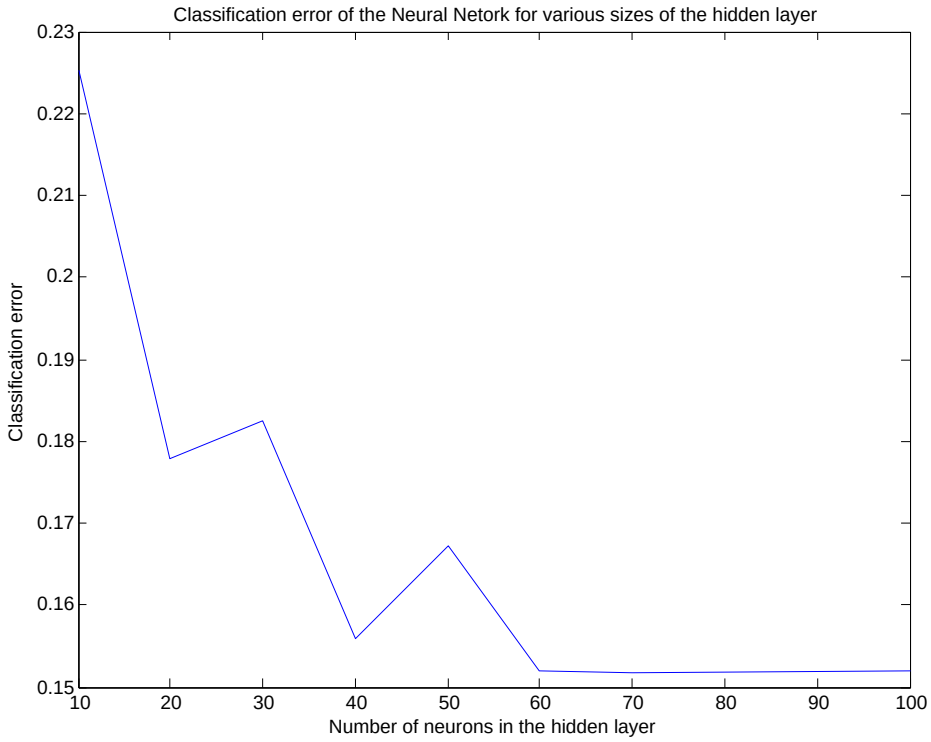

Figure 4.2: Classification error of the NN classifier for various parameter configurations, during validation stage.

4.1.3 SVM validation

For the SVM classifier, my initial intention was to also employ dimensionality reduction on the features, to obtain faster training times. However, after assessing the performance for various dimensions, as shown in Figure 4.3, and considering still manageable training durations, I have decided to use all 2268 HOG dimensions for the SVM classifier.

The plot from Figure 4.3 shows the evolution of the SVM classification error for various dimensions and using several kernel functions, obtained by 4-fold cross-validation. Having used the SVM implementation of the LIBSVM library [13], the parameters of these kernel functions have their default provided values. Thus, the kernel functions whose associated performance is depicted in Figure 4.3 are:

- Linear kernel : $K(u, v) = u^\top \cdot v$
- Polynomial kernel : $K(u, v) = (\gamma \cdot u^\top \cdot v + c)^d$, $\gamma = \frac{1}{\#\text{dimensions}}$, $c = 0$, $d = 3$
- Radial kernel : $K(u, v) = \exp(-\gamma \cdot \|u - v\|^2)$, $\gamma = \frac{1}{\#\text{dimensions}}$
- Sigmoid kernel : $K(u, v) = \tanh(\gamma \cdot u^\top \cdot v + c)$, $\gamma = \frac{1}{\#\text{dimensions}}$, $c = 0$

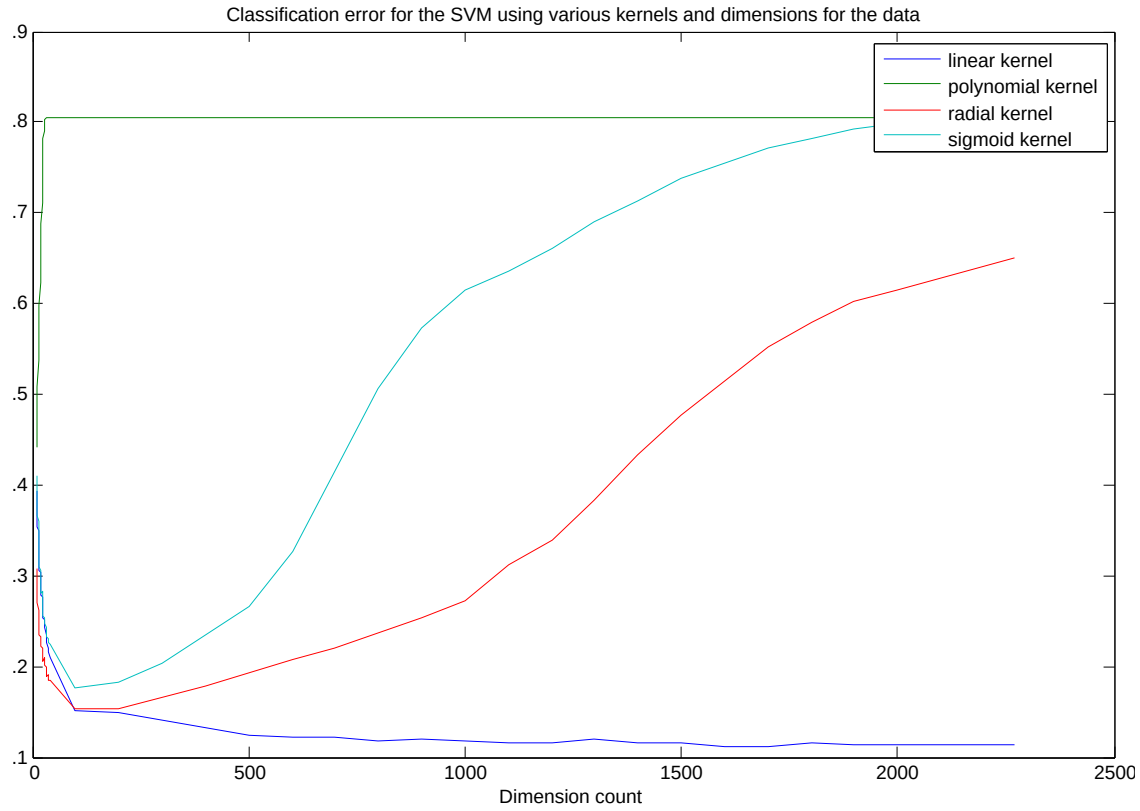

Figure 4.3: Classification error of the SVM classifier for various parameter configurations, during validation stage.

The evolution of the error observed was unexpected, as relatively low error values are yielded by the radial and Sigmoid kernels for a small number of dimensions, while for higher dimensions, under which less information is lost due to the dimensionality reduction, the error increases. The only kernel whose yielded error evolution follows an expected trend is the linear kernel.

The most plausible explanation for this behaviour is the fact that I have used the default parameter values of the kernels, as provided by LIBSVM, which may not be suitable for the higher dimensional representations of the data. This explanation is also confirmed by the fact that the only kernel behaving in an expected way is also the only kernel lacking any parameters. Should this be the case, a proper course of action would have been to also try various configurations for these parameters. These would have resulted in prohibitive computation times, and since the performance of the linear kernel is similar to the ones obtained by the other classifiers (judging by the classification error under optimal parameters), I would not have expected significantly better results from the other kernels.

Thus, for the final version of the SVM classifier, I have used a linear kernel without reducing the dimensionality of the data.

Table 4.1: Number of data points per class (specified in Figure 3.1) for the datasets used during training.

Dataset	Type	C1	C2	C3	C4	C5	C6	C7	C8	Total
TUD Multiview Pedestrian [22]	Body	400	749	644	749	400	622	545	622	4731
CVC Pedestrian [23]	Body	129	30	117	78	114	25	141	62	696
MIT Pedestrian [24]	Body	0	0	478	0	0	0	446	0	924
VIPeR [25]	Body	355	90	218	17	6	31	419	126	1262
Benfold [26]	Head	91	313	202	196	85	290	159	139	1475
HIIT6 [27]	Head	2000	0	2000	0	2000	2000	2000	2000	12000
QMUL [28]	Head	2256	0	2256	0	2256	0	2256	0	9024

4.2 Evaluation

4.2.1 Dataset description

During the training of the classifiers I have used several datasets, to have a greater variety of appearances. This, in turn, would be beneficial in achieving a better generalization of the training data and a good exploitation of the existing patterns, which in turn would support an efficient classification. Some characteristics of the datasets used during training are given in Table 4.1.

For testing my approach, I have used video sequences from the Collective Activity dataset [29]. These depict multiple human targets moving unrestricted in an urban environment. The ground truth annotation is available once every 10 frames, for the body only.

4.2.2 Setup

To assess the influence of each individual component of the method, I have decided to measure the error of the estimated angle class for various subsets of the classifiers and/or additional cues.

Thus, for the first method, I have considered versions of the algorithm involving only 1, 2 or all 3 of the HOG based classifiers with no additional cues, with only the velocity cue, only the face detection cue and with and without temporal smoothness.

Because of the nature of the second method, only versions involving various combinations of the HOG based classifiers were tested.

The error is measured as the angle difference between the estimated angle and the ground truth. The mean and standard deviation of the errors for all targets at all frames of a sequence are computed and presented in the following section.

4.2.3 Results and discussion

The results for the experiments run to evaluate the performance of method 1 are presented in Table 4.2.

The first goal of these experiments was to assess the contribution of each HOG-based classifier to the final angle estimation. Overall, the performances of the individual classifier were relatively similar, slight variations being observed in the response of each of them. This can be explained by the different ways in which each of them make use of the training data, being either projected through PCA (in the case of the GMM classifier), used as it is (in the case of the NN classifier) or projected in a higher dimensional space (in the case of the SVM classifier). The variability in the individual responses also insures the capability of a combination of classifiers to yield better results. A certain dependence on the video sequence can also be observed, as all the classifiers obtained better results on `Seq 42` than `Seq 15`. Since these classifiers take into consideration only the visual appearance of the targets, modeled by the HOG descriptors, the only explanation for this behaviour is the fact that the targets from `Seq 42` resemble more closely the targets used for the training of the classifier. This visual resemblance can further be explained by a closer similarity of the angle of the camera at which the images were captured, as well as a similarity of the resolution of the images.

The error obtained by combining the response of multiple classifiers proved to be better than the individual responses, if there is no large variation between the individual responses. Thus, in the case of `Seq 42`, all the combined responses yielded better results than the individual ones. As expected, the combinations including the more performant classifiers, such as GMM+NN, outperform the ones with the lower performing ones, such as GMM+SVM. In the case of `Seq 15`, the more pronounced poor result of the SVM classifier has a detrimental impact on the combined responses. Thus, only the GMM+NN combination has a better performance than any of its components, all others being roughly similar or even worse than the individual components.

The second goal of these experiments was to assess the impact of the individual cues considered. The performance of the method when only the velocity is used proved to be better than the responses of any of the individual or combined HOG-based classifiers, for the considered video sequences `Seq 15` and `Seq 42`, thus highlighting the importance of this additional cue. However, one might expect that for more particular video sequences in which the targets are mostly stationary, the velocity cue would provide less information and thus yield poorer results. The next configuration tested was the combination of the response of the HOG-based classifiers and the velocity cue. A significant improvement was observed over the response of the HOG-based classifiers, for both videos. However, in the case of `Seq 15`, where the HOG-based classifiers yielded poor performance, the overall result when taking into account the velocity cue was worse than in the case of using just the velocity. This was not the case for `Seq 42`, where the performance decreased dramatically, the mean error being lower than either of the constituents' responses.

Next, the face detection cue was assessed, also in combination with the response of the HOG-based classifiers. For `Seq 15` the performance improved in a similar fashion to the velocity cue, suggesting a similar informational gain. However, in the case of `Seq 42` the performance dropped over one of the HOG-based classifiers, most probably due to the high number of false face detections. When combining the two cues, velocity and face detection, the performance increases in the case of `Seq 15`, where the two cues taken individually generate similar results, while in the case of `Seq 42`, the performance is still lower than in the case of using just the velocity cue, due to the poor performance given by the false face detection.

The last element tested was the effect of the temporal smoothing. When combined with the response of the HOG-based classifiers, the performance increased, moderately for `Seq 15` and more significantly for `Seq 42`. The larger improvement in the second case can be explained

Table 4.2: Mean and standard deviation of the error for various versions of the method 1 of the approach, on two video sequences from Collective Activity dataset [29].

Method 1	Seq 15	Seq 42
GMM	63.0446/28.1862	65.5102/31.1502
NN	69.7277/30.5169	56.3265/29.4306
SVM	82.2030/33.6133	59.6939/30.1965
GMM+NN	61.7079/ 28.7315	52.6531/28.4356
GMM+SVM	63.0446/28.9955	55.4082/29.1968
NN+SVM	70.6188/31.0409	50.5102/27.5921
GMM+SVM+NN	66.8317/ 30.2219	54.1837/28.8972
GMM+SVM+NN + Temporal	61.9307/29.3805	40.4082/25.1518
Velocity only	43.8861/21.8800	44.3878/22.5420
Velocity + Temporal	46.7822/23.0858	36.7347/20.7998
GMM+SVM+NN + Velocity	48.3416/24.7429	36.4286/23.2556
GMM+SVM+NN + Face	47.6733/24.8099	59.0816/30.2519
GMM+SVM+NN + Velocity + Face	37.4257/21.1888	42.5510/25.3564
GMM+SVM+NN + Velocity + Face + Temporal	38.9851/22.0431	23.2653/18.9088

by a higher number of punctual misclassification, whose influence is reduced. When combined with only the velocity cue, the performance drops slightly for the first video, but increases for the second. This can be explained by a better velocity estimation in `Seq 15`, in which case the temporal smoothing only delays in response. The increase in the second case is also probably explained by punctual inaccurate estimations of the velocity. Similar trends are followed in the last configuration, involving all classifiers and cues, where the temporal smoothness factor has little influence on the performance from `Seq 15`, in which the estimations provided by the classifiers, the velocity and face detections seem to be more reliable. On the other hand, in the case of `Seq 42`, the performance increase is significant, as the error drops to almost half, due to the fact that punctual misclassification, inexact velocity estimation and false face detections, are smoothed out.

For the second method, fewer experiments were run, since by its nature, it relies heavily on the additional cues to determine final correct angle. Thus, the experiments focused on evaluating the performance of the classifiers and the influence of the temporal smoothness cue. The results obtained can be observed in Table 4.3.

The performance of the individual classifiers, as well as their combination, were either identical, as in the case of `Seq 15`, or very similar, as in the case of `Seq 42`. This is to be expected, since the HOG-based classifiers were trained to discriminate between only 4 classes, which were also more clearly separated in the feature space (as suggested in Figure 3.6). Thus, the classification task was easier, and the differences between the classifiers have a minor impact. The improvement made by the temporal smoothness cue is similar to the one observed in the case of the first method, minor for `Seq 15` and more significant for `Seq 42`.

Table 4.4 shows the error obtained by the complete method 1 and method 2 on several other video sequences. These sequences vary with respect to the duration, resolution, amount of camera movement, number of targets present in the frames, amount and variation of the targets' movement. These differences are also reflected in the results of the approach, with a

Table 4.3: Mean and standard deviation of the error for various versions of the method 2 of the approach, on two video sequences from Collective Activity dataset [29].

Method 2	Seq 15	Seq 42
GMM	34.7525/19.6425	48.9796/26.9089
NN	34.7525/ 19.6425	48.9796/25.5363
SVM	34.7525/19.6425	47.7551/26.3288
GMM+SVM+NN	34.7525/ 19.6425	47.1429/26.4905
All: GMM+SVM+NN + Temporal	33.4158/20.2192	35.8163/23.0919

Table 4.4: Error/Standard Deviation of method 1 and 2 of my approach for various video sequences from Collective Activity dataset [29]

Video	Method 1	Method 2
Seq 04	66.8382/25.2751	63.1985/24.1551
Seq 09	57.2485/25.1205	64.4379/27.3866
Seq 12	60.8084/23.5161	64.0419/24.4563
Seq 21	46.9565/16.4061	44.3478/15.1965
Seq 24	56.8681/22.3062	79.1209/28.5124
Seq 30	30.6250/18.2988	48.7500/22.6591
Seq 42	23.2653/18.9088	35.8163/23.0919
Seq 44	55.2273/21.4151	77.7273/27.7219
Seq 15	38.9851/22.0431	33.4158/20.2192
Seq 16	75.5447/28.1500	68.8827/25.9820
Seq 22	62.9348/24.2086	64.2391/24.9484
Seq 23	50.5618/24.9676	63.7079/28.5938
Seq 29	39.9057/24.4069	85.9670/37.4331
Seq 33	9.2308/7.7639	17.3077/10.0755

mean error ranging from 9.2308 degrees to 75.5447 for the first method and from 17.3077 to 85.9670 for the second method.

Unfortunately, the datasets used did not include ground truth values for the head angle orientations and thus the performance of the head angle estimation could not be evaluated. Furthermore, a direct comparison between my approach and the original approach of [2] could not be made, because of the lack of availability of the annotations used in their experiments. However, I have tested my own implementation of their approach on one of the videos evaluated for my approach, the results being presented in Table 4.5. I must mention that since the computation time is proportional to the squared number of training and testing data points, only a fraction of the training set was used. For a better comparison, the results of my approach (method 1 and method 2) using the same reduced training set are also presented.

Table 4.5: Evaluation of the performance on a video sequence (Seq 15 from Collective Activity dataset [29]) of the original method from [2] and both methods of my approach. The annotated data used represents only a fraction of the available training set (100 instances for each of the 8 classes for the body and 20 instances for each of the 8 classes for the head).

Method	Body label data	Head label data	Body Error/StdDev	Average processing time per frame (ms)
Original method of [2]	100 x 8	20 x 8	79.52/30.37	29948
Own approach (method 1)	100 x 8	20 x 8	56.58/22.73	73
Own approach (method 2)	100 x 8	20 x 8	48.11/20.55	57

Chapter 5

Conclusions and future work

5.1 Conclusions

In this thesis I have implemented and evaluated a novel approach for the task of body and head orientation estimation of freely moving human targets from a video sequence captured with a single monoscopic moving camera. The main ideas upon which this approach is built are inspired by the method described in [2], several contributions are made in order to improve its performance.

One contribution aims at improving the computation time required for usage of an online setting and consists in reformulating the problem under a different framework, allowing for a separate, offline training phase. This new framework also allows for the usage of multiple classifiers, their individual responses being combined for a more robust prediction. Another contribution refers to the usage of additional cues, such as face detections and temporal smoothness, as well as an improved method for taking into account the velocity cue.

The impact of each classifier and additional cue was evaluated and a discussion of the results was given. Only a brief comparison of the performance of the original method was possible, due to the unavailability of annotations used by the authors in their paper. Overall, the performance of the approach was good both in terms of angle error and computation time, and visually the results were satisfactory.

5.2 Future work

One limitation of the current method is the fact that the movement characteristics of the targets are only taken into account in a rudimentary fashion, by the velocity (computed based on possibly noisy 3D estimations of the targets' positions in the world) and the temporal smoothness cues. Thus, an improvement might concern the usage of an additional descriptor, specialized in encoding these motion characteristics. One such descriptor is the Histogram of Optical Flow, which takes into consideration the movement directions of various salient points situated on the target. These can prove especially useful in the case in which the target changes its facing while standing around a fixed point in space (and thus the velocity direction estimate would not be informative).

Another limitation is the poor detection of the head within the entire body bounding box. Although no annotations for the head bounding box were available for the video sequences used during testing, and thus no objective measure of how poor these detections were can

be made, visually the results indicated there is room for improvement. Thus, employing a different classification method, using different / additional descriptors and an improved training data set can yield better results.

The approach described in this thesis is built upon a new framework which aims at reducing the computation time required by the original method of [2]. Although this improvement was achieved, this new framework fails to capture an interesting and powerful feature of the original method, namely semi-supervised nature of the classifier. This refers to the usage of the characteristics of the evaluated targets in the construction of the classifier, thus adapting it to the input data. A direction for future work consists in exploring ways in which this characteristic can be also exploited in the newly introduced framework.

Bibliography

[1] W. Choi, C. Pantofaru, S. Savarese. "A general framework for tracking multiple people from a moving camera." (2012).

[2] C. Cheng, J. Odobez. "We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video." CVPR 2012 IEEE Conference on. IEEE, 2012.

[3] K. Smith, J.Odobez, S. Ba. "Tracking the visual focus of attention for a varying number of wandering people." Pattern Analysis and Machine Intelligence, IEEE Transactions on 30.7 (2008).

[4] C. Vondrick, A. Khosla, T. Malisiewicz, A. Torralba. "Inverting and Visualizing Features for Object Detection." (2012).

[5] D. Tosato, M.Spera, M.Cristani, V. Murino. "Characterizing humans on Riemannian manifolds." (2012).

[6] WL.Lu, JJ. Little. "Simultaneous tracking and action recognition using the pca-hog descriptor." Computer and Robot Vision, 2006. The 3rd Canadian Conference on. IEEE, 2006.

[7] S. Munder, DM. Gavrila. "An experimental study on pedestrian classification." Pattern Analysis and Machine Intelligence, IEEE Transactions on 28.11 (2006).

[8] CM. Bishop, "Pattern recognition and machine learning.", Chapter 14, p.655, *Vol. 1. New York: springer*, 2006.

[9] CM. Bishop. "Pattern recognition and machine learning.", Chapter 9, p.435. , *Vol. 1. New York: springer*, 2006.

[10] CM. Bishop. "Pattern recognition and machine learning.", Chapter 11, p.541, *Vol. 1. New York: springer*, 2006.

[11] CM. Bishop. "Pattern recognition and machine learning.", Chapter 5, p.227, *Vol. 1. New York: springer*, 2006.

[12] CM. Bishop. "Pattern recognition and machine learning.", Chapter 7, p.227, *Vol. 1. New York: springer*, 2006.

[13] CC. Chang, CJ Lin. "LIBSVM: a library for support vector machines." ACM Transactions on Intelligent Systems and Technology (2011).

[14] J. Wang. "Geometric structure of high-dimensional data and dimensionality reduction." Springer, 2012.

[15] IT. Jolliffe. "Principal component analysis." Springer verlag, 2002.

[16] H. Hotelling. "Analysis of a complex of statistical variables into principal components." The Journal of educational psychology (1933): 498-520.

[17] TF. Wu, CJ. Lin, RC. Weng. "Probability estimates for multi-class classification by pairwise coupling." The Journal of Machine Learning Research 5 (2004): 975-1005.

[18] S.Liao, X. Zhu, Z. Lei, L.Zhang, SZ. Li. "Learning multi-scale block local binary patterns for face recognition." Advances in Biometrics. Springer Berlin Heidelberg, 2007. 828-837.

[19] E. Maggio, A. Cavallaro. "Video tracking: theory and practice". *Wiley*, 2011.

[20] N. Dalal, B. Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

[21] G. Bradski, A. Kaehler. "Learning OpenCV: Computer vision with the OpenCV library." O'Reilly Media, Incorporated, 2008. http://www.robots.ox.ac.uk/~lav/Papers/benfold_reid_cvpr2011/benfold_reid_cvpr2011.html

[22] M. Andriluka, S. Roth, and B. Schiele. "Monocular 3D pose estimation and tracking by detection." CVPR 2010. TUD Multiview Pedestrians dataset. http://www.d2.mpi-inf.mpg.de/node/428

[23] D. Gernimo, A.D. Sappa, A. Lpez and D. Ponsa. "Adaptive Image Sampling and Windows Classification for On-Board Pedestrian Detection". Proceedings of the International Conference on Computer Vision Systems. Bielefeld, Germany, March, 2007. CVC-CER-01 Pedestrian Database.

[24] MIT Pedestrian Dataset. http://cbcl.mit.edu/software-datasets/PedestrianData.html

[25] Viewpoint Invariant Pedestrian Recognition (VIPeR) Dataset http://vision.soe.ucsc.edu/node/178

[26] B. Benfold, I. Reid. Guiding visual surveillance by tracking human attention. In BMVC, 2009. Benfold dataset.

[27] IIT (Istituto Italiano di Tecnologia) Head Orientation Dataset. https://sites.google.com/site/diegotosato/ARCO/iit

[28] QMUL (Queen Mary University of London) Head Pose Dataset. https://sites.google.com/site/diegotosato/ARCO/qmul

[29] W. Choi, K. Shahid, S. Savarese, Collective Activity Dataset. http://www.eecs.umich.edu/vision/activity-dataset.html