Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences
Department of Computer Science

Fraunhofer
IAIS

Master Thesis

Person Tracking and Following by a Mobile Robotic Platform

Christoph Brauers

**A thesis submitted to the
University of Applied Sciences Bonn-Rhein-Sieg
for the degree of
Master of Science in Autonomous Systems**

1. Supervisor: Prof. Dr. Gerhard Kraetzschmar
2. Supervisor: Prof. Dr. Hartmut Surmann

Submitted: 14. August 2010

I, the undersigned below, declare that this work has not previously been submitted to this or any other university, and that unless otherwise stated, it is entirely my own work.

DATE

Christoph Brauers

ABSTRACT

Robots integrated into a social environment with humans need the ability to locate persons in their surrounding area. This is also the case for the WelcomeBot which is developed at the Fraunhofer Institute IAIS. In the future, the robot should follow persons in the buildings and guide them to certain areas. Therefore, it needs the capability to detect and track a person in the environment.

In this master thesis, an approach for fast and reliable tracking of a person via a mobile robotic platform is presented. Based on the investigation of different methods and sensors, a laser scanner and a camera are selected as the primary two sensors. For the laser scanner, a classification and tracking method is implemented utilizing the laser scanner distance values as well as the remission values. For the camera, a very successful person detection algorithm is used and combined with some fast tracking algorithms known in the field of computer vision. Finally, a combined approach for both the camera and the laser scanner is presented. All methods are evaluated in different environments and have also been tested in a scenario where the robot is following a person automatically.

CONTENTS

ABSTRACT iii

LIST OF FIGURES vi

1. INTRODUCTION 1
   1.1 Motivation 1
   1.2 Context of work 1
   1.3 Problem statement 2
   1.4 Structure of work 2

2. STATE OF THE ART 3
   2.1 Person detection 3
      2.1.1 Person detection in mobile robotics 4
      2.1.2 Vision based approaches 5
      2.1.3 Laser scanner based approaches 9
      2.1.4 Depth imagery approaches 13
      2.1.5 Combined approaches 14
      2.1.6 Other approaches 15

3. CONCEPT AND METHODOLOGY 17
   3.1 Concept 17
      3.1.1 2D laser scanner 17
      3.1.2 Dioptric camera 19
      3.1.3 Fusion of Data 20
   3.2 Methodology 21
      3.2.1 Laser scanner based person detection 21
      3.2.2 Camera based person detection 23
      3.2.3 Approach: Combined approach 25

4. IMPLEMENTATION 26
   4.1 Laserscanner based people tracking 26
      4.1.1 Preprocessing 27
      4.1.2 Segmentation 27
      4.1.3 Classification 28

4.1.4 Tracking of legs . . . 31
4.2 Camera based person detection . . . 32
4.2.1 Camera calibration . . . 32
4.2.2 Detection of persons . . . 34
4.2.3 Tracking of persons . . . 38
4.2.3.1 Mean shift Color Tracking . . . 38
4.2.3.2 Template matching . . . 40
4.2.4 Combination of tracking and person detection . . . 41
4.3 Combination of the camera and the laser scanner . . . 42
5. EVALUATION . . . 45
5.1 Laser scanner based detection . . . 45
5.1.1 Description of the laser scanner interface . . . 45
5.1.2 Classification and tracking in a static environment . . . 46
5.2 Camera based detection . . . 48
5.2.1 Test of the HOG algorithm . . . 48
5.2.2 Test of the mean shift color tracking algorithm . . . 51
5.2.3 Test of the template matching tracking algorithm . . . 53
5.3 Detection in a dynamic environment, while following a person . . . 55
5.3.1 Dynamic environment while following a person with a laser scanner 56
5.3.2 Combination of the HOG detection and the template matching algorithm to follow a person . . . 58
5.3.3 Combination of laser scanner based and camera based methods . . . 59
6. CONCLUSIONS . . . 61
6.1 Summary . . . 61
6.2 Future work . . . 62
BIBLIOGRAPHY . . . 63

LIST OF FIGURES

2.1 Different robots used in the human machine context. The left robot is the exhibition robot developed by the Fraunhofer Institute for Opel [Fraf]. In the center the Scitos A5 from Metralabs is shown [Met]. On the right side the CaroBot build by the Fraunhofer Institute is depicted. [Frae] 4

2.2 Part based training example from [ARS08]. 8

2.3 Multilayer person detection by Mozos et. al. [MKH10]. 10

2.4 Laserscanners used by Carballo et. al. [COS09]. 12

2.5 Diagram of the segmentation used by Spinello et. al. [SATS10]. 14

2.6 Combination of camera based HOG algorithm and laser scanner data to reliably classify persons in the environment [SAM+09]. 15

3.1 Raw data of a laser scanner with two legs in the scan. 18

3.2 A camera with fish-eye optics (left), a picture taken with a normal camera (center) and an exemplary image taken at the same distance with fish-eye optics (right). 19

3.3 The concept for combining two restricted sensors, which complement each other. 21

3.4 Raw laserscanner data with two legs in the scan. 22

3.5 Comparison of classification by the Viola et. al. [VJ01] (blue) and the Dalal et. al. (green) [DT05]. 24

4.1 Laser scanner process chain. 26

4.2 Segmentation process. 28

4.3 The picture shows the approach of Xavier et. al. [XPC+05] to classified circular objects. s_0 - s_3 denote different laser scanner measurements, the angle α is in each case the same. β is equal to $180 - \alpha$ and the radius can be calculated by $r = \frac{f}{sin(\beta)}$. 29

4.4 Remission values for different materials: Except for the metal, all materials exhibit similar remission values. 30

4.5 Image taken with a fish-eye lens (left) and the compensation of the distortions (right). 33

4.6 The picture shows the different steps in for the classification with the HOG algorithms [DT06]. 35

4.7 Different tests with camera images classifying a person. 37

4.8 Mean shift algorithm: Tracked object (left), backprojection of the histogram (right) and histogram of tracked object (lower right). 39

4.9 Template matching algorithm moves a template (see small picture at top) over the image (see right). The result of each comparison is shown in left image. . 40

4.10 Timing of the different threads implementing HOG, template matching and the Camera. 42

4.11 Calibration tool for the extrinsic camera parameters. 43

5.1 Interface for testing the laser scanner based person detection. Left image shows different graphs of filtered laser scanner distance values with respect to the according angle. Right image depicts the detected legs (red points) in the Cartesian coordinate system and the tracking direction (green circle). 45

5.2 The left image shows different objects that were used in the classification. In the right image the red dots mark the different leg classifications. 46

5.3 Test scenario with leg like objects in the laser scanner range and different occluding objects. 47

5.4 Different tracks recorded of persons walking in the environments presented in Fig. 5.3. 48

5.5 Examples of the classification with the HOG descriptor. The left four images show positive classifications, the rightmost image shows a false classification and a case where a person has not been detected. 49

5.6 Different extreme positions that could now be detected, due to the black border. 50

5.7 In the image, a person can still be detected in a distance of 500 millimeters away from the robot and angles of 67,5 degrees to each side. 51

5.8 Mean shift color tracking algorithms successfully tracking a dummy person (left images) and failed to track a person (right images). 52

5.9 The left image shows the initialization scene for the color histogram. In the center and image is depicted demonstrating that the dummy person was lost by the tracking. Here, the initial tracking position is the white rectangle and the currently tracked position is marked by a red rectangle. The rightmost image shows how frequent the colors of the histogram occur in the environment. 53

5.10 Template matching used for tracking a person in a picture. The red rectangle marks initial region (left picture), the white rectangle is the region where the template matching algorithm has detected the template. 54

5.11 Template matching for detecting persons in different distances and angular positions in a more homogeneous environment. 55

5.12 Different testing environments for the laser scanner based approach. 56

5.13 Different tests with moving objects in between the laser scanner and the tracked person. 57

5.14 Testing interface of the HOG and the template matching approach. The upper images show the template matching, the lower images show the HOG classification. 59

5.15 Left: image of the laser scanner leg detection. Right: region of interest extracted from the camera picture with the HOG classification (green rectangle). 60

Chapter 1

INTRODUCTION

1.1 Motivation

Autonomous mobile robots are moving out of the laboratories and into public domain. There are already some pilot projects in which the interaction between robots and humans in environments like hospitals, museums, office buildings and supermarkets is tested. They either guide the way, follow a person while carrying heavy items or are just used as a mobile information system. In all of these scenarios, however, the robot has to interact with the person and know where he or she is located. This is one of the reasons, why person detection is gaining more and more importance in the field of mobile robotics. Some developers even say it already is a key technology for social interactive robots [AM10]. This could also be the reason why this topic filled a whole workshop of one of the largest robotic conferences (ICRA) [ICR09] in the last year.

Social interaction is also very important for the robot that is used in this master thesis. In the future, it is planned that the robot guides and follows persons through the institute. Therefore, however, it has to be aware of a human (for guiding a person) or in the case of following a person it even has to know exactly where he or she is located in the surrounding. Since the knowledge of the precise location implies that the robot is aware of a person, this master thesis will concentrate on the localization of a person in the environment.

1.2 Context of work

The Fraunhofer Institute-Center in Birlinghoven has three institutes and two research units which are working on solutions for the industry and the society. One of these institutes is called Intelligent Analysis and Information Systems (IAIS) [Frab] and is involved in the design and development of robots. Projects like ProfiBot [Frac] and VolksBot [Sur] produced different robotic platforms (e.g. HighTecBot, PeopleMover) that have been utilized in several internal and external projects.

The WelcomeBot robotic platform is a successor of the platform in the ProfiBot project. The aim of the project is to develop and design a service robot that guides and follows visitors from the entrance hall to certain places in the institute. In this context, already some diploma theses and research projects were done. This master thesis will be based on these and extend the robot with the ability to precisely and reliably locate a person in the surroundings of the robot for following him/her to a specific place. This

master thesis should be a basis for further implementations used for guiding a persons and interacting with visitors in a socially convenient way.

1.3 Problem statement

The main challenge in following a person with a mobile robot lies in the ability to detect and track a person precisely, reliably and in a fast manner in the environment. The detection and tracking performed by a mobile robot, however, has it's own challenges. Thus, for following a person with the robot, it must be able to detect and track a person by his/her back since a distinct feature like, for example, the face is in most cases not visible. Another challenge originates from the mobility of the platform. Since the robot is moving around in the environment, the parameters like lighting and background of the environment are constantly changing in the perspective of the robot. Therefore, the detection and tracking methods should cope with this problem. Another challenging problem for the mobile robot is the fact that the persons do not wear any special device or clothes to be detected from the robot. The only identification is done directly before the guiding sequence. Thus, the robot cannot rely on special clothes or keys to locate the person, but has to track it while moving around. Other challenges originate from particular parts of the implementations. Thus, for example, the tracker has to be able to compensate at least for short temporal occlusions of the person and the classifier has to be able to distinguish between persons and objects. There are a lot more problems which originate from the usage of particular sensors.

1.4 Structure of work

This master thesis is structured into 6 parts. In the following chapter, an overview is given of the different sensors and methods available to detect a person in the environment. Chapter 3 then presents the concepts and methods that are used in this master thesis for recognizing a person in the environment. Chapter 4 continues with a more precise description of the different algorithms, discusses eventual problems and shows different extensions. The evaluation of these methods is performed then done in chapter 5. First, the different properties of the sensors and methods in a static environment are describes and then some tests in a dynamic environment while following a person are presented. The last chapter then summarizes the master thesis and gives an outlook on future work.

Chapter 2

STATE OF THE ART

2.1 Person detection

Person detection gained a lot of attention in the last few years. It has various applications in areas like surveillance, robotics, image and video indexing and automotive safety. The growing need for automation is one of the motors for creating new advanced algorithms. Thus, there are already a lot of methods that are partially used in museums, theaters, sports arenas, parks, airports and traffic to monitor the behavior of humans and automate functionalities of the environment. One of these applications is for example in sports. Here, visual detection and tracking methods help to generate statistics about soccer games by monitoring the movement of players [Spo]. Algorithms gather informations like the estimated speed of a person, his or her contacts with the ball or how often he or she has shot onto the goal. The technique works in most cases with the help of static cameras that are pointed to the soccer field. The system merges the images to one big picture and displays it on the screen of the operator. The latter has to calibrate the whole system prior to the game on the exact scales of the field and assigns each player a specific profile. In the game, a tracking algorithm detects the colors of the players and their individual movement on the field. Based on this information, the statistical data can then be calculated.

The algorithms work quite well in a static environment like it is present in a soccer stadium. But there are also other applications, where similar algorithms are used. For example they are successfully implemented for counting pedestrians in traffic, at airports, museums and many other places.

Although these methods are very popular and effective for tracking people in a static environment like on a soccer field, they are not necessarily applicable in the context of robotics.

In industrial robotics for example it is sometimes not even necessary to exactly detect where a person is located in the environment. Most of the times it is more important to detect the presence of a person than the actual position. Therefore, the method that has been previously described could be applicable, but is in most cases much too complex or too expensive. In industrial robotics more often very simple sensors like Passive Infrared Detectors (PIR), laser barriers or fixed laser scanners are used to detect whether a certain area is free or not.

However, in mobile robotics, a lot of applications require a very precise and fast localization of a person. Especially when the robot should follow a human, it is essential

Figure 2.1: Different robots used in the human machine context. The left robot is the exhibition robot developed by the Fraunhofer Institute for Opel [Fraf]. In the center the Scitos A5 from Metralabs is shown [Met]. On the right side the CaroBot build by the Fraunhofer Institute is depicted. [Frae]

for the robot to know exactly where the person is located.

2.1.1 Person detection in mobile robotics

The detection of a person is an important task in mobile robotics. Especially when the interaction between the robot and the person should be in a socially convenient way, it is almost inevitable to have a certain kind of localization [FND03]. Areas where such robots are deployed are in health care, domestic areas, museums, surveillance applications and many more. For example the Carobot [Frae][SM99] (see right robot in Fig. 2.1) developed by the Fraunhofer Institut IPA [Frad] should help people in domestic household, bring them items and carry heavy things for them. In areas like museums and exhibition halls robots are used to guide persons around [Fraf] [TBB+99] [BCF+98] and present exhibits to the guests. In other scenarios, security robots [TCD05] autonomously patrol in certain areas or robots guide persons to specific products in a supermarket [Met] (see center robot in Fig. 2.1). There are a lot more applications where robots are used in a human machine context.

Person detection and localization therefore is an important task for a mobile robotic platform that is interacting with persons. However it also comes up with some additional difficulties in contrast to other scenarios.

The name of “mobile robotics“ already implies that the robot is not fixed to the ground but most of the times moving autonomously in the environment. As a consequence,

it cannot rely on the environment to have static properties. Thus, for example, the lighting conditions, the background of the scenery and a lot of other parameters are constantly changing while the robot moves around. Therefore, the actual detection method must somehow cope with these changes. There are two options: Either, the changes are ignored by, for example, filtering them from being processed at all or the changes are handled by incorporating them into the actual detection method. Both methods are used in mobile robotics to cope with constantly changing environments. While one algorithm might perform quite well in a static environment like a soccer stadium, it might fail in the context of a mobile robot.

In this master thesis the robot detects, tracks and follows a person from a mobile robotic platform. Therefore, it is important that the robot can cope with dynamic environments and knows at any time where a person is located and where he or she is moving. The following will concentrate on methods that are applicable for a mobile robot and could be used in this context. It is structured as follows:

- Vision based approaches: Detection of persons based on the analysis of a camera images.
- Laser scanner based approaches: Utilizing two dimensional laser scanners for recognizing shapes of persons.
- Depth imagery approaches: Approaches for detecting persons in three dimensional data.
- Combined approaches: Combination of different sensors to overcome the weakness of one or another.
- Other approaches: Approaches that do not rely on the sensors described before.

2.1.2 Vision based approaches

The probably best known and most famous approaches for detecting persons in the environment are based on vision systems. Most of them can either work online with real time images or offline with recorded video sequences from surveillance cameras. For the application in mobile robotics though it is important that the algorithm can be applied online in a fast manner.

General difficulties in person detection with cameras originate from different articulation, viewpoint and appearance of a person. However, in mobile robotics the detection of a person has some more challenges. Since mobile robots are moving around in the environment, the background sceneries and lighting conditions are continuously changing. Thus, the algorithms for this scenario either have to filter these changes or model them into their actual detection.

The detection of a person itself is often based on the recognition of single parts of the human body. The most discriminating part is of course the face of a person. This has also been discussed in a lot of papers [HL01] [ZCPR03] and there are already very popular and successful methods for detecting persons based on their faces.

However for the task of following a person, like it is described in this master thesis, the face is an unsuitable feature. Since the robot is following the person, the person is most of the times turned with the back to the robot. The face would only very occasionally be visible. Thus the next part of this section will present algorithms that are able to detect the back of a person and can cope with dynamic environments.

Vision based identification approaches can be mainly separated into three categories:

- Background subtraction methods
- Sliding window techniques
- Methods using part based body models

Background subtraction methods

Background subtraction methods are very popular for detecting persons in video surveillance applications. Most approaches first acquire a reference image when no person is present in the scenery and use this for subtracting the background in one of the subsequent image frames. However, this particular method is only applicable for a static environment and not necessarily in a mobile robotic one. Since the environment is changing over time, other approaches for background subtraction are used.

To overcome this limitation, some methods are modeling the current ego-motion of the robot. Therefore, they first calculate the positions using the odometry or measures of other sensors. Then they transform the image according to the previously gathered informations. Finally they apply a frame differencing algorithm to extract the position of moving objects or persons in the environment.

One of these methods is used in Feyrer et. al. [FZ99]. The algorithm for estimating and subtracting the ego-motion is described in [MB94]. They first compute the current movement of the robot that has occurred between two camera shots with respect to the orientation and position. This information is then used to transform the successive image frame with a homogeneous transformation matrix to model the actual movement in the picture. The resulting image is then compared with the original image and all non-matching areas are assumed to represent moving objects. These informations are then further analyzed with face detection and skin color detection algorithms to check the presence of a person.

Another paper that discusses a similar approach is written by Jung et. al [JS10]. They do not detect persons explicitly but describe a method that recognizes moving objects

in an outdoor environment of a robot. They also calculate the actual ego motion of the robot and perform similar processes to integrate them into the actual detection.

Sliding window techniques

Sliding window techniques scan the image for certain features that differentiate a person from objects. At each position and scale the classifier distinguishes whether the window contains a person specific feature and label the part either as human like or not. Since these techniques scan the whole image they are often computationally expensive.

Although the robot in most cases perceives the back side of a person, in general face detection algorithms are of interest because in some cases they can also be used for classification of other body parts. One of the more recent and popular algorithms in this area was published by Viola et. al. [VJ01] in 2001. It uses features similar to Haar wavelets and the AdaBoost [FSA99] technique, which combines several weak classifiers to a strong one, to identify faces in a picture. The learning method AdaBoost is first fed with a lot of images of faces (positive examples) and images showing various other objects (negative examples). Simple rectangular features are then detected in an integral image and combined by AdaBoost to a cascades of classifiers. Afterwards, these classifiers are then used to detect a face in an image.

This algorithm though is not only used for detecting faces. Since the detection is based on the positive and negative images that have been trained to the algorithm in the learning phase also other objects can be detected with this algorithm. For example, such an algorithm was also used to classify clocks, doors or doorsigns in robotic environments. More interesting though, especially for this master thesis, is a classifier that has been published by Kruppa et. al. [KS] [KCSS03]. They trained the classifier for detecting the upper, the lower and the full body of a person.

A similar classifier was also used by Viola et. al. in 2003 [VJS05] to classify persons in video data. They trained a classifier with a set of full body person pictures. However, to make the recognition of a person more reliable, they added an extra classifier that was trained with motion patterns of humans. As a result they could decrease actual false positive detection of persons drastically in their experiments.

Another very recent and popular algorithm that is used to detect persons in pictures has been introduced by Dalal et. al. [DT05] in 2005. It uses so called histograms of oriented gradients to classify persons in an image. This detection method is based on the principle that "local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions."[DT05] Therefore, first they trained a classifier on a large dataset of blocks of histograms of oriented gradients (HOGs). The latter were computed based on edge directions of positive images, showing persons in the

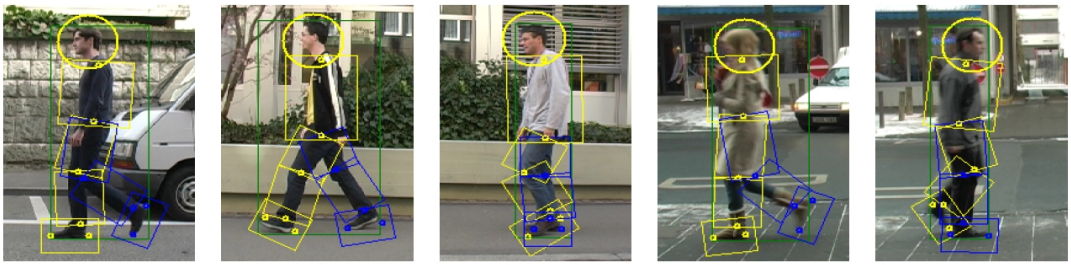

Figure 2.2: Part based training example from [ARS08].

environment, and negative images which depicted everything else but no humans. The resulting classifier was then used in the detection phase to scan the images at all scales and locations for corresponding HOGs. When these resemble each other up to a certain percentage, the analyzed region is classified positive. The algorithm was trained and tested with the INRIA Person dataset [pd], which consists of over 1800 images with humans that have been shot from several viewpoints under varying lighting conditions in indoor and outdoor scenes.

In [SAM+09] Schiele et. al. compared different sliding window techniques with each other and found out that the Histogram of Oriented Gradients [DT05] outperforms most other algorithms.

Part based models

The third group of detection techniques is summarized under the term "part based models". These methods are composed of two main components. The first one is a low level feature descriptor. It classifies regions that correspond to single body parts like a limb, arm or a leg in a picture. The second component describes how these different parts are connected to each other such that they model a full human body. Based on these models an evidence aggregation is then performed on the image classifying regions that correspond to the human body. Since this technique is based on the detection of single parts of the body it is more robust to partial occlusions of a person. A comparison of sliding window techniques and this technique is also made in [SAM+09]. The result was that the part-based people model can outperform sliding-window based methods in the presence of partial occlusion, but also required higher resolution images for recognizing persons correctly.

An algorithm that is utilizing this technique is described in [WN05]. The authors utilize an enhanced version of the boosting method used in [VJS05] for training different low level feature descriptors. Therefore they use so called edgelets which are described as silhouette oriented features to build robust low level features. They train the algorithm based on different parts of the body: the head and shoulder region, the torso, the legs and the whole body. After that, they fed these classifiers to a nested structured of detectors

to classify persons in the surrounding.

Another method that is also based on the part based model technique is described in [ARS08]. In this paper they model the whole body and its movement with a Gaussian process latent variable model. Therefor they first describe single limbs of the human body and train their low level feature detector by annotating each limb of a person in every training image (see Fig. 2.2). Though they do not differentiate between the left and right leg of a person, they can compute the person's speed and make an assumption about the actual articulation of the person.

2.1.3 Laser scanner based approaches

Two dimensional laser scanners have a wide variety of applications. They provide highly frequent measurements from a wide view angle and are invariant to illumination changes of the environment. Although they cannot detect objects that do not reflect the laser beams like in the case of glass or mirrors, they provide very accurate measurements from every other object in the environment.

This is one of the reasons why they are used for mapping and also for detecting persons. For example, [CZZS06] implemented a person detection based on several laser scanners that are distributed in a hall at a height of 16 centimeters. Each laser scanner measurement is send to a central processing unit that summarizes them all in one map of the environment. Since the laser scanners were mounted at fixed locations, the static environment can be filtered out simply. Therefore, it can be assumed that moving objects in the environment are corresponding to humans. Persons though were detected by their legs. To this end they searched in this map for objects that resembled two pillows that are located near to each other. By the usage of several laser scanners even occlusions did not seem to be a problem since at least one of the laser scanners could provide information about the persons in the environment. With this system Cui et. al. could track multiple person in the environment and predict their future movement. In most of the cases these systems are not suitable for the context of a mobile robot, since it cannot be assumed that the environment is static.

The placement of the laser scanner has an important role in the context of person detection. Dependent on the mounting position, different parts of the human body can be perceived. Three different settings have been mainly used over the last few years.

The most often used position is located above the ankle of the foot and beneath the knee at an approximate height of 20 to 50 centimeters. Robots that are equipped with such a laser scanner are described in [KLF$^+$02] and [AMB07]. They recognize the legs of persons based on different feature criteria that will be discussed later in this chapter. Since the foot is not rising above 20 centimeters for normal walking speed, a single leg is always visible in this configuration of the laser scanner. This may though vary for

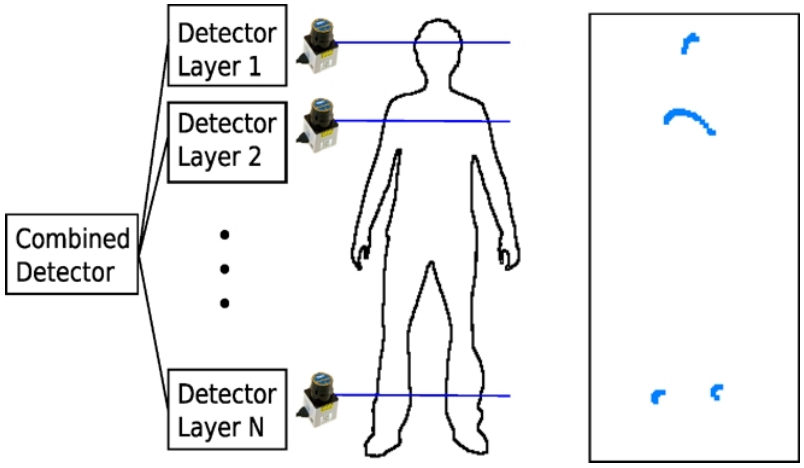

Figure 2.3: Multilayer person detection by Mozos et. al. [MKH10].

different persons with different heights. Additionally, by tracking the movement of both legs, a lot more information about the person's behavior can be analyzed, like walking speed, orientation to the robot and so on. A disadvantage though is the fact that at this height the detection of a single leg can be very easily confused with other objects like tables, chairs and other pillows in the surroundings.

The second mounting position is at a height of approximately 100 centimeters up to 150 centimeters from the ground, which could be slightly above the waist up to the chest of a person (see Fig. 2.3). Robots that use such a mounting are described in [MKH10] and [MKH09] and have at least one laser scanner that is mounted at that height. At this height, the probability for another objects having similar dimensions like a human are very low. Thus the detection of a person can be much more reliable than the one that is based on the detection of legs. Tracking of body parts should also be much simpler. Since the movement of the upper body is more homogeneous, meaning with a more constant speed in comparison to the movement of single legs, a tracking could be designed simpler. However, a disadvantage for this mounting is the assumption that every person is equally tall and has similar dimensions at this height to those for whom this system has been calibrated. Especially when it comes to very small and tall persons this could lead to some problems.

The third mounting position is located at approximately height of 1.80 meter and tries to detects the head of a person. This method is though used only for a few robots [MKH09] [COS09] and in most cases only in combination with several other laser scanners. At this height the probability for detecting another object with similar dimensions decreases even more. Therefore, based on a detection of a head shape in this region it could be very reliably estimated that a detected shape has to be a person. The disadvantage is, that not every person is equally tall and therefore the method would not fit for all

person heights.

From all of these mounting positions, the first one (leg height) though is used most frequently for detecting persons. However, in many cases the developers merge laser scanner readings from different heigths [MKH09] [MKH09] [COS09] to generate more reliable person classifiers (see also Fig. 2.3).

The different extraction methods that are used to identify a person in the laser scan reading are summarized in the following part and can be subdivided into three categories:

- Feature based methods
- Learning based methods
- Gait recognition methods

Feature based methods

Feature based methods rely on the extraction and tracking of object features such as the dimensions, shape, contour and velocity. They use direct methods like voting or cascades of filters to infer whether the actual detected feature is corresponding to a person or not. Mendes et. al. [MBN04] for example utilize a voting scheme to gather information about objects that are detected in the environment. They segment the laser scanner readings with a line fitting algorithm and identify legs of a person by two small segments that have a distance less than 50 centimeters. In [KLF+02] they group neighboring points of a laser scanner into segments if they do not exceed a certain threshold. Then they classify these segments according to a set of different thresholds whether it corresponds to a leg or not. Legs that are located near to each other are additionally grouped together. After that, they combine these measurements with predictions that are based on measurements made by a camera of the surrounding. Xavier et. al. [XPC+05] use the shape of a human leg as a feature to classify whether a certain object corresponds to a leg or not. Therefor they implemented a test system that uses the "Inscribed Angle Variance" of a triangle to calculate the actual radius, position and how the laser measurements are differing from the fitted circle in a very fast manner.

Carballo et. al. [COS09] use a scoring function to decide whether a detected segment is corresponding to a leg or not. Therefor they analyze the width, the linearity, the curvature and the elliptical shape to form a good classifier for a leg or a person. They classify the legs and the chest region (see Fig. 2.4)for these patterns and combine them afterwards to make a strong assumption whether the detected features are corresponding to a person or not.

Belotto [BH05] uses a direct method to filter a pattern of local minima that correspond to two adjacent legs in the laserscan. Therefor he defines certain thresholds for

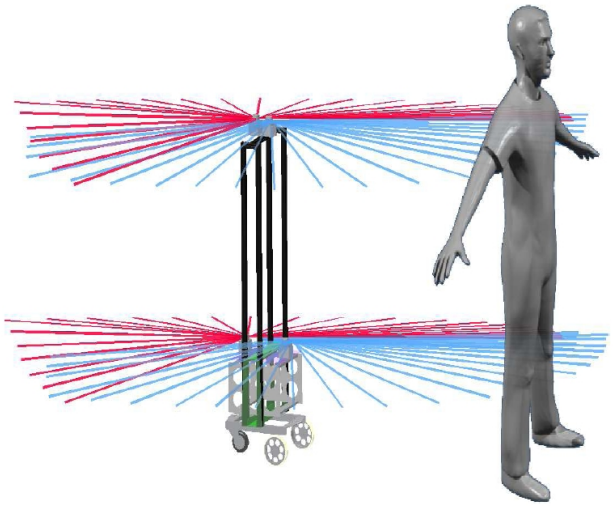

Figure 2.4: Laserscanners used by Carballo et. al. [COS09].

the distance of the legs between previous scan data and actual one to determine where a certain pattern occurs.

Learning based methods

Learning based methods are similar to the methods that have been presented before. However, these methods calculate the actual thresholds for each classifier automatically. Therefore, different positive and negative examples of legs in the surrounding are trained with a learning techniques like AdaBoost or SVM. AdaBoost though takes a lot of weak classifiers like the shape, the width, the radius, the circularity, the linearity and many other features of a leg and creates a strong classifier by weighting how these different classifiers are combined. This approach is for example applied in [AMB07]. They use 14 different classifiers to build a strong person detection method based on the laser scanner. This algorithms was also evaluated by [RC09] in an similar approach. They found out that in 90 percent the legs were classified correctly. But these methods are also applied to multi level laser scanner approaches. [MKH10] used three laser scanners on the height of the head, the upper body and the legs to build a more reliable classifier for detecting persons in the environment. Spinello et. al. [SS08] even extend the amount of classifiers by using a total of 50 weak classifiers for AdaBoost. Some of the features they used are: number of points, standard deviation, mean average deviation, width, height, linearity, circularity, radius, boundary length, mean curvature, mean angular difference, kurtosis, PCA based shape factors, n-binned histogram, PHG Boundary regularity.

Gait recognition methods

The third method is based on the recognition of the gait of a person. Instead of analyzing and segmenting the actual laser scan these methods build a occupancy map of the environment recording the actual movement that has been made relative to the robot. Therefor [PSE00] for example first calculates the ego motion of the robot based on the odometry. Though the odometry always has a slight error, they claim that this can be disregarded with respect to the minor movements that are made between two consecutive scans. They estimate which objects in the surrounding have been moving and assume that every movement that is not corresponding to the static environment is probably originating from a person. Though this method is quite efficient in the case when only persons are moving in the environment it will fail as soon as a person is standing still in front of the robot. Then the person will not be able to be detected and be interpreted as a part of the static environment. Another disadvantage is that this method is not able to handle any occlusions that could occur in the environment and also other moving objects could be confused with a person.

2.1.4 Depth imagery approaches

Depth image sensors perceive the surrounding environment in three dimensions. Thus they generate a three dimensional image of the surrounding environment including any persons. Popular sensors in this area are stereoscopic cameras, time of light cameras and 3D laser scanners. The generation of that data is yet very time consuming and computationally expensive.

However, the usage of three dimensional information of the environment makes it possible to build a much more detailed model of a person. Thus, the model can be more unique than it would be for example with only a single laser scanner or a single camera. The problem though is how to find a specified model in the large amount of data. The algorithms for a robot should be fast, precise and not require a lot of computational power. To accomplish this, most methods that are utilized in a robotic scenario are nowadays only processing minor parts of the images. The most popular methods segment the image by slicing them into different layers. The following part will concentrate on these methods.

One of these methods is described in [SM09]. They developed a detection system for a robot that should follow a specific person in a dynamic environment. First, they segmented the image into a fore- and background image using the depth information of a stereo camera. In the different layers they searched with a template matching algorithm for a set of previously learned depth templates of a head. To verify whether the detected image is really a person or not they additionally implemented a SVM-based person verifier which is based on manually classified intensity images of persons.

A similar approach was done by [DGHW00]. They also segmented the image into

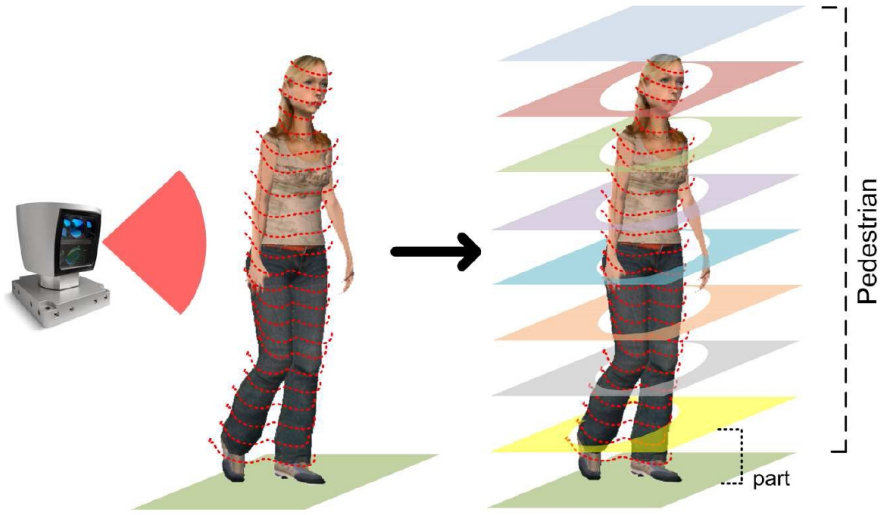

Figure 2.5: Diagram of the segmentation used by Spinello et. al. [SATS10].

a fore- and background image using the different layers of the stereo image. In their approach they are using image processing algorithms like skin color detection and face detection algorithms to verify whether the actual layer is really corresponding to a person or not. Due to the face recognition the person though has always to be facing the robot and therefor has to move backwards while guiding the robot to the specific location.

Another method is developed by Spinello et. al. [SATS10]. It is based on the three dimensional information provided by a 3D laser scanner. They segment the image by slicing the data into horizontal layers parallel to the ground (see Fig. 2.5). This algorithm then performs a similar search for person specific shapes like it was described for single layer or multilayer laser scanners before. However, this method cannot be used in real time since to the generation of a three dimensional image from the data of a rotating laser scanner is still very slow.

2.1.5 Combined approaches

The previous sections have presented different approaches that utilize a single sensor to detect a person in the environment. Although there are a lot of very reliable algorithms, the sensors itself still have their drawbacks due to a limited view angle or objects that they cannot detect (e.g. glass). This is one of the reasons why methods have been developed that use multiple sensors on one robot. [WBoG02] presents an approach fusing sonar sensor data and camera data to overcome occlusions that would be inferred by just using the camera. The most popular sensor fusion is done using a two dimensional laser scanner and a camera. Since both sensors are mounted on the robot in different locations occlusions are also unlikely to appear in both perceptions.

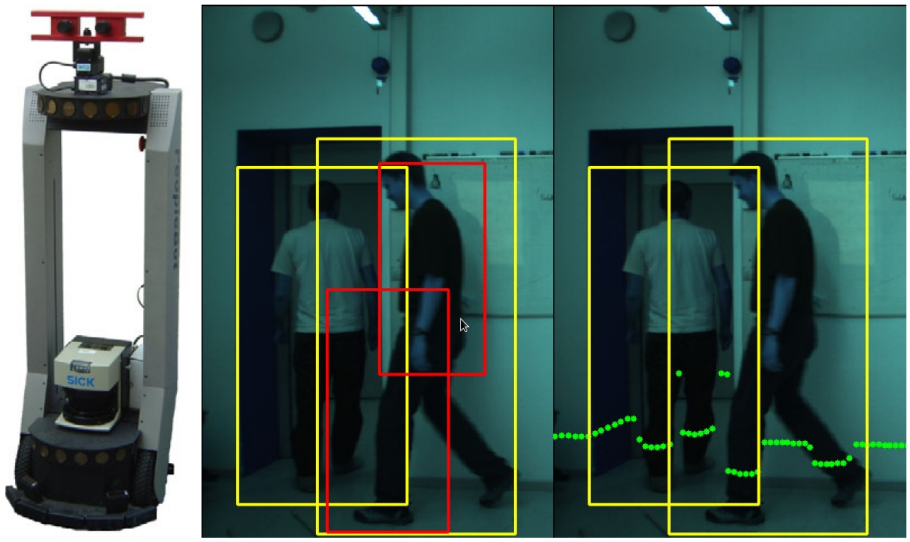

Figure 2.6: Combination of camera based HOG algorithm and laser scanner data to reliably classify persons in the environment [SAM+09].

Zivkovic et. al. [ZK07] present one of these systems which fuse the information gathered by a camera and a laser scanner. Therefor they first segment the data that has been perceived by the laser scanner using a Canny edge detector. After that, they classify the single segments with the use of a trained AdaBoost classifier. The camera image is in parallel processed by a part based approach (see section 2.1.2) trained for the upper and the lower body part of the body. The part based representation is then used to combine the information from the camera and the laser scanner to form a unified classifier.

Another similar approach is described by Schiele et. al [SAM+09]. They also utilize a camera and a laser scanner. First, they extract leg features from the laser scanner data and project these into the camera image. The camera image is processed by the Histogram of Oriented Gradients [DT05] algorithm and classified for potential person. Since the classification is quite noisy under certain circumstances they filter false detection by the following assumption: if the classified region is containing a projected laser scan in the lower three third of the area, the classification is positive and can be used.

2.1.6 Other approaches

The following part gives an overview over various other approaches that could be used to detect a person in the surrounding of a mobile robot. Some of them are not directly related to mobile robotics, but also do not have any restrictions which would hold them from being applied to this scenario.

One method is described by [Kra06] as “methods that imply the possession of an identifying object like a key, a magnetic card or a passport.” These systems require

persons to always wear a tag with them that discriminates them from other objects in the surroundings. Such systems are successfully implemented in companies for automatically registering and checking a person into the system and can be implemented very simple. The drawback though is the fact that every person that should be detected has to wear a key. Implementations use for example radio signal transmitting devices, artificial light emitting devices or other special equipment.

Christoph Ueberschaer [Ueb09] implemented such a system for identifying guests in the near proximity of the WelcomeBot platform. Therefore he equipped the guest with a Bluetooth tag that can be uniquely identified using the MAC address of the devices. With a special receiver on the robot it could receive this identification number and then query additional information about the user from a server. But although the receiver was of industrial standard and had a high precision, it was not possible to determine the position of the person with a higher accuracy than 5 meters.

Another approach is based on an acoustic sensing systems. These systems use several microphones that are attached to a robot [VMRL03] to detect and localize persons in a room. The localization of a person is though mostly restricted to an angular direction than an actual position. With the occurrence of echoes, reflections and absorbing material the localization of a person could be even more complicated.

Chapter 3

CONCEPT AND METHODOLOGY

This chapter describes the concept and methods on which the detection and tracking of a person is based. First, an overview of the purposed concept is given and then the different methods that are used to detect a person are described in more detail.

3.1 Concept

The main goal of this master thesis is the development of a robotic platform that is able to detect, track and follow a person in the environment. A robust and fast method is required for recognizing and identifying persons in the environment and tracking them over time. The robustness and the rapidness with which these can operate, however, is highly dependent on the actual sensor that is used for this scenario.

To meet these requirements different sensors have been compared. In the "state of the art" section already several sensors and methods were presented that are successfully used to detect persons rapidly and robustly in the environment. The outcome of the analysis of the different methods though was, that the most often used sensors for detecting persons are a laser scanner, a single camera and a stereo camera. All these sensors have been successfully applied in different applications either as single sensor or in combination with other sensors to detect a person reliably in the environment.

Based on this outcome, for this master thesis it was decided to test two of those sensors for the detection of persons: the 2D laser scanner and the dioptric camera. The stereo camera has not been chosen in this scenario since the usage of the sensor would have required an extra amount of time for becoming acquainted with the system. In addition to that, parts of the equipment for the system were not available at the beginning of the master thesis and the computational power for generating 3D images was considered to be too computationally expensive for the system.

Therefore, the following will concentrate on the description of the 2D laser scanner and the camera.

3.1.1 2D laser scanner

The 2D laser scanner has been used in a variety of applications from robot localization to the detection of persons in the environment. It measures the distance to nearby reflecting objects with high precision and is mostly invariant to lighting changes in the environment. The laser scanner operates with frequencies of up to 25Hz per scan line,

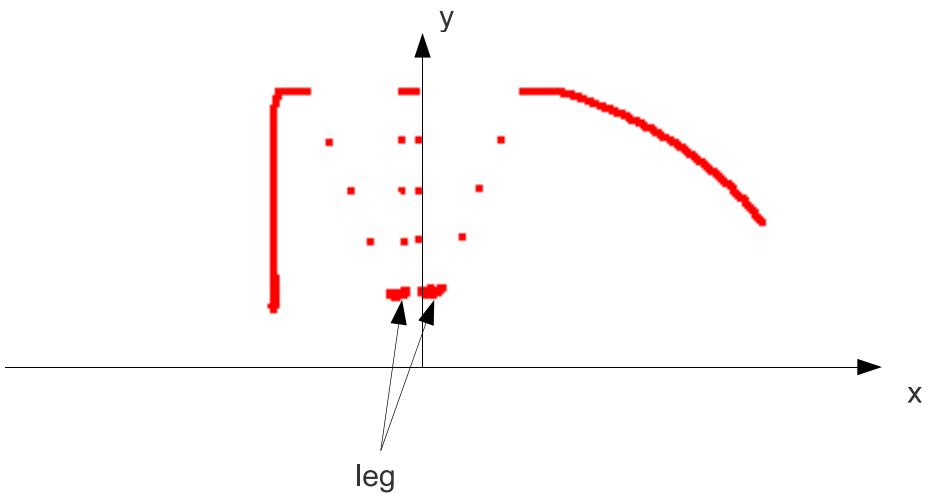

Figure 3.1: Raw data of a laser scanner with two legs in the scan.

depending on the actual mode that is chosen. The angular resolution of the scanner is up to a half degree precise and the depth precision at least one centimeter. It has an overall scanning range of 270 degrees.

The scanner, however, is not able to recognize mirroring or glossy objects that are not reflecting the laser beams as expected. Also, the information that it provides about the surrounding environment is very limited in its expressiveness. Since the scanner only perceives one layer of the surroundings, objects in the environment are represented by two dimensional shapes (see Fig. 3.4). The interpretation of the shapes can sometimes be quite difficult and ambivalent. For example, objects like the leg of a person and the leg of a table cannot be directly distinguished by the data. Therefore, other information about the environment is required.

The time for processing the data of a 2D laser scanner can be assumed to be very small, although it is dependent on the actual filters and algorithms that are utilized. Supposing the most precise case of a scanner with a resolution of a quarter degree and an scanning range of 270 degrees, the resulting number of values that would have to be processed by an algorithm would be maximally 1028. These are, in comparison to the amount of data that is perceived from sensors like a camera very few and can be processed very fast. Therefore, the 2D laser scanner is also known to be a fast sensor and used in a lot of robotic applications like, for example, fast tracking of persons.

To sum up, the laser scanner can be described as a fast and precise sensor, whose produced data lacks a certain kind of expressiveness, which could lead to a misclassification of objects.

Figure 3.2: A camera with fish-eye optics (left), a picture taken with a normal camera (center) and an exemplary image taken at the same distance with fish-eye optics (right).

3.1.2 Dioptric camera

The camera is the other sensor that is used in the robotic platform. The camera model that has been chosen for this master thesis is able to shoot high resolution color images of the surrounding environment with 30 fps and two mega pixel resolution. To enlarge the field of view of the camera a 0.25 wide angle lens[1] is attached to the front of the camera. Thus, it is possible for the robot to perceive objects in a field of view of up to 140 degrees (for comparison see Fig. 3.2). The full body of a person can be detected in a range of less than one meter in front of the camera. This will be necessary for one of the following methods used for recognizing a person in the environment. However, due to this modification of the optics there are quite big distortion in the corner regions of the lens. This effect will be discussed in a later chapter (see section 4.2.1), which will explain the process to generate an undistorted image out of the acquired image.

A drawback of the camera in comparison to the laser scanner is the missing depth information. In most cases, the distance to an object only can be inferred from the physical size of the item compared to its size in the image. This information, however, is highly dependent on the resolution of the image, the degree of distortions and how precise the object was detected in the picture. Therefore, generally only a rough approximation can be made about the distances to objects in the surrounding.

Similar to the laser scanner, the camera is perturbed by mirroring and non reflecting objects. In addition to that, most algorithms for the camera are highly sensitive to lighting changes. Although the camera model that has been chosen in this thesis has an auto exposure functionality, this function cannot be used. Due to the wide angle lens that has been applied to the front of the camera, approximately 35 percent of the resulting image is covered by the lens and is therefore black (see right image in Fig. 3.2). This has the effect that the camera internal auto exposure algorithm will assume wrong lighting conditions of the environment and therefore adjust the exposure according to a dark scenario. The

[1] The value 0.25 specifies the factor by which the focal length of the camera is reduced.

auto exposure functionality though can be disabled and set to a fixed exposure, which will be explained in section 4.2.1.

The main advantage of the camera is the fact that the images of it contain much more detailed information about the objects in the environment than the data of a laser scanner. Therefore, it is possible to uniquely identify certain objects in a camera image and even over several images by color, shape and size. The actual methods that are used for object classification vary from simple methods using color histograms to advanced methods based on SIFT features [Low04]. The methods that are used to detect a person will be discussed later in the methodology section.

The time for processing these pictures, however, is much higher in comparison to the laser scanner. Although this is still highly dependent on the technique being used, it can be generally assumed, that the time for analyzing a camera picture is much higher than the one used for a laser scanner. Thus, for processing a small image with a resolution of 320x240 already 76800 points have to be processed, which is a lot more than it is required for the high precision laser scanner.

The properties of the camera can finally be summerized as follows. The camera generates high resolution images of the surrounding with a high frequency of 30 fps, but is most of the times at least slower than the laser scanner in processing the data. It has a smaller angular range than the laser scanner and most algorithms are highly sensitive to lighting changes. However, the data that is contained in the images allows algorithms to identify single objects more uniquely and thereby track, for instance, a person more reliably over time.

3.1.3 Fusion of Data

The previous two sections presented the single properties of each of the sensors that are utilized in this master thesis. By comparing these with each other it can be observed that most of the disadvantages of one sensor compensated by the other one. By combining both sensors it therefore should be possible to compensate the weaknesses and produce a fast and reliable detector. This master thesis will present a concept for combining the results of both sensors.

Thus, for example, the low processing time that is due to the large amount of data of the camera would be compensated by the high frequency updates of the laser scanner. Additionally the bad depth perception could be enhanced with precise measurements received from the laser. Finally, scanner data that is sometimes misclassified could be verified by the more reliable information that is provided by the vision system.

The main idea behind this master thesis is it to have a slow and probably computationally expensive component that has a reliable estimator and a fast component that is able to track an object with high update rates and good precision of the position.

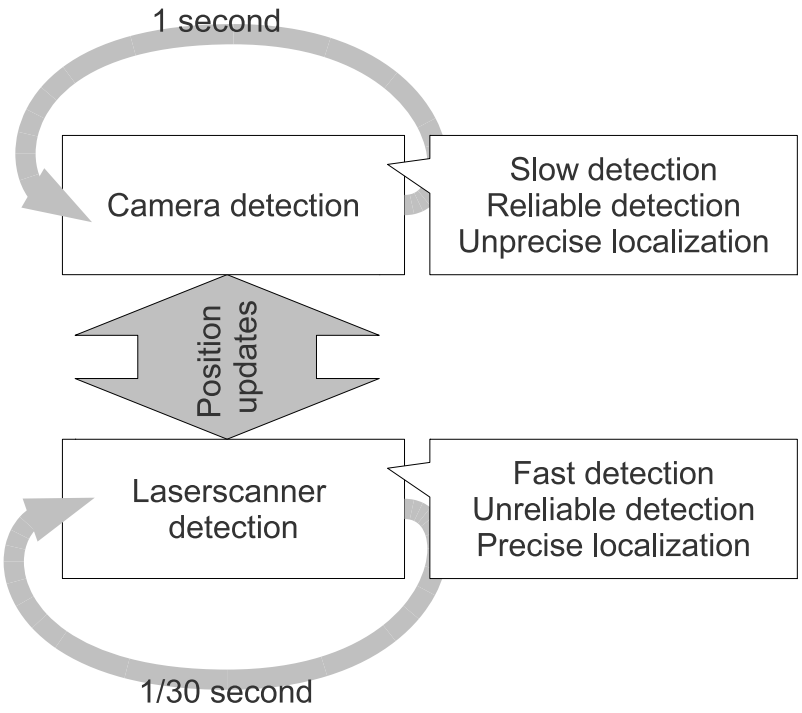

Figure 3.3: The concept for combining two restricted sensors, which complement each other.

This idea is also sketched in Fig. 3.3.

3.2 Methodology

In the "state of the art" chapter several methods that have been successfully used for detecting persons in the environment are described. Some of those were already implemented on mobile robotic platforms, and some are not directly related to the topic but would generally be applicable to them.

The following part will cover the approaches that have been chosen for this master thesis. It is subdivided into three subparts. The first part concentrates on the description of the algorithm that is used to detect legs in the laser scanner data. The second part then explains the approach for recognizing persons in the camera data and how this is extended for the use in an robotic scenario. Finally, the third part presents different methods on how both sensors' information can be fused together.

3.2.1 Laser scanner based person detection

The approach that is described in this section utilizes a 2D laser scanner for detecting persons in the environment. The sensor is mounted at the front of the robot at knee height such that it is scanning parallel to the ground. Doing so it is able to perceive the shapes

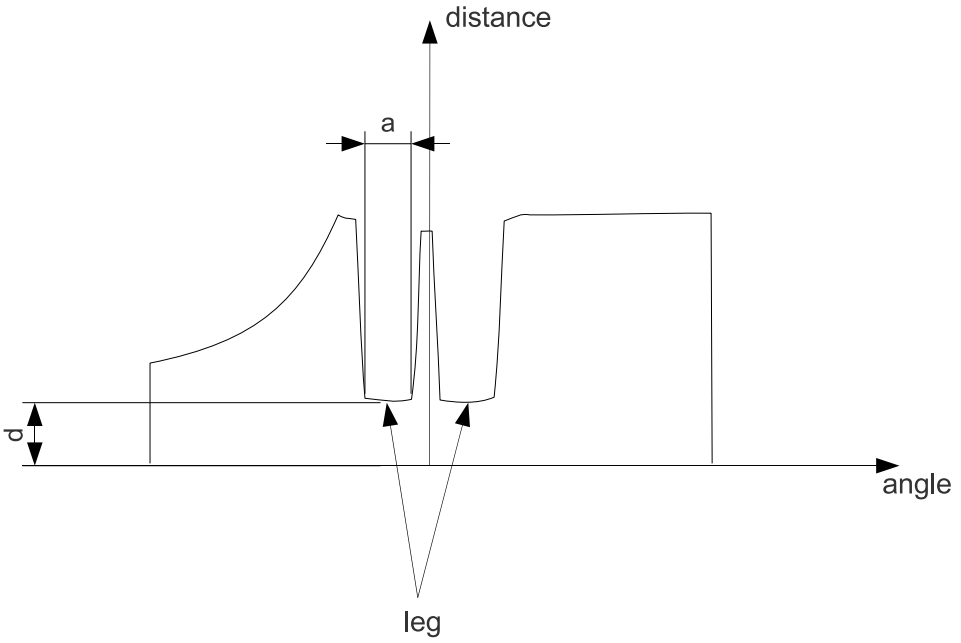

Figure 3.4: Raw laserscanner data with two legs in the scan.

of human legs in the surrounding. Although a mounting on waist or chest height would have the advantage that a lot of obstacles like chairs and tables would not appear in the scanning data, it was decided to mount the scanner at knee height for the following reasons: On this height the sensor data can also be used for other tasks like obstacle avoidance and mapping. The information about the leg position allows to make assumptions about the actual direction the person is moving and how this person is oriented towards the robot.

The detection of a person with a 2D laser scanner is based on the assumption that a person can be identified by one or two local minima that are detected in the laser scan data. Fig. 3.4 shows the laser scanner data produced by two legs next to the scanner in an ideal scenario, meaning with no distortions, other objects and noise of the scanner. The algorithm for detecting these patterns can be divided into three parts: segmentation, classification and tracking.

Segmentation is the process for generating connected regions of adjacent scan values that belong together under a certain criteria. This criteria is in this case a predefined threshold between adjacent scan values. Towards this end, the scan values are divided into subparts, if the distance between a previous and a following scan value increases by a certain amount.

The second phase for detecting legs is classification. It is used to identify whether segments, that have previously been extracted, belong to a know class. To do this, different kinds of so called classifiers are defined. They assign each segment a label whether it is a leg or not. The groups of classifiers that have been tested in this master thesis are as

follows. The first classifier is based on the width of each segment. It detects too small or too large legs and excludes them from further processing. The second classifier analyzes the shape of the segments. Therefore, it compares the segments with the ideal shapes of a circle and a line and outputs values that indicate how good these values fit to the expected shape. Based on the result, a segment is then denied or accepted to be of the type leg. The third classifier then detects whether in the surroundings another leg is located that could belong to the first one. If it detects a second leg, the probability for the first detected leg to be identified as a leg rises. At last it is tested whether and how good the remission values of the laser scanner can be used to classify a persons leg from other objects.

The data that is resulting from the classification process is then used by the tracker. For identifying and tracking corresponding legs in consecutive scans, a tracking algorithm should be implemented that searches in the neighboring environment for similar objects and associates them with each other. In addition to that, it should be investigated how the remission values of the laser scanner can be used to associate different legs with each other.

3.2.2 Camera based person detection

Cameras have been used in a lot of applications for detecting persons in the environment. The "state of the art" section presented several of those. Two of them are quite good established in robotic applications for recognizing humans. These are the algorithm described by Viola et. al. [VJ01] and Dalal et. al. [DT05] and both have been considered to be used in this master thesis.

A general reason for choosing these methods is the fact that an implementation of them is available in the OpenCV library [Ope]. The latter one is an open source computer library with a lot of programming functions for real time computer vision. It is implemented in C and C++ and was originally developed by Intel. It contains a lot of functions for image processing and classifiers for the HOG algorithms for example.

For the final approach that is described in this part of the work, however, it has been decided to use the algorithm described by Dalal et. al. [DT05] which is known under the term "Histogram of Oriented Gradients" (HOG).

The reason for choosing this one is originating from a comparison with the algorithm by Viola et. al. [VJ01]. Classifiers for detecting persons in the environment are either supported by the computer vision library OpenCV (HOG) or can be downloaded from Kruppa et. al. [KS][KCSS03] (Viola). Therefore, a comparison of both algorithms was done. The images were taken from the INRIA dataset and several independent images of persons at the institute were shot and classified with both algorithms. Some of the results can be seen in Fig. 3.5. The comparison showed that the performance of the HOG algorithm was in the overall result better than those from the Viola and Jones algorithm.

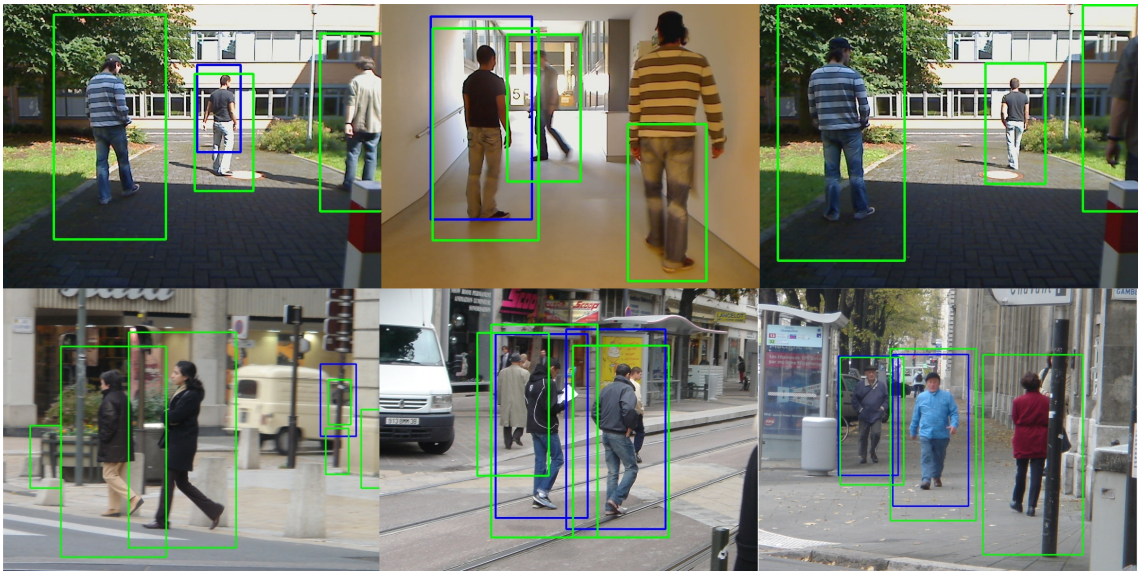

Figure 3.5: Comparison of classification by the Viola et. al. [VJ01] (blue) and the Dalal et. al. (green) [DT05].

In most cases, the HOG algorithm classified persons correctly while the other algorithm did not classify a person at all. Especially for the classification of the back of a person the HOG classifier was more reliable. These results have also been verified by a paper from Schiele et. al. [SAM+09]. They also compared both algorithms with each other. However, they reimplemented both algorithms and then used both SVM and AdaBoost as learning techniques to have a common basis on which both algorithms can be compared. The outcome of the comparison was that the the HOG descriptor and a similar shape context descriptor consistently outperformed the other Haar-like features independent of the learning algorithm. Due to these reasons the HOG algorithm was chosen for the implementation in this master thesis.

The HOG algorithm, however, also has restrictions. It works best if the whole silhouette of a person is visible in the image. To fulfill this precondition a wide angle lens has been attached to the front of the camera (see also section 3.1.2). The second restriction regards the processing time that is used for calculating the probable position of a person. The processing time has been estimated in tests with a resolution of 800x600 calculated on a 2.2 Ghz processor to approximately 1.5 seconds. This is too slow for following a person in real time.

Therefore, the approach in this master thesis is using the HOG algorithm as a temporal verifier. It should be executed in the background and detect persons in the image. In the intermediate frames, a faster algorithm should track the person while the slower computational expensive component is computing in the background. The algorithms that should be evaluated for the usage are the color tracking algorithm based on the mean shift

algorithm [FH75] and a template matching algorithm.

The idea here is similar to the overall approach, to have computational expensive methods that offer a reliable classification of an object and a fast tracking algorithm that is able to track certain parts of the image, for example based on the color, but is not directly able to detect a person.

3.2.3 Approach: Combined approach

This section describes the final approach for detecting persons in the environment of the mobile robot. It combines the two methods that have been described before to build a fast and reliable person tracking algorithm. The main idea of this approach is it to use the slow but reliable data of the camera to verify the fast and precise but unreliable data that is provided by the laser scanner. To ensure this, the following two methods should be considered.

The first approach calculates the regions corresponding to a person with the previously described HOG algorithm. The data of the laser scanner is computed in parallel and is projected onto the image plane. Doing so, it could be compared if the laser scanner and the camera based approach are classifying a similar region of the image and thereby deny or accept a classification of the laser scanner. A difficulty, however, is the fact that the laser scanner data is much faster updated than the data from the camera. Therefore, it can happen that the classifications are in a different image area.

The second approach only classifies certain regions of the perceived camera image. Therefore, first the laser scanner estimates a potential region of a person based on the leg detection. This information is then, similar to the previous approach, projected onto the image plane. After that a HOG detection is performed on this region of interest for potential persons that are located in this area. On a positive classification, the HOG algorithm would verify the detection of the laser scanner. This approach has the advantage, that it could perform a lot faster, since only a small region has to be analyzed with the HOG algorithm.

Chapter 4

IMPLEMENTATION

4.1 Laserscanner based people tracking

The following part describes the method for locating persons in the laser scan data. The main idea of the algorithm is it to identify local minima in the data and then classify these regions according to different criteria. Finally, the legs are tracked over several scans.

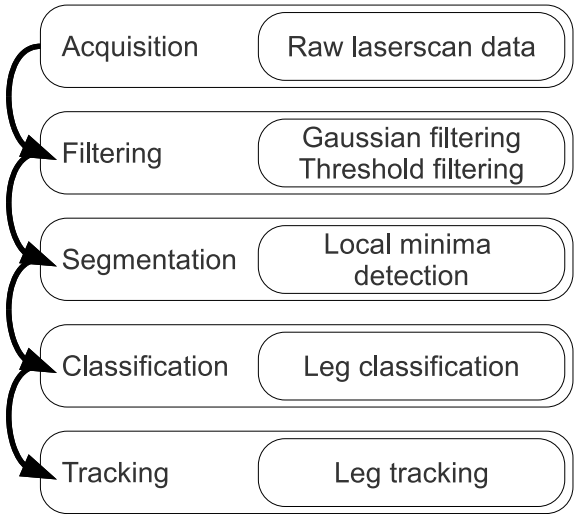

Figure 4.1: Laser scanner process chain.

The process for extracting leg features out of a laser scan is divided into four parts: preprocessing, segmentation, classification and tracking (see also Fig. 4.1). The "Preprocessing" step filters the raw data of the laser scanner for errors and prepares the data for the following processing steps. The "Segmentation" separates this data by the use of certain criteria into several connected regions. These are stored into a vector of objects and are then used for the "Classification". In the classification, the extracted segments are again searched for potential legs by using characteristic features of a leg like size, shape and so on. The results are stored as leg objects in a vector. Since not all potential legs in the environment of the robot should be tracked over time, a final tracker observes only specific legs, identifies them over time and tracks them while they are moving.

The algorithms and filters have all been implemented in C++ using the FAIRlib [Fraa] for accessing the laser scan data.

4.1.1 Preprocessing

In the "Preprocessing" step, the data is filtered and adjusted for all subsequent algorithms. The laser scanner LMS 100 from Sick, which is used in this master thesis, has a maximal statistical error of 20 millimeters and a typical error of 12 millimeters. This means that the data of the sensor varies in this range even when the laser scanner is at a fixed location.

To reduce these errors, a Gaussian filter is applied to the scan data. This has two effects. On the one hand, due to the smoothing effect, minor errors that are originating from the scan data are smoothed away and do not affect any of the subsequent algorithms. On the other hand, however, smoothing also removes sharp edges in the laser data, which could be good features for the classification. However, it was decided that the effect is negligible with regard to the actual benefit of reducing the errors.

In addition to that, two thresholds are applied to the data. The first one has a limit of 10 centimeters and excludes all laser scanner readings that are beneath this one. This removes smaller errors and does not have impact on the leg detection.

The second threshold filters all laser scan values that exceeds three meters. This has the following reason: the classification of legs is based on the amount of laser readings that are available for the classification algorithms to be processed. Although the laser scanner that is used in this master thesis supports a high angular resolution of 0.5 degrees, with an increasing distance, the spatial resolution decreases rapidly. Thus, a potential leg with a radius of 100 mm that is located in three meters distance from the laser scanner would only be represented by five laser points or even less. Since it has to be assumed that also errors can occur and degrade the current results, each leg that is behind this distance cannot be classified with a high reliability. Therefore, it was decided to limit the actual range for the laser scanner to a maximum of three meters.

4.1.2 Segmentation

The detection of legs is done by identifying local minima in the laser scan data d_i (see Fig. 4.2). Different methods are known in the literature for minima detection. For example [RC09] uses the so called jump distance to locate regions that belong to a leg. Towards this end, this algorithm looks in the neighboring scans of the laser scanner whether the distance of two adjacent scans is above a certain threshold and then splits the data at this point. In this master thesis, a similar method is used to identify discontinuities in the laser scanner data. A filter with the following kernel values is applied to the preprocessed scan data:

$$kernel = [1, 0.666, 0.333, 0, -0.333, -0.666, -1]$$

This filter has two effects. First, the ascending values of the filter smooth the whole

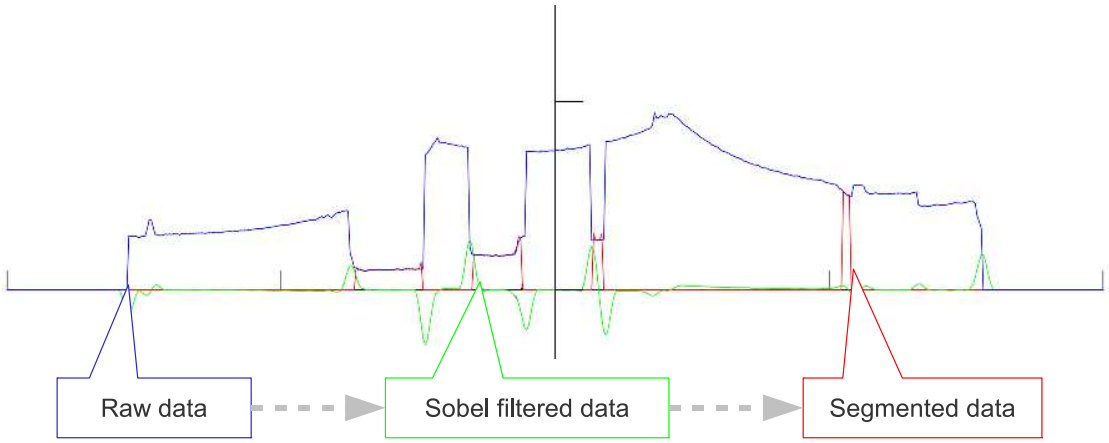

Figure 4.2: Segmentation process.

data in the region it is applied to. Second, it generates a graph representing the 1st derivative at each point of the data according to its predecessors. This is also visualized in Fig. 4.2 as a red curve. This data has a positive value when the slope of the data is negative and a negative value when the original data is increasing. This data is now filtered for a certain pattern which indicates a minimum.

To this end, the algorithm scans as long as it detects a rising positive edge followed by a falling edge (negative value). On success, it saves this part of the data to a vector of segments. The information that is stored in each of these single objects O_n is as follows:

- A unique index *uid*.
- The starting index i_{start} and ending index i_{end} of the detected segment in the laser scan data
- The raw laser scan data of the segment with the distances d_i and angular values a_i
- The point vector s_i with the coordinates x_i and y_i of each scan value with respect to the robotic platform
- The remission value r_i of each scan value
- An approximated width w_n of the segment in millimeters, based on the distance between the first and the last index

4.1.3 Classification

After a successful segmentation, the classification of the previously detected parts can be performed. The following methods represent different classifiers, that classify a leg according to the size, the shape, the reflection value of the scanner and whether or not there is another leg that belongs to the first one.

Width of leg

The first classification filters potential legs with a size criteria. Therefore, it uses the approximate size of the foot based on the rightmost and leftmost segment n. The allowed distance is defined as 100 millimeters with a range of 50 millimeters, such that the leg can have a maximum size of 150 and a minimum size of 50 millimeters. This simple filter already removes a lot of false classifications that come from mirroring and glass objects. Also, it excludes small pillows in the data that could originate from chairs and tables.

Shape of leg

The next criteria classifies the legs according to their shape. Towards this end, it fits the previously filtered data with a line and a circle and calculates which one has the minor error.

The first algorithm evaluates how well the objects fit to a line. It takes the left and rightmost scan data of the detected leg object as the beginning and the end coordinate of the line. After that, it calculates the deviation of the laser scan data from this defined line. Finally, a heuristic verifies whether the error is bigger than a defined threshold. If it is below the threshold, the object is classified as a potential leg.

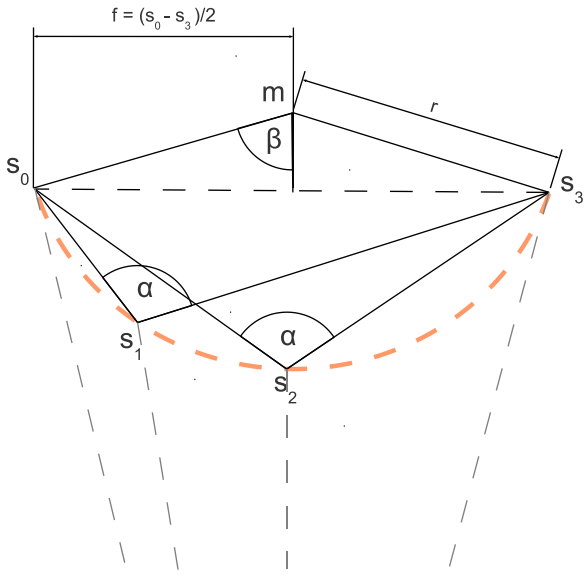

Figure 4.3: The picture shows the approach of Xavier et. al. [XPC+05] to classified circular objects. s_0 - s_3 denote different laser scanner measurements, the angle α is in each case the same. β is equal to $180 - \alpha$ and the radius can be calculated by $r = \frac{f}{sin(\beta)}$.

The next classification method tries to fit a circle to the acquired object data. Therefore, first, the RANSAC [FB81] algorithm was considered, but seemed to produce too much overload for these small segments. A second implementation is described in

[XPC+05] as the "Fast Line Arc Fitting Algorithm". It uses the fact, that in a circle, for two fixed points s_0 and s_3 (see Fig. 4.3) all points in between this segment (s_2 and s_3) are enclosing the same angle α. For the classification, an average angle and a standard deviation is computed an both are compared with some predefined parameters. This algorithm has been tested with different simulated and real objects and turned out to be a good classifier.

Remission of leg

The laser scanner returns two values: the laser scanner distance values with respect to the angle and the so called remission values with respect to the angle. In the following the latter values have been tested for the usage in a classifier. The remission values represent how well a surface reflects the laser beam. In case of the Sick laser scanner LMS 100, they are calibrated on the Kodak standard, a known standard in photography. The actual value, however, depends on three parameters: the angle in which the object is facing the laser scanner, the distance to the object and the reflectivity of the object's material.

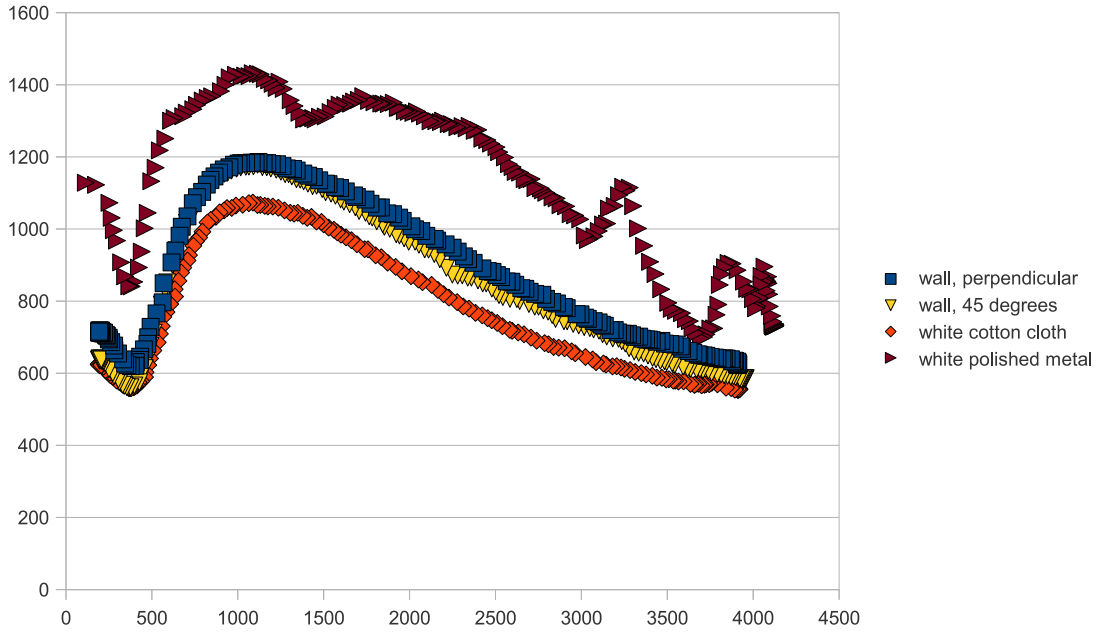

Figure 4.4: Remission values for different materials: Except for the metal, all materials exhibit similar remission values.

To test this behavior, a graph was recorded for objects with different materials and different orientations with respect to the laser scanner. Some of the results are shown in Fig. 4.4. It was observed that also for varying materials the graph has a similar shape. First, the values rise till a certain distance of 1000 millimeters and then fall again. The

tilting of the clothed wall up to 45 degrees also did not affect the measurements seriously, which, however, is due to the material of the wall. To confirm this result, in addition to the clothed wall another very shiny material was recorded. This is stated in the graph as "metal" and shows a rather distorted graph. The distortions, however, are due to minor changes of less then a degree while moving towards this material. This is showing how the actual material can greatly influence the actual remission values.

Based on these results, also the remission values of different trousers and legs were recorded in the near vicinity of the robot. However, it was observed, that no direct classification could be done based on the remission values. The remission values for different trousers were varying in the amount and in the shape. Such, for example, the remission values for blue jeans had a very linear shape, although the laser scanner readings measured a circular one. In contrast for a naked leg the shape was again curved while for a trousers with grey polyester, for example the shape was very noisy shape and was not directly linear. To filter these effects the remission values for one leg were smoothed by computing a mean value over all of them. This value, however, was quite stable for different distances. For example two equal legs of a trousers would have similar mean remission values when viewed from the same distance. This property will be used in the following sections to associate similar legs with each other. However, a direct classification of the legs is not possible using the remission values.

Association of legs

The forth classifier searches for two similar legs in the laser scan data that seem to belong to each other. Therefore, the classifier first tries to find a corresponding leg in the vicinity of the previously detected one. It searches in the objects for legs that are located in a certain distance and then compares their remission values with each other. Since the remission values of similar objects or legs in the near vicinity should also have similar values, except for a slight offset, the comparison of the values can be taken to associate legs with each other. Although this classification can be seen as quite robust, since a lot of other objects will be ignored, it is sensitive to occlusions of one leg from the other. It assumes that both legs of a person are visible to the robot. However, this classifier can be used as a verification of the other classification methods, i.e., it may verify of a person is really at the detected position.

4.1.4 Tracking of legs

The tracking of the legs is done using two assumptions. The first one is based on the high acquisition frequency of the laser scanner. It can be assumed that a leg of a normal walking person will only move a very small distance between two acquisitions of the laser scanner. Therefore, the distance between the detected legs in successive scans

should be very small. According to this observation it can be assumed that the leg that has been detected in a previous scan is that one that is near to the one that is detected in the current scan.

This assumption, however, is violated as soon as the tracked person is walking near to an object that is classified as a leg. To compensate this behavior, another assumption is made, which is based on the remission values of the laser scanner. This value indicates how well an object has reflected the laser beam. The value though is highly dependent on the actual angle of the object towards the laser scanner, the distance to this object and especially the reflectivity of the material. The latter property can though be used to associate different legs in successive laser scans with each other. Therefore, it is assumed that the remission value of one leg in two subsequent scans is only varying in a very minor way.

To sum up, it can be said, that the association of one leg in subsequent laser scans is based on the euclidean distance of both legs and their similarity of the remission values.

4.2 Camera based person detection

The camera on the WelcomeBot platform is a Logitech Pro 9000. It has a Carl Zeiss optics, a maximal resolution of 1600x1400 pixel and an auto white balance and an auto exposure functionality. To guarantee a fast processing of the camera images in this work a maximum pixel resolution of 800x600 is used for the processes described below.

The following part is divided into four subsections. In the first section, the calibration and the compensation of camera distortions is described. The second part will concentrate on the implementation of the Histogram of Oriented Gradients algorithm. The third part compares different methods that could be used for tracking the person in the images, while the main process is analyzing the image for persons. In the last part, an implementation is described how the HOG detection algorithm of the second part and the tracking can be combined with each other.

4.2.1 Camera calibration

In the “Concept“ chapter it was already described that the camera was extended with a fish eye lens, such that it is possible for the robot to perceive a field of view of up to 140 degrees. A raw image of this camera view is shown on the left side of Fig. 4.5. Since the algorithms that are implemented for person detection do not work properly on such an image, its distortions have to be compensated first, such that the resulting image looks like the picture on the right hand side of the Fig. 4.5.

To transform such a picture, OpenCV offers different functions that automate this so called ”undistortion” completely. Therefore, however, first the intrinsic camera parameters and distortion coefficients of the specific camera configuration have to be calculated. The

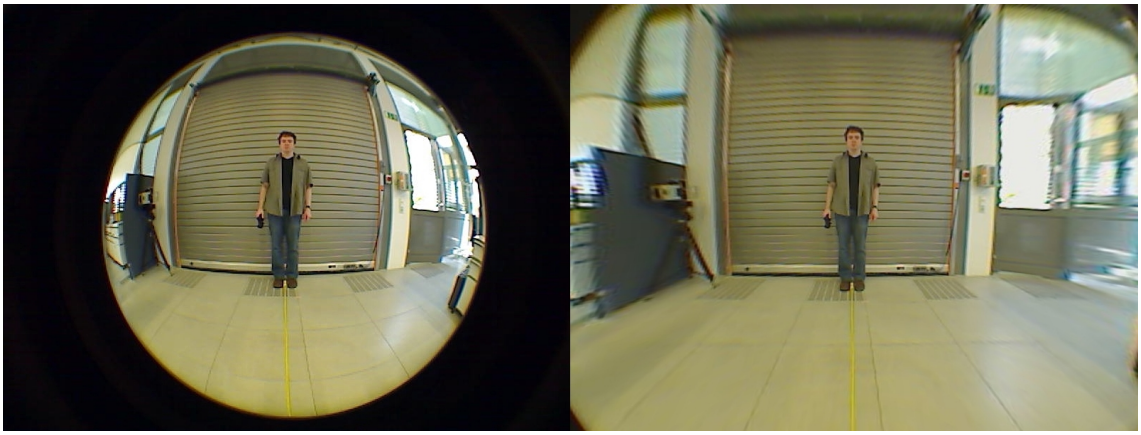

Figure 4.5: Image taken with a fish-eye lens (left) and the compensation of the distortions (right).

intrinsic camera parameters contain the focal length of the camera and model a possible displacement between the image center, i.e., where the optical axis hits the sensor, and the center of the sensor. In other words, the displacement describes the misalignment of the lens with respect to the center of the sensor. The distortion coefficient vector describes the radial and tangential distortions of a lens. Especially the radial distortion for the camera in this master thesis is quite large due to the wide angle lens.

To calculate these parameters, a lot of camera calibration tools are already available which automate the calculation of the intrinsic and distortion coefficients. Most of these tools are used in a similar way. They search for a certain pattern in the camera picture, recognize its position and orientation and calculate the according parameters.

The computer vision library OpenCV also offers a similar camera calibration tool. It uses a checkboard pattern for the calibration process. The user only has to define the number of inner corners for each board dimension and an optional square size. Then he or she holds this pattern in front of the camera in different locations and orientations, while internally the locations of each checkbox is calculated. Based on these informations, the two parameter vectors are computed. These can then be stored into a configuration file or for example utilized with the "undistort" function of OpenCV to generate an image without distortions (see right image in Fig. 4.5).

Looking at the resulting image, however, it can be observed that the corner regions are very blurry and still a little bit distorted. The first effect is due to the usage of the undistort method. Since the image in the corner regions has to be reconstructed using a very limited number of pixels of the original camera image (see left Fig. 4.5), the pixels in that region have to be interpolated, which results in blurry regions. The distortions which are still occurring in the corner regions will be discarded. In the center of the image, however, the resolution remains the same and it is now possible to see the full pose of a

person that is standing in less than 1 meter distance, which is required for the subsequent algorithms.

The Logitech 9000 camera can automatically adjusts the white balance and the exposure time based on internal measurements. This imposes another difficulty, especially when using a color tracking method. Since the robotic platform is moving around in the environment, the actual lighting conditions are constantly changing over time. Therefore, the auto white balance and auto exposure would adjust their values constantly. Algorithms, however, that are based for example on color tracking would therefore fail because they rely on constant pixel values. In addition to that, the auto exposure functionality does not work properly any more, because the wide angle lens has been attached to the front of the camera. Thus, approximately 35 procent of the original image (see left image in Fig. 4.5) is covered in black. The internal auto exposure therefore would always assume that the image is shot in a too dark region and adjust the brightness according to this incorrect values.

Due to these reasons, it has been decided to implement a function which is able to automatically enable or disable both functionalities before the camera is used. In addition to that, it can also adjust the according values to a fixed value. Since the camera is conforming to the USB video device class standard (UVC), the parameters can be easily adjusted using IO control commands of the UVC driver interface of the camera.

To simplify the usage of the camera in the following experiments, three methods have been implemented to automate the whole calibration and the configuration process. The intrinsic parameters and the distortion coefficients, which have been previously calibrated by the usage of the checkboards pattern, can be read very easily with a class called "CCalibrateCamera". The image that is passed to the class is updated according to the parameters and can then be used as a normal input image. A similar interface is implemented by the second class, which is called "CCameraProperties". This adjusts the exposure and whitebalance according to defined parameters. The last class is called "CCameraSetup" and automates the whole calibration process of the camera by reading the according intrinsic camera parameters, the distortion coefficients, exposure and white balance values from a configurations file. The output of this class is again the adjusted image that can be used for all the further tests.

4.2.2 Detection of persons

This section describes how the detection of a person with the camera is realized in the context of the robot. The detection is based on the algorithm that has been presented by Dalal et. al. [DT05] in 2005.

The algorithm works in several steps. First the edge directions of the image are computed in x coordinates ($dl_x(x, y)$) and y coordinates ($dl_y(x, y)$) using a simple 1-D

mask [1;0;1] sliding over the image. Then for each of the resulting pixels a gradient direction is calculated with the following formula:

$$\delta I(x,y) = arctan\left[\frac{dl_y(x,y)}{dl_x(x,y)}\right]$$

The new image now represents at each position the certain image gradient. Dalal et. al. then combined several of these pixels into so called cells of a certain size (e.g. 3x3 pixels). These vote with their gradient magnitude of each pixel ($I(x,y)$) in a histogram with 9 bins. The 9 bins theirby represent 0 - 360 degrees (depending on the application also 0 - 180) evenly distributed, such that the first bin takes all degrees in the range of 0 - 40 degrees, the next one 41 - 80 and so on. The resulting histogram is the so called histogram of oriented gradients. These cells are then again combined to several overlapping blocks (see Fig. 4.6) which have in most cases a size of 2x2, depending on the application. These blocks are then normalized and serve as the features for detecting, for example, persons in an image. Several of such images (positive images containing a person and negative images depicting everything that is not person-like) are then fed to a learning technique. Dalal et. al. used here in their original implementation a Support Vector Machine to generate a classifier. The trained classifiers are then saved and are used then for detecting a person.

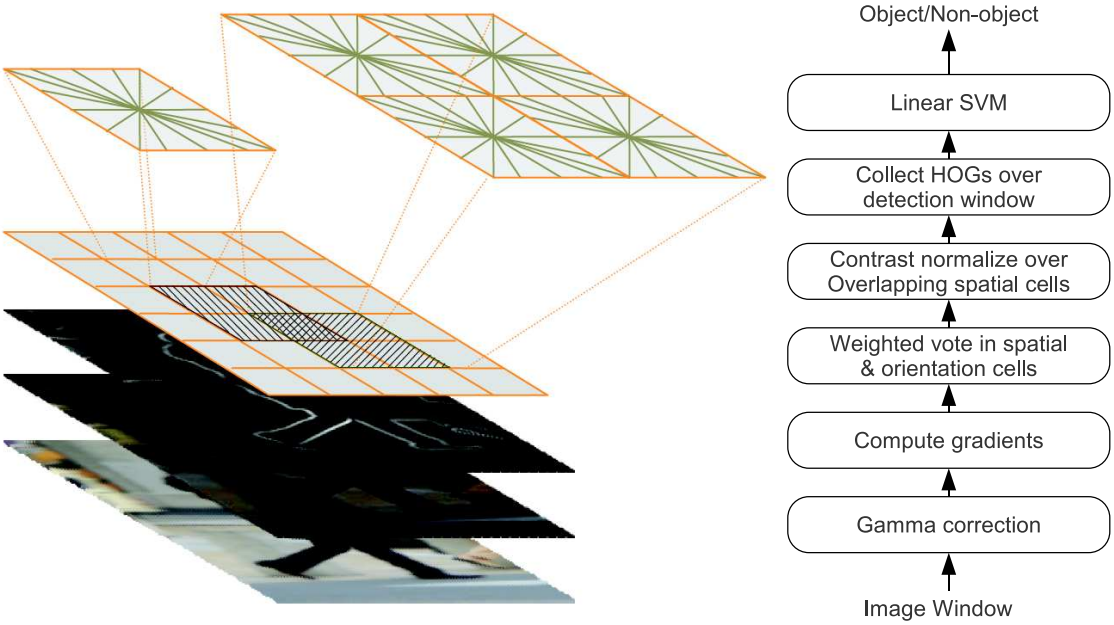

Figure 4.6: The picture shows the different steps in for the classification with the HOG algorithms [DT06].

In the detection phase, the following process is used to identify person in an image.

First, the image region under consideration is cropped from the original image. This region is preprocessed with different filters and then transformed according to the process

previously described to generate blocks of histograms of oriented gradients. These blocks are then compared with the trained classifier. If the correspondence, which is indicated by a threshold in the SVM, is high enough, the image region is classified as a person.

The original source code of the algorithm is published in the INRIA Object Localization Toolkit [Dal], which can be downloaded for free from the Internet. For this master thesis, however, it has been decided to use an implementation that is included in the OpenCV library. It is an adapted version of the original toolkit and is mostly compatible with that one. A suitable classifier for recognizing persons is also included in the code and used as the standard classifier.

The implementation supports different settings to adjust the method of operation of the algorithm. Thus, for example, it is possible to define a classification threshold. However, changing the the threshold increased the number of classifications, but also decreased the quality of them, meaning the correctly classified persons. A similar effect was observed when adjusting the value for the different scales that are used by the algorithm. This had the effect that the whole process was a lot faster, since the algorithm was only checking for a few scales over the whole image if a person is displayed in the area of consideration. However, this again degraded the quality of the classifications. Therefore, it was decided to leave the parameters for the HOG algorithm as they have been recommended by the authors of the code.

For a simple usage in this master thesis, however, a wrapper was written for the algorithm. Thus, the whole parametrization of the algorithm as well as necessary conversions is done inside of the class. The software interface is designed very simple and only takes an image as an input parameter. The resulting classifications are given in a vector containing the rectangular regions where a person was detected.

The algorithm was tested in different environments with a resolution of 800x600 using the previously described camera calibration. It returned suitable results that classified humans correctly in the image. However, the regions that were classified in this image did not always correspond to a person. It was observed that the classification problems did not originate from the wide angle lens or the calibration, since a similar effect was also observed when using a normal camera without any modifications.

The first approach to fix these misclassifications was to filter all regions that did not correspond to certain constraints. The first constraint was originating from the fact that most of the persons that are classified with this algorithm are in the near vicinity of the robot. Therefore, the classification should at least fill half of the image. Since most of the misclassifications had a very small rectangular height, this filter just rejects all classifications that do not fill half of the image and therefore correspond to a person that is standing in the near vicinity of the robot. Tests of this classifier showed in an increase in the correctly classified persons.

The second constraint originates to the mounting of the camera. Since the camera was mounted parallel to the ground and at the approximate height of the waist of a person, it could be assumed, that if a person is classified by the algorithm, the bounding box has to be above and beneath the center of the image. Therefore, all classifications that did not correspond to this constraint were rejected and removed from the actual results. Thus, it was possible to filter a lot of false positives out.

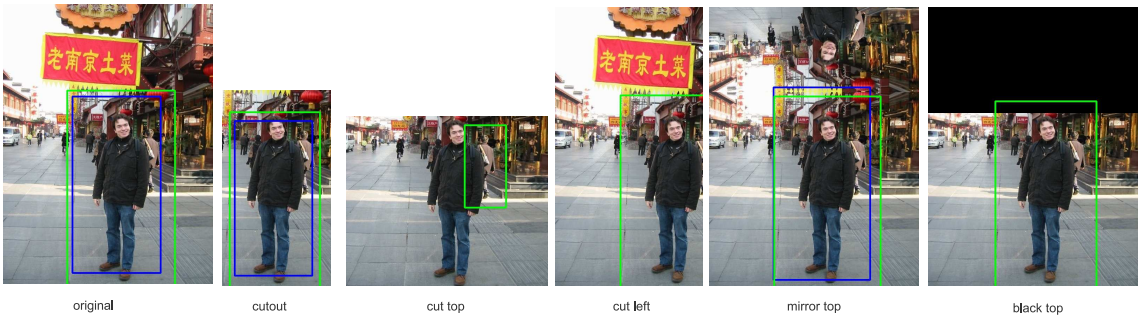

Figure 4.7: Different tests with camera images classifying a person.

An effect that was observed while using the algorithms was, that if the head of a person would appear very near to the upper corner of an image this could be a reason for the algorithm to fail. To verify this effect some tests were performed. Some of the examples are shown above (see Fig. 4.7). According to the position where the image was cut off, the classification was successful or not. This, however, can quite fast happen in the context of the work. Since the persons that should be recognized with the robotic platform are very near to the robot, their actual size will almost fill the whole image.

A reason for this error could be due to the specific training of the classifier. Since the classifier has been trained on persons that are standing upright and have a certain amount of free space above and beneath them, this could affect the actual classification of the algorithm.

To overcome this effect, it was tested whether a mirroring or a black border on top of the image would solve the problem (see two rightmost pictures in Fig. 4.7). The classification of the HOG algorithm now worked again and was not seriously tampered by this padding. Since both methods worked properly, it was decided to use the black padding of the corner regions for computational reasons. Using this padding above and beneath the window the actual detection could be improved by a large amount.

The processing time, however, was still negatively affected by the larger detection region that would now have to be analyzed for a person. Therefore, different scalings and their effects on the detection rate were tested (see section 5.2.1). With a reduction of the original image to 400x300 pixels and a padding of 50 pixels on each side, the best results were achieved. Doing so, the actual image now had a size of 500x400 pixels and

it was decided to be taken for the implementation. The processing time, however, could be reduced by a factor of 4, from 2 seconds to 500 milliseconds. This is a lot faster than the original implementation but still too slow in the context of following a person in the environment.

Another difficulty of the algorithm originated from the camera, which is somehow buffering the images in the background while the HOG algorithm is processing the results. Although this effect was minimized with the reduction of the image size it still could be observed that the camera image processed by the HOG algorithm is delayed with respect to the actual frames. Similar effects were also observed using other algorithms that are using a lot of processing time for calculating image results. To overcome this problem, threads were introduced to the image processing pipeline. This way, the camera could run in one thread and poll the images, while another thread is performing computations on the current image with the resulting classifications. The implementation was done using the pthread library, which is implementing kernel-level threads. This has the advantage that the different threads are distributed across different cores and are therefore speeding up the application. Although this does not yet speed up this application in one of the following approaches this will be of importance to decrease the processing time.

4.2.3 Tracking of persons

This section describes two approaches that were chosen to track a person in an image. The following part describes an approach using the mean shift method for tracking a color and an approach using a template matching method.

4.2.3.1 Mean shift Color Tracking

The mean shift algorithm was first presented by Fukunaga et. al. [FH75] in 1975. It locates local extrema in a density distribution by iteratively calculating the mean in a certain search window and then relocates the center of the window by the amount of the so called mean shift vector. Bradski et. al. [BK08] describes this algorithm as "just a hill climbing applied to a density histogram of data" [BK08]. This algorithm is used quite often in computer vision to track objects in an image by its color. It should also be used for the tracking of a person in this master thesis. The following paragraphs describe how it works and where difficulties are occurring.

The first difficulty that should be considered when tracking a color in a camera image are the lighting conditions. Although in the case of this master thesis the time period between an update for the tracking algorithm is quite low (500 milliseconds), lighting changes in the image can still occur and should be handled. One reason for these changes could be the auto white balance and auto exposure, which were therefore disabled in advance (see also section 4.2.1). Thus, changes from the camera itself should not affect

the tracking anymore.

However, to reduce the lighting problems that could be caused by the surrounding environment, the camera image is converted into the HSV (hue, saturation and value) color space . This color space is used in a lot of applications for tracking colors, since when only considering the hue and the saturation layer of an image, the actual values are mostly invariant to lighting changes in the environment. Therefore, the image is first converted from the RGB color space to the HSV color space. The image is also filtered for values that have a very high saturation and hue value. These values are mostly originating from very light spots, like lamps or the reflections, and therefore can be excluded.

The mean shift algorithm calculates a vector describing how the local maximum has moved in the subsequent images. Therefore, it requires a density distribution as an input or, in the case of the OpenCV, a picture describing where the most pixels are located that have a corresponding color to the objects that should be tracked. For this reason, first, a histogram of the object or person is created. The resulting histogram, however, only contains values for the hue and the saturation.

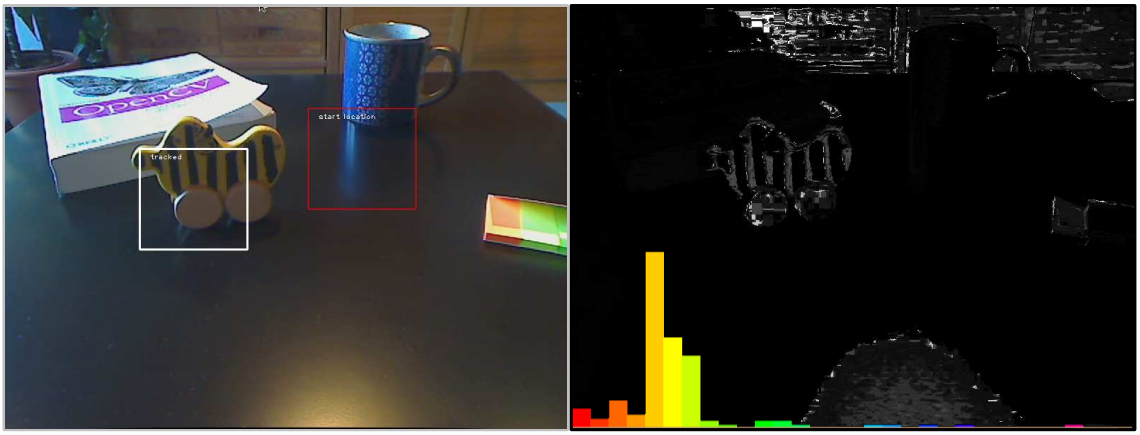

Figure 4.8: Mean shift algorithm: Tracked object (left), backprojection of the histogram (right) and histogram of tracked object (lower right).

Now, the current camera image is processed. It is filtered for all colors that do not correspond to the previous defined color histogram. This is done by using a so called backprojection. With this function, it can be recorded how well the pixels in the camera image fit to the distribution of pixels in the histogram. The resulting image is shown in Fig. 4.8. The image now represents all those pixels that have a similar value to those in the corresponding color histogram of the object or person. This image and an additional approximate initial position can now be used with the mean shift algorithm to calculate how the object has moved in the subsequent image (see left Fig. 4.8).

The mean shift color tracking is a very fast method for tracking an item in an image. It has a processing time of less than 10 milliseconds and converges in most cases in the

right direction. However, it works best with objects that have a very distinct color that is not occurring in the surrounding environment.

The tracking of a person, however, is causing some additional problems. Although the time between two updates of the Histogram of Oriented Gradients is quite small and therefor the actual lighting changes and the movement of the person would also be small, it could still happen that this algorithm fails to track the person since another color in the image is identical to the color of the person. This especially occurs if the clothing of the person for example is similar to a color of a surrounding object and the person is moving near to that. In this case, the algorithm will fail to work.

4.2.3.2 Template matching

An alternative to the mean shift color tracking algorithm was found in the template matching algorithm. This algorithm can be compared with the sliding window techniques presented earlier in this master thesis (see section 2.1.2). It matches a so called template, which is an image of the object that should be tracked, to an input image and computes the region that corresponds the best to this. This is done by sliding the patch over the input image and using different matching methods for comparing whether the area under consideration is fitting best to the template (see Fig. 4.9). In OpenCV, three different matching methods are offered: squared difference matching, correlation matching and correlation coefficient matching. In addition to that, for each method a normalized version is also available to reduce the effects of intensity differences between the template and the image.

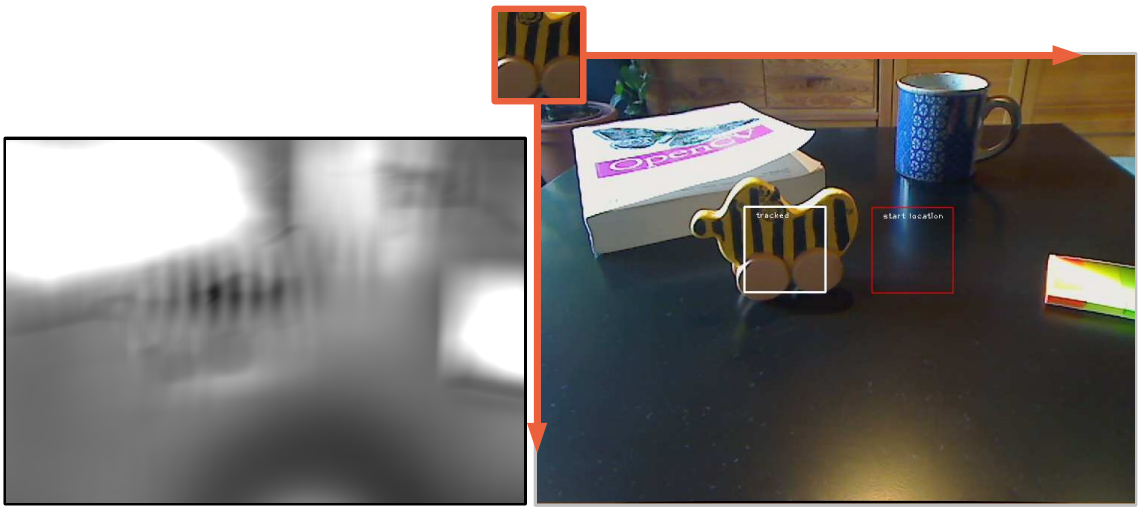

Figure 4.9: Template matching algorithm moves a template (see small picture at top) over the image (see right). The result of each comparison is shown in left image.

For this master thesis, the different methods were compared to each other. The

final decision, however, was made in favor of the squared differencing method with a normalization. This one returned the best results under different lighting conditions as well as different environments. The other methods, however, found less often the correct match for the template in the image.

The square differencing matching method calculates the difference between the actual template patch and the image region by squaring the differences between both images. Therefore, it moves the patch T(x,y) over the whole image I(x,y) and calculates the value R_{sqdiff} for each matched window.

$$R_{\text{sqdiff}}(x,y) = \sum_{x',y'} \left[T(x',y') - I(x+x',y+y')\right]^2$$

Fig. 4.9 shows some examples for sliding window positions. The resulting image $R_{\text{sqdiff}}(x,y)$ is depicted on the left side of Fig. 4.9. The dark pixels are corresponding to areas where the image fits best.

4.2.4 Combination of tracking and person detection

This part describes how the recognition of the Histogram of Oriented Gradients and the tracking algorithms are combined. For the robotic platform, it is important that it always knows where a person is located in the environment. The HOG algorithm can distinguish this position quite good, but is also quite slow. Due to this reason, different tracking methods were introduced earlier in this chapter, that are able to track a person with a higher frequency.

To enable these different methods to work concurrently, each algorithm and additionally the camera are handled by separate threads. The thread handling was implemented using the pthread library, which is a POSIX standard for thread implementation and uses the OS internal scheduler. This way, the different threads are distributed across the CPUs on a multiprocessor machine, which results in a reduced overall processing time of the method.

The programming using different threads requires, first of all, to prevent data corruption. Additionally, a timing slot has to be defined by which the different applications can submit their data. Since the camera is submitting the data the fastest way it defines the actual minimum time slot between each update. The implementation of the Histogram of Oriented Gradients and the template matching algorithm and the mean shift algorithm have a flag signaling when the data is ready to be returned. As soon as this is the case, they are submitted to the main thread and visualized in the application. The time dependent processing of the different steps is visualized in Fig. 4.10.

The data transmission between the different threads is managed as follows: Each thread has a flag which signals new data. As soon as the HOG thread for example signals

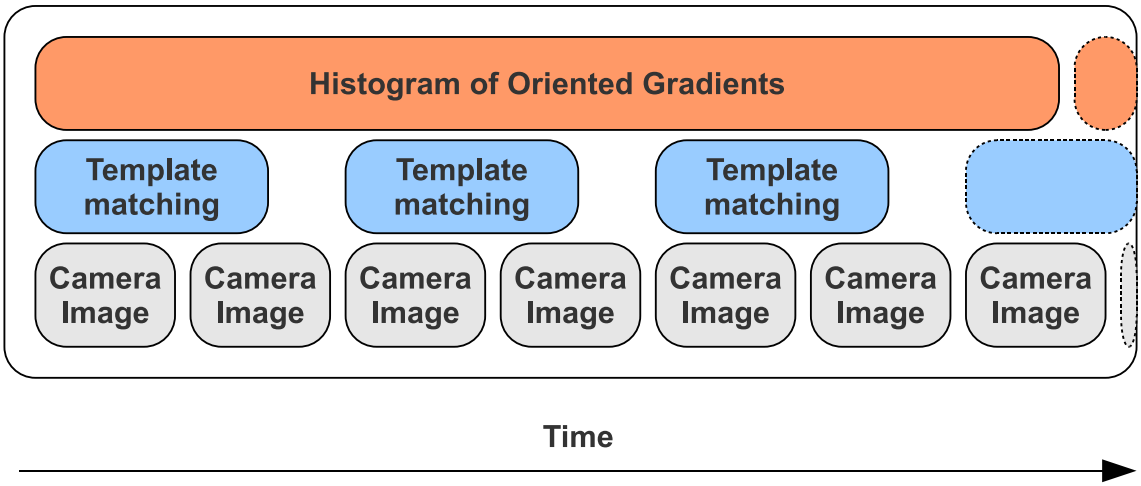

Figure 4.10: Timing of the different threads implementing HOG, template matching and the Camera.

a new update of the location of a person, the image of the person is submitted either to the mean shift or the template matching thread and the HOG gets the new current image from the camera. The tracking methods then update their histogram or the according template to the image that has been submitted and continue to track. However, since the HOG algorithm is quite slow, it could happen that the location where the person was detected in the previous scan is not in the near environment of the location where the person is now located. This does not affect the template matching algorithm, since it performs a global search for the image area that fits best to its template. In turn, the initial position is very important for the mean shift algorithm. Since it always needs an initial position where the person is located and from where it can shift to the next maximum, this is affecting the algorithm. To overcome this problem it is assumed that the previous position of the mean shift algorithm is quite precise and so this will also be taken for the next iteration. Since the update frequency of the method is very high, possible inaccuracies in the position can be excluded. It turned out that detections in the top or bottom are an indicator that the algorithm has failed to detect the person, since in the most cases the person appears in the center of the image and at the top and bottom of the image either the ceiling or the floor is shown.

4.3 Combination of the camera and the laser scanner

As it has been described in the concept chapter, the final idea is the combination of the fast laser scanner and the slow, computational expensive camera approach. Therefore, first of all, threads for both input devices are introduced to enable the different program parts to work concurrently. To merge and compare the data of the camera and the laser scanner, a similar approach to the one presented by Schiele et. al. [SAM+09] was chosen.

In that paper, the authors projected the laser scanner values into the camera picture, and were able to validate the results of the HOG algorithm. In this master thesis the laser scanner values are also projected into the camera picture. The HOG classifications then validate the laser scanner classification.

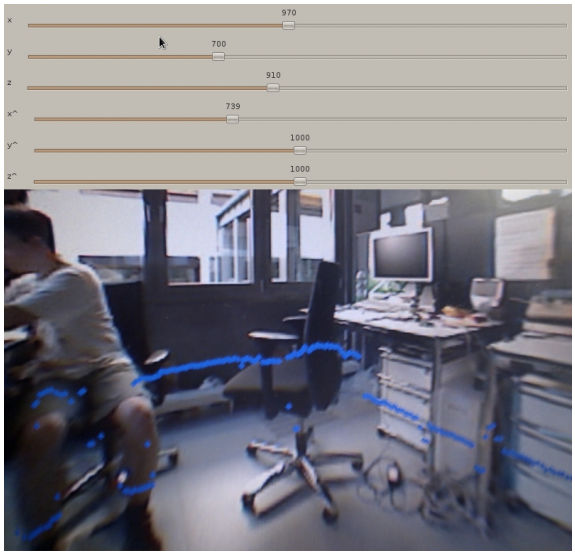

Figure 4.11: Calibration tool for the extrinsic camera parameters.

For the projection of the laser scanner values the camera intrinsic and extrinsic parameters are needed. The intrinsic parameters, including the focal length and the displacement of the projection center, have already been calculated with the calibration tool of the OpenCV library earlier in this work (see section 4.2.1). The extrinsic values define how the camera is located in the laser scanner coordinate system. They consist of two vectors describing the translational and the rotational displacement. Although the height between the two sensors can be distinguished quite easily, the values in the other two coordinates as well as the rotational displacement are hard to measure precisely. Therefore, a small configuration tool was written that simplifies the calibration. Fig. 4.11 shows the interface of the tool. It projects the laser scanner value onto the image plane. For each coordinate, it has a slider which is used to adjust the displacements in x, y, z coordinates and the rotation angles around the x, y, and z axis. The user can then adjust those sliders according to the different displacements and verify directly, whether the values are projected correctly onto the image. The best fit of settings with respect to the projection of the scan values is saved and is then used in the following algorithms.

With the use of these informations now the verification with the HOG algorithm can be done. There are actually two ways to do this. In the first approach the HOG algorithm calculates all potential persons in the whole image and marks them accordingly with the rectangular regions. These informations are then compared with the projected

laser scanner classifications, which are either confirmed or rejected. However, this theoretically requires the HOG algorithm to process the whole image, which is not necessary in most cases. Therefore, an alternative implementation only considers those areas in which probably persons are located and which have been recognized by the laser scanner based detection. Only these regions are cropped from the whole image and analyzed with the HOG algorithm. This way, the processing time needed for the HOG thread reduces to a minimum and is also not so error-prone to misclassifications.

Chapter 5

EVALUATION

This chapter describes different tests that have been performed with the methods presented earlier in the "Implementation" chapter 4. It is divided into three subsections: laser scanner based approaches, camera based approaches and a section describing tests that were performed with the robot in a dynamic environment while following a person. The first two sections mainly describe the evaluation for the single laser scanner and various camera based methods in a static environment. The last section then shows tests how different methods and combinations of those can be used in a dynamic environment to follow a person.

5.1 Laser scanner based detection

This section shows the results for the classification and the tracking methods used with the laser scanner. First, an evaluation interface is described that visualizes the scanner values, the classification and the tracking results. After that, the classification and the tracking algorithms are analyzed in a static environment. A test that utilizes these methods to follow a person in a dynamic environment is described in section 5.3 in combination with other approaches that have been tested in this scenario.

5.1.1 Description of the laser scanner interface

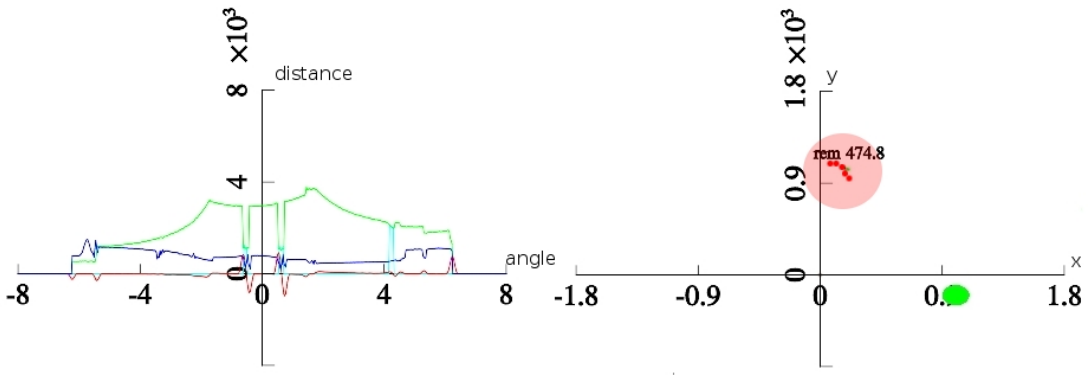

Figure 5.1: Interface for testing the laser scanner based person detection. Left image shows different graphs of filtered laser scanner distance values with respect to the according angle. Right image depicts the detected legs (red points) in the Cartesian coordinate system and the tracking direction (green circle).

For visualizing the laser scanner values, a specific evaluation interface has been designed in this master thesis. It offers a lot of functionalities to visualize data. Fig. 5.1 shows the laser scanner evaluation interface. The left image depicts a graph with the current distance values with respect to the according angle. This data has already been partially filtered. The raw laser scanner data is marked as a green graph, the extracted segments are colored red, the turquoise graph is the differential filtered data and the remission values are blue. The right image of Fig. 5.1 shows the different laser scanner measurements of the leg segments in the Cartesian coordinate system. The detected legs are marked with red points and the actual tracked position is marked by a green filled circle. Each leg detection is also labeled with the additional value named "rem", which represents the mean remission value of the detected leg.

This interface has been used in the evaluation of the system to observe which items are detected correctly and when the tracking algorithm has probably lost the legs of a person. The following tests refer to this interface for an explanation of the test results.

5.1.2 Classification and tracking in a static environment

Figure 5.2: The left image shows different objects that were used in the classification. In the right image the red dots mark the different leg classifications.

In this section, different tests are presented that were performed with the classification and the tracking algorithm of the laser scanner (see section 4.1). First, the performance of the classification algorithm in a static environment was evaluated. Therefore, different items were placed in front of the laser scanner. These were mainly various pillows, bottles and boxes (see left image in Fig. 5.2). They were chosen based on their sizes and shapes. These items were then scanned with the laser scanner and classified according to the criteria presented earlier in this master thesis. The objects that were successfully identified as legs are presented in Fig. 5.2 on the right side marked by red dots. Generally speaking it can be said that the big objects were rejected from the classifier, while the smaller objects, like the pillow on the left side or the red rectangle in the center were

classified as potential legs. The tests were repeated several times with a similar result. After that, persons were placed in front of the laser scanner. These were wearing different trousers, like various kinds of jeans, trousers made out of polyester and also some shorts, such that actually the bare legs could be detected. Since some of those trousers, however, could not be detected directly with the predefined values, the thresholds defining when a leg is detected as such were slightly adjusted. This way, all legs could be successfully classified as such in the end. On the one hand, this had the effect that more objects are identified as possible legs, but also that all trousers and persons legs could in the end be classified correctly.

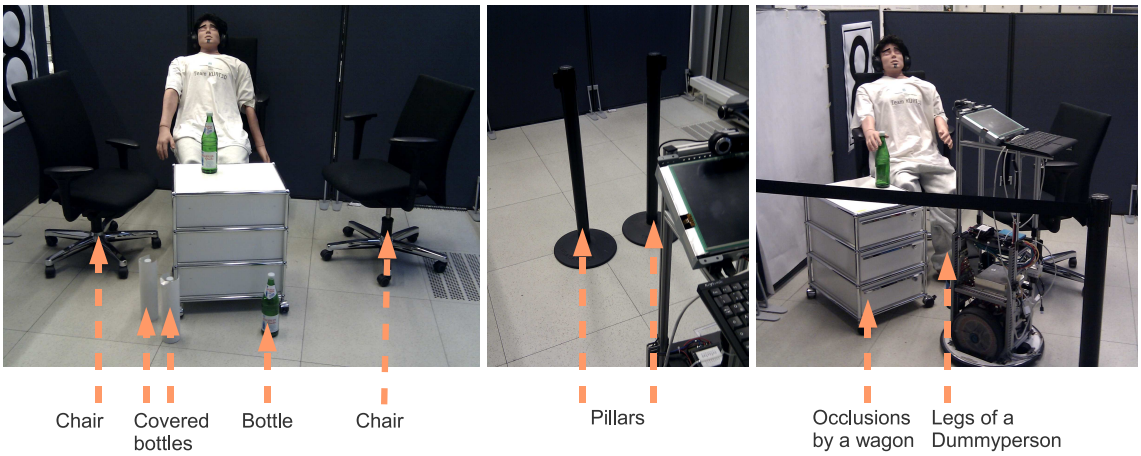

Figure 5.3: Test scenario with leg like objects in the laser scanner range and different occluding objects.

Next, the tracking algorithm was evaluated. Therefore, different test scenarios were set up in the test environment (see Fig. 5.3). The objects that were used are: green bottles, with and without a white paper wrapping, chairs, a second dummy person sitting in the back, pillars, small boxes, tables and many more (see annotations underneath Fig. 5.3). For each scenario, the laser scanner was placed at a fixed location, such that it was not moving during the test. Then the tracker was initialized by scanning the legs of the person. The person then walked through the environment and was tracked while walking.

Some of the tracks in the environment are shown in Fig. 5.4 as green lines. The leftmost picture was generated in the environment that is also depicted in Fig. 5.3 in the leftmost image. Here, the method did not loose the tracking of the person in favor of any other object[2]. The same was done in the second scenario (see center image in Fig. 5.3). Also here, the tracking of the person was not lost, even when he was walking behind the two pillars. However, the actual tracking of the person can also fail. This is demonstrated

[2]The detected leg objects in the environment are marked as red dots (see Fig. 5.4). If the tracker would, even partially, have lost the person while tracking, the green line would tend to cross or even cross those red dots.

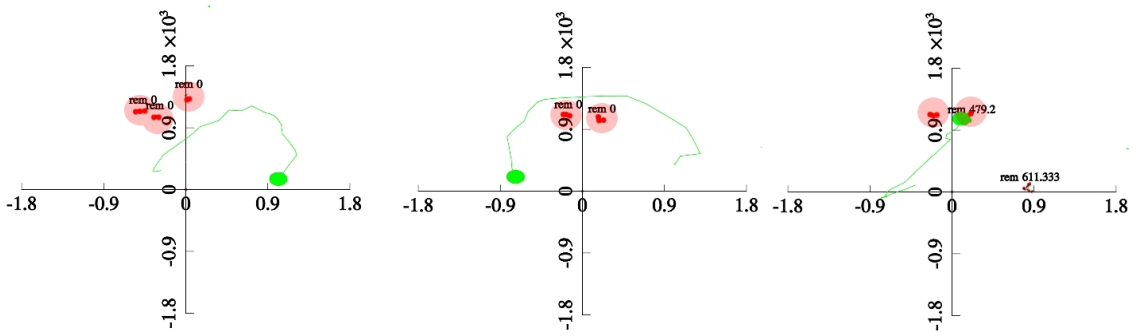

Figure 5.4: Different tracks recorded of persons walking in the environments presented in Fig. 5.3.

in the rightmost picture of Fig. 5.4. This time, the person was walking directly in the middle through the two pillars. Therefore, the distance between the legs and the object (tracking radius) was not large enough any more and additionally the remission values were quite similar such that in the end the pillars were detected as possible legs.

The tests have shown that the tracker is quite stable, also when the person is walking behind some pillars or other small objects. However, in the case when a person or an object is very near to the tracked person, the tracker can also fail. Some more tests in a dynamic environment are described in section 5.3.1.

5.2 Camera based detection

In the following section, several tests are presented that show the performance of the different camera based detection and tracking techniques in detail. The section is divided into three parts, describing the HOG detection of a person, the mean shift color tracking and the template matching algorithm. In each of these sections, the different algorithms are tested for their performance in a mostly static environment.

Interesting aspects of these tests are, how the wide angle lens with the corresponding reduced pixel resolution and distortions in the corner regions is affecting the results of each algorithm. In addition to that, different resolutions of the camera are evaluated to check, whether they affect the performance of the algorithm.

5.2.1 Test of the HOG algorithm

The Histogram of Oriented Gradients is used to locate persons in pictures of different environments. The descriptor that is used in this thesis was trained with the INRIA dataset [pd], which contains different sets of training and testing pictures. With the pictures also some tests were performed in this master thesis and the overall performance was very good. However, these photos were all taken with a camera that had a normal camera lens with no distortions and a reasonable pixel resolution. The camera that is used

in this master thesis is equipped with a wide angle lens whose distortions are corrected by a computer algorithm. This has the effect that the images have a quite small pixel resolution in the corner regions and some minor distortions. Therefore, it would be interesting if the algorithm would still work in the corner regions of the image. This is evaluated in the following part.

For this master thesis, an individual small test dataset of several persons was created (100 pictures). It contains photos of persons in different poses and in five scenarios: outside at a sunny day, under well defined light in front of a gate, in a hallway with artificial light, in a room with bright backlight and in front of posters, robots and other things standing in the back. For acquiring the pictures, the same camera configuration was used that has also been described in the previous chapters.

Figure 5.5: Examples of the classification with the HOG descriptor. The left four images show positive classifications, the rightmost image shows a false classification and a case where a person has not been detected.

The first tests were executed with the maximum resolution of 800x600 pixels. These were very successful and most of the times, all persons were classified correctly. Some of the example classifications are presented in Fig. 5.5. These show correct classifications in the four left pictures and a misclassification and a not classified person in the rightmost picture. The misclassification in this image is most likely due to the very bright backlight from the windows. The missing person in the front is most probably not classified, since he was standing too close to the camera. This problem will be addressed below in this section. The overall result of the classification was that 86 percent of all persons in the environment were recognized and also located correctly. Reasons for failed classifications were mostly due to the fact that either the person was standing too close (see rightmost image Fig. 5.5) or too far away from the camera. Additionally problems could be caused by motion blur from the movement of single persons and the bad lighting conditions in some scenarios. Although the detection rate with this resolution is quite good, the processing time for one picture takes at least 1500 milliseconds on a 2.4 Ghz processor. On a robotic platform this is a quite long time period.

Therefore, the following part evaluates how the reduction of the resolution can also

reduce the processing time used for the algorithm. An effect that should be considered, however, is that with a lower resolution the HOG descriptor will more probably fail to detect a person in the environment. Therefore, as a second measurement it is evaluated in which approximate maximum distance a person can still be detected. In addition to that, all correct classifications in all image are counted. This is though only a guideline, since this is again highly dependent on the fact in which distances the persons are standing in the environment on the pictures. The maximum number of persons in all 18 example pictures is 51. For the test it has been decided to use the resolution 800x600 as the maximum size and then try 640x480, 400x300, 320x240, 200x150 and 160x120. The resulting processing times are as follows:

Image resolution	Average processing time	Correct classifications	Approximate maximum distance
160x120	5 ms	1	1 m
200x150	30 ms	3	1 m
320x240	190 ms	20	2 m
400x300	300 ms	34	3 m
640x480	1000 ms	37	4 m
800x600	1500 ms	40	5 m

Resulting from these tests, it has been decided to use a resolution of 400x300. In most cases, the persons in the robotic environment are located in an approximate distance of 3 meters, the “correct classifications“ for this resolution are still good and the processing time for one picture seems to be reasonable in a robotic application. However, the processing time varies from processor to processor and should again only be a guideline for comparing the different resolutions with each other.

Figure 5.6: Different extreme positions that could now be detected, due to the black border.

In the implementation part in section 4.2.2, a method was presented that allows the HOG algorithm to recognize persons also in very extreme locations. This was done by simply adding a border to the outer regions of the image. A validation of this was done in

some tests that have been performed with this implementation on the previous dataset. Doing so, it was possible to detect persons in the near vicinity of the robot – also up to extreme angles. Fig. 5.6 shows some of the results. However, due to the reduced resolution of 400x300 the actual recognition of persons in the depth was worsened, such that in the right most image of 5.6 one person for example is not classified at all.

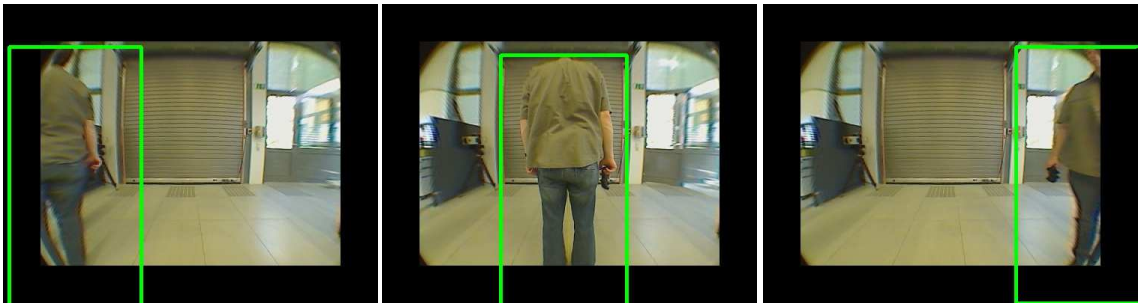

Figure 5.7: In the image, a person can still be detected in a distance of 500 millimeters away from the robot and angles of 67,5 degrees to each side.

It has been figured out, that with this method, a person can still be recognized by the HOG detector, if it is standing in a distance of 500 millimeters (see center image Fig. 5.7). The maximum distance was approximately 3 meters. For the angular resolution this test was repeated and a person was moving as long to the left and right till it could not be detected anymore. The maximum degree up to which a person in 1 meter distance could still be detected is approximately 67 degrees according to the measurements. However, all these results were taken in a more or less ideal environment and also depend on a lot of other factors. It is important how well the person can be detected in general, in which distance the person is standing in this angular position and which other outer environmental properties are affecting the recognition.

5.2.2 Test of the mean shift color tracking algorithm

In this section, the mean shift color tracking algorithm is evaluated. The lighting conditions during the tests were stable and the robot was placed at a fixed location, with the camera mounted on top. The only moving objects in the surrounding were the persons that should be tracked. The algorithm is based on the detection of certain colors in the environment. Therefore, the test persons were wearing each different clothes during the test. These were a white t-shirt with white trousers, a high visibility vest with white trousers and, finally, a green shirt with blue trousers. The color histogram was generated from a certain region of the shirt of each person. This has the advantage that this color is in most cases more unique than, for example, the color of the trousers in the environment.

Fig. 5.8 shows two examples where a person or a dummy person were successful and not successful tracked in the environment based on the color of their shirts. In the

Figure 5.8: Mean shift color tracking algorithms successfully tracking a dummy person (left images) and failed to track a person (right images).

pictures, the red rectangle marks the starting position where the person was located in the beginning, and the white rectangle marks that position where the person is currently located based on the results of the method. On the right side of each photo, also the backprojection images of each window are shown. This shows, how frequent certain colors of the histogram occur in the environment. In the first case, where a dummy person with a high visibility vest is sitting in the scene, the colors of the vest occur practically not elsewhere in the scene. This was the reason why the tracking with these clothes was most successful and did not fail in this scenario for 10 test runs. However, in most real life situations these clothes are not very likely worn by persons. Clothes that are worn by the person in the second picture are more representative for daily clothes. Here, the green shirt is tracked during the procedure. This test was, however, not so successful, and failed in 40 percent of the tests. The reason is, that a lot more regions in the image, e.g. the gate in the back, had a similar coloring like the actual tracked item. Therefore, the mean shift algorithm more often drifted to these other positions.

In the third test case, the dummy person was tracked again, now wearing a white t-shirt. However, in this test instead of the actual person, the robot was turned away from the person in the direction of the whiteboard (see Fig. 5.9). In this scenario, the weakness of the color tracking algorithm can be shown again. In the backprojection image one can see on the right side that a lot of regions have a similar color than the one that has been recorded for the clothes of the dummy. Therefore, the mean shift algorithm was also very likely to drift to another location that has a similar coloring than the one of the shirt.

To sum up, it can be said that the actual color tracking algorithm is depending very much on the color of the clothes a person is wearing while he/she is being tracked.

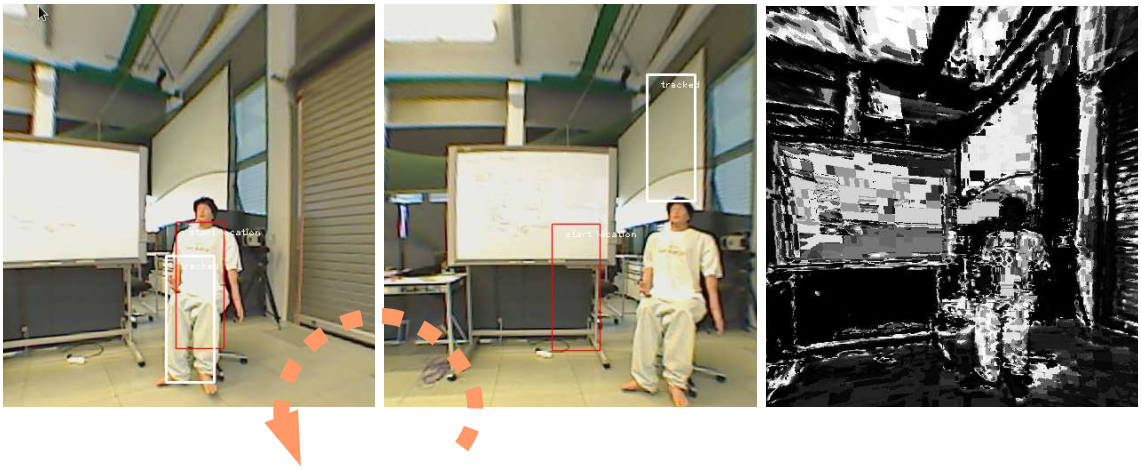

Figure 5.9: The left image shows the initialization scene for the color histogram. In the center and image is depicted demonstrating that the dummy person was lost by the tracking. Here, the initial tracking position is the white rectangle and the currently tracked position is marked by a red rectangle. The rightmost image shows how frequent the colors of the histogram occur in the environment.

5.2.3 Test of the template matching tracking algorithm

In this section some tests that have been performed with the template matching algorithm are presented. The tests were mainly executed in a static environment with fixed lighting conditions and a static background scenery. The main concept of the algorithm is that it takes a template image and then compares it at each position of another image with the latter image. However, it does not rotate or scale the image, such that the corresponding object has to have a similar scale and orientation in the picture under consideration. Therefore, in the case of tracking a person, the region in between the chest and the knees of a person is taken as a template. This has the advantage that this region is often in the same upright orientation – especially when the person is walking in front of a robot. Although the distance of the persons to the robot might differ and therefore their shapes are scaled, the algorithm showed a certain scale invariance.

For the testing environment, again, different backgrounds were chosen. The robot and the camera were located at a fixed position. First, the template was initialized by the back of a person (see left image, white rectangle Fig. 5.10). Then the person was moving around, mostly with the back turned to the robot, such as if the robot would follow the person (see right pictures in Fig. 5.10). Since the person moved mostly perpendicular to the robot, the template matching method worked best and did not loose the person in 10 test runs [3]. Also, if the person moved less than half a meter to the front or the back the

[3]The scaling and the orientation of the template picture in comparison to the real image did not differ much in all of these pictures. Therefore, the template matching works best here.

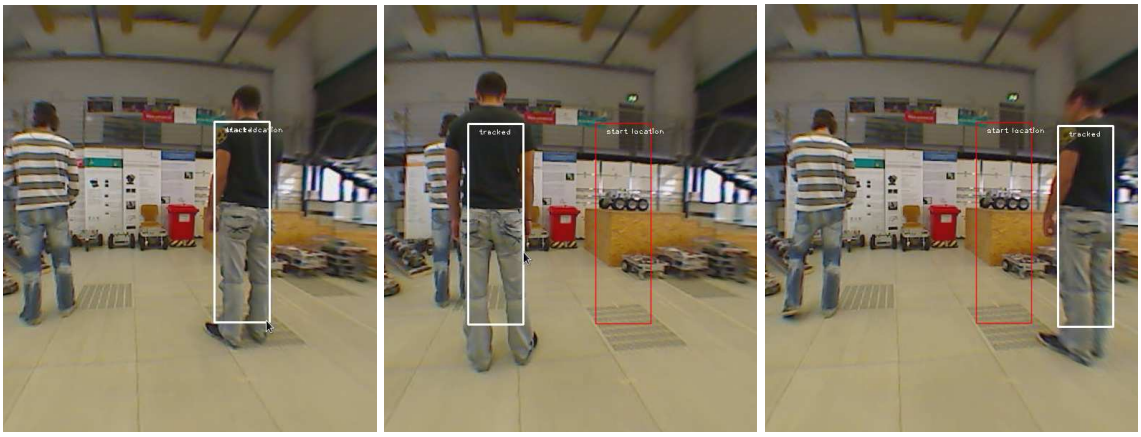

Figure 5.10: Template matching used for tracking a person in a picture. The red rectangle marks initial region (left picture), the white rectangle is the region where the template matching algorithm has detected the template.

template matching algorithm would still work properly.

The algorithm, however, is very sensitive to the actual surrounding. In Fig. 5.10, a very cluttered background is presented, which influenced the actual results a lot and caused the method to loose the person as soon as this one was leaving a certain range. In a more homogeneous environment the template matching will work a lot better. To prove this behavior, another test was performed in a different environment (see Fig. 5.11). In this scenario, less other objects in the environment that were similar to the persons back were available. Therefore, the tracking in this particular scene was very stable and it even worked to detect the person with the same template at the maximum distance of three meters. In the other scenes, the template matching algorithm would already have mismatched and taken another region due to the scale. It was also tested up to which angular position the template could still be correctly identified. This was ascertained to be 55 degrees. It also proved that the actual distortions and the reduction of the resolution in the corner regions of the camera were not affecting the current method too much.

The processing time of the template matching algorithm was also evaluated. It is almost linear to the amount of pixels that have to be processed. Such by reducing the resolution from 800x600 to 400x300, only a forth of the processing time was needed to analyze the whole image. However, the detection though also degrades with each reduction in the resolution. It was decided to choose a similar resolution to the one that has been chosen for the HOG algorithm (400x300 pixel). The resulting processing time is 40 milliseconds on a 2.2 Ghz processor.

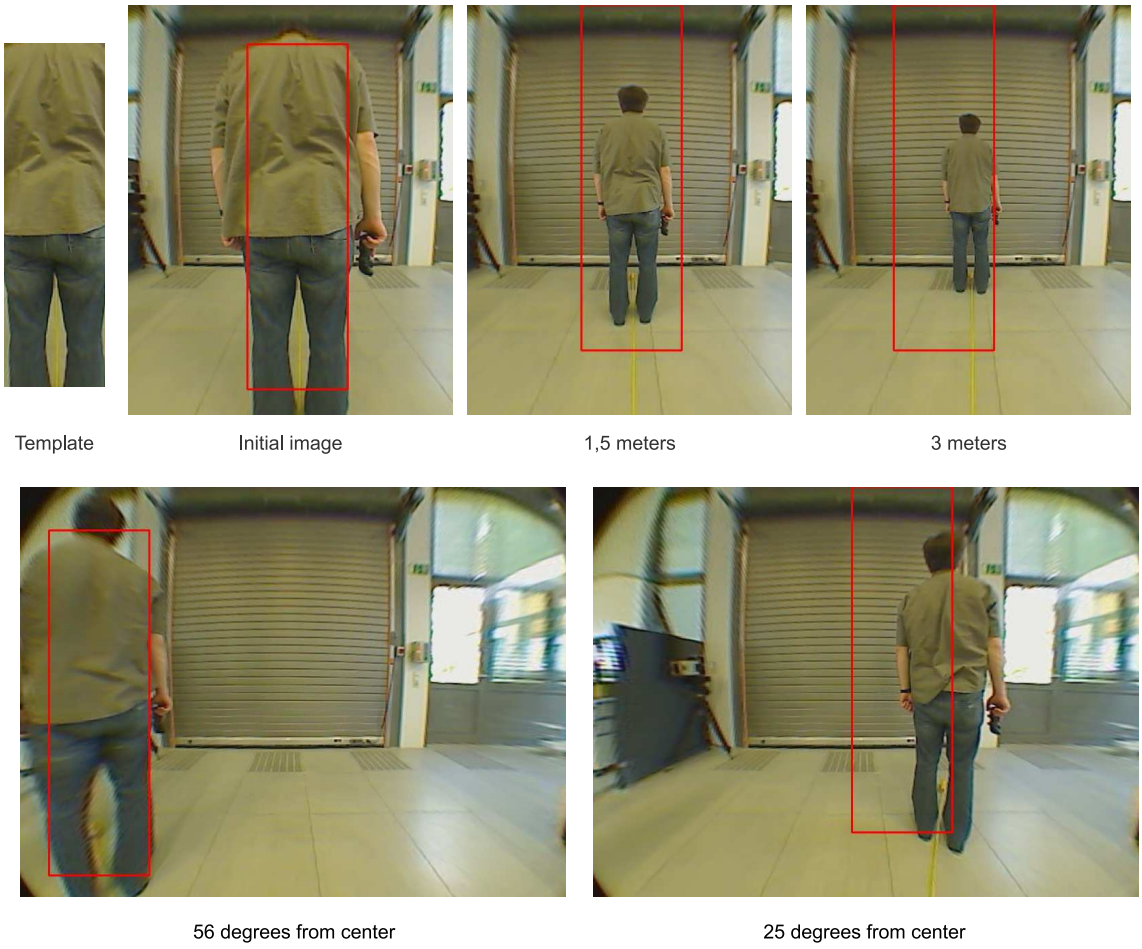

Figure 5.11: Template matching for detecting persons in different distances and angular positions in a more homogeneous environment.

5.3 Detection in a dynamic environment, while following a person

In the following sections, tests of implementations are discussed that utilize the laser scanner and the camera to follow a person in the environment. The "following procedure" that is used for pursuing a person is a simple reactive approach. This means, that the sensor values are directly passed to the actuation unit and define the rotational as well as the translational speed of the platform. The rotational speed is defined by the angle in which a person has been detected and the translational speed is a value calculated from the distance value to that person.

To test how the different approaches perform in the environment, each test run was recorded on video and was evaluated afterwards. The following sections will show some of the features and give an overview of where eventual weaknesses are and under which environmental circumstances they emerge.

This section can mainly be subdivided into three parts. In the first part tests

are described that use the single laser scanner approach to locate and track a person in the environment. In the second tests then the approach is evaluated that utilizes the Histogram of Oriented Gradients in combination with the template matching algorithm.[4] In the last part tests are described that were performed with the approach combining the laser scanner and the HOG detector.

5.3.1 Dynamic environment while following a person with a laser scanner

In this section, tests are described that have been performed with the laser scanner based approach while following a person. The main difficulty in this scenario are the changing background sceneries. In each frame, due to the movement of the robot different objects and persons in the environment may appear in the laser scanner range and may cause misclassifications and therefore loose the tracking of a person.

Figure 5.12: Different testing environments for the laser scanner based approach.

The environments in which the robot was tested were a long hallway with different doors at the sides, a wide area of free space and the same area cluttered with various items in the environment. Fig. 5.12 shows the different scenarios where the tests were performed. The first property that was tested is how often the tracking algorithm fails to track a person in a normal following scenario.

Therefore, the leading person walked in front of the robot in the hallway and in the free wide area. He made very fast direction changes to irritate the robot's tracking algorithm. This experiment was performed with different persons, wearing different trousers. The result was, that the tracking – as there are no obstacles in the environment – is robust and could only be irritated if the person was walking out of the laser scanner range.

In the following, different properties of the tracking method that have been tested are described. These are: robustness to other objects besides the robot, in the near vicinity of the tracked person, short temporal occlusions, objects crossing in between the laser scanner and the tracked person and crossing persons walking in between the laser scanner and the person.

[4]The mean shift algorithm did not prove to be very stable in the previous tests (see section 5.2.2), such that it was not considered for an implementation of the following algorithm.

Therefore, objects with different sizes and shapes were inserted into the environment. These were in this scenario: chairs, tables, walls, trashbins, different sized boxes, a pillar, a dummy person and other objects standing at the outer regions of the test fields (see right image in Fig. 5.12). The robot was again initialized to track the person in the environment. The person then walked in between the different objects, with a varying distance to the objects of sometimes even less than 15 centimeters. This was done for at least 5 minutes.

In this walk in the test field, the method was confused mainly by two objects, the dummy person sitting in the chair (two times) at the front and a box standing at the front right (once). Both times, however, the person was walking towards this certain object till he was quite near to it and then turned very quickly into another direction. Another reason was that the person was moving out of the scope that has been previously set to three meters, and therefore the dummy for example was tracked afterwards.

Figure 5.13: Different tests with moving objects in between the laser scanner and the tracked person.

In the next test, it was evaluated how temporal occlusions of the legs are affecting the tracking of a person. Therefore, different objects were moved perpendicularly to the moving direction, such that the persons' legs were occluded for a short period of time (see left images Fig. 5.13). For short periods of 1 second this did not affect the actual methods. However, the following could be observed: the longer the time period and the bigger the object that occluded the legs was, the more probable it was that the actual tracking algorithm would fail.

The next property that was then tested with the robot is how a person walking in between the laser scanner and the person being tracked is influencing the actual tracking. Therefore, different tests were performed. The robot was in most cases moving in a straight direction when another person was walking in between the robot and the leading person (see right image in Fig. 5.13). This test was successful in 75 percent, such that the robot did not follow the other person. In 25 percent, however, the robot failed and was following the other person. This was most probably due to the fact that the robot was turning around in these situations, while the persons were crossing the line. This way, the tracking was more likely to take the crossing person for the person to follow.

Also, different types of trousers affected the amount of how well the crosswalk could be compensated or not. If the reflectivity of the trousers the leading person was wearing was very high, the possibility to loose the robot in a crosswalk situation was less likely. And also like in the test before, it depended on how long the person was actually occluded by the other one. The longer this period was and the more of the person was occluded, the more likely it was that the tracking algorithm failed.

5.3.2 Combination of the HOG detection and the template matching algorithm to follow a person

In the implementation part in section 4.2.4, a combination of the HOG and the template matching algorithm was presented. The HOG algorithm has a quite long processing time (see also section 5.2.1). As a consequence, the resolution was already reduced to 400x300 and a border was attached as mentioned before to allow better detection of persons in the near vicinity. However, the detection is still very slow for a single HOG detection and takes 500 milliseconds (with borders) and is also increasing with every other process running in the back. Therefore, it was also considered in the implementation chapter to use a fast tracking method to find corresponding regions in the image in a fast way. The HOG detection therefor should only be executed each 500 milliseconds. In the meantime, pictures that are acquired from the camera should be evaluated using the template matching algorithm, which allows to track the position in a fast way.

This implementation is tested in the following. The HOG algorithm processes the image and outputs a region in which a person was detected. Since the region that is provided by the HOG algorithm is quite large, it is shrinked such that primarily only the area, where the persons' body is visible, is passed to the template matching algorithm. The latter then detects the position in the image, which corresponds at best to this template.

The tests that have been performed with this implementation were using the x position of the detected template to specify the direction the robot should drive to, which is approximately the angle in which the person is standing. So if the person was detected on the left side of the window, the algorithm passed a rotational speed value to drive slightly to the left. If the person was detected in the right area, it would head to the other direction.

This scenario was tested in the open area (see right image in Fig. 5.12), where no obstacles are in the environment. The person moving in front did not wear any extra marker or special clothes and was only detected by the HOG algorithm in the images. Some of the results are presented in Fig. 5.14: It shows the screen of the detection algorithm while the robot was following a person in the environment. In the lower images, the detected person is marked with a green rectangle around him. This region is then shrinked and used as a template for the template matching algorithm. The resulting image is shown

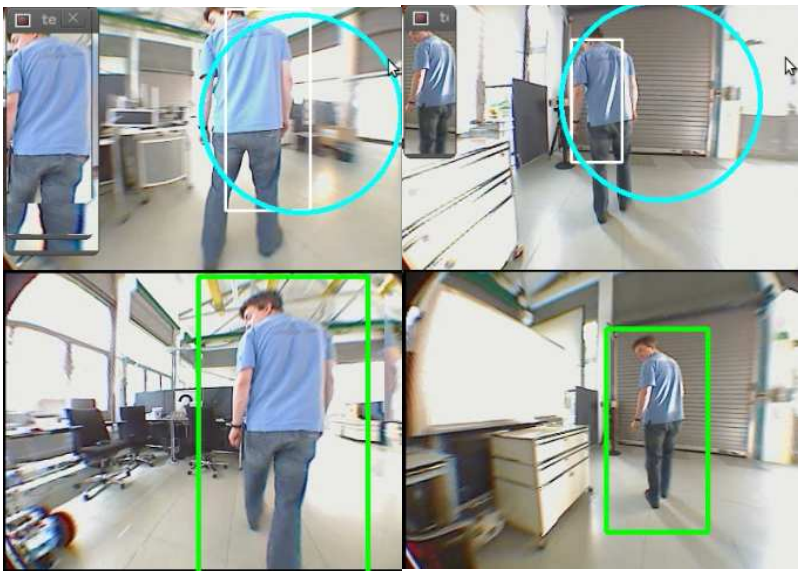

Figure 5.14: Testing interface of the HOG and the template matching approach. The upper images show the template matching, the lower images show the HOG classification.

on each image in the left upper corner. In the upper image, the region detected by the template matching algorithm is marked with a white rectangle.

Following a person with this technique worked quite well, but had in some cases difficulties to distinguish between other objects and the person: Sometimes either the HOG algorithm detected another object as possible person in one cycle or the template matching algorithm failed to find the corresponding person to the template that has been previously saved. Nevertheless, it was possible to lead the robot from one side of the area to the other side, turn around and move backwards. One of the major problems, however, is still the processing time. Due to this, the actual detection was often lagging a second behind such that the robot was not able to react fast on any movement. Therefore, in the end this algorithm can be used in a wide area, like it has been done in this scenario, with a very slow driving speed, but is not able to follow a person in real time.

5.3.3 Combination of laser scanner based and camera based methods

This final test shows a combination of the laser scanner based tracking and the camera based detection of persons. The implementation was already presented earlier in this work (see section 4.3). Now it should be evaluated how this would work in a real "following scenario".

The approach that was implemented, is mainly based on the laser scanner tracking of a person's legs, which is then verified (less than 150 milliseconds frequency, due to smaller image region) with the HOG algorithm, whether the tracked leg really corresponds to a

person. Therefore, the position in which a person has been detected with the leg detection projected into the camera picture. This position defines the center of a region of interest that should be classified with the HOG algorithm (see Fig. 5.15). The HOG detector then classifies the region accordingly and returns whether it contains a person or not. This information is then shown in Fig. 5.15 as a green rectangle.

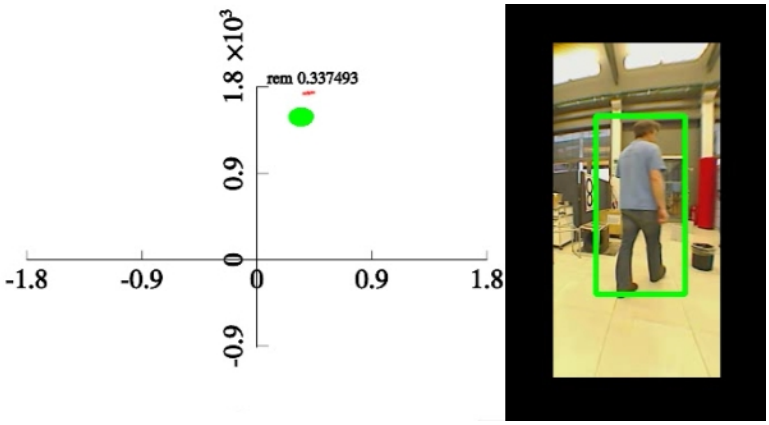

Figure 5.15: Left: image of the laser scanner leg detection. Right: region of interest extracted from the camera picture with the HOG classification (green rectangle).

The method has also been tested in the cluttered environment (see right image in Fig. 5.12) and performed quite well. The person was in the region of interest visible on the screen all the time. The HOG algorithm though identified a person in only 86 percent of the examined areas. Especially when walking directly in front of the robot and with slight turns to the left or right the classification was performing very well and updated almost every 150 milliseconds. The remaining 14 percent that the person has not been detected are due to the following reasons: When the person is walking very close (less than 500 millimeters) in front of the robot, the HOG is not able to classify any region. This was especially then the case when the person was walking through the cluttered environment. In addition to that, another problem was occurring when the person directly turned on spot to another direction. In this case, in addition, to the very low resolution in the corner regions of the image due to the wide angle lens, also motion blur occurred in the image, which reduced the overall detection rate.

However, it has been proven by this test, that the camera can be used in combination with the laser scanner. Thereby, it is possible to verify whether the segments that have been classified by the laser scanner as potential legs really belong to a person. As a future development it would also be interesting to use the template matching algorithm to verify whether the detected person is really corresponding to the traced one, which was detected in the first run.

Chapter 6

CONCLUSIONS

6.1 Summary

In this master thesis, an approach for fast and reliable tracking of a person via a mobile robotic platform was presented.

First, different person tracking methods, which are applicable for the specific scenario of a mobile robotic platform were investigated. Based on these results, the laser scanner and the camera were chosen for the identification of persons in the surrounding. The laser scanner offers fast and precise depth measurements, while the camera was chosen, since a lot of methods exist that are able to detect a person more reliably in the surrounding than a laser scanner.

The developed tracking algorithm of the laser scanner is mainly based on the classification of leg features, taking their shapes and sizes into account. For the tracking of a leg in two consecutive scans, the algorithm searches in the near vicinity of the previously detected leg to find a similar one with a middle remission value that fits best. The remission values though define the reflectivity of an object and are mainly dependent on the detected material. Therefore, they can be used as an approximate measure to associate legs in subsequent scans.

The approach for the camera is based on a state of the art person detector named Histogram of Oriented Gradients. It is used in a lot of applications to detect persons and is also able to correctly classify a person from the back. The algorithm scans the whole image at different scales for a set of a certain gradient distribution that has been trained with a supervised learning technique. However, the algorithm requires that a person can be viewed in full pose in the camera image. Therefore, the camera was equipped with a wide angle lens. Since the algorithm is quite slow computing the results, two additional, fast tracking algorithms were evaluated for the use in this scenario. These were the mean shift color tracking and template matching algorithm. The latter was finally chosen to track a person while the slow and computationally expensive HOG method is processing the image in the background.

Both approaches, the laser scanner classification and tracking and the HOG method in combination with the tracking algorithm, were evaluated in different test scenarios. The main outcome was that, on the one hand, the laser scanner was fast and precise in tracking persons' legs but in some particular situations missed the real person and was confused with other legs or objects. However, the tracking worked well and could also be used for

following a person in a cluttered environment. On the other hand, the camera algorithm was evaluated to be more reliable in detecting persons in the environment, but it was not as fast and precise in locating a person in the distance as the laser scanner.

Therefore, a third and final concept was presented and implemented, combining both approaches. Doing so it was possible to track a person in a fast manner based on the laser scanner leg detection, while in the background, another process was verifying that these legs really belong to a person using the HOG detection method.

6.2 Future work

For the laser scanner, different approaches could be evaluated and compared with each other. Such, for example, the classification could be done with a supervised learning technique to generate an even more reliable classifier for the legs. The movement of the person could also be modeled. Thus, it would be possible to make an assumption in which direction the person is walking next. However, this implies a complex model, since both the sensors and the person are moving.

High quality cameras could be tested, with wide angle lenses that have less distortions. It could also be evaluated how combined, multiple cameras would improve the detection rate of the HOG detector. In addition, it could be investigated if it is possible to identify a person from the back. A possible approach would be, for example, the previously presented template matching algorithm or a local color histogram of the clothes.

For the fusion of both sensors, different approaches could be tested. For example the results from the classification of the HOG method and the laser scanner could be fused with the use of a particle filter.

A lot of the proposals given in this section are computationally expensive and require a faster computer than the one being available at the time of this master thesis. However, since hardware is improving fast, a soon realization might be possible.

BIBLIOGRAPHY

[AM10] Kai Arras and Oscar Mozos. Editorial. *International Journal of Social Robotics*, 2:1–2, 2010.

[AMB07] K.O. Arras, O.M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *IEEE International Conference on Robotics and Automation*, pages 3402–3407, Roma, 2007.

[ARS08] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, 2008.

[BCF+98] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proceedings of the National Conference on Artificial Intelligence*, pages 11–18, Madison, Wisconsin, United States, 1998.

[BH05] N. Bellotto and H. Hu. Multisensor integration for human-robot interaction. *The IEEE Journal of Intelligent Cybernetic Systems*, 1, 2005.

[BK08] Dr. Gary Rost Bradski and Adrian Kaehler. *Learning opencv, 1st edition.* O'Reilly Media, Inc., 2008.

[COS09] A. Carballo, A. Ohya, and Yuta S. Multiple people detection from a mobile robot using double layered laser range finders. In *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[CZZS06] J. Cui, H. Zha, Z. Zhao, and R. Shibasaki. Laser-based interacting people tracking using multi-level observations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1799–1804, Beijing, 2006.

[Dal] Navneet Dalal. INRIA Object Localization Toolkit (OLT). http://www.navneetdalal.com/software/.

[DGHW00] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185, 2000.

[DT05] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, San Diego, CA, USA, 2005.

[DT06] Navneet Dalal and Bill Triggs. Object detection using histograms of oriented gradients. In *European Conference on Computer Vision (ECCV)*, Graz, Austria, May 2006. Workshop.

[FB81] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FH75] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:32–40, 1975.

[FND03] Terrence W. Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4):143–166, 2003.

[Fraa] Fraunhofer Institut - IAIS. Fraunhofer Autonomous Intelligent Robotics Library – FAIRlib. http://sourceforge.net/projects/openvolksbot/.

[Frab] Fraunhofer Institut - IAIS. Fraunhofer institut - intelligent analysis and information systems. http://www.iais.fraunhofer.com/.

[Frac] Fraunhofer Institut - IAIS. Project description of the project profibot. http://www.iais.fraunhofer.de/profibot.html.

[Frad] Fraunhofer Institut - IPA. Fraunhofer-institut für produktionstechnik und automatisierung ipa. http://www.ipa.fraunhofer.de/.

[Frae] Fraunhofer Institute for Manufacturing Engineering and Automation IPA . Museum robots. http://www.care-o-bot.de/english/MuseumRobots.php.

[Fraf] Fraunhofer Institute for Manufacturing Engineering and Automation IPA and Opel. Opel exposition robots. http://www.care-o-bot.de/english/OpelRobots.php.

[FSA99] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society for Artificial Intelligence*, 14:771–780, 1999.

[FZ99] S. Feyrer and A. Zell. Detection, tracking, and pursuit of humans with an autonomous mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 864–869, Kyongju , South Korea, 1999.

[HL01] E. Hjelmås and B.K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–274, 2001.

[ICR09] 2009 ieee international conference on robotics and automation - icra 2009 - call for participation. *Robotics Automation Magazine, IEEE*, 15(4), dec. 2009.

[JS10] Boyoon Jung and Gaurav Sukhatme. Real-time motion tracking from a mobile robot. *International Journal of Social Robotics*, 2:63–78, 2010.

[KCSS03] H. Kruppa, M. Castrillon-Santana, and B. Schiele. Fast and robust face finding via local context. In *Joint IEEE Internacional Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 157–164, 2003.

[KLF+02] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lomker, GA Fink, and G. Sagerer. Person tracking with a mobile robot based on multi-modal anchoring. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN)*, pages 423–429, Berlin, Germany, 2002.

[Kra06] K.F. Kraiss. *Advanced man-machine interaction: fundamentals and implementation*. Springer Verlag, 2006.

[KS] Hannes Kruppa and Bernt Schiele. Haar-based detectors for pedestrian detection. http://alereimondo.no-ip.org/OpenCV/35.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[MB94] D. Murray and A. Basu. Motion Tracking with an Active Camera. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 16, pages 449–459, 1994.

[MBN04] A. Mendes, L.C. Bento, and U. Nunes. Multi-target Detection and Tracking with a Laserscanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 796 – 801, 2004.

[Met] MetraLabs. Scitos a5. http://www.metralabs.com/.

[MKH09] O.M. Mozos, R. Kurazume, and T. Hasegawa. Multi-layer people detection using 2d range data. In *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[MKH10] Oscar Mozos, Ryo Kurazume, and Tsutomu Hasegawa. Multi-part people detection using 2d range data. *International Journal of Social Robotics*, 2:31–40, 2010.

[Ope] OpenCV. Open computer vision library. Website. http://sourceforge.net/projects/opencvlibrary/.

[pd] INRIA person dataset. Dataset of upright people in images and video. http://pascal.inrialpes.fr/data/human/.

[PSE00] E. Prassler, J. Scholz, and A. Elfes. Tracking multiple moving objects for real-time robot navigation. *Autonomous Robots*, 8(2):105–116, 2000.

[RC09] D.L. Rizzini and S. Caselli. Experimental Evaluation of a People Detection Algorithm in Dynamic Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[SAM+09] B. Schiele, M. Andriluka, N. Majer, S. Roth, and C. Wojek. Visual People Detection–Different Models, Comparison and Discussion. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, Kobe, Japan, 2009.

[SATS10] Luciano Spinello, Kai Oliver Arras, Rudolph Triebel, and Roland Siegwart. A Layered Approach to People Detection in 3D Range Data. In *AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, USA, 2010.

[SM99] C. Schaeffer and T. May. Care-o-bot-a system for assisting elderly or disabled persons in home environments. In *In Proceedings of the Association for the Advancement of Assistive Technology in Europe*, Düsseldorf, 1999.

[SM09] J. Satake and J. Miura. Robust Stereo-Based Person Detection and Tracking for a Person Following Robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[Spo] SportsVu. Tracking technology for precise and insightful sports information. http://www.stats.com/sportvu/football.asp.

[SS08] L. Spinello and R. Siegwart. Human detection using multimodal and multidimensional features. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3264 – 3269, Pasadena, CA, 2008.

[Sur] Dr.-Ing. Hartmut Surman. Project description volksbot. http://www.volksbot.de/.

[TBB+99] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: a second-generation museum tour-guide robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 1999 – 2005, Detroit, MI , USA, 1999.

[TCD05] A. Treptow, G. Cielniak, and T. Duckett. Active people recognition using thermal and grey images on a mobile security robot. In *International Conference on Intelligent Robots and Systems (IROS).*, pages 2103 – 2108, Edmonto, Alberta, Canada, 2005.

[Ueb09] Christoph Ueberschaer. Personenerkennung und personenidentifikation mittels bluetooth bei autonomen robotersystermen. Diploma thesis, Hochschule Bonn-Rhein-Sieg, 2009.

[VJ01] P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, volume 1, 2001.

[VJS05] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[VMRL03] J.M. Valin, F. Michaud, J. Rouat, and D. Létourneau. Robust sound source localization using a microphone array on a mobile robot. In *Proceedings International Conference on Intelligent Robots and Systems*, volume 2, pages 1228–1233, 2003.

[WBoG02] T. Wilhelm, HJ B öhme, and HM Gross. Sensor fusion for vision and sonar based people tracking on a mobile service robot. In *Proceedings of the International Workshop on Dynamic Perception*, pages 315–320, 2002.

[WN05] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Tenth IEEE International Conference on Computer Vision (ICCV 2005)*, pages 90–97, 2005.

[XPC+05] J. Xavier, M. Pacheco, D. Castro, A. Ruano, and U. Nunes. Fast line, arc/-circle and leg detection from laser scan data in a player driver. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3930–3935, 2005.

[ZCPR03] W. Zhao, R. Chellappa, PJ Phillips, and A. Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, 35(4):399–458, 2003.

[ZK07] Z. Zivkovic and B. Kröse. Part based people detection using 2D range data and images. In *IEEE Int. Conf. on Intell. Rob. and Sys.(IROS)*, pages 214 – 219, San Diego, CA, 2007.