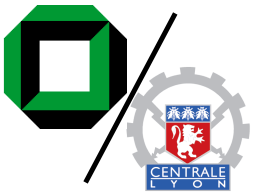

UNIVERSITÄT KARLSRUHE (TH)
FAKULTÄT FÜR INFORMATIK
INSTITUT FÜR ANTHROPOMATIK
Prof. Dr. Rainer Stiefelhagen

cv:hci

DIPLOMA THESIS

Counting Persons in Videos

SUBMITTED BY

Sebastian Wirkert

AUGUST 2009

ADVISORS

Prof. Dr. Rainer Stiefelhagen
Prof. Dr. Liming Chen
Prof. Dr. Emmanuel Dellandréa

Computer Vision for Human-Computer Interaction Research Group
Institute for Anthropomatics
Universität Karlsruhe (TH)
Title: Counting Persons in videos
Author: Sebastian Wirkert

Sebastian Wirkert
Kranzlweg 12
94157 Perlesreut
email: sebastian.wirkert@googlemail.de

Statement of authorship

I hereby declare that this thesis is my own original work which I created without illegitimate help by others, that I have not used any other sources or resources than the ones indicated and that due acknowledgement is given where reference is made to the work of others.

Karlsruhe, 31. August 2009

.....
(Sebastian Wirkert)

Acknowledgments

First and foremost, I would like to thank my professors, Prof. Dr. Rainer Stiefelhagen, Prof. Dr. Liming Chen and Prof. Dr. Emmanuel Dellandréa for giving me the opportunity to write my diploma thesis in this interesting subject. Special thanks goes to Mr. Stiefelhagen for proof-reading the thesis on such short notice. I would like to thank Xi, Chao, Quingin, Pierre, Kiryl, Karima and Di for supporting me and acting in my videos. A special thanks to Xi for introducing me to the subject and giving me much valuable help. Thanks to the Chinese guys for the food, it was very interesting. Thank you Anne for your love and support during the whole course of the thesis, you are the greatest. Another big thanks goes to my correctors Sebastian, Margit and Florian, who made this thesis much more accessible. Last but not least I would like to thank George Orwell, whose visions inspired so many computer scientists.

Abstract

In this thesis, a novel approach to track and count humans is presented. Using the video input of surveillance cameras, it reliably tracks and counts humans, e.g. in a shopping mall setup. In essence, the system combines person detection with a tracking algorithm. Unlike many other approaches, it does not rely on background subtraction for person detection but instead uses a sophisticated method, the Histogram of Oriented Gradient (HOG) person detector [1]. To speed up the HOG detection and reduce false detections, scale information is incorporated into the HOG framework, resulting in the scale-aware HOG. The tracking algorithm is the main contribution of this thesis. The developed tracking can restore inter-person dependencies via variational methods and uses a newly developed occlusion handling method dubbed GOETHE (General Occlusion Estimation for Tracking Humans Efficiently), which performs in close to linear time. An extensive evaluation is conducted, which measures the performances of the scale-aware HOG, GOETHE and counting and tracking in general. The counting accuracy is measured on the CAVIAR dataset [2], showing an accuracy of approximately 87%.

Zusammenfassung

Automatisiertes Zählen von Personen ist ein Gebiet mit vielen praktischen Anwendungsmöglichkeiten. So kann es beispielsweise helfen, Fahrpläne öffentlicher Verkehrsmittel zu optimieren oder die Auslastung eines Kaufhauses zu evaluieren.

Häufig wird Personenzählung mittels einfacher Lichtschranken durchgeführt. Der Genauigkeit sind hier allerdings Grenzen gesetzt, da sie weder zwischen Mensch/nicht Mensch noch nebeneinander gehenden Personen unterscheiden können. Der Trend geht also zu Personenzählung mittels Videokameras. Hier bietet es sich an, das ohnehin gegebene Netz der Sicherheitskameras auszunutzen. Herausforderungen hierbei sind das Identifizieren von Menschen im Bild sowie die Behandlung von Verdeckungen, die Menschen für das System vorübergehend unsichtbar machen.

In dieser Diplomarbeit wird das Histogram Of Oriented Gradients (HOG) Framework zur Personendetektion verwendet. Als Ergänzung zum original HOG werden Skalierungsinformationen verwendet, um die Berechnung zu beschleunigen und Falschdetektionen zu verringern.

Nachdem eine Person detektiert wurde, wird sie mit einem Partikelfilter verfolgt. Trackingsysteme wie der Partikelfilter haben traditionell Probleme mit der Verfolgung mehrerer Personen. Verfolgt man jede Person einzeln, verliert man die Möglichkeit, Abhängigkeiten zwischen den Personen zu modellieren und Verdeckungen zu handhaben. Will man alle Personen mit einem einzigen Tracker verfolgen, so steigt der Aufwand exponentiell mit der Anzahl an Personen. In dieser Diplomarbeit wird ein Partikelfilter vorgeschlagen, das zwar jede Person größtenteils einzeln verfolgt, jedoch Interaktionen und Verdeckungen mit niedrigem Aufwand berücksichtigt. Die Methode zur Handhabe von Verdeckungen wurde dabei in dieser Diplomarbeit neu entwickelt. Sie wurde GOETHE (General Occlusion Estimation for Tracking Humans Efficiently) genannt und zeichnet sich dadurch aus, dass sie beliebig lange Person-Person und Person-Welt Verdeckungen in pseudolinearer Zeit berechnet. Dies ist eine Verbesserung zu existierenden Verfahren zur Handhabe von Verdeckungen.

Das Zählsystem wird mit dem CAVIAR Datensatz evaluiert. Eine Zählgenauigkeit von ca. 87% in diesem schwierigen Datensatz zeigt die praktische Tauglichkeit des entwickelten Systems.

Contents

1 Introduction

Almost everyone of us "modern human beings" has been counted one time or another, probably without taking notice of it. It could have been in a store, public transportation or even at a national park [3]. First let us explore some of the potential application areas of a person counting system.

The store manager wants to monitor person counts for various reasons. He wants to discover how frequently the store is visited at what times. This information enables him to efficiently schedule employees. Furthermore, the people counting system can show how much the new ad campaign increases the number of customers. With a real time counting system, shop assistants can be interactively assigned to the areas in the store in which they are needed most. Additionally, areas with only few persons can be identified and thus weak points in the store layout can be discovered. Information denoting from where to where persons went can be used to optimize the store layout. In the example of public transportation, the information can again be used to determine the optimal schedule. Additionally, overcrowding can be discovered. Crowd counting systems are another class of people counting system which can estimate the number of people in large crowds. They can for example be applied to estimate the numbers of people who participated at a demonstration or a festival.

Note that for different kinds of information, different requirements are posed for the counting system. Sometimes for example real time counting ability is needed, whereas the crowd estimator has to be able to handle a big number of people. Because the application range is so broad, one system usually cannot provide all of the above applications. Therefore, let us first have a look at the different ways of counting humans.

1.1 Classes of Person Counting Systems

Many different people counting systems have been proposed. They can be roughly classified into vision and non-vision based and tracking and non-tracking based, as shown in figure 1.1. Each class has certain strengths and weaknesses which are described below. In chapter 2 some interesting examples are presented.

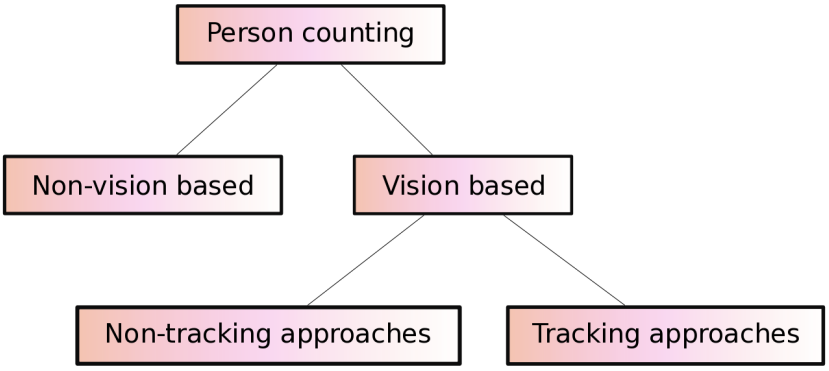

Figure 1.1: Different classes of people counting systems.

1.1.1 Non-vision Based Systems

This class sums up all systems not based on video data from a camera. Examples are systems based on infrared or pressure sensors. Infrared sensors usually use light barriers. They infer the number of people from discontinuations of the light barrier. Pressure sensors are hidden in the ground and use weight changes people cause by walking over them for counting. There are several common disadvantages these systems have to deal with. One point is, that they do not use available infrastructure provided by existing security cameras. Moreover it is not easy to distinctly count humans. Additionally, infrared sensors are likely to get obscured by insects etc. and are only able to perform on narrow paths. Another situation where infrared sensors perform poorly is when people walk side by side. However, unlike video based approaches, these systems are insensitive to environment factors like light changes. An example for a non-vision system is shown in [4].

1.1.2 Vision Based Systems

Vision based systems ideally use infrastructure provided by the security network. They count persons passing, utilizing the information given by a security camera's video stream. Unlike non-vision based systems, they are not generally limited to narrow corridors and doors. Due to the information from the image, it is possible to distinguish humans from other objects like bags and dogs. However, occlusions between humans and varying environmental conditions are typical challenges for vision based systems. Some systems like [5] overcome this problem by mounting the camera showing directly down to the floor.

This approach, however, limits the area that is covered by the camera. Furthermore, this is not the way security cameras are set up, thus these systems cannot use the infrastructure provided by the security system.

1.1.2.1 Non-tracking Approaches

These approaches estimate the number of people without any tracking of groups or individuals. This is reasonable when the number of people in a frame is very large, since most tracking approaches do not scale well with an increasing number of people. Due to their ability to count large amounts of people, these systems are called crowd monitoring systems. If the number of persons is small, one should however take advantage of the additional information provided by video data. The main drawback of this approach is, that it only delivers the total number of persons in a frame. Information on how many people enter or leave a location (e.g., a store) cannot be obtained. Examples for non-tracking based approaches are given in [6, 7, 8].

1.1.2.2 Tracking Approaches

Systems using this approach track objects over time in a video. An object can be a human being, but also clusters of persons or low level features which are later clustered to humans. In contrast to non-tracking based approaches they can provide information from where to where a person went. Thus, most of the scenarios described above can be handled by tracking based approaches in theory. Some popular problems are the curse of dimensionality, the correspondence problem and track initialization/deletion. An extensive discussion can be found in section 3.2.2. Tracking-based approaches can for example be found in [9, 10, 11, 5].

1.2 Diploma Thesis Approach

In this diploma thesis follows a tracking based approach. It is chosen, because it can provide most of the desirable capabilities described at the beginning of the diploma thesis. The proposed system is able to identify humans in video and track them over time so that the number of persons in a specific location can be determined. The biggest drawback of the tracking based approach is the reduced ability to count many people at the same time, due to computational costs. Nevertheless it is tried to design a system which can process a maximum number of persons at once. In order to design the system, one has to first set his goals:

- The system should make use of a *sophisticated method* for *detecting humans*. Most existing approaches do not distinguish human/non-human objects and are prone to falsely count other objects (e.g., [9, 10, 11, 5]). In this diploma thesis, a state of the art approach to robustly detect humans is used.
- As described at the beginning, *real time performance* adds important capabilities to the system. If possible, the system should be able to track a modest number of people in real time.
- Since the system should be applicable in practice, the *setup* should be as *easy* as possible. For example camera specific classifier training should be avoided because this is labor-intensive and therefore restricts the system's practical usability.
- If possible, the system should *extract track information*. By detecting from where to where people went, many other kinds of information can be obtained.
- It should be possible to cover *multiple regions* by using only one camera: If there are multiple entrances and exits in a room, all should be observed by only one camera instead of having to mount one camera for each exit.

For the rest of the thesis a horizontally aligned, static camera with a tilt angle of less than 45^o is assumed. It is assumed that the camera is not, or only mildly, distorted. This is a common setup for most surveillance applications. Other assumptions made during the design process will be explicated.

1.3 Thesis Outline

This thesis is organized as followed: first, some interesting systems for person counting are introduced in chapter 2. Then, chapter 3 explains the theoretical basics necessary to understand the concept. The concept in chapter 4 is the heart of this thesis. Here the designed people counting system is explained. The experiments follow in chapter 5. Herein the components of the counting system are evaluated. The thesis closes with the conclusion in chapter 6. It summarizes the achievements and problems of the developed system. Furthermore a prospect into possible future extensions is given.

2 State of the Art

This chapter divides in two basic blocks: an overview of modern person counting systems and an overview of state of the art multi-target tracking systems. Due to the abundant literature it is not possible to describe every system. To give a good outline, first interesting systems are described briefly. Then two to three particularly interesting, recent examples are explained in more detail. A broad range of different systems is covered. For the in-depth described approaches a short summary is given and the strengths and weaknesses are pointed out quickly. Then a more detailed description follows.

2.1 Person Counting Systems

A sophisticated non-vision based system is presented by Li et al. in [4] . They collect input given by a photoelectric sensor and classify it using a BP neural network. Their system shows good results with a counting accuracy of up to 95%, but suffers from the typical difficulties like people walking in a row.

The system of Laurent et al. [11] focusses on counting people in transport vehicles. They identify people by skin blob ellipses of their head, which they obtain by skin-color segmentation. Then these skin blobs are used for tracking and counted after they pass a counting line in the image. The counting accuracy of the system is about 85%. Problems are false counts from hands and the need for a robust skin-color model.

In [5], Septian et al. developed a system that counts people using an overhead mounted camera, which looks straight down to the floor. This prevents the problems arising from occluded persons. They segment and track foreground blobs. In their experiments they show a counting accuracy of 100%, but the database is very small. Problems with this approach are e.g., that it does not sophistically detect humans and the inability to use it in a normal security camera setup due to the untypical camera angle. Also, the overhead mounted camera cannot cover large areas, which is normally an advantage of vision-based systems.

Zhao et al. [12] track persons' faces using data coming from a face detector. The faces are tracked by a scale-invariant Kalman Filter. Instead of relying on a counting line or area like other approaches, they count people by classifying their trajectories. That way, a counting accuracy of 93% can be achieved. Drawbacks are the need of newly training

this classifier for each camera and the need of persons moving towards the camera. An advantage is, that by using a face detector, they are among the first approaches which are able to reliably differentiate humans and non-humans.

Now let us have a look at other interesting approaches in more detail.

2.1.1 Counting Crowded Moving Objects

In [10] the authors present a tracking based approach to segment moving objects in densely crowded videos. Their approach shows encouraging results for videos with up to 50 people. Briefly, they track a large number of low level features. These features are clustered considering the spatial distribution of the features over time. Afterwards, the clusters are counted, denoting the result of the segmentation.

Because this approach tracks many low level features, it is resistant to some features getting lost. Features might get lost due to occlusions or unpredicted movement. That means, the approach neither depends on a sophisticated model to deal with occlusions nor requires a complex motion model. Due to its general formulation, it is applicable to many different scenarios. A drawback is, that because of the lack of an elaborate model characterizing human, it is likely that not only humans are counted by the program. Furthermore, the system can only estimate the number of people in the videos but not give any additional information about how many people went where. The system moreover cannot detect persons who are fully occluded.

2.1.1.1 System Description

The proposed system consists of two basic steps, namely tracking and clustering, which are described below.

Tracking The KLT tracker [13] is used to track low level features over time. It tracks a window W in a video by determining the affine transformation which minimizes the dissimilarity between two frames:

$$\int_W [J(A\mathbf{x} + \mathbf{d}) - I(\mathbf{x})]^2 w(\mathbf{x}) d\mathbf{x} \tag{2.1}$$

With A and $\mathbf{d}$ denoting the affine motion parameters. The uniform weight $w(\mathbf{x}) = \frac{1}{|W|}$ is a normalization factor assuring that every window size has an equal weight. If several windows are available for one feature, the KLT tracker only selects the one with the best quality. Furthermore, the parameters for the windows are limited to a set which is obtained by training data. The computation is further fastened by utilizing integral

images. Due to occlusions and exit of objects regularly respawning of features has to be performed. Because of computational costs respawning does only take place in carefully selected places of space and time. A technique to smoothen the trajectories is applied to reduce their inhomogeneity. Therein, the trajectories are conditioned by other near trajectories.

Clustering The clustering is in essence a graph partitioning problem, where the nodes represent the features and the edges represent the membership to a common object. Building this connectivity graph is thus the goal of clustering. Clustering the low level features using a single frame is not possible, since the features' spatial relation of one frame could be coincidental. To perform the clustering *necessary* and *sufficient* conditions are formulated.

necessary An object's maximal width and height is determined experimentally. Furthermore, the variation between two trajectories (corresponds to features) is determined by training data, accounting for how "loose" their connection is. For two trajectories to be connected, their maximum distance has to be lower than the object's maximum size and the variance has to be lower than the determined variance. This condition has to be fulfilled during a certain period of time for the features to be possibly connected.

sufficient If a set of trajectories T is rigidly connected, they share an affine transformation which propagates them to the next time-step. If an affine transformation exists for every time in a certain period, the trajectories in the set T are considered as connected. The sets T are determined by performing RANSAC [14].

Now, sets of trajectories T which are surely connected and possible connections between two features exist. Two sets T^i and T^j are now iteratively merged, if all features in the two sets are connection candidates. This continues until every possible pair has been analyzed. Figure 2.1 shows the resulting feature groups after tracking.

2.1.2 Privacy Preserving Crowd Monitoring: Counting People without People Models or Tracking

Chan et al. [6] presents a non tracking based approach to count large numbers of people and to distinguish in which directions they are walking. To achieve this, first the crowd is separated into groups with different motions using the mixture of dynamic textures model [15]. Then for each region a set of simple features is extracted which are afterwards classified by a Gaussian Process [16]. Figure 2.2 shows an overview of the system.

One advantage of this system is, that it is privacy preserving because no separation of single persons takes place in the classification process. It can deal with a large number of

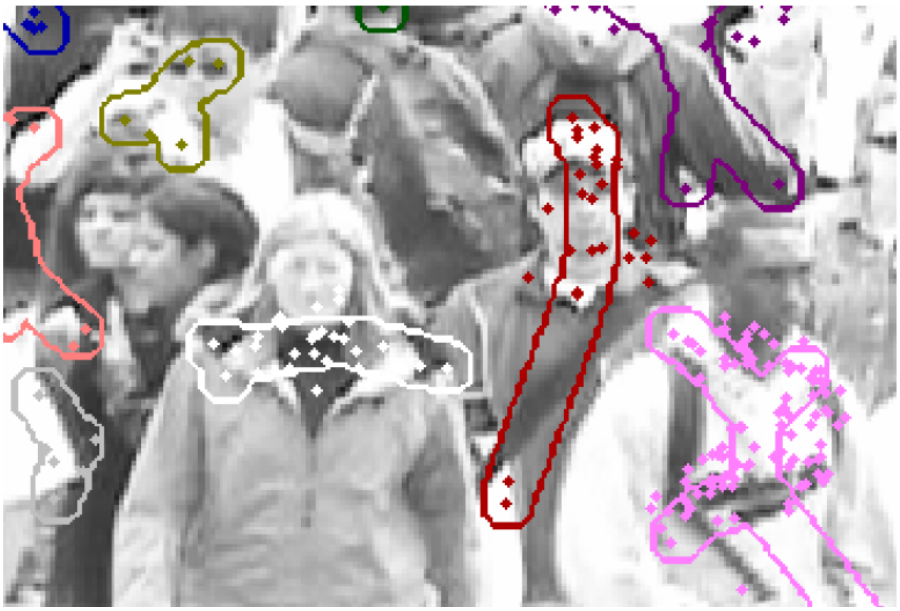

Figure 2.1: Result of Rabaud et al.s system. The dots are the features. If they are bordered, the system marked them as connected and counts them as one person. Good results can be seen, except for two persons on the left, which are merged [10].

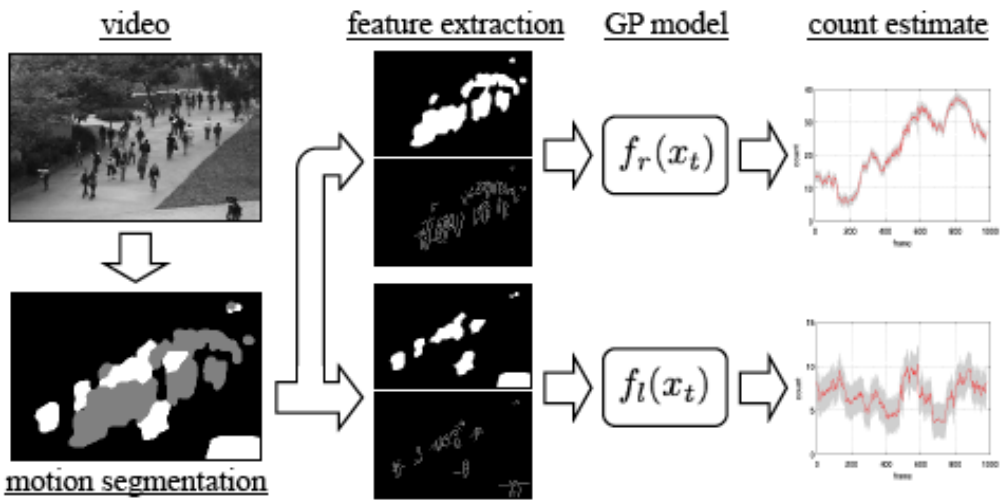

Figure 2.2: Overview of the crowd counting system of Chan et al.. First the video input is segmented into crowd of different directions. Then for each direction features are extracted and classified by a Gaussian Process. The Gaussian Process is trained for each direction separately. The result denotes the number of persons in the current image.

pedestrians, but the downside again is, that no counting with respect to "who went where" can be performed. However, this may be solved by simply tracking the different regions. Tracking regions instead of persons has the advantage that a large number of persons can be summarized in one region, resulting in a lower computational effort. The problem might be that regions can merge and split. Another problem of the presented approach is that both the mixture of dynamic textures model and the Gaussian Process have to be learned. This is contrary to the requirement of an easy setup, defined in section 1.2.

2.1.2.1 System description

First, the picture is split into regions of different direction using the mixture of dynamic textures model. This model is learned with the Expectation Maximization (EM) [17] algorithm and is described in detail in [15]. Before the feature extraction can take place, the effects of perspective must be considered. If not, closer objects have more influence because they are bigger. To reduce this effect, a simple approach is chosen to generate an approximate perspective map. The pixels are now weighted according to the distance information provided by this map. Afterwards 28 features are extracted, which can be classified as *segment features* (e.g., the total number of pixels in the segment), *internal edge features* (e.g., the total number of edge pixels in the segment) and *texture features*

(e.g., the homogeneity of the texture at different angles). People have different appearances depending on their walking direction. Thus for each walking direction a separate Gaussian Process has to be trained to map feature output to crowd size. The kernel function for the Gaussian Process is modeled by a linear and a RBF kernel. It was chosen because normally the features should linearly correspond to crowd size, but some non-linearities arise due to various reasons like occlusion, segmentation errors and spacing within a segment [6].

2.1.3 Detecting and Counting People in Surveillance Applications

Liu et al. present in [9] a tracking based approach to count people. They use a combination of foreground detection, crowd segmentation and tracking. The persons are counted when they cross a "virtual" gate.

The advantage of this system is, that it can really detect where people are going as opposed to just deciding how many persons are present. The algorithm presented is not only applicable to human tracking, but can be extended to other classes as well. The downside of the approach is, that it heavily depends on foreground detection and does not detect humans by a sophisticated model, but by foreground blobs of human proportions. There are some problems caused by foreground segmentation. First of all, shadows are often also detected as foreground. Second, a person can be split in several not connected blobs. Multiple persons in close proximity form one big foreground cluster. The system tries to solve this problem by using a crowd segmentation technique. This works, when the cluster is composed by only few persons. But when strong occlusions are present or the crowd gets large it fails, since it uses cues based on edges of the foreground map.

2.1.3.1 System Description

The two most important steps of the system are the tracking and the model based segmentation, described in the following sections. The tracker follows the persons and uses the segmentation algorithm in case the persons clutter. Virtual gates are defined in the picture. These virtual gates can for example be lines, but also more complex geometric forms. When a person crosses such a virtual gate, it is counted.

The Tracker The tracker uses an adaptive appearance based approach. Here a color model and a probability for every pixel in the model to belong to the foreground are adaptively trained. This information is used to refine the information provided by the foreground segmentation to make a measurement of the current location of the person. Persons are tracked by simply shifting according to their velocity and comparing it to the current measurements. If a measurement is in close proximity to the prediction, it is set as

the new location of the person. To overcome the negative effects clutter has on the tracking, large foreground regions and regions with a close proximity of persons are forwarded to the segmentation process. This segmentation process splits the cluttered regions. Since this segmentation is not always correct, group targets are tracked by Kalman Filters and a simplified multiple hypothesis tracker [18] is used.

The Model Based Segmentation Features are extracted based on the foreground segmentation. These features are used to detect cliques which represent a person. Now a large set of hypothetical persons is available. A combination is searched that assigns each feature to at most one person. To find the combination with the biggest likelihood, the EM algorithm is employed. A short graphical summary can be seen in figure 2.3.

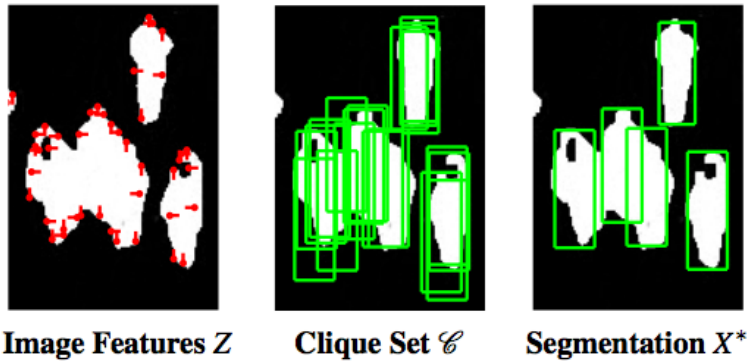

Figure 2.3: Overview of the model based segmentation. The left image shows the features detected in the foreground region. In the center image possible persons implied by the features are drawn in. The right image displays the most probable combination of persons found by the EM algorithm.

2.2 Multi-target Tracking Systems

When designing a tracking based counting system, you are primary designing a multi-target tracker. The tracker has to be able to track varying number of targets. Yet it suffers from problems arising due to person interaction and occlusion. Person interaction and occlusions are big problems for tracking algorithms, because they imply a strong dependency of the individual targets. These dependencies are too costly to calculate exactly, thus an efficiently calculable solution has to be found. In this section, some systems are presented and their capabilities are described. Two systems of special interest for this diploma thesis are then described in more detail.

In case of multiple, non-labeled measurements, the problem of data association is important. When the targets are close both in space and appearance, it is not an easy task to associate the right measurement to the right tracker. Reids Multiple Hypothesis Tracker (MHT) [18] and the joint probabilistic data association filter (JPDAF) [19, 20] handle this problem. While the MHT can deal with a changing number of targets, the number of targets in the JPDAF remains fixed. These methods are not usable for the purposes of this thesis, because their runtime increases exponential with the number of targets.

The probability hypothesis density (PHD) filter first developed by Mahler in [21] retains the joint nature of the multiple target tracking and models the appearance and disappearance of targets directly in the filter as opposed to a superordinate process. The resulting filter is the multi-target equivalent to a constant-gain Kalman filter. The resulting equations are still not efficiently computable. For that reason, implementations approximate the PHD, for example using particle filters [22]. Although the PHD models the multi-target problem principled and efficient, the problem is coarsely approximated by utilizing the constant-gain Kalman filter.

In [23], a probabilistic exclusion principle is presented, which prevents persons from occupying the same space. Additionally, the technique of partitioned sampling is introduced in this work to handle the occlusion problem. Partition sampling decomposes the joint structure of the occlusion problem. If it is for example known that target A occludes target B, first the configuration for target A can be calculated and later used to infer the configuration for target B. The problem with this approach is, that it assumes that the spatial distribution of two targets is known. But in practice, one could have two hypotheses for target A, one in the front and one in the back of the picture, while target B stands in between. Now the decomposition as suggested by partitioned sampling is not possible anymore.

In [24] Yu et al. developed a filter using Pairwise Markov Random Fields (PMRFs) to avoid coalescence of different targets. Coalescence means that two trackers lock on the same target. This phenomenon occurs, when two similar looking targets stand very near to each other. To avoid coalescence, PMRFs model pairwise interactions between two targets. This interaction can for example prevent two persons from occupying the same space. That way, coalescence can be prevented.

2.2.1 Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors

In [25] Wu et al. present an interesting approach for tracking multiple humans. They use a part-based detector to identify humans. If possible, the new detections are simply matched to similar old detections. If this fails, a mean shift based tracker [26] is employed

to find the person. This system and its predecessor in [27] are the only systems known to the author evaluated on the CAVIAR dataset.

The advantage of this system is that it can handle partially occluded humans (both inter-person and inter-scene) by utilizing part-based detectors. Their edgelet feature provides a robust description of human. Another advantage is that by utilizing a state of the art human detector, they can discriminate humans and non-humans. The biggest drawback of the presented approach is the inability to cope with fully occluded persons.

2.2.1.1 System Overview

The system divides into two parts: first a part-based detection is performed. Then the detection results are used to reliably track humans.

Part-based Detection The part-based detector developed in this work uses edgelets as features. An edgelet is a local shape feature, that should detect the silhouette of a human. Examples are lines and circles. These edgelets are weak classifiers, which are used in a boosting method [28] to train a cascade-of-rejectors. That way, several classifiers are trained, as visualized in figure 2.4. For every body part several views are trained. These views share the same root node in the cascade-of-rejectors, making the computation more efficient.

The full body part and the head-shoulder detector are used to find hypotheses for humans in the image. Then the torso and legs detectors are used to scan in the region of these hypotheses. Now combined responses are formed, which "fuse" together part detectors belonging to the same human: the hypotheses for the humans are analyzed to see whether there are other part-detectors supporting the hypotheses or not. To account for occlusions, an occupancy map is built from the hypotheses which marks occluded body parts with do not care. Then a bayesian approach is chosen to find the best fitting mapping given the image observation. That way, false alarms and false negatives can be filtered out and the so-called "combined responses" from the part-based trackers are built.

Tracking Based on Detections In the first stage of tracking, the detection results are matched to the existing trackers. This is done by defining an affinity measure which regards position, size and color appearance. The persons are matched to the detection with the biggest affinity.

Potential tracks are initialized every time a detection has not been matched to a tracker. The potential tracks become a confident trajectory, if they have been matched to a detection for a certain amount of time. The amount of time needed is determined by the affinity

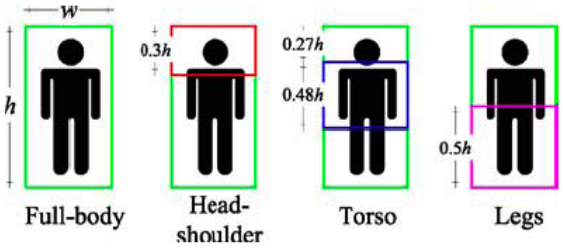

Figure 2.4: The trained detectors.

of the match and the probability for consecutive false measurements. Track deletion is done in a similar way.

At every time-step, persons are tracked by matching the detections to the existing trajectories. When this fails, a mean shift tracker is used to track the parts individually. The probability distribution tracked by the mean shift tracker is composed of the color based appearance model, a Kalman based dynamic model and the detection confidence. To improve the performance of the probability from the appearance model, principal component analysis (PCA) is used to model shape constraints.

2.2.2 Approximate Bayesian Multibody Tracking

A very interesting work is presented by Lanz in [29]. It is an advancement of the work done by Lanz and Manduchi in [30], who developed the Hybrid Joint-Separable (HjS) filter. It presents a principled approach to handle occlusions. The HjS is a Bayes filter with a particle filter implementation. It can restore both dependencies from occlusions and inter-person dependencies. The runtime is quadratic with respect to the number of persons.

The system has multiple advantages. Due to the Bayes filter fundament, a principled, theoretically sound probabilistic approach is used. The explicit modeling of occlusions allows arbitrarily long occlusions without the tracker failing. The modeling of inter-person dependencies prevents two trackers from locking onto the same target. The drawback of the system is the quadratic complexity, which makes it unusable for applications with many persons present.

2.2.2.1 System Overview

This description requires knowledge about the Bayes filter [31] and makes use of the notations introduced in chapter 3.2.1. The basic idea of the HjS filter is the following:

an assumption when using separate filters for multi-target tracking is that different targets have no influence on each other concerning the measurement. This is not true, mostly due to occlusions. A fully joint filter on the other side is not feasible because of computational costs. Lanz therefore assumes the less drastic approximation, that the joint belief and prediction can be represented by its marginal components:

$$p(\mathbf{x}|Z) \approx \prod_k p(x^k|Z) \tag{2.2}$$

where

$$p(x^k|Z) = \int p(\mathbf{x}|Z) d\mathbf{x}^{\neg k}. \tag{2.3}$$

with $\mathbf{x}^{\neg k}$ representing the joint state vector with the kth component removed. The same definition is applied for the prediction. Lanz proves that this formulation is the best possible estimation in the single target spaces. Based on this assumption, Lanz derives the *marginal process* and *measurement* models.

The marginal process model is efficiently computed by describing the inter-target dependencies as a Pairwise Markov Random Field and using Belief Propagation [32] to compute the marginals.

Lanz expresses the marginal measurement model in the log-likelihood domain, which he proves is a first order approximation of the real likelihood. This leads him to two terms: the foreground term and the background term. While the foreground term expresses the contribution of the visible parts of one target, the background term subsumes the contribution of the occluded parts and the background. These two parts are computed iteratively, making use of a particle filter [33] implementation of the Bayes filter: the particles are sorted by distance from the camera. Then the particles are processed from nearest to furthest. Hypotheses about how probable occlusions are for the current particle are determined by using iteratively built maps which account for the previously regarded states.

The process of building these maps is similar to the approach in this thesis. However, the principal approach and the way to compute the maps is different, resulting in faster, but less principled, calculations.

3 Theoretical Foundations

In this section, the theoretic building blocks necessary to understand the concept are explained. First a method for finding persons in a video is introduced. Then probabilistic tracking is illustrated, showing the theoretic framework, the Bayes filter, an implementation of the Bayes filter called particle filter and giving a short introduction in multiple target tracking and its problems. At last, a method to model pairwise dependencies between objects called Pairwise Markov Random Fields is introduced.

3.1 Person Detection

There are various methods of person detection. The person detector cannot be chosen independently from the following tracking, since the two processes are closely connected. The tracker developed in this thesis follows a color histogram of the detected person, thus the person detector should give a rectangle around the person as a result.To be able to decide for an object detection framework, one must first set the demands which should be met. In this case, these are:

- The object detector must be able to *detect humans*. Most existing people counting approaches do this by simply identifying foreground blobs with human proportions. For this diploma thesis a more sophisticated approach is searched.
- Since humans in surveillance videos can occur in any orientation, the detector must be *orientation invariant*.
- People have different size, furthermore persons far from the camera appear smaller. Thus the tracker has to be *scale invariant*.
- Due to large distances from the camera, persons can have a very small appearance in surveillance videos. The detector should be able to use as much information as possible, i.e. utilize the *whole human body* as input instead of e.g. only the face.
- The detection should perform fast, so that the system can work in *real time*.
- The method should be *general* enough to work under several different camera angles.

We chose the Histogram of Oriented Gradients (HOG) developed by Dalal et al. in [34] since it is a state of the art object detection which can meet almost all of the demands stated above: the HOG human detector can be used to detect upright standing, fully visible humans of any scale and orientation. The original HOG detector does not run in real time, but with enhancements proposed in [35] real time performance can be achieved.

3.1.1 Histogram of Oriented Gradients for Human Detection

The HOG object detector is a rectangular detection window, which is stridden over the whole image. This detection window extracts the HOG features as described in the next section. These HOG features are then classified into human/non-human by a linear Support Vector Machine (SVM) [36]. Generally the HOG architecture can be split in two phases shown in figure 3.1. In the following, first the feature extraction is explained, then

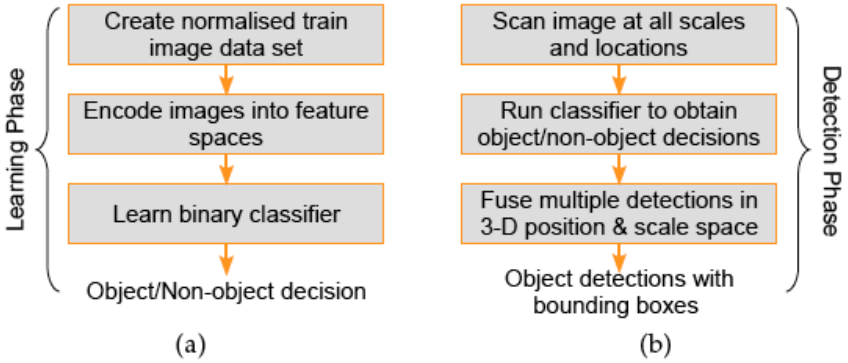

Figure 3.1: The two basic phases of the hog. First the classifier is trained in the learning phase. Then the obtained classifier is used for person detection [34].

the learning phase is discussed. Finally the generalization to multiple scales is explicated.

3.1.1.1 HOG Feature Extraction Chain

First, the detection window is normalized to reduce the influence of illumination and a gradient image is computed using the dominant color channel. Now the detection window is divided into several cells, each holding a 9-bin histogram. The bins in this histogram represent different angles of gradients. The gradients of a cell are then sorted into the corresponding histogram. Now a second structure is introduced: the block. A block is a collection of several neighboring cells (typically 2x2). The features of one block are the normalized histograms of all its cells. The blocks overlap, thus one cell is appearing in

several blocks, but due to normalization denotes different results. The feature vectors of all blocks are then concatenated, forming the combined feature vector. The whole chain is visualized in figure 3.2.

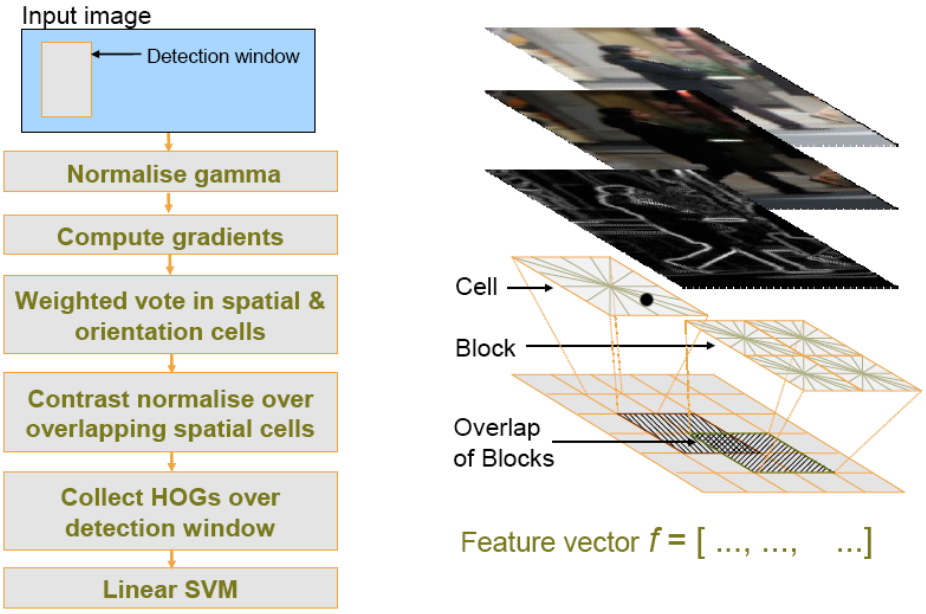

Figure 3.2: The feature extraction steps performed for every HOG detection window [34].

3.1.1.2 The Learning Phase

In the learning stage a SVM able to discriminate person and non-person detection windows is created. The input is the combined feature vector created as in the last section. First a preliminary SVM is trained by some positive and negative examples. The negative examples are randomly sampled from pictures without persons. Now this preliminary SVM is used to search for false positives in the pictures. These false positives and the original person, non-person images are used to re-train the SVM classifier. The SVM classifier will output a number which can be negative or positive, depending on the class. The farther away from zero, the more certain the detection will be. Naturally, this SVM classifier can also be trained for other classes than humans. In this work the detector developed in [34] is used to detect fully visible, upright standing humans. For some camera perspectives, as for example a top-down view, the SVM has to be newly trained.

3.1.1.3 Multi-Scale Object Localization

To account for multiple scales, the so-called scale pyramid is built. Here, the image is simply scaled to several different sizes and the detection window is stridden over all these images. With a robust classifier, an object is detected even if the scale or position do not fit exactly. False detections are present, but far less frequent. This means, that there are multiple detections for one object and also some detections for non-person objects. We, however, want only one detection for a real person and no detection false positives.

Therefore the technique of non-maximum suppression is used: first, all negative detections are discarded by setting their score to 0. Then all positive measurements are mapped into a 3D space, composed of xy coordinates and scale. To every measurement a smoothing kernel is applied, which also accounts for the detection probability given by the SVM score. Then a mean shift procedure is used to detect the modes. The modes are created by many measurements in close spatial proximity. False measurements do not create modes, due to their lower frequency, they are not accompanied by many other measurements to support them. Again, this process is visualized in figure 3.3.

3.2 Tracking

We regard tracking as the pursuit of one or more objects over time using the information denoted by noisy measurements (in the following also called observations). The objects can be physical entities like aircrafts, cars and people but also abstract ideas like the development of a country.

Normally, these problems are formulated in a discrete-time, state-space approach. Discrete-time means that the measurements are only available at certain points in time. State-space means the objects is described by a state-vector $\mathbf{x}$, which contains all relevant information. An aircraft can for example be represented by its position, orientation, velocity and acceleration. The development of a country could be represented by indexes like the Human Development Index and Gross domestic product (GDP).The measurements $\mathbf{z}$ are also characterized by a vector. Examples for the measurements are radar data or the images of a video camera. For the country development key figures like literature rate, life expectancy, GDP, etc. can be measured.

What makes tracking difficult is that the observations are subject to noise and objects can behave unexpectedly. For example an image can be distorted, a radar can miss an object or falsely detect an additional one. Tracked persons can suddenly change their direction due to a change in mind, which can not be foreseen by the tracker. Because of the uncertainty, it is often reasonable to use a probabilistic framework to describe the problem. The goal is now to estimate the state $\mathbf{x}$ at time t as good as possible out of all the measurements available until time t, $\mathbf{Z} = \{\mathbf{z}_{time=1}, \mathbf{z}_{time=2}, ..., \mathbf{z}_{time=t}\}$. The

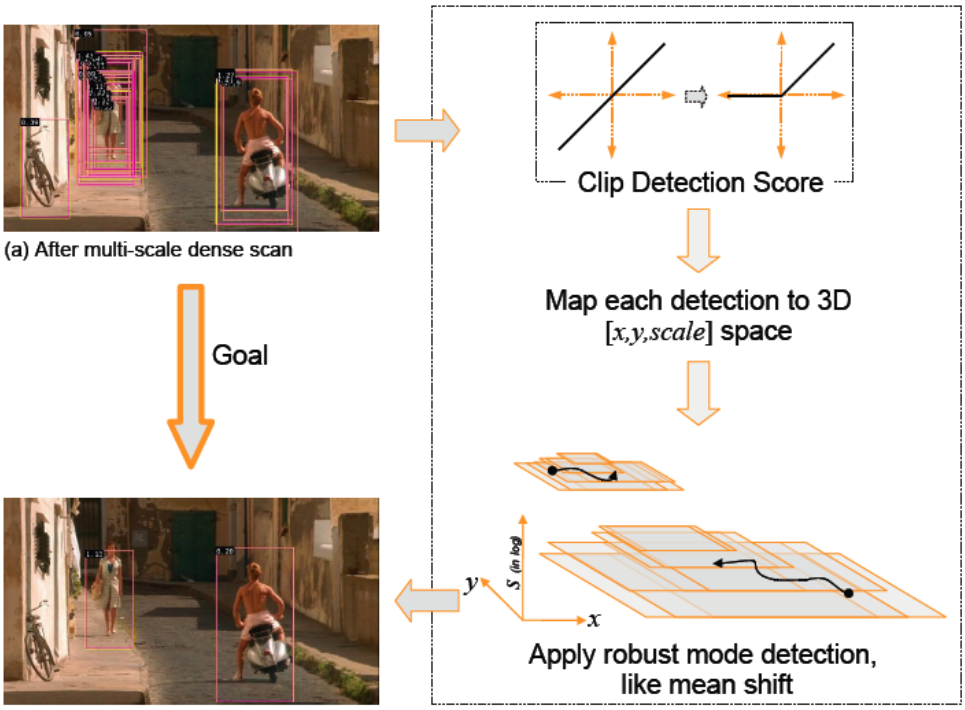

Figure 3.3: We search for modes caused by many measurements in close proximity. Note that the bicycle is falsely classified once. Since there are no other measurements to support the hypothesis, the bicycle does not create a mode [34].

posterior probability density function (pdf) $p(\mathbf{x}_{time=t}|\mathbf{Z})$ embodies this estimation. As it can be see, the information available increases with time. To make the computation of the posterior possible, one needs a recursive ansatz. In the next section a probabilistic, recursive framework to compute the posterior is illustrated: the Bayes Filter [31].

3.2.1 The Bayes Filter

Before the Bayes Filter is explained let us introduce some notation valid for the rest of the thesis. A vector without time index denotes the vector at time t. Time $t-1$ is referred to with a superscript $^-$. E.g. $\mathbf{x}$ denotes the state at time t, $\mathbf{X}^-$ denotes the set of all states until time $t-1$.

A Bayes Filter calculates the posterior using recursion. This recursion is based on the Markov assumption. The Markov assumption states that the state at time t is only influenced by the preceding state. Therefore, all important information must be contained in the preceding state. This information is used to make assumptions about the current state. To perform this *prediction*, a *process model* is necessary. It describes the evolution of the state in time. A *measurement model* is used to relate the current measurement to the state. In the *filter step* the so obtained information is combined, giving an estimate for the current posterior. Figure 3.4 clarifies this relation.

Now it is time to introduce the actual equations which describe the Bayes Filter. Applying Bayes' rule to $p(\mathbf{x}|\mathbf{Z})$ and using the Markov assumption denotes the equation for the filter step:

$$p(\mathbf{x}|\mathbf{Z}) = c\, p(\mathbf{z}|\mathbf{x}) p(\mathbf{x}|\mathbf{Z}^-) \tag{3.1}$$

where c is a normalization constant, and $p(\mathbf{z}|\mathbf{x})$ is the measurement model. The measurement model determines how likely a measurement is given the state. For example, one can map a person onto an image using his state and then compare how the image matches the person. We call $p(\mathbf{x}|\mathbf{Z}^-)$ the prediction. To be able to evaluate it, the Chapman-Kolmogorov equation is applied. Again the Markov assumption is used to simplify the equation:

$$p(\mathbf{x}|\mathbf{Z}^-) = \int p(\mathbf{x}|\mathbf{x}^-) p(\mathbf{x}^-|\mathbf{Z}^-) d\mathbf{x}^- \tag{3.2}$$

The probability for state $\mathbf{x}$ is calculated by adding all probabilities for the previous states to reach state $\mathbf{x}$. The probability that state $\mathbf{x}^-$ translates to state $\mathbf{x}$ is represented by the process model $p(\mathbf{x}|\mathbf{x}^-)$. For the previous airplane example this means how position, velocity and acceleration of the airplane change from one time-step to the next with some noise present. The noise is necessary to model unpredictable movements. E.g., in case of the airplane, complex changes in acceleration cannot be modeled.

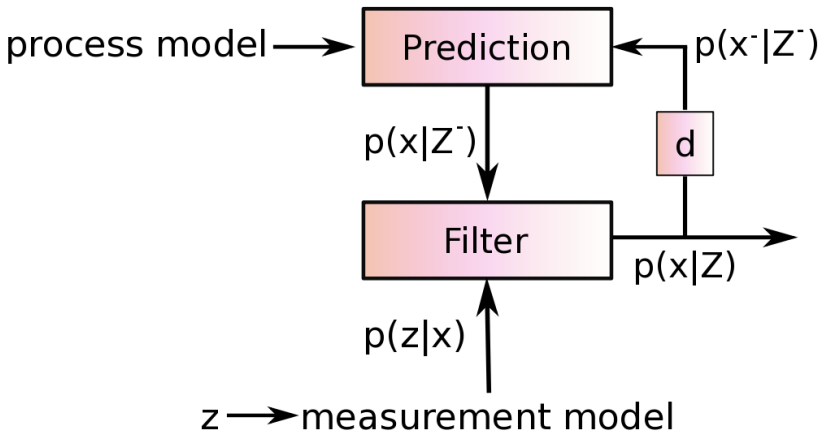

Figure 3.4: The Bayes recursion graphically displayed. "d" stands for delay and indicates the passage of time from the previous time-step to the current time-step.

Summing up, process model and the measurement model have to be defined to specify the Bayes Filter. Furthermore, an initialization $p(\mathbf{x}_{time=0})$ is needed to start the recursion. This has been a quick outline of the Bayes Filter. For a more complete derivation see [37].

The Bayes Filter in its general form presented here is not solvable analytically. An approach to solve the problem analytically is the Kalman Filter. If $p(\mathbf{z}|\mathbf{x})$, $p(\mathbf{x}|\mathbf{x}^-)$ and $p(\mathbf{x}_{time=0})$ are linear and gaussian, it solves the Bayes equations exactly. Since these restrictions do not hold in case of this work, it will not be further described here. The interested reader is referred to [38]. The next section describes the particle filter implementation of the Bayes Filter. The particle filter is a numerical filter which can describe arbitrary posteriors. After introducing the particle filter, some of the problems which occur then tracking multiple targets are explicated.

3.2.1.1 The Particle Filter

The basic idea of the particle filter [33, 39] is to describe the posterior pdf $p(\mathbf{x}|\mathbf{Z})$ not directly but by a set of N weighted samples $\{\mathbf{x}^{(l)}, w^{(l)}\}_{l=1}^{N}$. Let us first assume, that one is able to draw these samples independently and identically-distributed (i.i.d.) from the posterior.

With Monte Carlo integration, useful values like expectation or variance can be determined given these samples:

$$I = \int f(\mathbf{x}) p(\mathbf{x}|\mathbf{Z}) d\mathbf{x} \approx \frac{1}{N} \sum_{l=1}^{N} f(\mathbf{x}^{(l)}) \tag{3.3}$$

The function $f(\mathbf{x})$ can be used to calculate the stochastic moments of the posterior. For example the function to calculate the first order moment is $f(\mathbf{x}) = \mathbf{x}$.

The more samples are drawn, the better they describe the actual posterior pdf. Since they are drawn i.i.d. from the posterior, more samples will be found in regions with high probability and vice versa. The problem is that in general it is not possible to draw samples from the posterior. One can, however, evaluate the posterior at any point in the state space. To be able to draw meaningful samples nonetheless, the technique of *importance sampling* is now illustrated.

In importance sampling, one does not sample from the posterior, but from another function q with the same support called *importance density*. This function should be as similar to the posterior as possible, thus ensuring the samples fall into the true posterior's high probability regions. Since the two functions are generally not the same, there are regions with "too many" or "too little" samples regarding the posterior. Importance sampling hence has to assign a weight to the samples to reverse the distorting effects introduced by sampling from the importance density. By doing this, samples drawn in underrepresented regions are assigned higher weight and vice versa. Figure 3.5 shows this dependency graphically.

What is needed now is a convenient importance density to draw samples from. In case of the particle filter, the importance density is chosen to factorize such that

$$q(\mathbf{x}|\mathbf{Z}) \triangleq q(\mathbf{x}|\mathbf{x}^{-}, \mathbf{z}) q(\mathbf{x}^{-}|\mathbf{Z}^{-}). \tag{3.4}$$

In this case, the weights can be recursively determined by the equation

$$w^{(l)} \propto \frac{p(\mathbf{z}|\mathbf{x}^{(l)}) p(\mathbf{x}^{(l)}|\mathbf{x}^{(l)-})}{q(\mathbf{x}^{(l)}|\mathbf{x}^{(l)-}, \mathbf{z})} w^{(l)-}, \ \sum_{l=1}^{N} w^{(l)} = 1. \tag{3.5}$$

The choice of a good importance density is a critical step in the design of a particle filter. A common choice for a first factor of the importance density is the process model $p(\mathbf{x}^{(l)}|\mathbf{x}^{(l)-})$. Then, the samples are re-weighted based only on the measurement:

$$w^{(l)} \propto p(\mathbf{z}|\mathbf{x}^{(l)}) w^{(l)-}. \tag{3.6}$$

This choice is very descriptive. It can be imagined the particles are simply propagated by the process model and then assigned a new weight based on the new measurement.

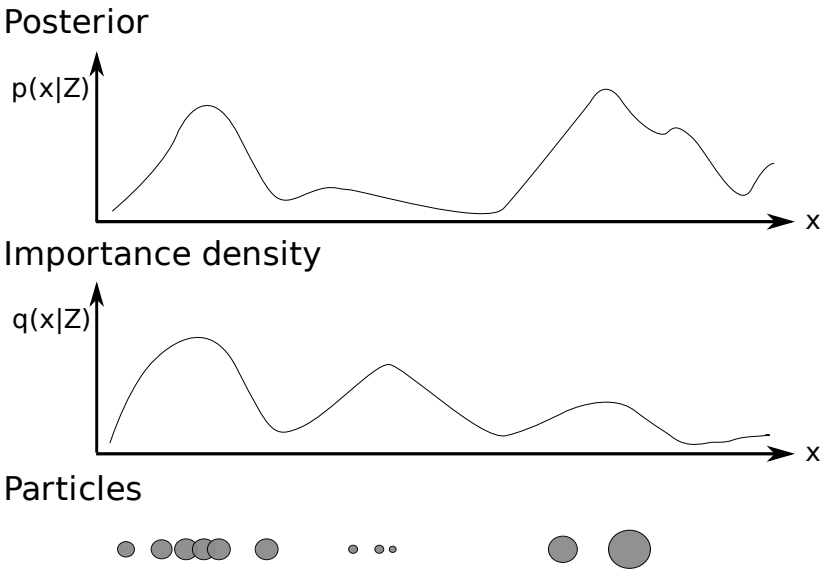

Figure 3.5: The particles are drawn from the importance density q. Although q somewhat resembles p it fails two predict the second maximum in p and instead is more likely to draw particles at a position, where p has low probability. Importance sampling now weights the particles to correct the errors made by drawing from q. The bigger the circle in the image, the bigger the particle weight.

Now, the Monte Carlo integral can be evaluated as

$$I = \int f(\mathbf{x}) p(\mathbf{x}|\mathbf{Z}) d\mathbf{x} \approx \frac{1}{N} \sum_{l=1}^{N} w^{(l)} f(\mathbf{x}^{(l)}). \tag{3.7}$$

It was shown, that the particle filter estimates converge to the true posterior as $N \to \infty$. However, the degeneracy problem is inherent in this formulation of the particle filter. It states that after a certain amount of time, only one particle will have non negligible weight. To solve this problem, the resampling step was introduced. Here, one samples N times from the old particles $\{\mathbf{x}^{(l)}, w^{(l)}\}$ to receive the new particle set $\{\mathbf{x}^{\star(l)}, 1/N\}$. This eliminates samples with low weight, which are unlikely to play a role in the future posterior, and duplicates samples with high weight. How often re-weighting is performed depends on the version of the particle filter. In this diploma thesis, the SIR particle filter is implemented, which performs re-weighting at every time-step. The algorithm for the SIR filter can be looked up in [33]. There an algorithm to perform re-weighting in linear time can be found. Figure 3.6 shows the basic steps of a SIR particle filter. For a complete derivation of the particle filter and more background information, see [33] or [39].

3.2.2 Multi-target Tracking

In general, multiple targets can be tracked exactly as one target by a Bayes Filter. However, then tracking a varying number of multiple targets, some specific difficulties arise. These difficulties and the two principle approaches to handle multi-target tracking are explained in this section.

3.2.2.1 Joint Approach

The joint approach tracks all the targets at once using one single joint state vector

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \tag{3.8}$$

while process and measurement model can contain arbitrary dependencies. This means, that e.g. group interactions can be modeled in the process model and occlusion can be foreseen in the measurement model. The big problem is the curse of dimensionality: the computational effort grows exponentially with the size of the state vector. Since the state vector grows linear with the number of targets, the computational effort grows exponentially with the number of targets. This disqualifies the joint approach for any practical application, where more than three or four targets are present.

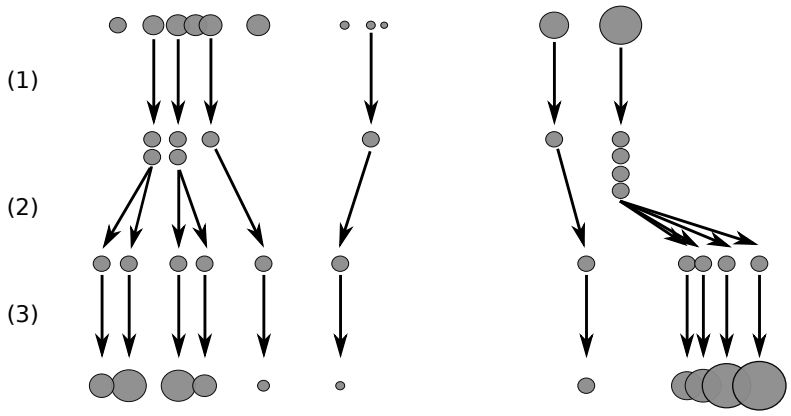

Figure 3.6: The basic steps of one SIR particle filter iteration graphically presented. N particles are taken from the previous time-step and draw i.i.d. N new particles, according to the old weights (1). Since they have been drawn i.i.d. these particles have evenly distributed weights now. Note how the improbable middle region is now only represented by one particle, whereas the plausible regions at the left and right are represented by many particles. In case of the left region, this is due to many moderately weighted particles. In the right region the reason for this are few, but heavy-weighted particles. In (2), the particles are propagated by the importance density. Finally, the new weights are assigned to the particles in (3) according to equation 3.5.

In equation 3.8 an important notation for the rest of the thesis was introduced: bold characters stand for the joint state, thin letters with little subscript index indicate that only the state of one person is meant. For instance $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$ describes the positions of all persons i.

3.2.2.2 Separable Approach

In the separable approach, it is assumed that the measurement model and the process model are fully independent concerning different targets. Assuming the prior $p(\mathbf{x}^-|\mathbf{Z}^-)$ factorizes, one derives

$$p(\mathbf{x}|\mathbf{Z}) \propto p(\mathbf{z}|\mathbf{x}) \int p(\mathbf{x}|\mathbf{x}^-)p(\mathbf{x}^-|\mathbf{Z}^-)d\mathbf{x}^- \tag{3.9}$$

$$= \prod_{i=1}^{n} p(z_i|x_i) \int \prod_{j=1}^{n} p(x_j|x_j^-)p(x_j^-|Z_j^-)d\mathbf{x}^- \tag{3.10}$$

$$= \prod_{i=1}^{n} p(z_i|x_i) \prod_{j=1}^{n} \int p(x_j|x_j^-)p(x_j^-|Z_j^-)dx_j^- \tag{3.11}$$

$$= \prod_{i=1}^{n} p(z_i|x_i) \int p(x_i|x_i^-)p(x_i^-|Z_i^-)dx_i^- \tag{3.12}$$

The individual targets are completely separated now. This means one tracker can be initialized for every target and track them separately, thus avoiding the curse of dimensionality. The downside is, that now the dependencies of states and measurements of different objects cannot be modeled anymore. People can now "pass through" each other, since the trackers do not know about the other peoples. Measurement dependencies occur, when one object occludes another. Since there is no possibility to model these dependencies, the separable approach is error prone. For example it is possible for the trackers to lock on the same target, if the targets are in close spatial proximity. In a joint tracker, a spatial exclusion principle could prevent this.

The later goal is to create a tracker which is able to restore the most important dependencies of the process and measurement model, while performing the rest of the calculations in the separate domain.

3.2.2.3 Common Problems

One common problem is the data association problem. With more than one measurement present, it is not clear which target corresponds to which measurement. This problem is

especially severe, when the objects cannot be identified by a certain attribute. A red car and a blue car in an image could for example be separated relatively easy, whereas the radar measurements of two aircrafts deliver no additional information to separate the two planes besides their position. Another problem is the initialization and deletion of new objects, when the number of objects is not known in advance. This problem is made even more severe due to possible occlusions, false measurements and other sensor errors.

3.3 Pairwise Markov Random Fields and Variational Methods

In the diploma thesis concept a principled approach to describe the interactions of human movement is needed. For example one wants to express that two people cannot run "through" each other. To model these dependencies, pairwise Markov Random Fields (PMRFs) were chosen. Additionally, some way to efficiently infer information from this model is needed. This is done via variational methods. Both, PMRFs and variational methods are introduced in the following.

3.3.1 Pairwise Markov Random Fields

In computer vision problems, often some observations z_i are given and the goal is to infer something about the system state x_i of object i causing the observation. Every object is linked to one observation and can be linked to an arbitrary number of other objects $j \neq i$. We express the described dependencies by an undirected graph $G = (V, E)$ with vertices V and edges E. Herein, a node i denotes either an object or an observation, the edges (i, j) indicate a dependancy between the nodes they connect. Therefore, the absence of an edge implies conditional independence: let $\mathcal{N}(i)$ be the set of nodes neighboring i, then $p(x_i|V) = p(x_i|\mathcal{N}(i))$.

We speak of a pairwise Markov Random Field (PMRF) [32], if the overall joint probability factorizes according to the graph:

$$p(x_1, x_2, ..., x_n, z_1, z_2, ..., z_n) = p(\mathbf{x}, \mathbf{z}) = c \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j) \prod_{k\in V} \phi_k(x_k, z_k) \quad (3.13)$$

with

$\phi_k(x_k, z_k)$ The evidence expressing the statistical dependency between the state and the observation. In the following referred to shortly by $\phi_k(x_k)$.

$\psi_{ij}(x_i, x_j)$ Influence the states x_i and x_j have on each other.

c A normalization constant.

Both ϕ_k and ψ_{ij} are positive, real valued functions [40].

Note, that using PMRFs one can only model pairwise interactions. Interactions with three or more nodes, which could be characterized by some function like ψ_{ijk} can not be expressed. Figure 3.7 shows an example of a PMRF.

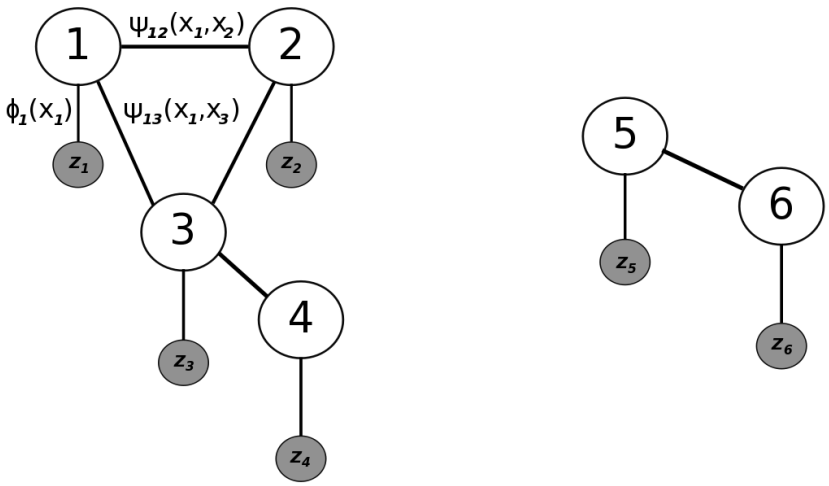

Figure 3.7: An example for the used graphs. The filled nodes denote observations, whereas the labeled nodes denote different objects. A line indicates that there is some dependency between the nodes. Every node is related to exactly one observation. The ϕ and ψ belonging to node 1 are drawn in exemplary. Node 1 represents all its possible states x_1.

Now PMRFs are defined but their use has yet to be explained. One is ultimately interested in the posterior probability $p(\mathbf{x}|\mathbf{z})$. For graphs with no loops, algorithms exist which calculate this posterior in linear time. However, when graphs contain loops, the exact solution becomes too complex to compute exactly and one has to resort to approximative solutions.

Two of these approximative solutions are Belief Propagation (BP) and variational methods. For this diploma thesis, variational methods were chosen, since they can approximate the solution based on a factorized distribution, which is necessary for the concept to work.

3.3.2 Variational Methods

In this diploma thesis variational methods [41, 40] are used to determine $p(\mathbf{x}|\mathbf{z})$. To do this, first the problem is formulated as an optimization problem, where a function $q(\mathbf{x})$, called *variational distribution*, is adapted to fit $p(\mathbf{x}|\mathbf{z})$. To make the solution trackable, two structures are used:

- $q(\mathbf{x})$ is chosen to factorize
- $p(\mathbf{x}, \mathbf{z})$ has the structure of a PMRF as explained above.

The results are the so-called mean field equations, which update each component of $q(\mathbf{x})$ iteratively until a local minimum is reached.

First, the problem is posed as an optimization problem: we optimize the *Variational Distribution* $q(\mathbf{x})$, so it fits the posterior $p(\mathbf{x}|\mathbf{z})$. One does that by minimizing the Kullback-Leibler (KL) divergence of the two distributions:

$$J(q) = arg\min_q KL(q(\mathbf{x})||p(\mathbf{x}|\mathbf{z})) = arg\min_q \int_x q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{z})} \tag{3.14}$$

The Kullback-Leibler distance is a way to measure the proximity of two probability densities. It is 0 if the two densities match and > 0 otherwise. It can be shown that $KL(q(\mathbf{x})||p(\mathbf{x}|\mathbf{z}))$ is a convex function and therefore has only one minimum that is global [41]. Written as it is, the minimization problem formulation is not very useful. Note that one cannot evaluate $p(\mathbf{x}|\mathbf{z})$ easily (in fact, the goal is to approximate it).

We can however convert the problem into the following, using some transformations given in [41]:

$$\begin{aligned} KL(q(\mathbf{x})||p(\mathbf{x}|\mathbf{z})) &= \log p(\mathbf{z}) - \int_X q(\mathbf{x}) \log q(\mathbf{x}) - \int_x q(\mathbf{x}) \log p(\mathbf{x}, \mathbf{z}) && (3.15) \\ &= \log p(\mathbf{z}) - H(q) + E_q[\log p(\mathbf{x}, \mathbf{z})] && (3.16) \end{aligned}$$

with

$H(q)$ The Shannon-entropy of the variational distribution.

$E_q[.]$ The expectation with respect to $q(\mathbf{x})$.

$\log p(\mathbf{z})$ A constant with no influence on the result.

To make the evaluation of $KL(q(\mathbf{x})||p(\mathbf{x}|\mathbf{z}))$ feasible, one needs to assume some structure on q. In this diploma thesis, q is chosen to factorize as follows

$$q(\mathbf{x}) = \prod_i q_i(x_i). \tag{3.17}$$

By doing this, the best q cannot be restored anymore, but merely the factorized distribution, which describes $p(\mathbf{x}|\mathbf{z})$ the best. Furthermore, $J(q)$ exceeds to be convex in this

subspace of q. As a result, one can get stuck in local minima when trying to solve the optimization problem and therefore is not able to restore the best factored representation of q.

However, now each q_i can be optimized on its own with the equation

$$KL(q_i) = -H(q_i) - \int_{x_i} q_i(x_i) E_q[\log p(\mathbf{x}, \mathbf{z})|x_i] - \sum_{k \neq i} H(q_k) + \log p(\mathbf{z}) \tag{3.18}$$

by assuming the other $q_j, j \neq i$ constant. $E_q[\log p(\mathbf{x}, \mathbf{z})|x_i]$ is the conditional expectation defined as

$$E_q[\log p(\mathbf{x}, \mathbf{z})|x_i] = \int_{\mathbf{x} \setminus x_i} \prod_{j, j \neq i} q_j(x_j) \log p(\mathbf{x}, \mathbf{z}). \tag{3.19}$$

According to [42] one obtains

$$q_i(x_i) \propto e^{E_q[\log p(\mathbf{x}, \mathbf{z})|x_i]} \tag{3.20}$$

by setting the derivation to zero.

Up to this point, no use has been made of the PMRF form of $p(\mathbf{x}, \mathbf{z})$. This is done now by setting $p(\mathbf{x}, \mathbf{z}) = c \prod_{(i,j) \in E} \psi_{ij}(x_i, x_j) \prod_{k \in V} \phi_k(x_k)$. Using this and the conditional independencies introduced in the graph, one can derivate the final mean field equations [24]:

$$q_i(x_i) \leftarrow c \phi_i(x_i) \prod_{j \in \mathcal{N}(i)} e^{M_{ij}(x_i)} \tag{3.21}$$

$$M_{ij}(x_i) = \int_{x_j} q_j(x_j) \log \psi_{ij}(x_i, x_j) \tag{3.22}$$

By updating q_i iteratively with this equation, the KL-divergence is decreased monotonically until eventually an equilibrium is reached [42]. Note, that the order in which the different q_i are updated as well as the initial value of the q_i have an effect on the result [41]. For the initialization, $q_i(x_i) = \phi_i(x_i)$ is a reasonable choice.

Figuratively speaking, the mean field equations calculate $q_i(x_i)$ by considering its evidence $\phi_i(x_i)$ and all mean effects of the neighboring nodes. The mean effect a node j has on the state x_i is dependent on how likely the transitions $\psi_{ij}(x_i, x_j)$ are weighted by the probability to be in state x_j, $q_j(x_j)$. This relation is sketched in figure 3.8.

3.3.2.1 Efficient Computation of the Variational Distribution

In [42], an efficient method to compute the variational distribution q using Monte Carlo Methods is proposed. For this thesis a similar algorithm is adapted. Suppose the local

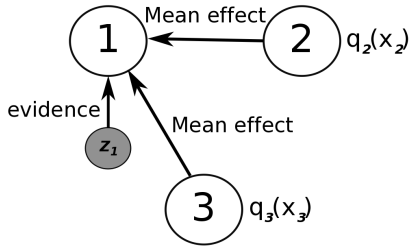

Figure 3.8: Update for $q_1(x_1)$ is done by combining the evidence which supports x_1 a priori and the mean effect each associated node has on node 1.

evidences $\phi_i(x_i)$ are each represented by N particles $\{x_i^{(l)}, w_i^{(l)}(0)\}_{l=1}^N$. Algorithm 1 shows now how to compute the new weights, corresponding to the variational distribution. The number of iterations is fixed to 5, since normally convergence is reached after at most 5 iterations [24].

Algorithm 1 Calculate variational distribution weights

$k \leftarrow 0$
while $k < 5$ **do**
  $k \leftarrow k + 1$
  for all node i **do**
    for all particle index l **do**
      $m_i^{(l)} \leftarrow \sum_{j \in \mathcal{N}} \sum_{m=1}^{N} w_j^{(m)}(k-1) \log \psi_{ij}(x_i^{(l)}, x_j^{(m)})$
    end for
    $w_i^{(l)}(k) \leftarrow e^{m_i^{(l)}}$
    normalize weights
  end for
end while

4 Concept

In this section, it is explained how the concept is composed. The basic components are outlined and then described in detail. Two basic assumptions were made:

- Persons are well characterized by their color histogram.
- Color histograms are not suited to find new persons in an image.

This means, that when somehow a person is detected, the further tracking can be done by following its color histogram. Thus, three parts were designed: one for detecting new persons, one for tracking the detected persons and one for linking the two instances, the manager. Additionally, the counting module is introduced. It decides which persons left the image. It counts the person and orders the manager to delete the person. Figure 4.1 shows the dependencies between the modules graphically.

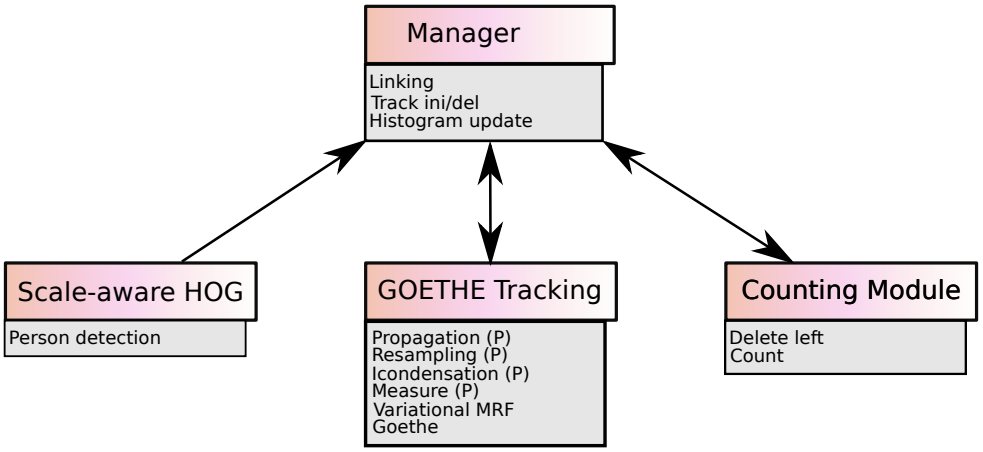

Figure 4.1: The concept modules with responsibilities and dependencies. Parallel performed functions are marked with a (P).

The scale-aware HOG described in the next section is used as the person detector. The main contribution of this diploma thesis however is the tracker. The presented approach can deal with inter-target occlusions and with soft dependencies in the process model while scaling linearly for a modest amount of people (ca. 1-10).

4.1 Scale-aware Histogram of Oriented Gradients

Though the HOG as described in section 3.1.1 delivers very good results off-the-shelf, some extensions were made here. The original HOG scans the image at every scale and every location. Starting by default at the original scale of the image and then scaling the screen until the detection window is as big as the screen. In the special case had in this thesis, this procedure can be improved by utilizing additionally available information about persons' scale.

Since a fixed camera view is assumed, some additional information about the persons' sizes can be utilized. The HOG is supported by three additional, easy to compute values:

- The *height of an average person at the bottom of the screen*. It is used to define the maximum height of an average person.
- A constant *scale factor*, defining how the size of a person changes at different distances in the image. It is computed using two sizes of the same person at different distances: $\frac{\Delta sizes}{\Delta y}$.
- The *minimum size* of a person which should be detected.

Now this information is utilized in two regards: the optimal start- and end-scale is computed using the two given sizes. Additionally, information about the expected size of a person is used. Since it is known how tall a person is at the bottom of the screen, the expected sizes over the whole image are interpolated, using the scale factor. This information is used to discard detection windows which differ substantially from the expected size. The advantages are twofold:

- The detection process is significantly *sped up*.
- *False detections* are *reduced*. Because false detection usually melt two persons to one or detect certain parts of the body like feet as humans.

The price payed is the loss of the ability to identify people which differ substantially from the average height, like children. Figure 4.2 shows some examples of rectangles regarded and sorted out.

4.2 GOETHE Tracking

The tracker developed in this thesis presents a novel method for handling occlusions, dubbed GOETHE (**G**eneral **O**cclusion **E**stimation for **T**racking **H**umans **E**fficiently). The term GOETHE tracking is used equivalent to tracking with GOETHE. The GOETHE tracking is the core contribution of this thesis.

GOETHE tracking is viewed as an alternative to the HjS-filter developed by Lanz [29]. As Lanz, a Bayes Filter which is able to handle occlusions is developed. The advantage of

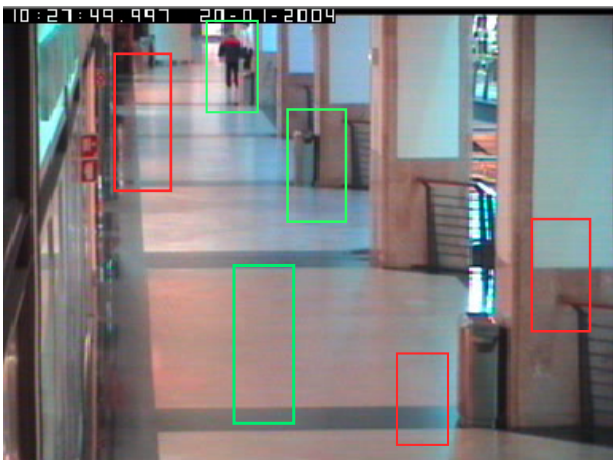

Figure 4.2: Green rectangles are regarded by the scale-aware HOG, red ones are not.

the system here is the almost linear runtime with respect to (w.r.t.) the number of persons, whereas the HjS has quadratic complexity.

As described before, the tracker follows histograms of persons in a video. But, since it is formulated as a Bayes filter, the transition to different metrics for describing humans or other objects can be easily made. Person initialization and deletion are not modeled in the tracker, but done by the manager. Like the HjS, the developed tracking algorithm is a mixture of the joint and the separable approach discussed in section 3.2.2.1 and 3.2.2.2. The goal is to derive an easy to calculate posterior. It should factorize regarding the different persons given a factorized prior $p(\mathbf{x}^-|\mathbf{Z}^-)$. Measurement and particle drift are performed for each tracker independently. To be still able to deal with target occlusion and restore dependencies in the process model, two additional terms are calculated jointly.

The core idea of GOETHE is to model the occlusion state in the state vector, thereby making the measurements independent of each other. The now more tightly connected process model is split, until an easy to compute occlusion term can be processed jointly. The dependencies between the position and motion of the persons are modeled by a MRF interaction model similar to whose presented in [42, 24, 29].

4.2.1 State Space

In the state vector, all known information about the tracked persons is collected. Therefore, the state space it is element of has to be chosen carefully. First, a simple standard state space is introduced. Then it is expanded to the actually used, newly developed state space which is needed for the occlusion handling concept to work.

4.2.1.1 The Simple State Space

In this first design of the state space, the person is described by his position, size, speed and color histogram.

$$x_i = \begin{pmatrix} pos_i = (pos_{x,i}, pos_{y,i}) \\ size_i = (width_i, height_i) \\ v_i = (v_{x,i}, v_{y,i}) \\ hist_i \end{pmatrix} \tag{4.1}$$

pos_i Position of the person given in cm in ground plane coordinates. $pos_i \in \mathbb{R}^2$.

$size_i$ The width and height of the person, given in pixels. $size_i \in \mathbb{N}^2$.

v_i Velocity of the person in cm/s, relative to the ground plane. $v_i \in \mathbb{R}^2$.

$hist_i$ The color histogram of the whole person.

This is how one person is characterized. Since one needs not to track one, but n persons, the state vector is composed by all tracked persons like

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}. \tag{4.2}$$

The state space is defined as the space, which contains all valid state vectors as defined above. This is a standard state space. The acceleration is not modeled explicitly, but implicitly by process noise. The definition of pos_i and v_i in ground plane coordinates ensures that the speed does not change due to scale effects.

4.2.1.2 The GOETHE State Space

The basic idea of the GOETHE state space is to model the occlusion in the state space. The occlusion can be one of several modeled discrete occlusion states like e.g. ”not occluded”, ”upper body occluded” or ”fully occluded”. To be able to do so, first several body parts are defined. An example is given in figure 4.3. Now, the occlusion states can be defined by setting body parts to visible or occluded for the specific occlusion state. Figure 4.4 shows some examples.

To be able to utilize the information denoted by the occlusion state one needs to define one histogram per body part instead of one histogram for the whole person. This is illustrated in figure 4.5.

The splitting has two additional advantages:

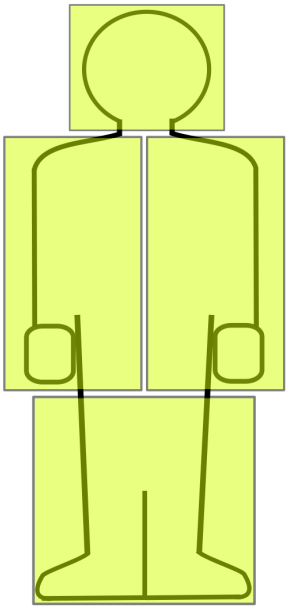

Figure 4.3: Person split up in several body parts. In this example: Head, left upper body, right upper body and feet.

- Different body parts like torso and feet usually have different color histograms due to e.g. different clothes. Therefore, the splitting introduces further model knowledge about the human body.
- The histograms can be updated separately. This enables us to update e.g. just the upper body, if the lower body is occluded.

Note that the shown body parts and occlusion states are only examples. In practice, one has to decide based on the size of the human in the image and the human detection method how many body parts and occlusion states are modeled.

To conclude, the GOETHE state vector formulates as

$$x_i = \begin{pmatrix} pos_i = (pos_{x,i}, pos_{y,i}) \\ size_i = (width_i, height_i) \\ v_i = (v_{x,i}, v_{y,i}) \\ hist_i \\ os_i \end{pmatrix} \tag{4.3}$$

with

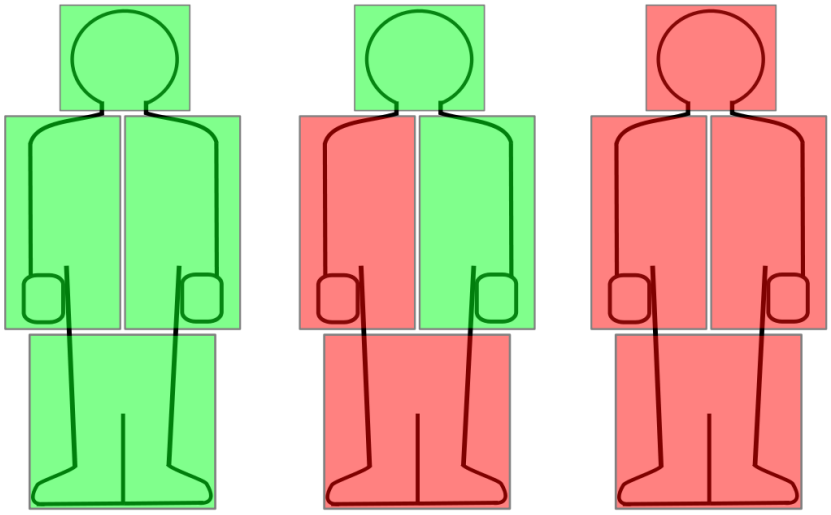

Figure 4.4: Some examples of different occlusion states. Green means that the body part is visible, red denotes occluded. The left picture shows the state ”not occluded”, the middle ”left body side occluded” and the right ”fully occluded”.

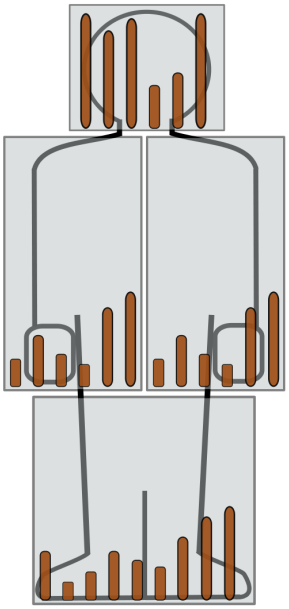

Figure 4.5: $hist_i$ now contains several histograms as opposed to one histogram per person.

$hist_i$ The histograms of the different body parts. E.g. $hist_i = \begin{pmatrix} hist_i^{\text{head}} \\ hist_i^{\text{left upper body}} \\ hist_i^{\text{right upper body}} \\ hist_i^{\text{feet}} \end{pmatrix}$ with $hist_i^{\text{body part}}$ being the actual histogram of the body parts.

os_i A discrete *occlusion state*. It is element of the modeled occlusion states. It is a vector, which assigns every body part a truth value. A truth value of 1 means that the body part is occluded, 0 means it is visible. Let us, for instance, model two body parts, upper and lower body. The truth value for the upper body is in the first row, the one for the lower body is in the second row. Now the occlusion state "lower body occluded" formalizes to $os_{lower\_body\_occluded} = \begin{pmatrix} occ_{upper\_body} \\ occ_{lower\_body} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

and pos_i, v_i and $size_i$ defined as before.

4.2.2 The Filter

As seen in section 3.2.1.1, one needs to specify a *measurement model* $p(\mathbf{z}|\mathbf{x})$ and a *process model* $p(\mathbf{x}|\mathbf{x}^-)$. The computations performed for one persons should be mostly independently from the other persons. When there are dependencies, these need to be

calculable fast. Furthermore, one wants both the measurement model and the process model to factorize w.r.t. different persons. The factorization is needed in order to keep the posterior and therefore the trackers separable. In addition, one needs to find a suited importance density. How process model, measurement model and importance density are found is explained in this section. Moreover, a particle filter specific data fusion technique is introduced in section 4.2.3.

4.2.2.1 The Measurement Model

Normally, the measurements of different persons are strongly dependent. When a person stands in front of another, it occludes the person standing in the back, therefore greatly influencing its measurement.

But, since the occlusion is modeled in the state, the measurements of different persons do not influence one another. Hence one can write

$$p(\mathbf{z}|\mathbf{x}) = p(z_0|\mathbf{x}) \prod_{i=1}^{n} p(z_i|x_i) \tag{4.4}$$

The first factor denotes the probability for observing the background. It depends on all the states and is discarded. Note that by doing this, the demand true Bayesian inference cannot be fulfilled, stating that the measurements are constant and not a function of $\mathbf{x}$ [43]. This is somewhat problematic and leads to the introduction of the occlusion maximum in section 4.2.2.2. There the problem is explicated in detail.

Now the question is, how to determine $p(z_i|x_i)$. Different body parts are assumed to be stochastically independent:

$$p(z_i|x_i) = \prod_{bp \in \text{(body parts)}} p(z_i^{bp}|hist_i^{bp}). \tag{4.5}$$

With $p(z_i^{bp}|hist_i^{bp})$ denoting the one body part measurement probability. According to [29, 44] it can be calculated using the equation

$$p(z_i^{bp}|hist_i^{bp}) \propto e^{-\lambda dist(z_i^{bp}, hist_i^{bp})} \tag{4.6}$$

with

$$dist(z_i^{bp}, hist_i^{bp}) = \begin{cases} dist_{bhattachayya}(z_i^{bp}, hist_i^{bp}) & \text{if } bp \text{ not occluded} \\ 1 & \text{if } bp \text{ occluded} \end{cases} \tag{4.7}$$

where

$dist_{bhattacharyya}(x, y)$ The Bhattacharyya Norm, which measures the similarity between two discrete probability distributions. Here the normalized histogram induced by measurement $z_i^{(bp)}$ is compared to $hist_i^{(bp)}$.

λ A parameter controlling the width of the distribution.

It was chosen like this due to the following reasoning: if a body part is visible, it should result in a good match in the image. Otherwise, the measurement can be arbitrary given the state. The measurement model has now been successfully factorized into easy to calculate terms. Note, that the measurements can be calculated in parallel. Now one needs to do the same for the more complex process model.

4.2.2.2 The Process Model

Normally, the process model of different persons is only loosely connected. Since the occlusion is now in the state, this is not the case anymore. Below it is show how to gradually detach the process model.

$$p(\mathbf{x}|\mathbf{x}^-) = p(\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\\ \mathbf{v}\\ \mathbf{hist}\\ \mathbf{os}\end{pmatrix}|\mathbf{x}^-) \tag{4.8}$$

$$= p(\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\\ \mathbf{v}\\ \mathbf{os}\end{pmatrix}|\mathbf{x}^-) \tag{4.9}$$

$$= p(\mathbf{os}|\mathbf{x}^-, \begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\\ \mathbf{v}\end{pmatrix}) p(\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\\ \mathbf{v}\end{pmatrix}|\mathbf{x}^-) \tag{4.10}$$

$$= p(\mathbf{os}|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) p(\underbrace{\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\\ \mathbf{v}\end{pmatrix}}_{\mathbf{x_{mov}}}|\mathbf{x}^-) \tag{4.11}$$

$$= \underbrace{\prod_{i=1}^{n} p(os_i|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix})}_{\text{occlusion model}} \underbrace{p(\mathbf{x_{mov}}|\mathbf{x}^-)}_{\text{motion model}} \tag{4.12}$$

In the second row the histograms are excluded, since they are not tracked but updated externally by the manager. Now one has two terms. One can be regarded as an occlusion

model, the other as a motion model. Both the occlusion and the motion are still dependent on the global state and need to be further simplified. Let us first assume a forward model for the persons' motion given by

$$p(\mathbf{x_{mov}}|\mathbf{x}^{-}) = p(\mathbf{x_{mov}}) \prod_{i=1}^{n} p(x_{mov,i}|x_i^{-}). \tag{4.13}$$

By doing this, the reasoning of [29] is followed, that the position, size and velocity of one person is highly correlated, but inter-person correlations are relatively weak. An example for such weak interaction would be a couple browsing through a store, staying in close proximity. It is assumed that these weak interactions can be described by $p(\mathbf{x})$, i.e., without dependency on the previous time-step. The strong intra-person interactions $p(x_{mov,i}|x_i^{-})$ can however not be made time invariant. The persons' velocity at time-step t depends e.g., on its previous velocity.

For calculating the intra-person interactions a constant velocity model is assumed. Acceleration is modeled via normal distributed noise. Because ground plane units are used to describe pos_i and v_i, the result is scale invariant. The change in size is calculated by assuming a constant scale factor in x- and y-direction. Because of the constant scale factors, only a planar ground can be modeled.

Further it is assumed, that the inter-person relations can be described by pairwise interaction potentials. Only persons in a close neighborhood should influence each other. The different persons therefore form a graph as shown in figure 4.6.

Figure 4.6: An example for the pairwise interactions modeled. Persons in a certain proximity are linked by an edge. This graph is constructed at every time-step.

This simplifies the motion model to

$$p(\mathbf{x_{mov}}|\mathbf{x}^-) = \prod_{(i,j)\in V} \psi_{ij}(x_{mov,i}, x_{mov,j}) \prod_{k=1}^{n} p(x_{mov,k}|x_k^-). \tag{4.14}$$

Now the motion model is factorized as far as possible. This formulation is used in section 4.2.2.3 to efficiently calculate the prediction via PMRFs. In the following, the occlusion term is further analyzed.

Detaching the Occlusion Model First, the occlusion term is split in occlusions of different body parts:

$$p(os_i|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) = \prod_{bp\in\text{body parts}} p(occ_{bp}|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) \tag{4.15}$$

with

$$p(occ_{bp}=1|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) = p_{occ}(bp|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) \tag{4.16}$$

and

$$p(occ_{bp}=0|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) = 1 - p_{occ}(bp_i|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}). \tag{4.17}$$

To make clear how to compute the occlusion of one body part, let us first have a look on how to compute the occlusion for the pixels in this body part. For that, the *occlusion map* is created. The occlusion map stores the probability for every pixel in the picture to be occluded by an object that stands in front of it[1]. The occlusion map thus is a picture with the size of the current observation. We initialize it all black, indicating that the probability for the pixels to be occluded is 0. Now all the persons in front of i are projected on the image plane and their pixel value is set to 1. To determine if a pixel is occluded or not, one can simply look up its value on the occlusion map. An example of these occlusion maps is given in figure 4.7.

Now one can define a body part as occluded, if at least half its pixels are occluded.

$$p_{occ}(bp_i|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) = \begin{cases}1 & \text{, if more than half of a body part's pixels are occluded}\\ 0 & \text{, otherwise}\end{cases} \tag{4.18}$$

We rewrite this definition to better fit the later purposes. Let O denote the number of body part pixels with an occlusion probability higher than 0. Let furthermore denote

[1] A person is defined to be in front of another if its y-value in image coordinates is higher. This assumption only holds if the camera is horizontally aligned

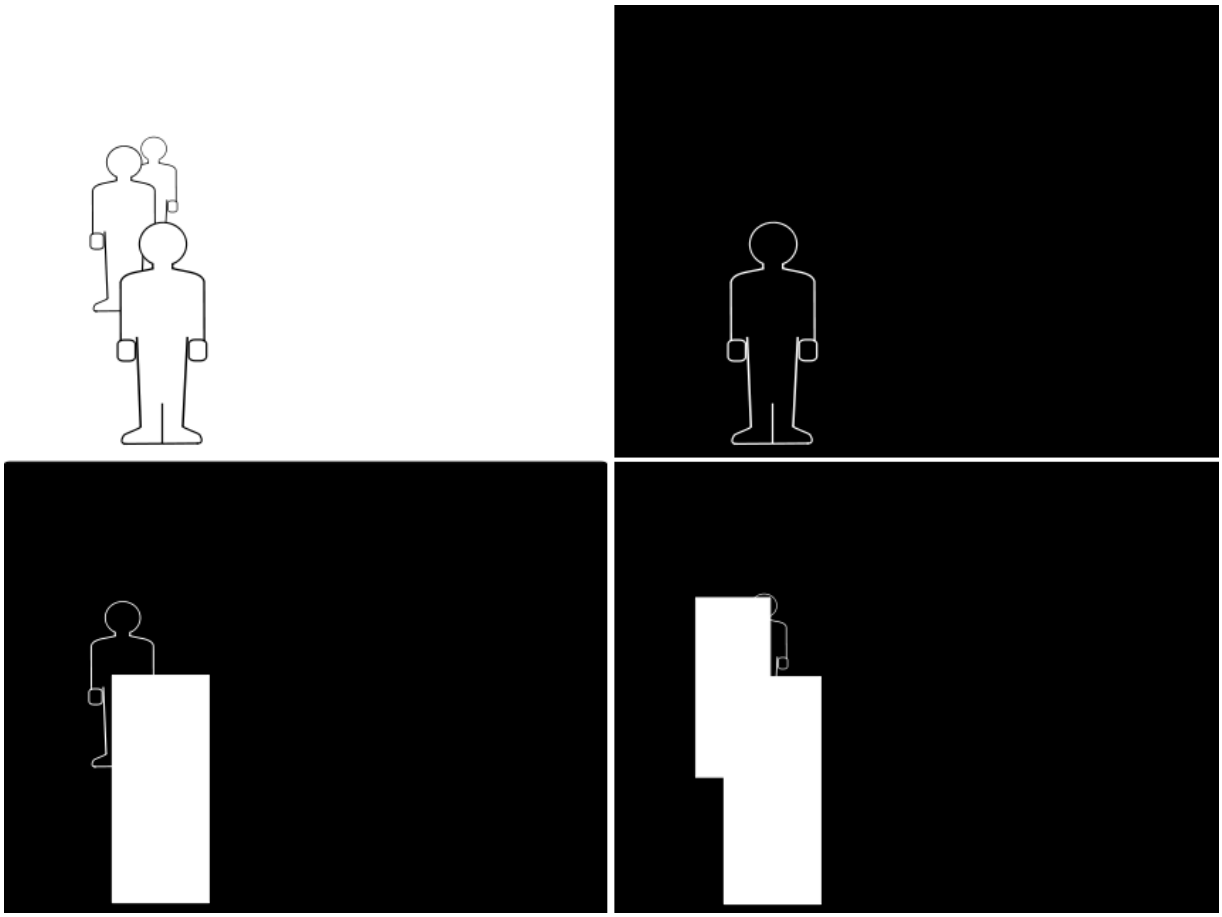

Figure 4.7: The picture in the upper left shows a configuration of people. The other three pictures show the occlusion maps. The silhouettes of the persons are drawn in to show how the persons fit in the image and are not part of the real occlusion maps.

$occmap(pixel)$ the occlusion probability of the pixel given its occlusion map (i.e., the value that is stored in the occlusion map at position $pixel$):

$$p_{occ}(bp_i|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix})=\begin{cases}\frac{1}{O}\sum_{pixel\in bp_i} occmap(pixel) & \text{, if more than half of a body part's pixels are occluded}\\ 0 & \text{, otherwise}\end{cases} \tag{4.19}$$

That means the average of all occluded pixels is taken as the occlusion probability, in case more than half of the body part is occluded. Since the occlusion probability for one pixel is always 1 when occluded, this is equivalent to the previous definition. This definition will be handy later on, when the occlusion map is approximated for person i. Note that this modeling of discrete occlusion states introduces an approximation error.

Now it was shown how the occlusion term can be evaluated. But because of the dependence on $\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}$ it still operates on a global space. Hence the trackers are not separable and the effort is still significant due to the curse of dimensionality.

Here is a small example scenario to visualize the curse of dimensionality in this situation: suppose, persons can only stand at discrete positions. Presume further, that three persons are tracked and in front of person 1 are 100 possible positions to stand. Now, there are 100^2 possibilities for both other persons to stand in front of person 1, 200 combinations for only one of the persons standing in front of person 1 and one combination with person 1 as the frontmost. This makes a total of 10201 possible combinations. Suppose a fourth person has to be tracked. Now there are 1020301. It is evident that the combinations, which have to be covered by samples in the implementation, explode exponentially.

These combinations should be subsumed into one *expected occlusion* for person 1. This expected occlusion is an occlusion map as described above. It summarizes all the occlusions which could happen in front of person 1. Therefore it is not dependent on any specific relation of person 1 to the other persons 2 and 3. Unfortunately, it is not possible to assign absolute values for the occlusion like 0 or 1, since one has not given one combination but rather has to guess using all possible combinations.

Given this occlusion map, the probabilities for the occlusion states are evaluated according to equation 4.19. Note, that now this definition is indeed useful, since one does not have only discrete values 0 and 1 for the occlusion probability anymore. Now, one can write

$$p_{occ}(bp_i|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) \approx p_{occ}(bp_i|E_{y,i},\begin{pmatrix}pos_i\\ size_i\end{pmatrix}) \tag{4.20}$$

$$\Rightarrow p(os_i|\begin{pmatrix}\mathbf{pos}\\ \mathbf{size}\end{pmatrix}) \approx p(os_i|E_{y,i},\begin{pmatrix}pos_i\\ size_i\end{pmatrix}). \tag{4.21}$$

$E_{y,i}$ is the occlusion map of person i up to distance y. It denotes the expected occlusions up to distance y. This means, that the occlusion of the trackers is now factorized w.r.t. the different persons. The goal is reached, if a way to compute the occlusion map efficiently is found. This is discussed in the next topic.

Computing the Occlusion Map Let us first establish that

1. A person can only be occluded by other persons in front of him.
2. A person cannot occlude himself
3. One pixel in the image can only be occluded by one person at once.

Because of the first fact, the occlusion map can be computed recursively. One starts with a black occlusion map (i.e., every pixel is assigned the value 0).

Say an occlusion map for all persons, E_y, has been computed until height $pos_{y,i}$ and a person x_i is given, which stands at height $pos_{y,i}$. Now the occlusion map has to be updated to accord for person x_i at the current position. The occlusion map denotes the expected occlusion over all possible former states. Thus the update must somehow "grade" the different states. This is done with the use of the current measurement z_i. Every pixel is regarded which is occupied by x_i in the image. Let us say pixel $pixel$ lies in the region of visible body-part bp_i and the value of the occlusion map at the pixel's position is $occmap(pixel)$. Now one can assign

$$occmap(pixel) := max(dist(z_i^{bp_i}, hist_i^{bp_i})(1-p_{occ}(bp_i|E_y, \begin{pmatrix} pos_i \\ size_i \end{pmatrix})), occmap(pixel)) \tag{4.22}$$

It was chosen like this, because one pixel can only be occluded by one person at once. The most probable is assigned, which resembles an "or" connection. Vividly speaking this means, that if a person fits good at the measurement, persons behind him will have a high probability of being occluded. Additionally for every pixel in the occlusion map the person causing the occlusion has to be saved. The occlusion map for person i, $E_{y,i}$, can be computed from E_y: $E_{y,i}$ is composed of all values of E_y not caused by person i. This comes from fact two, a person cannot occlude himself. Figure 4.8 shows the concept of the occlusion map graphically.

Note that one can also model scene knowledge by setting the scene elements' pixels to occluded after their height is surpassed in the picture. This becomes handy if one wants to model persons leaving through exits.

In practice, a problem already mentioned briefly in section 4.2.2.1 occurs. A Person's particles tend to be "sucked in" by persons walking besides him, if the other person can be measured better in the image. This is a consequence of not modeling the background in the measurement model. If the background was regarded, the hypotheses saying the

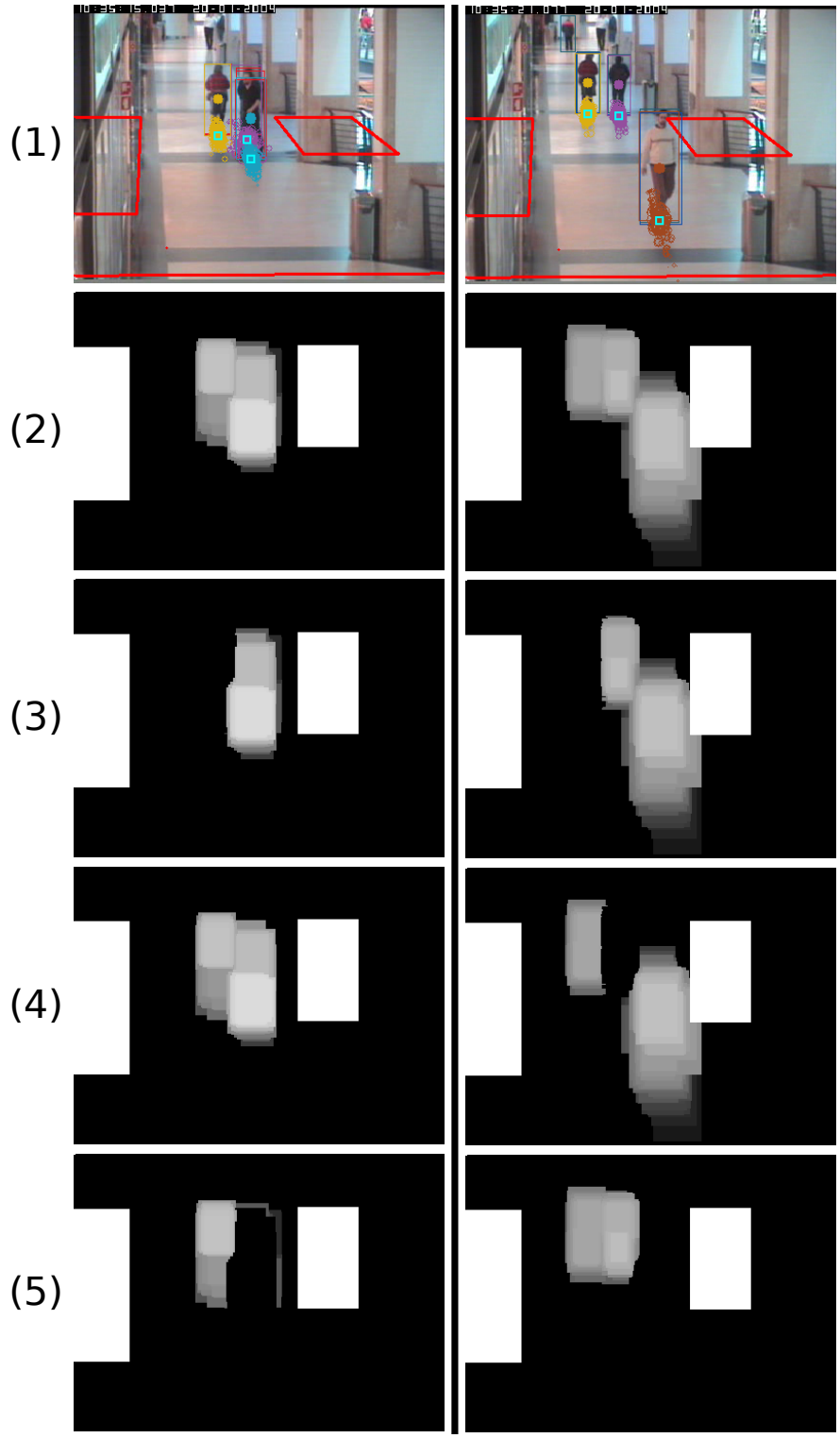

Figure 4.8: Two examples for occlusion maps. The maps show the final results after processing all the particles. Row (2) shows the global occlusion map. The other maps show the occlusion maps of the different person, which can be generated out of the global occlusion map and the map storing which pixel is occluded by which person. Row (3) shows the occlusion map of the yellow person, row (4) the occlusion map of the purple person and row (5) the map of the third person. In the first column it can be clearly seen, that the person in row (4) creates almost no occlusion volume, because it is completely occluded by the person in row (5). In all occlusion maps one can see the world occlusion caused by the store entrance and the pillar. Note that these are the occlusion maps before being thresholded by occ_{max}.

person is behind the other would not be probable, because the background was altered by the visible person.

To overcome this problem, a maximum for the occlusion occ_{max} is set which cannot be surpassed. Ideally, the maximum's value is higher than typical responses from the background and lower than the minimal measurement response a person can receive. Thus particles stay at the person, if he is visible. For correctly assigning the occlusion values to the right person, the real value for the occlusion probability of a pixel is stored internally. Note that Lanz [29] has a similar problem, which leads him to introduce the occlusion penalty d_{occ}, a constant penalty factor for occluded pixels.

4.2.2.3 The Prediction

Let us now substitute both motion and occlusion in the prediction, assuming the prior is available in a factorized form:

$$p(\mathbf{x}|\mathbf{Z}^-) \approx \prod_{i=1}^{n} p(os_i|E_{y,i}, \begin{pmatrix} pos_i \\ size_i \end{pmatrix}) p(\mathbf{x_{mov}}|\mathbf{Z}^-) \tag{4.23}$$

with the *motion prediction*

$$p(\mathbf{x_{mov}}|\mathbf{Z}^-) = p(\mathbf{x_{mov}}) \prod_i p(x_{mov,i}|Z_i^-) \tag{4.24}$$

$$= \prod_{(i,j)\in V} \psi_{ij}(x_{mov,i}, x_{mov,j}) \prod_{k=1}^{n} \int p(x_{mov,k}|x_k^-) p(x_k^-|Z_k^-) d\, x_k^- \tag{4.25}$$

The motion prediction models how one thinks a person moves from $t-1$ to t, whereas the occlusion denotes the occlusion person i will be in, given the positions of the other persons in the picture. If the motion prediction can be calculated effectively, the goal is reached.

Calculating the Motion Prediction The motion prediction is modeled as a PMRF as described in section 3.3.1. The PMRF is built newly in every frame, depending on the persons relative positions. As evidence $\phi_i(x_{mov,i})$ for node i, $p(x_{mov,i}|Z_i^-)$ is used. As interaction potential, a exclusion principle

$$\psi_{ij}(x_{mov,i}, x_{mov,j}) = \begin{cases} 1 & \text{, if } |x_{mov.i} - x_{mov,j}| > thresh \\ \varepsilon & \text{, otherwise} \end{cases} \tag{4.26}$$

with ε near zero can be implemented. It assigns low values if persons stand very close to each other, thus avoiding that two persons occupy the same place.

Since one is interested in a factored representation, variational methods as in [42, 24] are used to infer the best factored representation of $p(\mathbf{x_{mov}}|\mathbf{Z}^-)$. The result is then

$$p(\mathbf{x_{mov}}|\mathbf{Z}^-) \approx \prod_{i=1}^{n} q_i(x_{mov,i}). \tag{4.27}$$

Now, the overall prediction factorizes to

$$p(\mathbf{x}|\mathbf{Z}^-) \approx \prod_{i=1}^{n} p(os_i|E_{y,i}, \begin{pmatrix} pos_i \\ size_i \end{pmatrix}) q_i(x_{mov,i}) \tag{4.28}$$

meaning the goal is reached, because the prediction is now fully factorized w.r.t. the different persons.

4.2.2.4 Integration in the Particle Filter Framework

This section discusses how to integrate the proposed filter into the particle filter framework. We initialize n different particle filters, each tracking one person. These particle filters contain "superparticles", which do not have a specific occlusion state. The result of the variational calculations $q_i(x_{mov,i})$ is chosen as importance density. It is evaluated by using algorithm 1. The particles are sorted by y-value. Now the super-particles are split into one particle for every occlusion state, meaning one always calculates every occlusion state. This is a negligible overhead after the values for the occlusion probability have been calculated. The weight is determined iteratively using the equation

$$w_i^{(l)} \propto p(z_i|x_i^{(l)}) p(os_i^{(l)}|E_{y,i}, \begin{pmatrix} pos_i^{(l)} \\ size_i^{(l)} \end{pmatrix}) w_i^{(l)-}, \sum_l w_i^{(l)} = 1. \tag{4.29}$$

Where E_y is updated with each processed particle. Figure 4.9 shows one iteration of the new particle filter graphically.

4.2.2.5 GOETHE Tracking Summary

Let us make a quick recap on what was done in this section. First it was shown, that the measurement model is easy to evaluate when the occlusion is modeled in the state. Then the process model was gradually detached until it finally was represented by one model for the occlusion and one for the motion. It was explicated how the real occlusion state can be approximated using the occlusion map. Furthermore it was shown how to compute the occlusion map recursively to keep the computational load manageable. Although Lanz [29] also utilizes occlusion maps, his method of computing them is different from

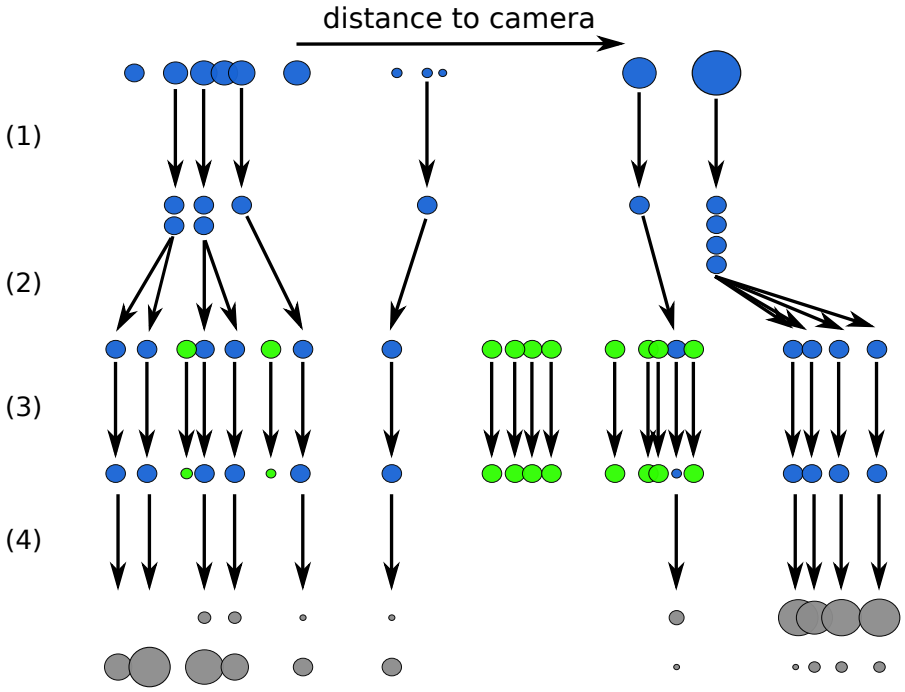

Figure 4.9: The basic steps of one iteration of the particle filter graphically presented. Blue particles are the super-particles. Their weight is determined by adding the weight of all the occlusion states. As in the SIR filter, the particles are drawn and propagated in step (1) and (2). For the next step, the green particles belonging to a second person are drawn in. In step (3), PMRF are used to model that 2 persons cannot occupy the same space. Thus particles which occupy a region, where another person probably stands, are weighted down. In step (4) the super-particles are finally split into several particles with an occlusion state. The occlusion probability is determined by the occlusion map. This occlusion probability combined with the probability for the measurement and the old weight to form the new particle weight. The first row symbolizes the occlusion state “fully occluded”, the second row “not occluded”. Note, that the particles in the front have low to no probability to be occluded. The particles on the right have a high occlusion probability, since the person characterized by the green particles stands in front of it. All these particles would have had a much lower weight, if the occlusion would not be considered.

the one in this thesis, leading to quadratic complexity, as opposed to pseudo-linear as in our approach (see section 5.3 for the runtime analysis). This method for managing the occlusion is the main contribution of this thesis. The drawback to the GOETHE approach is that it is less principled than Lanz's HjS filter. The GOETHE approximation for the occlusion map is based more on logical reasoning than on mathematical derivation.

After dealing with the occlusion, the motion was treated. The persons' motion was modeled by a pairwise Markov Random Field. By doing this, it is possible to model some weak inter-person interaction. Variational methods are then used to infer the factorized motion prediction. Now the Bayes Filter is completely separated and evaluates as

$$p(\mathbf{x}|\mathbf{Z}) = \prod_i p(z_i|x_i)p(os_i|E_{y,i}, \begin{pmatrix} pos_i \\ size_i \end{pmatrix})q_i(x_{mov,i}). \tag{4.30}$$

4.2.3 Icondensation

Up to this point, a fully functional tracker was developed. However, the information provided by the object detector is not directly utilized. Therefore, Icondensation [45] is used as a technique for data fusion naturally fitted to the particle filter framework. The basic idea is the following: as before, the goal is to estimate the posterior pdf as good as possible by a finite amount of samples. When drawing the samples as described before, it can happen that they are not concentrated in the interesting regions of the state space. This is possible because the object moves unexpectedly, is occluded for a long time or "bad" samples are drawn by chance.

Icondensation now draws some particles in the area of an additional measurement, thus ensuring that the particles cover that area of the state space. This area is defined by a normal distribution around the measurement. The covariance is obtained from the training data. Since the drawing of the particles has been influenced by the measurement, importance sampling is again used to guarantee the samples converge to the posterior pdf. One here looks how probable it is to draw the sample from the original samples and divide it by the probability to draw the sample from the object detector measurement:

$$\lambda_i^{(m)} = \frac{\sum_{l=1}^{N} w^{(l)-} p(x_{mov,i}|x_i^-)}{g(x_i^{(m)})}. \tag{4.31}$$

Where m denotes the index of the particle drawn by Icondensation. $g(x_i^{(m)})$ denotes the probability to draw $x_i^{(m)}$ from the measurement. This correction factor is simply multiplied to the weight in equation 4.29. Since the measurement does not contain any information about the velocity of the object, it has to be drawn out of a fixed velocity distribution.

In Icondensation, the particles are split into three groups: one group that is drawn classically, one which is drawn from the measurement and corrected using importance sampling, and one that is drawn from the measurement without correction factor. The last group is introduced for making it possible to recover after big failures. An example for the Icondensation particle drawing can be found in figure 4.10. How many particles are

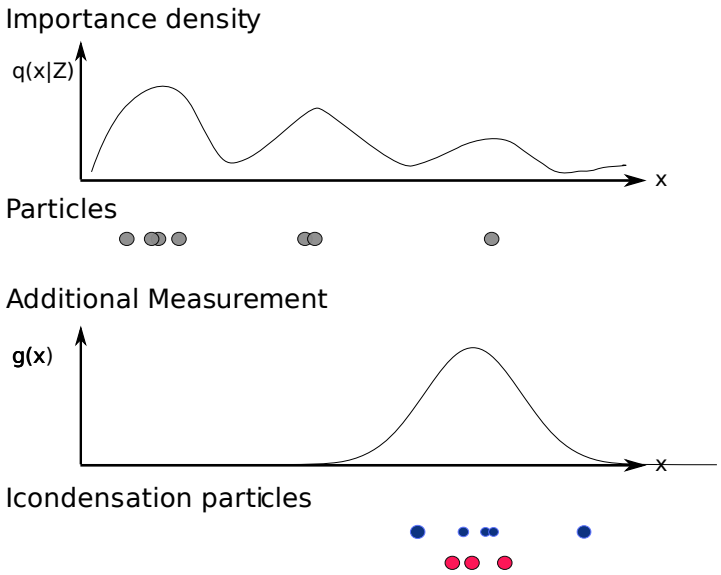

Figure 4.10: The Icondensation particle drawing. The normally drawn particles are drawn in grey. The blue particles are Icondensation particles drawn according to the measurement. The particles' weight is dependent on the importance density and the additional measurement. The red particles are the ones which are drawn out of the additional measurement but not re-weighted.

drawn from each group is a fixed ratio in the original Icondensation. We, however, use variable ratios provided by the manager described in the next section. Figure 4.11 shows Icondensation in this application.

In practice, Icondensation results in a matching of the tracker to the measurement, if the available measurement fits well to the human model of the tracker. If not or if no measurement is present, the person is tracked normally with the tracking algorithm. Icondensation is especially useful, if a person was occluded for a long time with a lot of possibilities for the occluded person to stand. Now the particles form hypotheses for all these "hiding places" resulting in distributed samples. After the person reappears, the samples can be re-concentrated very fast on the person using Icondensation (see figure 5.20 in the experiments chapters). This is important, since particles otherwise tend to stay in the occlusion,

especially if the person is not recognized very well in the image. In that regard, Icondensation and occlusion handling form a natural couple. The occlusion handling keeping the particles in place and the Icondensation re-concentrates them after the person reappears.

Figure 4.11: Icondensation in action. The circles on the ground represent the particles, the bigger the circle, the bigger the particle weight. The small bordered rectangle on the ground shows the expectation of the tracker. The rectangles around the persons show measurements from the HOG. A line in the color of the particles to the rectangle means, that the measurement has been matched to the tracker of this color. As one can see, the tracker is first a little off when predicting the person. When the measurement is available, the particles are drawn mostly from it, guiding them into the right direction.

4.3 Manager

The manager connects the HOG person detector, the tracker and the counting module. The manager is responsible for the linking of HOG measurements to the trackers, track initialization/deletion and the update of the tracker histograms. These functions are explained one by one in the following.

4.3.1 Linking

Both track initialization/deletion and histogram update are dependent on the linking built in this step. At every time-step the manager collects all the measurements from the HOG and all the predictions from the tracker. Then it tries to connect the predictions to the measurements. Every prediction is connected to at most one measurement. This problem can be formulated as a bipartite graph, with the measurements on one side and the predictions on the other. Every measurement is connected to every prediction with an edge of a weight $dist_{linking}$ defined later. What one wants to find is the best matching between the

two groups. For this, the Hungarian algorithm [46, 47] is applied. Because the Hungarian algorithm requires quadratic matrices, the missing fields are filled with zeros. If the best matching of a certain measurement/prediction is lower than a certain threshold, it is not considered a successful match.

This means one does not pursuit several different hypotheses like in a MHT or JPDAF, but rather a nearest neighbor approach is followed. The main reason are the much higher computational costs when using a MHT or JPDAF. Additionally, the information given by the persons' histograms usually is sufficient for a reliable linking.

A proximity measure is defined for measurements and predictions $dist_{linking}$ between 0 and 1. This distance measure is composed of three factors:

1. A factor d_{prox} weighting down pairs which are far away from each other.
2. A factor d_{size} comparing the pairs' sizes.
3. A factor which determines the histogram similarity of the pair d_{hist}.

The distance factors d_{prox} is defined as

$$d_{prox} = e^{-\frac{1}{2}((\frac{\Delta x}{\sigma_x})^2+(\frac{\Delta y}{\sigma_y})^2)} \tag{4.32}$$

and the size factor according to

$$d_{size} = e^{-\frac{1}{2}(\frac{\Delta size}{\sigma_s})^2}. \tag{4.33}$$

The distance of the histograms is evaluated by comparing each body part pair using the Bhattacharyya norm. E.g., the upper body of the measurement is compared to the upper body of the predictions histogram. Then the minimal coincidence is the resulting d_{hist}. A special case occurs when the measurements suggest that a body part overlaps with another measurement. Then its distance is set to 1, meaning this body part is not regarded. The reason is, that the histogram is likely to contain colors from another person occluding it or standing near to it. The overall distance is evaluated according to

$$dist_{linking} = d_{prox} d_{size} d_{hist}. \tag{4.34}$$

If $dist_{linking}$ is below a certain threshold, the measurement-tracker pair is not taken into account.

4.3.2 Person Initalization/Deletion

Not every measurement has a valid corresponding tracker prediction. Every measurement which is not matched initializes a new preliminary tracker. If this tracker is not linked in the next frame, it is deleted again. This way, almost all false positives are filtered, since

it almost never occurs that the HOG performs two false classifications in a row. Now the tracker has a property called trust threshold. This indicates the time in a row a tracker can exist without a measurement being linked to it. If it is surpassed, the person is deleted. This is first set to 5, after another measurement it is set to 25. This is a quite high value. But note that usually no measurement is available if the person is occluded, resulting in no linking of the person. The threshold can be adjusted according to the occlusion state of the person to reduce this problem. The ability to use the knowledge denoted by the occlusion state is another advantage resulting from modeling the occlusion in the state.

An example of the linking and initialization is given in figure 4.12. Further ameliorations of this concept might use a more sophisticated, probabilistically motivated approach as in [25] for initialization and deletion.

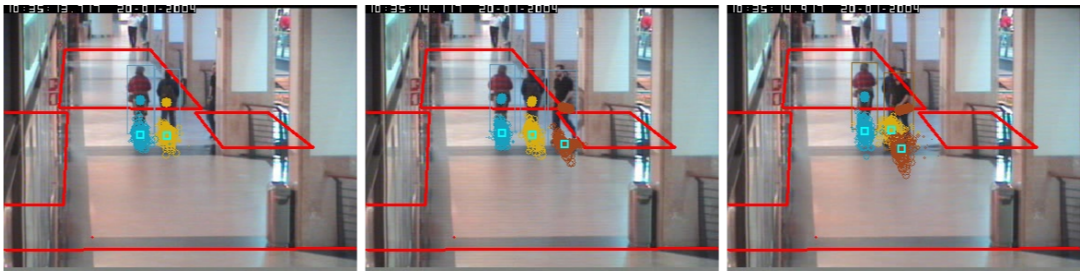

Figure 4.12: The linker in action. When the third person arrives, the tracker cannot match it to a tracker, thus it initializes a new tracker. In the third frame, the person with the yellow particles is not measured. Using the Hungarian algorithm, the linker matches the two available measurements to the correct trackers.

4.3.3 Histogram Update

Since the tracker does not update the persons' histograms itself, the manager updates them to account for changes in lighting, direction and available information, which changes with the persons' size in the image. For every prediction, which is linked to a measurement, the color histogram is updated. The old and the new color histograms are fused. How much the bins are updated, depends on how good the match between the prediction and the measurement is:

$$bin_{tracker} := (1 - dist_{linking})bin_{tracker} + dist_{linking}\, bin_{measurement}. \tag{4.35}$$

If another measurement suggests that the person is occluded, only the not occluded parts of the person are updated. This way, the color histograms are kept up-to-date. Furthermore, the measurement is passed to the tracker, so it can initialize new particles based

on it via Icondensation. The better the match, the more particles are initialized via Icondensation. Making both the histogram update and the number of Icondensation particles depending on the quality of the match prevents the tracker from failing, if a wrong assignment is made.

To make the histogram creation and updates more robust, a standard background subtraction method is utilized. Pixels belonging to the background are not considered, when a person's histograms are created. Note that this background subtraction is not an essential part of the system. It does not matter if it is not very correct contrary to approaches which rely on background subtraction.

4.4 Counting Module

All entrances are modeled as "counting regions". These counting regions are rectangles defined for every camera in ground plane coordinates. An example is given in picture 4.13.

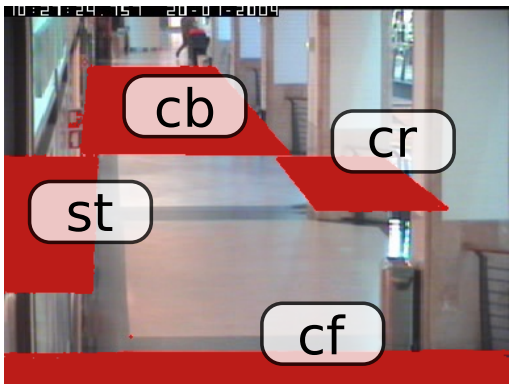

Figure 4.13: The counting regions in an example. If a person spends some time in a counting region, it is counted. If it is initialized in a region it is not counted until it first leaves the region. If it is initialized outside a region the person's entrance is set to the nearest region.

Every time a person is initialized the counting module determines the nearest region and defines it as the person's entrance. If the detected person is too far away from any entrance, the entrance is set to "unknown". When the person steps again on a counting region, it is assumed that it left through the entrance defined by this region. To be counted, the person must spend a certain amount of time outside of the areas, followed by a certain amount of time inside one area (here, both are set to 3 consecutive frames). This avoids multiple counting, when a person walks along one of the regions or stands near one of them (see

also figure 4.14). The counting module then sends this information to the manager, which deletes the person from the tracked objects.

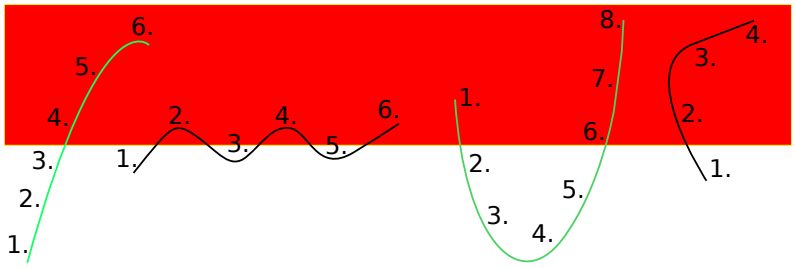

Figure 4.14: For being counted, a person has to stay a certain amount of time outside and inside the red counting region. In this example, green trajectories are counted, black trajectories are not.

The object manager stores the entrance and exit of a person in a counting matrix, which has one field for every entrance/exit combination. Information about occlusions of scene objects is modeled in the tracker. By doing that, a person can be counted *after* he left the picture, e.g., through a door. His particles are "gravitated" towards the regions through which he left.

5 Experiments

In the experiments the basic parts of the system are evaluated. First a look at the HOG descriptor is taken in section 5.1. Here it is seen how the HOG performs on the CAVIAR dataset and how the inclusion of scale information changes the result. Then the tracker is evaluated in section 5.2. Some challenging situations were staged by us to test if the tracker can handle them. Finally section 5.3 evaluates the system with the data from CAVIAR to see if good counting and tracking performances can be achieved.

All computations are carried out on a workstation with two Core(TM)2 Duo processors with a clock rate of 3.0GHz. The program is coded in C++ using the OpenCV library [48] for most of the image manipulation tasks. The Boost library [49] is utilized in various occasions. The HOG library available at [50] is also used.

5.1 Scale-aware HOG Evaluation

The HOG has been extensively evaluated in [1]. Nevertheless, the suitability of the HOG for a security camera application is evaluated by testing the HOG on the CAVIAR dataset. It is also tested how the runtime and accuracy of the original HOG compares to the scale-aware HOG .

5.1.1 Experimental Setup

Both the traditional HOG and the scale-aware HOG are run on the 26 corridor videos in the CAVIAR dataset. It contains a total of 36405 frames at a sampling rate of 25Hz. For this evaluation, it is down-sampled to 5Hz. Note that no general statement about the HOG performance is made, since the frames are correlated.

Two ratios are computed:

- The *false detection rate* informs us about the number of false positives with respect to the total number of detections.
- The *missed detection rate* is the ratio of all missed detections to the sum of all missed detections and true positives.

A problem is when to declare a person correctly detected. Demanding a perfect match of the detection window and the ground-truth would be asking too much. However, just demanding that they somehow overlap would be a too loose constraint. Therefore, the false detection rate and missed detection rate are computed as a function of the mutual overlap: rectangle one has a total area of $a1$, rectangle two an area $a2$ respectively. Let both have an area ac in common. The mutual overlap is now defined as

$$min(\frac{ac}{a1}, \frac{ac}{a2}). \tag{5.1}$$

By this definition, both rectangles have to overlap each other to a certain degree, so that it is for example impossible for the HOG to always detect the whole image and get counted as a good detection. How to compute the mutual overlap is visualized in figure 5.1. Additionally it is made sure, that every person is matched to only one detection. If more than one detection is available, the person is matched to the best.

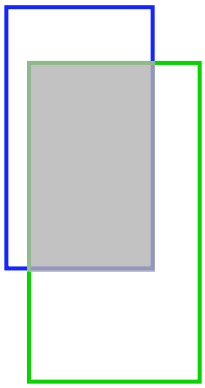

Figure 5.1: The definition of the overlap visualized. In this case, the mutual overlap is the ratio between the grey area and the area of the green rectangle, because the ratio of the grey area and the blue area would be bigger.

This alone leeds to lots of missed detections due to very small or occluded humans. The effects of these on the final result are investigated by filtering missed detections having their origin in small or occluded humans.

5.1.2 HOG Results

First a look at the false detection rates is taken. These are important, since false detections can result in the initialization of new trackers, leading to possibly faulty results. As one

can see in figure 5.2, the scale-aware HOG can significantly lower the false detection rate. This is, because it does not detect false detections which are significantly bigger/smaller than the average person. Figure 5.3 shows the same situation and the results of both detectors to visualize this issue.

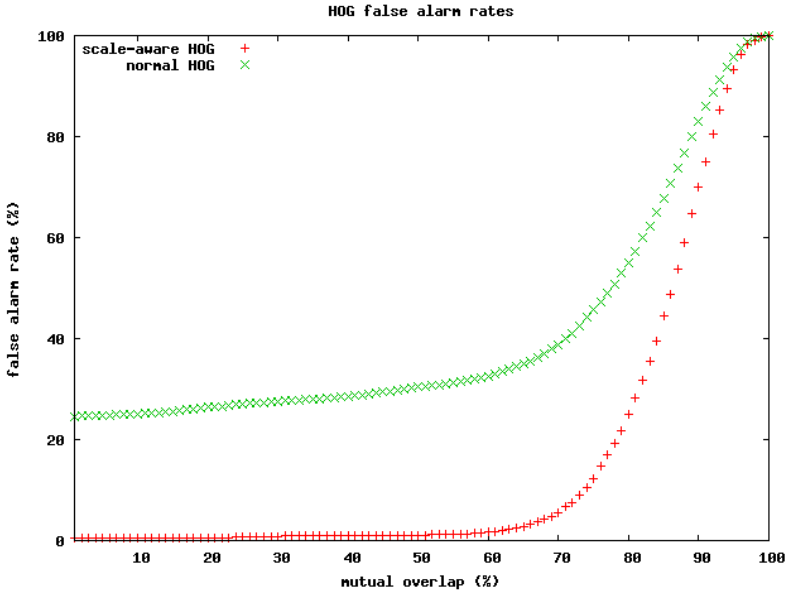

Figure 5.2: False alarm rate on both the original and the scale-aware HOG.

Then comparing the missed detection rates in figure 5.4, one can see, that the original HOG misses less persons than the scale-aware HOG. There are two reasons for this. First, often there are small parts of humans, which are detected as humans. These explain the lower missed detection rate at small overlaps. Second, persons differing substantially from the norm (e.g. kids) can be detected while the scale-aware HOG might not search for them. However, no example of this case is found in the test data.

If one asks for a mutual overlap of 70-80%, it can be seen, that the missed detection rates are similar, with the non-scaled HOG performing slightly better. At this overlap, the normal HOG performs significantly worse than the scale-aware HOG regarding false classifications. Note that missed detections are not very severe in most cases, since the tracker normally can follow the object without any HOG measurements. However, if a person is detected not even once, the system is not able to track it.

Both missed detection rates are pretty high. There are three reasons for this. For one, occluded persons are counted as missed detections. The HOG cannot handle occlusions, thus these misses are natural. Also, there are many persons in the image which are very

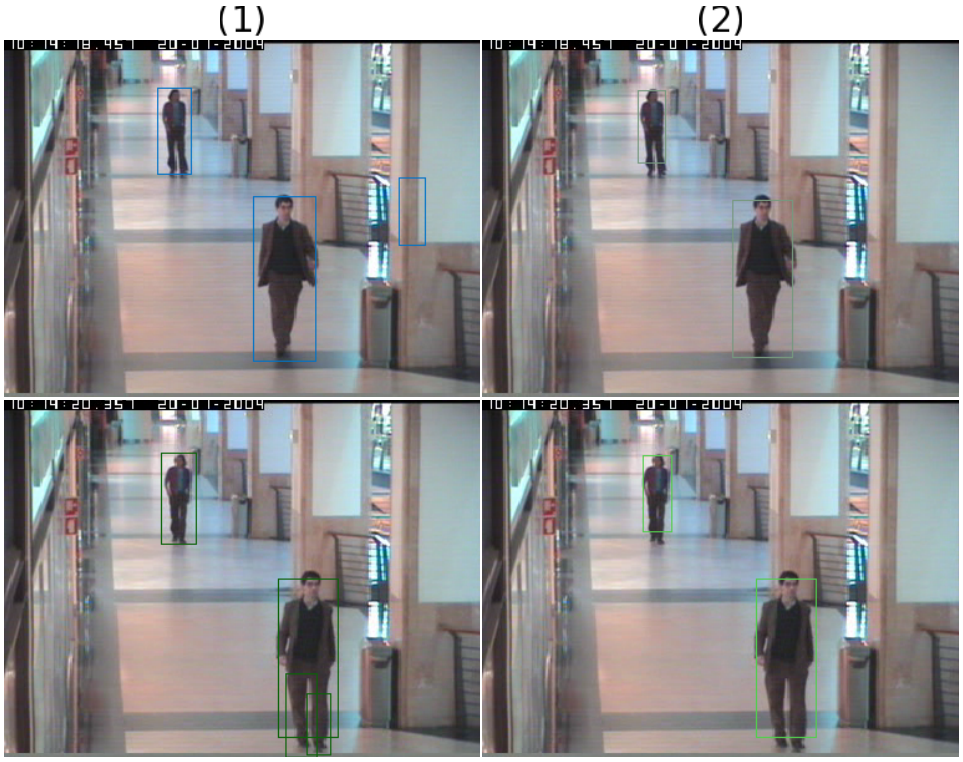

Figure 5.3: Comparison of original HOG and scale-aware HOG. the first column shows misclassifications of the original HOG. The scale-aware HOG shown in the second column is more robust to these misclassifications.

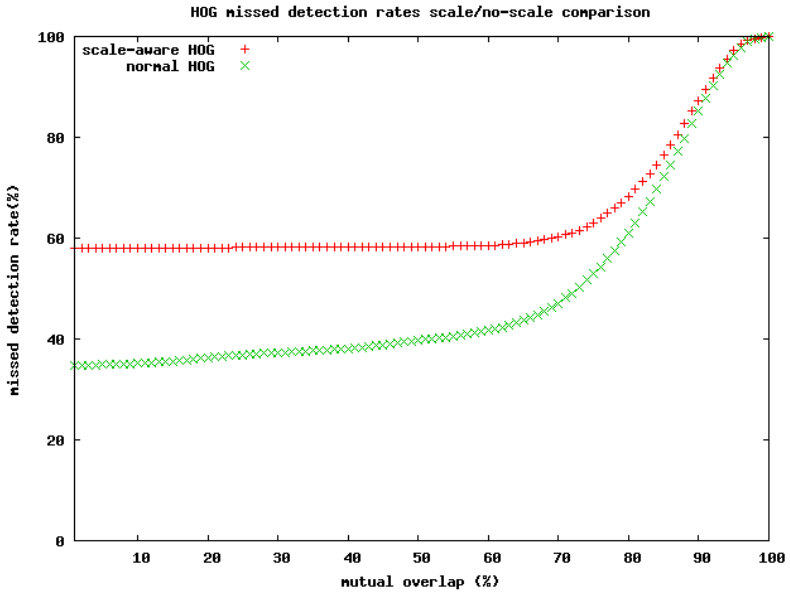

Figure 5.4: The original HOG misses less persons than the scale-aware HOG.

small (<40 pixels). The HOG does not give stable results if persons are that small. The third reason is also connected with occlusions. If a person is going through an exit and is only partly visible, the labeled rectangle is only the visible part (see figure 5.5). In graph 5.6, these causes will be filtered out and compared to the original result.

We can conclude that the HOG gives very good overall results with a very low false positive rate and a reasonably low missed detection rate. The missed detections are not very critical, since the tracker can follow the person even if it is occluded and not visible for some time or simply not detected from time to time. The only problem arises, when a person is never detected or not detected for a long time.

5.1.2.1 Computational Time

The computational time in the example is reduced almost by the factor 2. While the normal HOG needs constant $6.12\,s$ to perform the detection for one picture, the scale-aware HOG only needs constant $3.64\,s$. Note, that the time safed is dependent on the scale range which has to be scanned in the scene. With large changes in scale, the time saved is also more significant. Also note, that there is no need to scan every possible size for humans in the image. The tracker only tracks persons taller than about 60 pixels

Figure 5.5: Picture showing the two types of occlusions: inter-person occlusions and occlusions caused by a person walking through an exit.

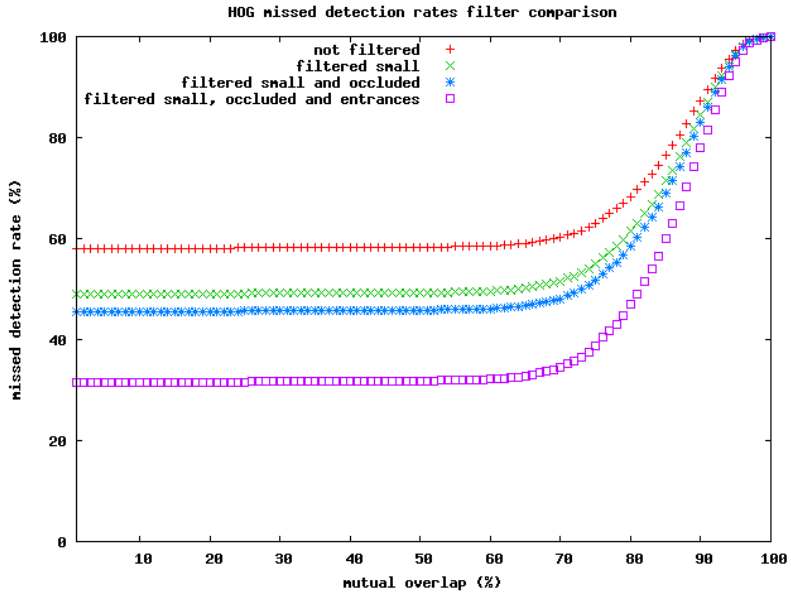

Figure 5.6: The missed detection rates if the natural causes for some missed detections are filtered. What stays are “really” missed detections, persons which should have been detected but were not.

robustly. If the HOG is used only to scan for these sizes, it only needs $2.68\,s$ to process an image.

Unfortunately, these runtimes are nowhere close to real-time. But the reason for the HOG to be chosen was also its principal ability to run in real time. Zuh et al. improve the HOG in [35] with blocks of varying sizes and a cascade of rejectors to speed up the calculations by the factor 20. This would speed up the calculations to about 5Hz, which is sufficient for real time performance, since the tracker also works with 5Hz and both object detection and tracking can be performed parallel. The time scope of this thesis did not allow to implement this approach, but it would be the next logical step in future development.

5.2 GOETHE Tracking Evaluation

For the evaluation of the tracking system some videos were filmed some videos which stage some difficult tracking problems. The focus is on evaluating how occlusion handling improves the tracking results. Also, an extensive analysis of computational time, both theoretical and real, is given.

5.2.1 Experimental Setup

Several videos sequences were filmed, designed to test the different characteristics of the tracker. The videos were filmed at a frame rate of 5Hz, containing a total of 590 frames. The data was labeled manually to obtain the ground-truth. Note that this ground-truth is not always 100% correct, since the position of occluded persons can only be guessed from the video camera data. For this dataset, Icondensation is turned off, since the goal is to evaluate the "pure" performance of the tracker. Also, initialization and deletion of persons is disabled, the persons are initialized manually.

A two part body model is used, modeling the upper and lower body. Possible occlusion states are "not occluded", "lower half occluded" and "fully occluded". The tracker in the experiments always uses HS-histograms with 32 hue and 30 saturation bins. The HS color-space is chosen, since it is less variant to illumination changes. The occlusion maximum is set to 0.15. The velocity uncertainty is set to $0.6\frac{m}{s^2}$ standard deviation. To test the success of the tracking, the distribution of the posterior mass around the ground-truth, plotted as a function of the localization error as in [29] is determined. In other words, the integral of the posterior pdf is evaluated dependent on the distance from the ground truth. The values are evaluated by summing the normalized particle weights in the area around the ground truth. This way, we can see how "far off" the posterior is when predicting the person. To make the result more stable, the average of 10 runs is taken. Additionally, key frames are analyzed to show interesting situations.

5.2.2 GOETHE Tracking Results

Here the filmed videos are introduced and analyzed one at a time. Afterwards, some additional experiments are carried out on all of the videos. The frame-by-frame results of the videos are available in the CD attached to the thesis.

Two Person Occlusion Two persons are present in this video. They walk towards each other and stop when they cross, leading to a long, total occlusion of one person. They do this several times.

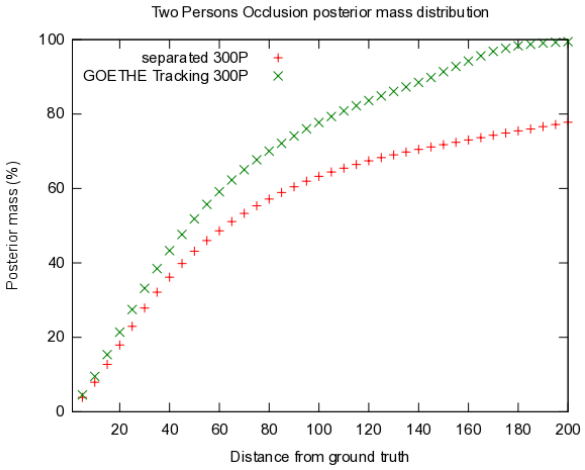

Figure 5.7: The posterior mass distribution of the GOETHE filter and the separate filter of the "Two Person Occlusion" video. Both utilize 300 (super)particles.

As one can see in graph 5.7, the system developed here outperforms the normal separate tracker. Figure 5.8 shows a keyframe analysis.

Three Person Random Walk This is a longer sequence without any choreography. Mostly the persons are only shortly occluded, but a difficult occlusion of one person happens once.

As on can see, the posterior masses of the two trackers are similarly distributed. This is because there are little long total occlusions present. In cases of one person just walking behind another and reappearing immediately, the separate tracker has no problem, because particle filters can handle some missing measurements by using the predictions from the last time-step. The separate tracker performs even better in these situations, be-

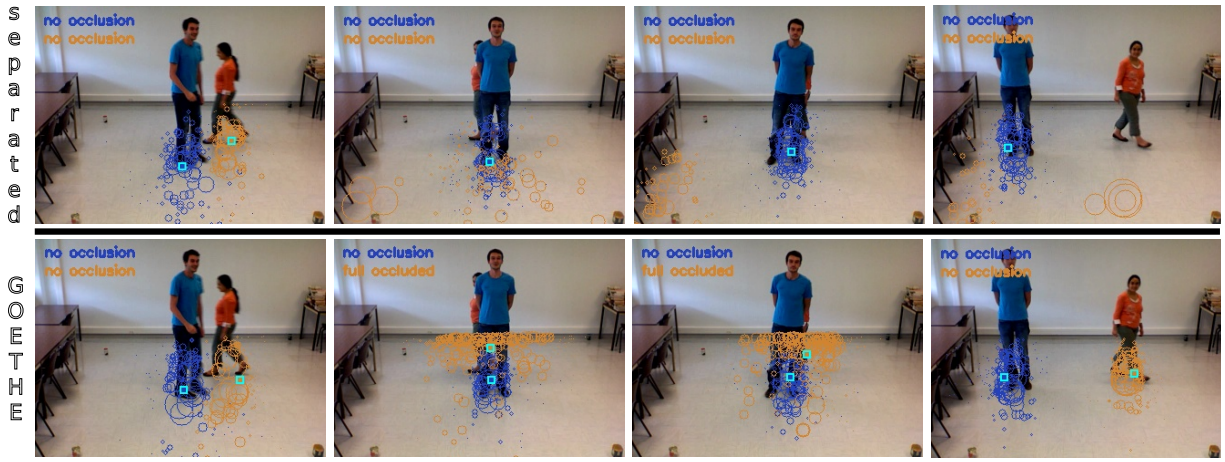

Figure 5.8: Keyframes (13, 24, 42, 53) of video 'Two Person Occlusion". The woman walks behind the man in the blue t-shirt, stays for a long time and leaves. The upper row shows the separate tracker. Here the woman is lost almost immediately after she enters the occlusion. In the GOETHE tracker, the particles stay behind the man. After she is reappearing, the tracker can lock onto her quicker. Also note, that the GOETHE tracking can guess the present occlusion state of a person.

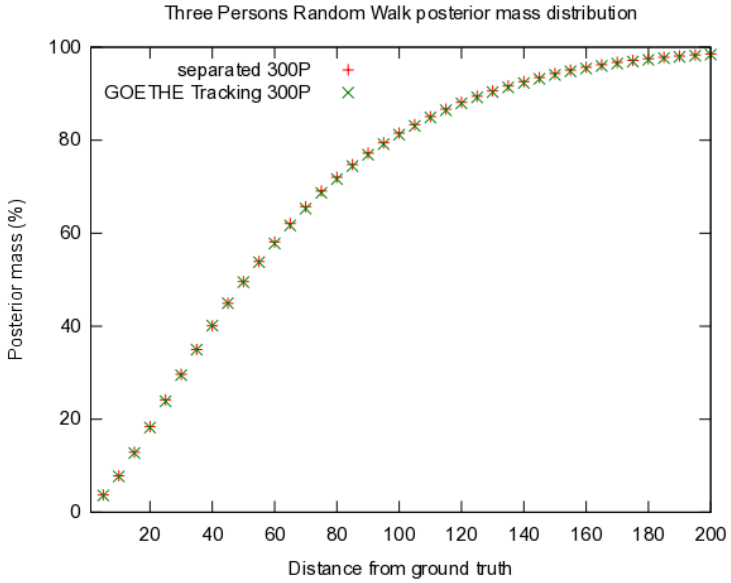

Figure 5.9: The posterior mass distribution of the GOETHE tracker and the separate filter of the "Three Person Random Walk" video.

cause the GOETHE tracker always keeps some particles in the occlusion volume of the passed person for some time.

There is one interesting scene, in which two persons pass behind another, leading for the separate tracker to lock on the wrong person, while the joint tracker keeps hypotheses for the person being behind one of the two persons (see figure 5.10).

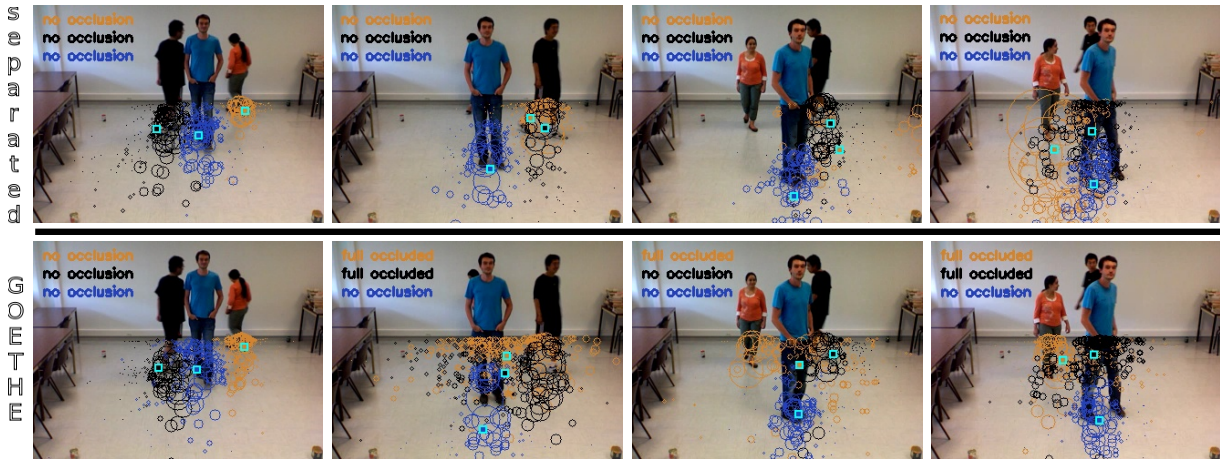

Figure 5.10: Keyframes (46, 52, 58, 59) of video "Three Person Random Walk". Two persons walk behind another. The black person reappears faster, dragging the particles of the separate tracker away from the yellow person. It is only able to lock on the target again, because the black person crosses the yellow person later. The GOETHE tracker pursues two hypotheses: the yellow person could be behind either of the two others; after it reappears, the particles are quickly drawn away from behind the blue person and the woman can be tracked again.

Four Person Random Walk The most challenging video in the set. Here four persons walk randomly. One very long total occlusion appears in which one person crosses almost the whole scene while being occluded.

Although here the posterior distribution is again not very different, the results in fact are. Then one has a look at the posterior distribution of the long occluded person in figure 5.12, one can clearly see the difference. The similar posterior distribution can be explained by the presence of four persons, three of which are not occluded for a very long time. This reduces the effect of the one occluded person on the posterior distribution. Also note that, as stated above, the separate tracker performs slightly better in cases of short term occlusions. The key frame analysis in figure 5.13 shows the results of both trackers during a long term occlusion of one of the persons.

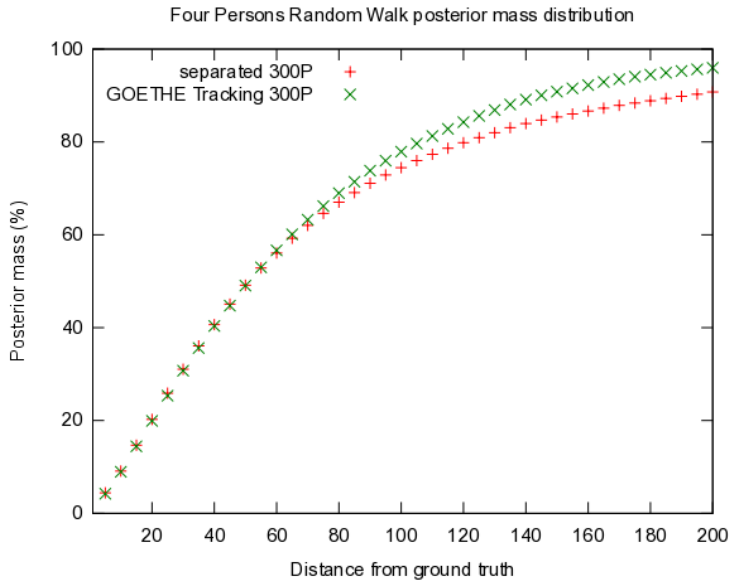

Figure 5.11: The posterior mass distribution of the GOETHE tracker and the separate filter of the "Four Person Random Walk" video.

5.2.2.1 Other Considerations

The tracker was also run on only 100 superparticles. While the posterior mass distributions look almost the same, the de-facto tracking result is worse. When turning unexpectedly, often persons are lost for a short time when only 100 superparticles are available. Additionally, less particles lower the ability to cover all of the occlusion volume, which leads to worse results during occlusions with many possible positions of occluded people.

The advantage of the pairwise MRF is keeping two similar looking persons separated. Unfortunately, the persons in the videos are very distinct. But have a look at figure 5.14, to see results from the CAVIAR dataset, where similar looking persons in close proximity are regularly present.

5.2.2.2 Computational Time

In this section, first the theoretic worst case complexity of the tracking algorithm components is derived and then supported by real time measurements.

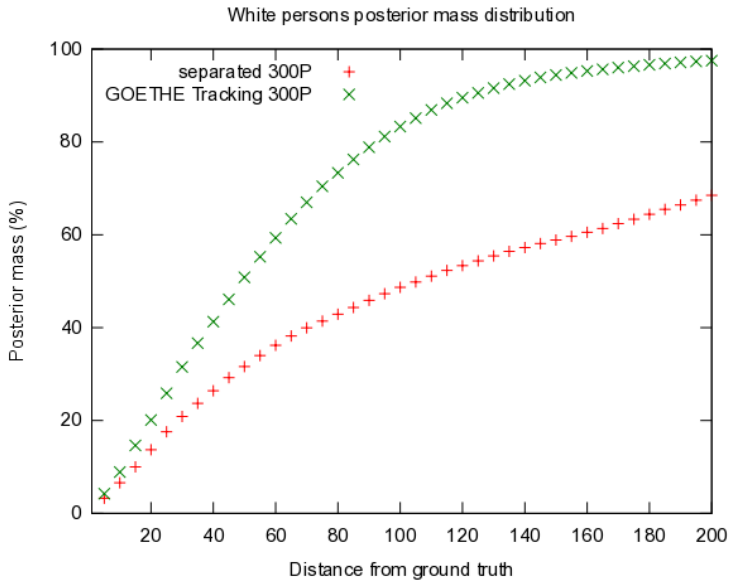

Figure 5.12: The posterior mass distribution of the GOETHE tracker and the separate filter of the occluded person in the "Four Person Random Walk" video.

- The normal *separable particle filter* runs $\mathcal{O}(NPR)$, with N denoting the number of persons present, number of particles P for one person and average rectangle size R.

- The *occlusion handling* concept has a total complexity of $\mathcal{O}(3NPR+NP\log(NP))$ which is element of $\mathcal{O}(NPR\log(NP))$. The $NP\log(NP)$ component is caused by the necessary particle sorting. Evaluating the values for p_{occ} of all particles takes another NPR, since all pixels in all rectangles have to be regarded. The building of the occlusion map needs $2NPR$. One NPR is needed to update the map itself and another one for storing which person caused the occlusion. This means, that the complexity of the occlusion handling is pseudo-linear concerning the number of persons. This is a great advantage to the system of Lanz [29], which has quadratic complexity.

- Contrary to what is stated in [42], the *variational computations* do not run in linear time but have a complexity of $\mathcal{O}(KNP(N-1)P) \in \mathcal{O}(K(NP)^2)$. K is the number of iterations, here set to 5. The other component is because in the worst case, every particle of a person has to be compared to every particle of the other persons.

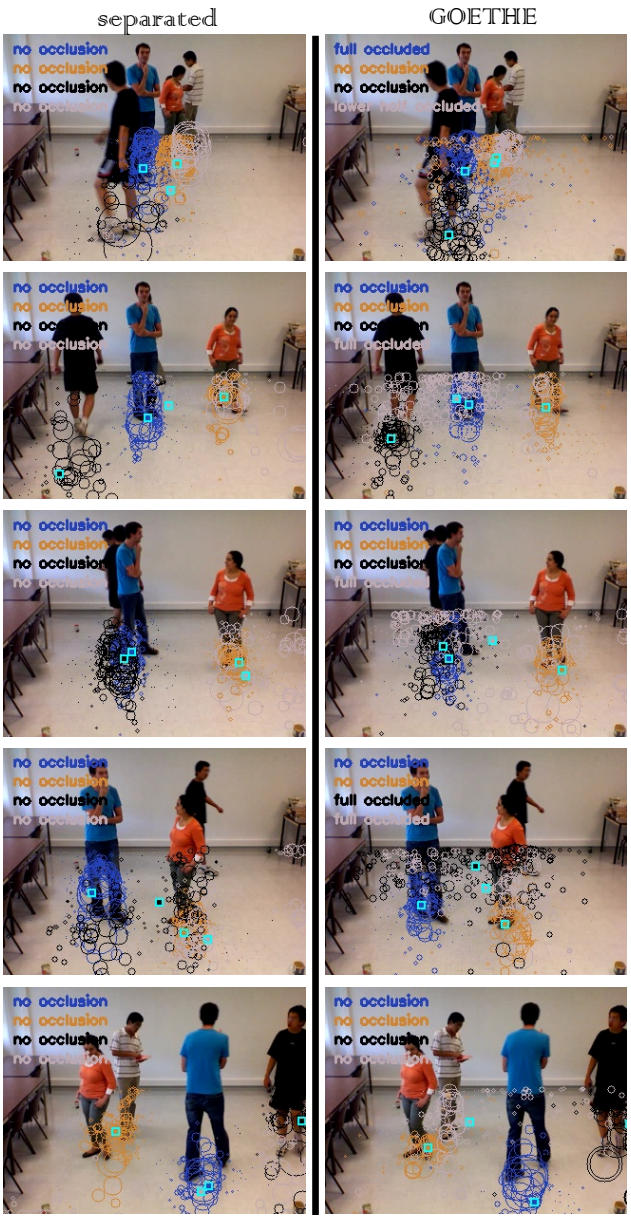

Figure 5.13: Keyframes (35, 41, 46, 52, 63) of video “Four Person Random Walk”. The person in the white t-shirt is occluded for a long time with many possibilities of where he could hide. As one can see the separate tracker in the upper row loses track fast, while the GOETHE tracking always supports multiple hypotheses where the person could be hiding.

Figure 5.14: PMRFs in practice. The upper row shows the tracker without use of variational methods. Here the left and right trackers mix. The second row shows the tracker with variational methods. Here the particles stay clearly separated.

nr. persons	meas	occ	var	ic
1	100/33Hz	100/33Hz	>100/50Hz	>100/100Hz
2	100/25Hz	100/16Hz	>100/50Hz	>100/100Hz
3	100/16Hz	33/14Hz	100/25Hz	>100/100Hz
4	50/12Hz	25/12Hz	100/25Hz	>100/100Hz
5	50/11Hz	25/11Hz	50/20Hz	>100/100Hz
6	20/11Hz	12/9Hz	50/20Hz	>100/ >100Hz

Table 5.1: runtime of the tracker (avg/max).

- *Icondensation* complexity is $\mathcal{O}((NP)^2)$ because the weight of every particle drawn by Icondensation has to be evaluated by regarding all the normally drawn particles.

This is the theoretical runtime. Table 5.1 sums up the real runtime of the filter. As it can be seen, the filter runs in almost linear time concerning the number of persons. The reason is, that the computations costs of the measurement process dwarfs the much faster arithmetic computations performed for Icondensation, variational methods or sorting, at least for a reasonable number of persons. In practice, the parallel performed measurement process has almost the same runtime as the theoretically much more demanding occlusion calculations. This comes from the much more complicated calculations performed during the measurement process, like creation of the histograms and calculation of the histogram norms. Unfortunately, the rectangle size has a big influence on the runtime of the measurement process and the occlusion. Therefore, the results are somehow less meaningful, because we like to only see the influence of a changing number of persons.

5.3 Overall Evaluation

This final evaluation tests how the different components of the system interact on the challenging CAVIAR public data set. Again frame-by-frame results are available in the CD attached to this thesis.

5.3.1 Experimental Setup

The system is evaluated on the CAVIAR corridor dataset on all 26 videos. The videos are down-sampled to 5Hz. The resolution of the videos is 384x288. Although the scenes are not very cluttered, very complex situations resulting in many occlusions occur. This makes the dataset a very challenging tracking problem. Several counting regions are defined in the video shown in figure 5.15. The same body and occlusion model as for the tracker evaluation is used. The occlusion maximum was set a little higher to 0.4. It can be set higher, because Icondensation takes care off pulling particles out of the occlusion mass fast. Without Icondensation, particles easily get "stuck" in the occlusion mass, if the occlusion maximum is set too high. Due to Icondensation, we can only use 100 superparticles and still obtain a stable tracking.

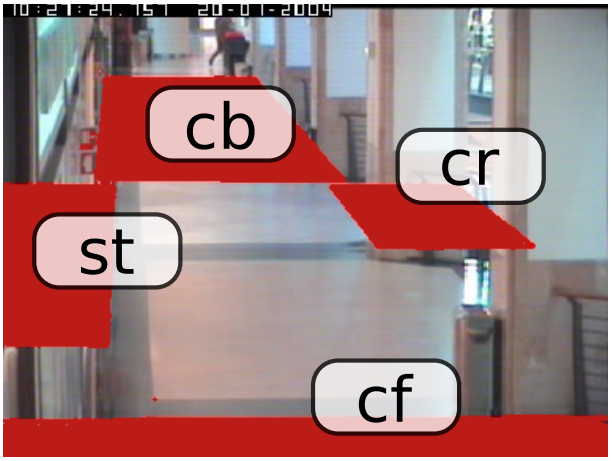

Figure 5.15: Used counting regions. Four model entrances are modeled. The stores (st), corridor back (cb), corridor right (cr) and corridor front (cf)

Two kinds of experiments are performed. One measuring the counting performance and one doing a more general evaluation of the tracking system.

For computing the counting success of a video the number of persons leaving through the exits is counted. It is compared to the real number to get the total errors for each exit.

This is done for each video and the errors are summed up. Then the information is used to compute the total errors for every exit. With this information, one can easily compute the percental errors from all the exits and the overall percental error. To receive a stable result, 10 complete iterations are taken into account.

The second part of this evaluation measures the quality of the tracker. This is not equal to the counting success, for which one only needs reliable tracking near the entrances. Furthermore, the error sources for bad counting results and bad tracking results are different. Note that if track information about from where to where the persons went should be extracted, a good tracking performance is obligatory. The only publication doing an extensive tracking evaluation on the CAVIAR dataset is [25], which is described in section 2.2.1 in the state of the art chapter. To be able to compare this system to theirs, the same seven criteria are used:

- The number of *mostly tracked* trajectories. A trajectory counts as mostly tracked, if more than 80% of its trajectory was tracked by a single tracker. A trajectory point is defined as tracked, if a tracker's expectation and the ground-truths intersection is 50% of their union. As in the linking process, the prediction is matched to the ground truth rectangles using the Hungarian algorithm.
- The number of *mostly lost* trajectories. A trajectory is mostly lost if more than 80% of the trajectory is lost.
- The number of *tracking fragments*. A fragment is a trajectory which is less than 80% of the ground-truth trajectory.
- The number of *false trajectories*. A false trajectory is a trajectory corresponding to no real object.
- The number of *identity switches*. An identity switch occurs when a pair of result trajectories exchanges its identity.
- The number of successful *short-* and *long-term occlusions*. An occlusion is successful, if the tracker assigned to the track is the same during the whole duration of the occlusion. An occlusion is labeled long-term if it persists for more than 50 frames.

These evaluation criteria are visualized in picture 5.16.

Additionally, the tracker's performance under occlusion is measured. An object is defined as occluded, if more than 50% of its ground truth rectangle is occluded. To avoid multiple counts when the occlusion "flips" around 50%, hysteresis is applied, defining the beginning of an occlusion after passing 50% and the end of an occlusion after dropping below 33%. Note that Wu et al. give no criteria in [25], so the values here differ from theirs, limiting the possibility of comparing the two systems.

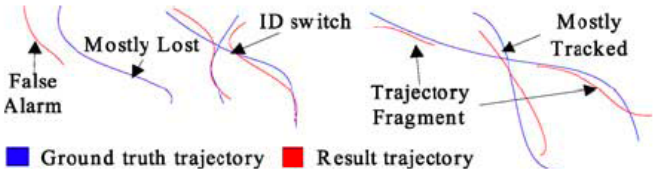

Figure 5.16: The criteria used for the overall tracking summed up [25].

There are some problems with the annotations as shown in figure 5.17. Also, it is not explained very clearly in [25] what the criteria for a tracked trajectory point is. The intersection of the tracker expectation and the ground truth rectangle is demanded being at least 50% of their union. This is Wu et al.s' value for a correct detection when evaluating the part-based detector. The tracker is also run demanding only 10%. That way, one can counteract the annotation issues and see if the tracker really looses the persons or just does not track them very accurately. To make the results more consistent with those of Wu et al. the CAVIAR data is not down-sampled for this experiment.

Figure 5.17: The problems with the annotation data. In the left image the ground truth rectangles are drawn over the whole persons, even if they are not completely visible. In the right picture, the ground truth rectangle of the occluded woman in the center is only the size of the visible parts.

5.3.2 Counting Results

As one can see in table 5.2, the counting results show a good overall performance of about 85-90% overall counting success. The results are especially good for the frontal entrance

	st	cf	cr	cb	total
GT	55	52	12	25	144
Avg	8.7(15.8%)	3.1(5.9%)	2.0(16.6%)	4.6(18.4%)	18.4(12.8%)
Worst	10(18.1%)	5(9.6%)	2(16.6%)	6(24%)	23(16.0%)
Best	6(10.9%)	2(3.8%)	2(16.6%)	5(20%)	15(10.4%)

Table 5.2: Absolute and percental errors compared to ground truth. Avg: Average 10 iterations; Worst: Worst iteration; Best: Best iteration; st: stores; cf: corridor front; cr: corridor right; cb: corridor back

with a counting success around 95%. This can compete with results from non-vision based systems as in [4].

The results for the frontal entrance are better than the results of the other entrances. The main reason is, that persons at the main entrance are supported by more evidence in the picture, resulting in better HOG and tracking performances.

We see that most errors are made at the corridor back entrance. The reason is, that persons get very small near this entrance, which worsens the HOG and tracking performance considerably. Also, one pixel error in the back of the picture equals more distance in ground plane coordinates than in the front. This of course further degrades the results of the HOG and linking.

The errors on the store and right corridor entrance are mostly caused by the need to count people after they entered, leaving no evidence for the persons in the image. The modeling of world-object occlusion make this possible. However, when persons stand near to entrances and do not find much support in the image, they are drawn into the exit, resulting in a false count (see figure 5.18). This can for example occur, if the persons histogram changed much since it was updated the last time. Another difficult situation occurs when a person enters an exit while another person stands near to him. Now the filter cannot decide, whether the person left or is just occluded by another person. This often results in a missed count (see figure 5.18). To reduce this problem, the probability for person-world occlusions is set a little higher than person-person occlusion. The intuitive reasoning is that at an entrance it is more probable for a person to leave than to be occluded by another person. This is realized by setting the occlusion maximum 0.1 higher for person-world occlusions.

5.3.3 Tracking Results

Table 5.3 sums up the results and compares it with the results of [25]. We do not regard persons which are not fully visible because they leave the screen. Some persons are never

Figure 5.18: Left column: Person does not leave but is counted; Right column: Person at the store leaves but is not counted.

seen fully. Because of this, the number of present persons is a little lower. As one can see, the method of Wu et al. does a better job in successfully tracking persons. The probable reason for this is their use of a part-based detector, which can detect humans even if they are mostly occluded. Note that the number of fragments in [25] is not consistent with their definition: the ground truth is 189, 140 tracks are mostly tracked. By definition, at least 49 fragments have to exist. The number of mostly lost tracks is comparable. The system developed in this thesis produces more false alarms, probably because of the less principled way of initializing and deleting tracks. In the system developed in this thesis, less identity switches are encountered. The reason may be, that an error in linking does not necessary cause an identity switch in our system.

We cannot directly compare the occlusions, because the method of defining occlusions used here might differ from theirs. However, they seem to be similar. Our system performs much worse, if one asks for the intersection of tracker prediction and ground truth to be 50% of the union. One reason is, that the GOETHE tracking can predict a person quite far from its real position: the prediction denotes the expectation value. The particles form hypotheses at all places a person can hide. Thus the variance is high when the occlusion volume is large. One can support this thoughts by the higher success rates, if an overlap of only 10% is demanded. The other failures are often caused linking errors. If persons stand very near during an occlusion, it is possible, that they are matched falsely. Due to clutter, persons histograms are sometimes mixed in case of occlusions, resulting in failure of the linking process. An example is shown in the following.

Now a look at some interesting situations is taken. Figure 5.19 shows an id switch. The reason for the switch is, that the two person stand very near to each other and the HOG can only detect one. Now the color histograms get mixed up, since the HOG measurement covers both persons. In the case that both persons are detected, this would not happen, since the histograms are not updated if two measurements overlap too much. Due to the mixed up histograms, the linking fails and the identities switch.

Figure 5.20 shows a recovery from a false measurement assignment. When the woman is measured for the first time, the tracker of the man is falsely assigned to her. This is because their color is similar and no measurement for the man is available. However, not many particles are initialized by Icondensation, because the correspondence is not very good (the woman not close to the prediction for the man and has different colored pants). After all three measurements are available again, the tracker can recover.

Figure 5.20 shows the occlusion handling system in action. The woman with the brown particles is fully occluded for a long time by the person with the white particles. While she is occluded, the particles assume her to be either behind one of the persons or inside the exit. After she is visible again, one can see how occlusion handling and Icondensation work together: while the occlusion handling keeps the particles in the right places, the Icondensation quickly reassigns them to the object, after it reappears. Note also, that the

	GT	MT	ML	Fgmt	FAT	IDS	SO	LO
Zhao et al. [27]	189	121	8	73	27	20	40/81	6/15
Wu et al. [25]	189	140	8	40	4	19	47/81	10/15
This Method (50%)	173	114	12	144	14	12	26/87	9/12
This Method (10%)	173	117	6	155	14	12	41/87	7/12

Table 5.3: GT: Ground truth (nr. of tracked persons); MT: mostly tracked; ML: mostly lost; Fgmt: trajectory fragment; FAT: false alarm trajectory; IDS: ID switch. SO: Short occlusion; LO: Long occlusions

particles of the occluded person do not spread to the entire image, but only at places the person could really stand due to occlusions. Also note, that this example counts as a failed occlusion, since the prediction is too far away from the woman during the occlusion.

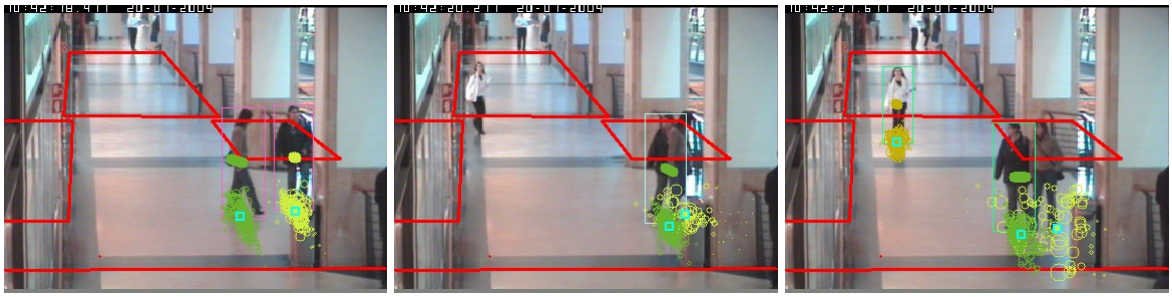

Figure 5.19: Example of an id switch.

5.3.3.1 Computational Time

In table 5.4 an overview of the total computational load is given. The values were computed using a complete iteration of the CAVIAR videos. Object detection is by far the most expensive part. Thus this will have to be improved first for achieving real-time performance.

5.4 Summary

In this extensive evaluation, an in-depth look at the most important components' performance was taken. Additionally, the overall system was evaluated, regarding both counting and tracking success.

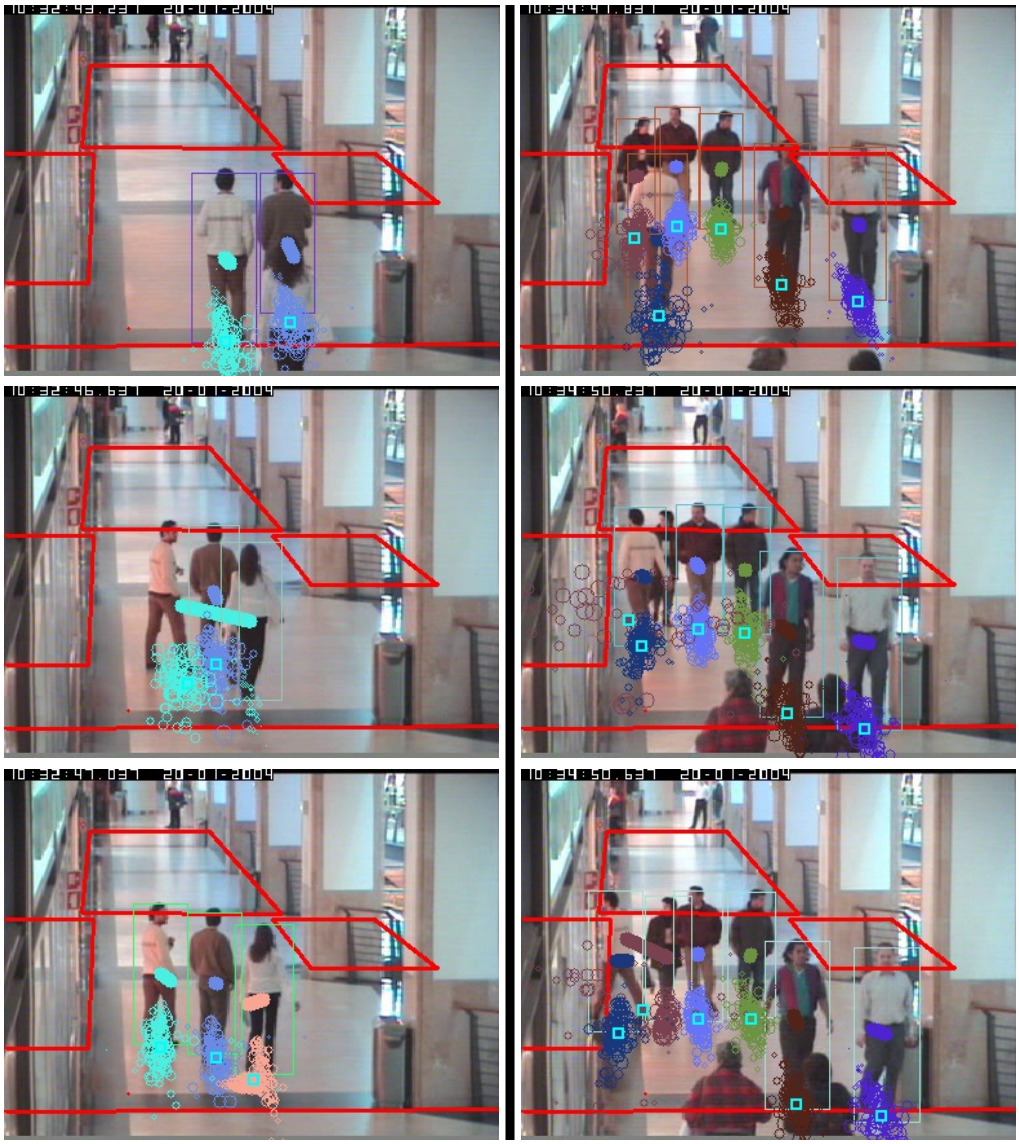

Figure 5.20: Left column: Example of false linking recovery; Right column: Example for the occlusion handling in action.

nr. persons	tracking	HOG
1	100Hz	$1/6.1$Hz
2	33Hz	$1/6.1$Hz
3	16Hz	$1/6.1$Hz
4	11Hz	$1/6.1$Hz
5	10Hz	$1/6.1$Hz
6	6Hz	$1/6.1$Hz

Table 5.4: runtime of the overall system (avg.).

It was shown that the HOG is indeed suited for real world tracking applications. It shows low false-alarm rates and missed detections are relatively rare. The inclusion of scale-information showed great success. To achieve real-time performance, some additional improvements to the HOG as proposed in [35] have to be made.

The occlusion handling system also shows substantial improvement to tracking without occlusion handling, as shown in section 5.2.2. It was shown, that the occlusion handling scales almost linearly, an important advantage to existing systems.

The results of the overall counting system are twofold. The counting results are very good. In a normal situation, when people move towards the camera, the results are as good as state of the art light-barrier systems. As an additional advantage, the tracker can observe more than one entrance at once. However, two factors worsen the performance. First, if persons are far away and supported by little evidence in the image, the results decline. Second, entrances modeled by person-world occlusion show some disadvantages. However, even in these cases the counting accuracy stays in a region of approximately 80%.

The tracking results are lower than that of the state of the art. A part-based person detector can probably improve the results. Maybe other features than only color histograms of rectangles should be taken into account to describe the person. This is not a problem for the tracking system, since it is formulated generally as a Bayes filter approach. Note that the factors which worsen the tracking performance do not need to have a negative effect on the counting. If an object is lost far away from an entrance and reinitialized, it has no negative effect on the counting. The tracking result on the other hand declines.

6 Conclusion

In this diploma thesis a system for counting humans was developed. Unlike usual light-barrier systems, it can count persons from multiple entrances and has no problem with persons walking side by side. It can discriminate between humans and non-humans. This is an advantage to other vision based systems counting systems, which usually depend on basic features like foreground detection or skin color blobs to detect humans. When people walk towards the camera, the counting accuracy is about 95%, which can compete with commercial light-barrier systems. Entrances farther away from the camera achieve lower performance, but counting accuracy still stays over 80%.

As all tracking based systems, the system cannot handle large crowds of people, but in setups like smaller stores or banks, it is applicable. Up to 9 persons at once are present in the public dataset used for the evaluation. The system has no problems tracking these amounts of people. However, real-time performance is not achieved, because the HOG used for detecting humans performs too slow. Improvements to the HOG, which allow the system to run in real time are possible, but were not implemented due to time constraints. Without the person detection, the system can track 6-7 persons at once in real time. The system does not require a complicated setup for each new camera or perspective. Some systems require for example training data for each new camera setup. The only thing needed in the developed approach is a ground plane homography and a relatively undistorted, horizontally aligned camera with a tilt angle smaller than 45^o. The tracking system is general and not restricted to person counting. The overall systems tracking performance is a little lower than the state of the art. A possible reason for this is the use of a full body detector, instead of a part-based approach.

Two novelties are proposed in this thesis. First of all, a HOG improvement, which uses information about the scale changes in images to sort out unlikely hypotheses and speeds up the computation. The biggest novelty is the GOETHE tracking system, which presents a principled approach to handle occlusions. The great advantage to other systems is its performance in pseudo-linear time, which makes the system practically relevant. Both scale-aware HOG and GOETHE tracking are evaluated in the experiments section. The scale-aware HOG shows much lower false detection rates than the original HOG. The missed detections are a little lower, but not substantially. The inclusion of scale information speeds up the detection substantially, up to the factor 2. The GOETHE tracking has been successfully evaluated, showing much better results than the normal, separate approach in case of long time full occlusions.

6.1 Future Work

In future work the HOG should be augmented with the methods proposed in [35] to achieve real-time performance. Another useful extension would be the use of a part-based detector. This relaxes the requirement for the persons to be fully visible at least once. It also makes the tracker more stable, because it will also detect partially occluded humans. Furthermore, a part-based approach would harmonize with the already split histograms used for the tracking. The inclusion of auto-calibration methods is thinkable to reduce the need for an undistorted, horizontally aligned camera. As in the work of Lanz, a generalized cylinder model could be used to describe humans as opposed to the simpler, rectangular based approach used here. The HOG can be trained for other camera perspectives, to be applicable in a larger variety of scenarios. The developed system only works, if the camera tilt angle is smaller than about 45%.

One problem of the approach is that the background $p(z_0|\mathbf{x})$ is not regarded in the measurement model (see section 4.2.2.1). This eventually leads us to the introduction of the occlusion maximum. This could be prevented, by guessing the *expected background* much like the expected occlusion. For this purpose, the occlusion map after all particles are processed can be viewed as an "occupancy map" denoting the probability of the background being occupied by a foreground object. Let p be a particle of person i. The expected background for every particle can be calculated using the space p occupies and the occupancy map of all particles not belonging to person i.

Interesting new applications would arise in the combination of the counting system and a person re-identification system. If many cameras are present in different areas, e.g. in a shopping center, the counting system can provide the person re-identification system with information about where which person went. This can improve its results by restricting the possibly present humans. On the other hand, the counter could create whole movement profiles of persons, if it knew which person in one camera view corresponds to which person filmed by another camera, covering a different area. Another combination possibility is to count how many people come in or out of a building as e.g. a museum and then use the re-identification system to match the incoming with the outgoing persons. In that way, it can be determined how long persons usually stay. It could also be determined if and which person did not leave the building after closing time, maybe to steal an expensive drawing.

Until now, all systems for tracking and re-identifying persons have some flaws. Thus, Orwell's imaginations will take a little while longer to come true.

Bibliography

[1] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[2] EC Funded CAVIAR project/IST 2001 37540.

[3] The Access Company. People counter information - extracts from the access company's report to the countryside agency on evaluation of people counters for national trails, August 2006.

[4] Na-Na Li, Jie Song, Rui-Ying Zhou, and Jun-Hua Gu. A people-counting system based on bp neural network. In *Proc. 4th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 283–287, Washington, DC, USA, 2007. IEEE Computer Society.

[5] H. Septian, Ji Tao, and Yap-Peng Tan. People counting by video segmentation and tracking. In *Proc. 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–4, 5–8 December 2006.

[6] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 23–28 June 2008.

[7] Li Chen, Ji Tao, Yap-Peng Tan, and Kap-Luk Chan. People counting using iterative mean-shift fitting with symmetry measure. In *Proc. 6th International Conference on Information, Communications & Signal Processing*, pages 1–4, 10–13 December 2007.

[8] Tao Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, page 459ff, 18–20 June 2003.

[9] X. Liu, P. H. Tu, J. Rittscher, A. Perera, and N. Krahnstoever. Detecting and counting people in surveillance applications. In *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 306–311, 15–16 September 2005.

[10] V. Rabaud and S. Belongie. Counting crowded moving objects. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 705–711, 17–22 June 2006.

[11] Sebastien Harasse Laurent, Laurent Bonnaud, and Michel Desvignes. People counting in transport vehicles. World Academy of Science, Engineering and Technology(4), 2005.

[12] Xi Zhao, Emmanuel Dellandrea, and Liming Chen. A People Counting System based on Face Detection and Tracking in a Video. In *6th IEEE International Conference on Advanced Video and Signal Based Surveillance*, September 2009.

[13] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, TR93-1399, Ithaca, NY, USA, 1993.

[14] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. *Sensor Fusion VI*, 2059:432–443, August 1993.

[15] Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):909–926, 2008. Member-Chan, Antoni B. and Member-Vasconcelos, Nuno.

[16] Carl E. Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning (adaptive computation and machine learning). December 2005.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[18] D. B. Reid. An algorithm for tracking multiple targets. In *Proc. IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, volume 17, pages 1202–1211, January 1978.

[19] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE International Conference on Robotics and Automation*, pages 1665–1670, 2001.

[20] Jaco Vermaak, Simon J. Godsill, and Patrick Perez. Monte carlo filtering for multitarget tracking and data association. *IEEE Transactions on Aerospace and Electronic Systems*, 41:309–332, 2004.

[21] R. P. S. Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.

[22] B. N. Vo, S. Singh, and A. Doucet. Sequential monte carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1224–1245, October 2005.

[23] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. 7th IEEE International Conference on Computer Vision*, volume 1, pages 572–578, 20–27 September 1999.

[24] Ting Yu and Ying Wu. Collaborative tracking of multiple targets. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 834–841, 27 June–2 July 2004.

[25] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, November 2007.

[26] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. The variable bandwidth mean shift and data-driven scale selection. In *Proc. 8th Intl. Conf. on Computer Vision*, pages 438–445, 2001.

[27] Tao Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. IEEE Computer Vision and Pattern Recognition*, volume 2, pages 406–413, 2004.

[28] Chang Huang, Haizhou Ai, Bo Wu, and Shihong Lao. Boosting nested cascade detector for multi-view face detection. In *Proc. 17th International Conference on Pattern Recognition*, volume 2, pages 415–418, Washington, DC, USA, 2004. IEEE Computer Society.

[29] O. Lanz. Approximate bayesian multibody tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449, September 2006.

[30] O. Lanz and R. Manduchi. Hybrid joint-separable multibody tracking. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 413–420, 20–25 June 2005.

[31] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, April 1970.

[32] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. Technical Report TR-2001-22, Mitsubishi Electric Research Laboratories, San Francisco, CA, USA, January 2002.

[33] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

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

[35] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1491–1498, 2006.

[36] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[37] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.

[38] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[39] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, February 2002.

[40] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[41] T. Jaakkola. Tutorial on variational approximation methods. In M. Opper and D. Saad, editors, *Advanced mean field methods: theory and practice.* MIT Press, 2000.

[42] Ying Wu, Gang Hua, and Ting Yu. Tracking articulated body by dynamic markov network. In *Proc. 9th IEEE International Conference on Computer Vision*, pages 1094–1101, 13–16 October 2003.

[43] J. Sullivan, A. Blake, M. Isard, and J. MacCormick. Bayesian object localisation in images. *International Journal of Computer Vision*, 44(2):111–135, 2001.

[44] Josephine Sullivan and Jens Rittscher. Guiding random particles by deterministic search. *IEEE International Conference on Computer Vision*, 1:323, 2001.

[45] Michael Isard and Andrew Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Lecture Notes In Computer Science*, volume 1406, pages 893–908. 1998.

[46] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[47] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[48] Opencv v1.0.0 (http://sourceforge.net/projects/opencvlibrary/) last checked 14.08.09.

[49] Boost library (http://www.boost.org/) last checked 14.08.09.

[50] Navneet Dalal. Olt-toolkit (http://pascal.inrialpes.fr/soft/olt/) last checked 14.08.2009.