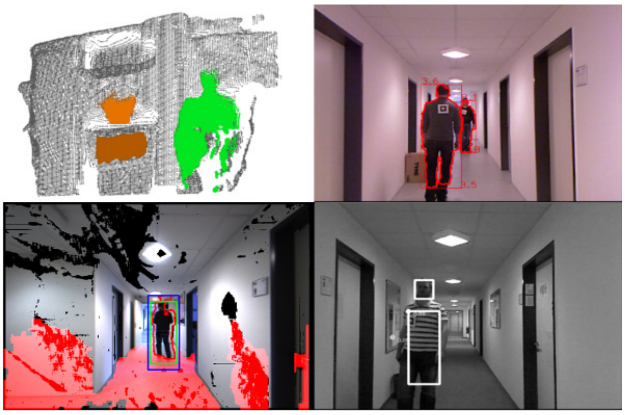
3D Human Detection and Tracking on a Mobile Platform for Situation Awareness

Niklas Beuter

Universität Bielefeld

3D Human Detection and Tracking on a Mobile Platform for Situation Awareness



Dissertation zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

der Technischen Fakultät der Universität Bielefeld

vorgelegt von

Niklas Beuter

vorgelegt am 06. Juli 2011

Gutachter:

Prof. Dr.-Ing Franz Kummert
Prof. Dr. rer. nat. Christian Wöhler

Prüfungsausschuss:

Prof. Dr. Barbara Hammer
Prof. Dr.-Ing Franz Kummert
Prof. Dr. rer. nat. Christian Wöhler
Dr.-Ing Hendrik Koesling

gedruckt auf alterungsbeständigem Papier nach ISO 9706

A few words

Many people supported me in diverse areas, while writing my thesis. Here, I want to write a few words to thank all these people. First of all, I want to thank my supervisors Franz Kummert and Christian Wöhler, who always had an open door for my requests and who took their time to support the work on my thesis.
The inspiring work with my colleagues at the Bielefeld University raised many ideas, which resulted in several publications. It was a great time with many opportunities and fruitful collaborations. I want to thank all my colleagues for the heartily athmosphere and the conspirative work. I will miss the good and mostly funny discussions, where not only research was in the focus.
Most notably I want to thank my wife, who backed me up wherever it was necessary and who stands for the water of my fountain. Her happiness and her wonderful lust for life are my inspiration and accordingly, the most important part in my life to realise the final goal of getting the PhD.

Abstract

The vision of robots supporting the human in daily life encouraged research in the area of mobile robots in the recent past. The robots are meant to share the same environment and they are supposed to deal with the same requirements like the human in order to be able to assist the human in his tasks. But, the dynamic and highly complex human environment makes it affordable to implement algorithms, which enable the robot to deal with the arising requirements. Thereby, such algorithms are based on sensing and interpretation of the environment. Here, the following thesis applies by achieving a solid basement for the robot's situation awareness. Situation awareness can be divided into four categories, whereas the first three categories sense and interpret the environment and the fourth predicts the gathered knowledge into the future in order to adapt the aspired actions. The thesis provides a solution to the first three categories, which implement the perceptual part of situation awareness.

The first category deals with the sensing of the environment. Here, a complete scene analysis by building an *articulated scene model* by observation of the *Vista space* is proposed. The model inherits different abstract scene parts, which are the static background, movable objects like e.g. chairs and moving objects like humans, which are all revealed by a single model building process. The second category deals with a temporal linking of information, which is especially difficult on a moving platform. In addition to a map, the most important information for a mobile robot is the information of positions and walking paths of present humans. Utilising the dynamic movements of humans the robot is able to calculate a safe path through the environment. Here, a *dynamic human detection and tracking system* is introduced, which is able to create temporal links between human occurrences even in the presence of challenging ego motion and scene changes. The third category of situation awareness is implemented through a *top-down visual attention system*, which directs the focus of attention onto desired objects, humans or locations. As the main purpose of a mobile robot is the interaction with the human, it is proposed to use a human model in combination with the top-down directed visual search. The model represents more precise the diverse appearance of the human. This way, the robot is able to recover aspired humans or interaction partners, if they were absent for a short time.

For each category experiments are conducted to show the performance of my solutions. The results show that each category implementation provides solid and stable information, which support the robot in achieving a broad situation awareness.

Contents

1 Motivation

During the last years, household and industrial robotics have been attracting notice to researchers and a rising number of consumers in order to support the human in his/her daily life. But, the human-friendly environments make great demands on technical systems. Especially robots mostly rely on very specific algorithms, dealing only with a rough picture of their environment. If robots are meant to provide human-like capabilities in a human-like environment, cognitive inspired algorithms are required. Humans have developed outstanding capabilities in perceiving and understanding highly complex and dynamic environments and situations. Their perceptual system and cognitive competences permit the ability to deal with daily life and the achievement of diverse tasks. One such important human capability is *situation awareness*.

> *"Situation awareness is the continuous extraction of environmental information along with integration of this information with previous knowledge to form a coherent mental picture, and the end use of that mental picture in directing further perception and anticipating future need"*
> Dominguez, Vidulich, Vogel, & McMillan, 1994 [56]

In short, the quotation emphasizes the fact that situation awareness is an important aspect for the human being to realize what is happening around. Thereby, it divides situation awareness in four main categories. First, one has to extract information from the environment (*"continuous extraction of environmental information"*). The human incorporates this analysis of the surrounding already in his/her early years in order to get useful and important information. Children e.g. observe the actions of their parents to learn which parts in a room can be moved or where they can open a door. Second, a temporal link has to be established in order to extract more information than through simple observation (*"integration of this information with previous knowledge"*). For instance, it is a daily subject for humans to detect and track other moving entities in order to prevent collisions or accidents. An employee on the way to work e.g. has to drive safely through the traffic without colliding with other road users and at work he/she has to navigate through the office and not collide with any other person. Thereby, the human uses his/her capabilities to segment other entities and to establish a temporal link between each occurrence. Predicting the recognised movement in the future the human can safely avoid collisions. The third category directs the focus of attention on specific areas in order to extract more detailed information from these areas (*"use of that mental picture in directing further perception"*). Thereby, attention is a process of restricting the incoming information to the most relevant information. Humans use this ability unconsciously as well as consciously to direct their focus of attention to specific areas being of special interest or have strong attractiveness. In this way the processing can

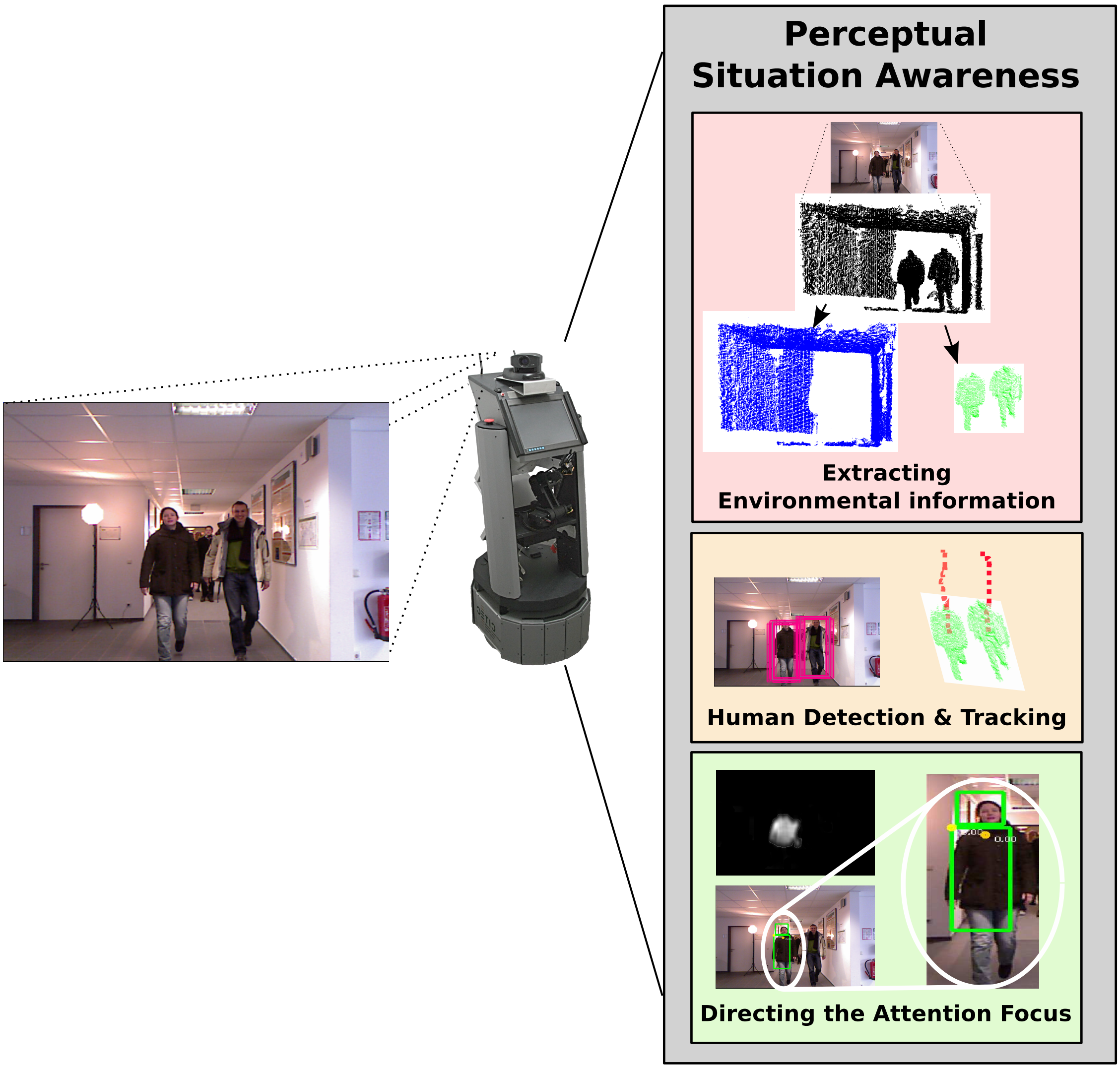

Perceptual Situation Awareness
Extracting Environmental information
Human Detection & Tracking
Directing the Attention Focus

Figure 1.1: *Situation awareness for a mobile robot.* The following thesis proposes algorithms which implement different parts of situation awareness for a mobile robot. Here, robot BIRON analyses and segments the incoming information in order to extract scene information as well as humans present in the scene. Tracking of all humans is applied, which reveals the current and former positions of all humans in the scene even in the presence of ego motion of the robot. To focus on one interaction partner biologically inspired visual attention is implemented enabling the robot to keep the focus on one entity.

be reduced and restricted to the most important information. Using the information from the environment, the temporal links between entities and directing the focus of attention, it is possible to predict future actions, which enable the fourth category of the quotation ("*anticipating future need*"). The prediction should adapt the own actions best to the actions from others and the sensed environment. Summing up, the first three categories form the important perceptual parts of situation awareness enabling the human to sense the world around (see Fig. (1.1)), whereas the last category is important for future planning strategies.

The coordination of the required skills does not form any problem for the human if he/she wants to achieve situation awareness. But, the transformation of these skills to technical systems rises a challenge, because technical systems naturally neither have software nor algorithms to handle the incoming data in a right and efficient way. Here, the following thesis applies by asking the following research question:

- ***How can perceptual situation awareness be achieved on a mobile robot?***

Deriving from this general question and regarding the different categories of situation awareness the following additional research questions arise:

- *How can a robot system be able to sense and perceive the environment without restricting the perception onto specific objects?*
- *How can a robot system be aware of humans and their movements during ego motion?*
- *How can a robot system direct its attention onto specific areas like a desired interaction partner?*

The thesis takes the questions into consideration and proposes solutions to realise all three categories of perceptual situation awareness.

The transfer from human capabilities in situation awareness to robot sensing is not directly feasible. First, the robot needs some kind of sensor for the perception of its environment. Second, the information has to be processed in a way that the robot can extract the essential information to build a mental picture of the environment.

One informative way of sensing is the visual perception. Known from the human eyes, vision provides rich information about structures, texture and colour of a scene. Considering the view of both eyes or different views from one moving eye, it is possible to extract additional depth information. The human uses the depth information extensively for e.g. path planning strategies. Depending on the distance to other entities, he/she varies his/her velocity and direction of the movement. Using only the 2D image information, the velocity of objects moving in the same direction or in the opposite direction are not directly detectable. Hence, additional distance information supports the detection of the accordant movement. The combination of digital cameras with additional distance information should provide a richer set of information, which can be used by a robot to gather information for his situation awareness.

Next, the visual information has to be processed to gain useful information out of the pure signal input. Figure 1.1 shows the different steps provided by this thesis to solve this task. Again, the human is the prototype for a good system being able to detect different parts of a scene. The first part extracts information about the environment. More precise, it extracts static non-changing parts like walls or cupboards, changeable parts like doors, chairs or other objects and, above all, other humans. Humans could be both potentially more dangerous for safe navigation compared to stationary objects and be a possible interaction partner. Here, the robot has to use algorithms which are reproducing these capabilities. Especially, the possibility to handle three dimensional data should be incorporated in order to reflect best the described human skills.

Motion tracking not only describes the second part of situation awareness, it is important for many computer vision problems starting from static camera scenarios like surveillance of a special area and ending with the most complex problem of moving cameras on some kind of platform. Motion tracking on an autonomous moving platform has several restrictions. First, the computational power is limited and second, the ego motion of the platform restricts the usage of most typical vision algorithms. In this thesis a mobile robot platform is used and concludingly, these complex problems have to be addressed. All robots that share environments with humans need to detect and track humans to avoid collisions and to ensure that the human does not become part of their background scene model. This is important for building a map of the environment, which helps the robot to localize itself, specific regions or objects. Finally, the robot needs some kind of memory and memory management to store the gathered information. Changes in the scene or appearing and disappearing persons have to be detected and related to the knowledge from the past.

In a human-robot interaction many situations arise, where the robot has to concentrate on specific objects or interaction partner. Here, it is of main importance that the robot is able to keep the focus of attention on the desired scene part. The direction of the attention enables the robot to extract further information out of these regions or to keep the interaction up with a desired interaction partner.

This thesis aims at providing algorithms, which enable a mobile robot to build a coherent mental picture of the important aspects of its surrounding. More precisely, solutions to implement the first three categories of situation awareness on a mobile robot are presented. The first solution enables a robot to extract efficiently comprehensive information from the environment by building a newly developed *articulated scene model*. The second proposed solution establishes temporal links between moving entities even in the presence of ego motion. Last, an algorithm based on biological inspired visual attention is presented, which provides an efficient way to restrict the visual processing to areas, which are most important to achieve situation awareness. All proposed solutions accomplish a stable and broad basis for future systems, which could implement the last category of information prediction. In the following, the aspired scenario and the used mobile robot platform are presented. Thereafter, the arising problems are further described and defined. Finally, it is stated how this thesis contributes to current research and a short outline will be presented.

1.1 Scenario Description

For a few decades now, the interest in household robotics made to assist the human in his/her daily life has been increasing. Special interest lies in the independent acting of the robots, because everyone should have the opportunity to use such robots at home without special knowledge of the technology. Therefore, the robots need the capability to securely move around in the human-like environment where they have to deal with the extra requests of highly dynamic surroundings. Persons move in narrow rooms or corridors, the background is often strongly cluttered and things might be rearranged. The solution to this problem is to facilitate situation awareness for robots [70]. My thesis is meant to work on an autonomous mobile robot which thereby can achieve situation awareness through implementation of the first three categories.

The development of the robot Biron (BIelefeld Robot CompaniON) started at the Bielefeld University in 2002. Its purpose is the interaction with humans. The robot platform is produced as a mobile robot where the base is a PatrolBot, which is 59 cm in length, 48 cm in width and 38 cm in height. The drive is a two-wheel differential drive with two passive rear casters for balance. Its solid foam-filled 19 cm diameter wheels are at the centre of rotation. The robot's weight is about 50 kilograms with batteries and the robot is manoeuvrable with 1.7 millimetres per second maximum translation and 300 degrees rotation per second. The robot uses several sensors like a laser range finder with 180 degrees, a pan-tilt camera unit, microphones and an interchangeable depth sensor. The depth is calculated either by stereo, by time of flight or by a mixture ensemble.

Figure 1.2: *Human-Robot-Interaction.* Robot Biron helps humans in their natural environment

The idea is to purchase a mobile robot for assisting the human in a private home environment, called *home-tour scenario*. Thereby, the robot has to pass through groups of humans, cluttered rooms and corridors. In order to move safely the robot has to detect obstacles and moving objects to calculate a safe path through the environment. Incorporating the detected object movements the robot could additionally apply smooth interaction spatial concepts and implicit body movements in such a way that the robot will be enhanced in reacting to social signals.

1.2 Problem Description and Definition

In general, situation awareness in a scene can be achieved by using one or several sensors under different conditions. The use of one camera or a calibrated pair of cameras to acquire depth is the most common case. There is also the possibility to acquire the desired information through a bundle of cameras, which observe a common scene. Here, I present the usage of active 3D cameras. The fast processing speed of these cameras enables the use on a mobile robot platform. The 3D cameras additionally provide an intensity image, which is subject to the same restrictions like usual 2D cameras. In all cases the algorithms have to deal with different and changing lightning conditions as well as with highly dynamic scenes.

The first category of situation awareness addresses the problem of what information should be of relevance when being incorporated in a scene-model. Here, traditional approaches try to build specific categories for each possibility. Yet, the possibilities are not predictable and the number of potential categories produce an invincible overhead.

Typically, objects and background scene fuse in the eye of the sensory input and it is hard to reveal each individual and the correct background scene. In order to deal with unknown and dynamic environments algorithms have to be developed that identify interesting objects on their own, differentiate between them and reveal the nature of the background scene.

The objects are interesting for the robot, as it could learn something about them or interact with them. The background scene is very important for a mobile robot, because it needs static and non-changing scene information to successful navigate through the human environment. If the algorithm is able to deliver such a complete picture of the viewed scene, the robot implements the first category of situation awareness.

One solution to the second category of situation awareness deals with the detection and tracking of humans in the scene in order to establish a temporal link between each entity. The detection and tracking of other possible moving objects is a complex problem on a moving platform. The cameras on the robot change their position due to ego motion and simultaneously the objects perform their own movements. Due to the ego motion many simplifications of the scene are not possible. The most common simplification is the subtraction of background [182], which reduces the complexity of the algorithm as only the foreground has to be analysed. This enhances the object detection and the tracking of objects as the foreground often consists only of the searched objects. If the camera is moving, this assumption is not true as not only the objects are moving, but also the background. One additional constraint of mobile platforms is related to the changing scene conditions. During the movement the position of the light sources relative to the robot are changing, which has a strong effect on the image intensity and illumination.

Directing the attention focus is essential for a mobile robot to adjust the sensor to the important input data. The third category of situation awareness is difficult to implement in a technical system, as the prototype origins from the human visual attention system. Theoretical models do exist in literature, but the approaches have to be adapted to a technical system. Hence, attention has to be achieved through adapted biological approaches which perform similar to the known attention from e.g. humans.

Here, it is required to copy the important weighting mechanisms, which weight a feature higher than others, especially if this feature has some kind of importance. A feature has got importance, if it is apart in a particular area, if it has got a strong colour contrast, if it has got a noticeable shape or if the feature is moving. The weighting of features is only the first step. If the focus should be directed onto a specific object, the discriminative features for this object have to be determined. Again, a weighting mechanism is needed in order to weight the features due to their distinction between object and background. Here, a most ideal weighting approach has to be found to discriminate even complex objects, like humans, from other humans or the background.

1.3 Contribution of this Thesis

In this thesis solutions for three of four parts for situation awareness on a mobile robot are presented which deliver an integrated solution to the perceptual parts of situation awareness. The fourth category of anticipating future needs is not addressed in this thesis. The presented algorithms handle with complex and actual topics in robotics, which results in a solid knowledge basis for further development in this area. The proposed system-modules implement state of the art algorithms, which are enriched with new methods and combined in a new and efficient way. In particular, the individual contributions are described as follows:

- The algorithms deal with two-dimensional and three-dimensional data simultaneously in order to achieve better and more stable results. The multi-dimensionality provides rich information to achieve a comprehensive situation awareness.
- All presented algorithms are designed to deal with the requirements of a mobile robot scenario in terms of processing speed and inter-process communication.
- The first part implements a newly developed *articulated scene model* which effectively incorporates information about the three-dimensional static background, movable objects and the human itself in one model Thereby, the robot uses the beneficial *Vista-space*, which describes everything viewable from the current point of view.
- The second part shows a fast and reliable solution to establish temporal links between human entities. Thereby, the complex problem of a mobile platform is addressed by a frame-based detection and a dynamic particle filter. The detection is speeded up by an interplay of a newly developed *u-v-disparity pre-detection* and a distance adaptive version of the reliable state-of-the-art *Histograms-of-oriented-Gradients based support vector machine classifier*. The tracking is done by a newly designed *dynamic particle filter with multi-dimensional observation model*, which effectively incorporates image features and three-dimensional data. The dynamic of the system results from the effectiveness of the interplay of detection and tracking, managed by a novel implemented *hypotheses management*, which also handles occlusions and cluttered scenes.
- Directing the focus of attention is done in the third part. A biological inspired *top-down attention* calculation is enriched with a *human-parts model*, which improves the directing of the attention focus.

The results show that each part is able to deliver important information for the robot's situation awareness. The articulated scene model extracts humans as well as static and movable scene parts out of the present observation. The mobile tracking part accounts the need for a mobile detection and tracking of humans even in the presence of ego motion. In this thesis a cognitive vision algorithm is shown which realises situation awareness through additional biologically inspired visual attention turning the focus of attention to a specific human in order to e.g. find a person again after a longer absence. All algorithms are fast and reliable enough to run directly on the mobile robot.

1.4 Overview

Chapter 2 describes the theoretical basement for the proposed systems. This includes the sensory input, the detection of humans and their appropriate tracking. Each part is expanded with additional information about algorithms working also in the three dimensional case. In Chapter (3) the algorithm proposal for a local scene analysis in 3D is presented. The Chapter (4) describes a solution, which enables a mobile robot to detect and track multiple humans during motion. In Chapter (5) the biologically inspired cognitive visual attention system for directing the attention focus is presented. All chapters include a detailed introduction to the employed algorithms as well as to each system's results. Finally, Chapter (6) finishes with a conclusion and a short outlook for further research.

2 Visual Basis for Situation Awareness

Situation awareness (SA) requires the ability to perceive the environment. Beside the static environment dynamic objects like humans are of main interest, as the first and second categories of SA rely partially or completely on the detection and tracking of moving entities. This is the most important input information for a mobile robot, as the human is a dynamically moving object, which could be e.g. an obstacle or an interaction partner. Other information like the background, action spaces or interesting objects are also addressed, but the focus lies on the visual perception of the human. Detection and tracking are generally important tasks in technical systems, as the detection is the essential step to be aware of something present and the path of a moving object provides rich information for many purposes. On the one hand a lot of data can be annotated and associated easily by tracking, like annotating video streams or analysing traffic scenarios. On the other hand many real-time critical tasks like surveillance [115] [47] [85], human-robot interaction [28] [172] [128] [111] [74] [160] [179] [60], driver assistance [89] [10], perceptual user interfaces [37], smart rooms [216] [123] [103], augmented reality [66], or object-based video compression [54] can be solved by detection and tracking.

In the following I describe the typical concept of detection and tracking systems to achieve situation awareness. Comparing different systems, which use detection and tracking, it becomes evident that most of them are designed through the following composition (see Fig. (2.1)). First of all, each system uses a sensory input to gather information about the environment. The information has to be processed in order to detect all appropriate targets and to build hypotheses. The system has to build a distinct model in order to distinguish each hypothesis from the background and other hypotheses. Using this information a tracking step reveals each position of the object over time. Additionally, the system has to deal with new, occluded or disappeared objects and has to keep track of the known hypotheses in subsequent frames. Finally, the system should deliver some kind of output, which is usually a trajectory for each found hypothesis, the actual position and the size of present objects.

In literature a large number of detection or tracking algorithms exist, but in a combined system there are additional requests for the different parts. They have to be working in real-time in order not to thwart the complete calculation. They have to provide their information in a common status and they have to add a communication layer to distribute their information. Last, a synchronisation is important to assign the correct data to each module at the correct time. The algorithms introduced in this chapter do not consider these requests, but my presented solutions in the subsequent chapters do. Anyway, the ongoing chapter provides a basic knowledge about each part of a typical detection and tracking system.

Visual input provides rich information about appearance, structure and light in the scene. Hence, the thesis is based on visual perception. In order to provide a solid

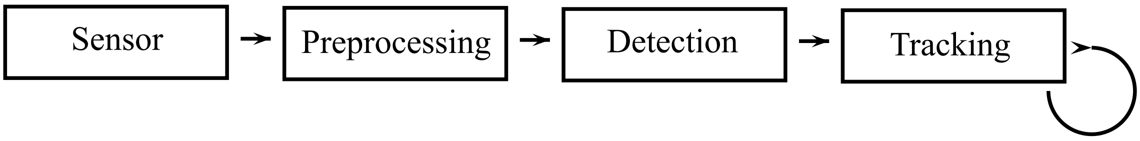


Figure 2.1: *Typical tracking system approach.* Most tracking systems are based on the presented system approach. The data is delivered by sensory input, which is pre-processed in different ways. Afterwards, objects are detected, which are handed over to the tracking, which continuously tracks the object in the subsequent frames.

knowledge on the basis of visual perception, the underlying sensory input is described in the ongoing chapter. Here, the aim is to use 3D information in addition to the usual visual perception to represent better the environment and to enhance the presented techniques (e.g. regarding the false-positive detection rate cf.Sec. (4.7.4)). Accordingly, the different methods of acquiring 3D information are also introduced. Thereafter, a detailed introduction in visual detection and tracking of moving objects with special emphasis on humans is given in order to provide all essential information which is needed to build a complete awareness system.

2.1 Description of Sensor Set-ups

Mobile robots need first of all a sensory input to gather information about the environment in order to achieve SA. This could be any type of sensor like a laser, sonar, lidar or a video camera. Additionally, a robot system could use several sensors to combine their information to a more meaningful one. This could be an amount of identical sensors or a mixture of sensors. This work is based on visual input, which constrains the following descriptions to the visual input by cameras. Vision is chosen, because it delivers manifold information like appearance, colour and shape, which is advantageous compared to the point information from laser, sonar or lidar. Cameras usually have the disadvantage of projecting the 3D world onto a 2D image plane. The third dimension is lost and the other dimensions are projectively distorted. But, a mobile robot needs to perceive its environment in 3D in order to successfully avoid collisions and to better discriminate between different objects. Utilizing additional 3D data the missing depth information from usual camera vision can be compensated. Hence, despite the basic knowledge about monochrome cameras the following sections give an overview about the principles of 3D vision, which is needed to reconstruct object trajectories in 3D.

2.1.1 Monochrome Vision

Most of the developed algorithms in computer vision are based on a single or monochrome camera input, because one do not has to deal with set-up calibration or data transformation from one sensor into the other. Digital cameras provide their data as Gray level or colour information, whereupon in this thesis I do not go into camera optics, light reflectance properties or sensor qualities, which all can be found in the accordant literature [63] [90] [198] [211]. Here, I describe the basis of projection and calibration of a sensor, because it is an important step to gather reliable information from it. The calibration is needed to transform a projected 3D world point using an ideal pinhole camera model into real camera coordinates considering the pixel sensor spacing and the relative position of the sensor plane to the origin.

Before I explain the calibration itself, it is important to mention the 3D to 2D projection using an idealized camera pinhole model. In computer vision the most common projection is the *perspective* projection. Here, 3D points x_k are projected onto an image plane $p = (u, v)$ by dividing the points x, y, z components by the depth z. Using inhomogeneous coordinates in Euclidean geometry this can be written as

$$p = P_z(x_k) = \begin{bmatrix} -bx/z \\ -by/z \\ 1 \end{bmatrix} \tag{2.1}$$

,with b distance from the optical centre to the principal point on the image plane. The more common usage are the homogeneous coordinates in order to circumvent the non-linear formulation of perspective projection in Euclidean geometry [211]

$$p = \begin{bmatrix} -b & 0 & 0 & 0 \\ 0 & -b & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_k \tag{2.2}$$

The projection of a 3D point removes the distance dimension, which is not possible to recover without the use of additional information. This disadvantage can be recovered using additional calibrated cameras, different viewpoints or range sensors, to mention the most common possibilities. They allow the calculation of the sensor-based depth or disparity value d. Then, it is possible to use the inverse of a $4x4$ projective matrix (see Sec. (2.1.2)) to recalculate the 3D point coordinates.

The calibration of a camera is an important step, because it represents more precisely the physical assembling of a real camera compared to the ideal camera pinhole model. Incorporating a calibration, the spacing of the sensor and its relative position to the camera are considered. Fig. (2.2) is meant to clarify the point.

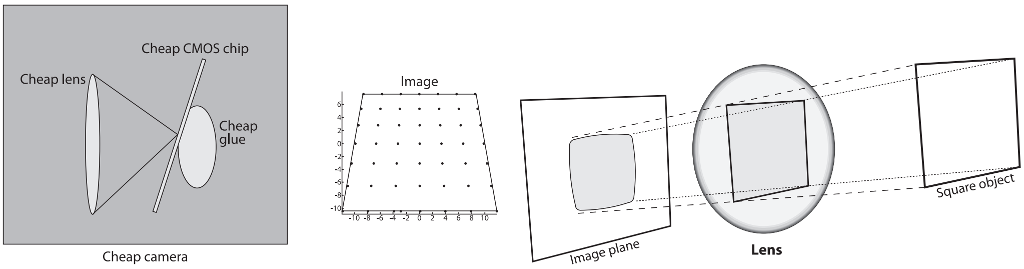


Figure 2.2: *Physical sensor set-up.* If the sensor is not parallel to the image plane, the image results in tangential distortion. Additionally, the projection of an object (here a square) undergoes a distortion due to possible inaccuracies of the lens. ((Images found in [36]))

The projection centre of the 3D point onto the sensor does not have to meet the image centre, because the sensor could be shifted to the image plane. This fact can be adjusted by integrating the real *projection centre* c_x, c_y into the calculation. Additionally, the projection does not fall directly onto the lens, but instead on the sensor lying a small distance behind. This small distance is called the*focal length* f. Both values are expressed in pixel coordinates.

To map a 3D point on the camera image plane, the following equation is used

$$p = \left[R|t\right] x_l \tag{2.3}$$

with the *extrinsic* parameters R (Rotation) and t (translation). The mapping from the camera coordination system into the sensor coordination system is given by the camera matrix K

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

where s is called *skew* and encodes any possible skew between the sensor axes due to the sensor not being mounted perpendicular to the optical axis. In practice, the skew is mostly set to $s = 0$. The optical centre is often set to the middle of the image $(c_x, c_y) = (W/2, H/2)$, which can result in a usable camera model with only a single unknown, the focal length f.

The parameters of the matrix K are called the *intrinsic* camera parameters. They can be computed through known 3D points, which can be found using a calibration pattern (e.g. a chessboard with known size). The principles and different algorithms to calibrate a camera can be found in [211]. In this work, the calibration method of *Bouguet* [34] is used.

The complete projection from a world point to the sensor is summarized with

$$p = K\left[R|t\right] x = Px_k \tag{2.5}$$

,with the projection matrix P

$$P = K\left[R|t\right] \tag{2.6}$$

combining the intrinsic and extrinsic camera parameters in one image formation process.

The described idealized camera model assumes a linear projection model, where straight lines in the real world project to straight lines in the image. This assumption does mostly not apply to real cameras, because they have some kind of distortion in their lens [198]. This leads to a visible curvature in the projection of the straight lines. The distortion can be divided into the two most important *radial* and *tangential* distortion (In fact, there are some more distortions, but their effect is much lower). The radial distortion effects that coordinates of projected points are shifted towards or away from the image centre by an amount proportional to the radial distance. The tangential distortion arises from manufacturing defects, which cause in a not exactly parallel lens to the imaging plane. The distortion artefacts have to be removed to apply the described camera model formulas (further information can be found in [90]).

With the principles of projective geometry and the knowledge about single camera calibration it is possible to build exact mathematical models for the transformation from one camera into another or from one viewpoint to another to be able to reconstruct three dimensional structures. The following section gives an overview of the specific mathematics and algorithms needed for the reconstruction.

2.1.2 Stereo Vision

Three-dimensional information is essential for a mobile robot in order to interact safely with his rapid changing environment. Usual monochrome cameras do not provide depth data, because this information is lost in the projection process. Hence, further steps are necessary to get depth or 3D information from monochrome cameras. On the one hand, depth information could be gathered through photometric approaches (shape from shading, shape from shadow, photoclinometry, photometric photo and shape from polarisation) or by the spread function of the optical system (shape from focus, shape from defocus). On the other hand, 3D information can be calculated by geometrical approaches, which minimise the Euclidean back-projection error [211]. Here, the use of two cameras, called stereo vision, and their geometrical correlation is described.

Stereo vision is well known from our human visual perception. We look at the world around us with two eyes, which enables us to perceive depth from the difference in the appearance from the left and right eye. Near objects have a bigger shift in both images than objects far away. This shift is called *disparity* and it is inversely proportional to the world distance from the observer. The calculation of the disparity of points is accomplished by searching for corresponding points in each image and measuring their distance. With the known intrinsic and extrinsic camera parameters (here, extrinsic means the rotation and translation between both camera centres) the 3D world coordinates can additionally be applied. The details of the stereo process can be divided into the following 4 steps:

- **Undistortion:** Mathematically remove radial and tangential lens distortion
- **Rectification:** Align for the angles and distances between cameras. The output are row-aligned and rectified images
- **Correspondences:** Search the same features in the left and right images. The outcome is a disparity map, where each value means the difference in x-coordinates on the image plane of the same feature in each image (for horizontal aligned cameras)
- **Re-projection:** Calculate the 3D world position through the disparities and the known camera set-up parameters.

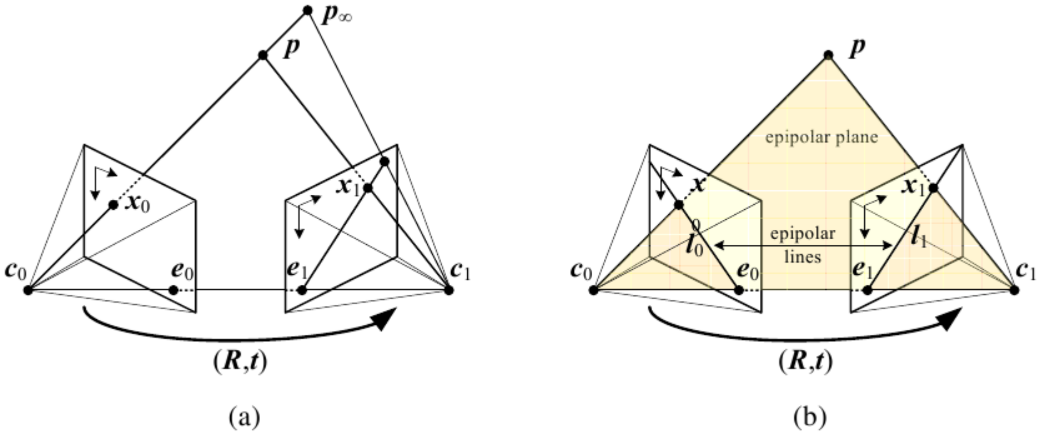


Figure 2.3: *Epipolar geometry.* (a) One epipolar line corresponding to one ray. (b) Corresponding epipolar plane ((Images found in [198]))

The first step is described in Sec. (2.1.1). The other three steps are further explained in the following.

To explain the rectification it is important to describe first the epipolar geometry, which motivates the rectification in order to speed up the search for correspondences. Figure 2.3 shows how one point x_0 projects to an epipolar line segment in the other image. If the point p projects on the point x_0 in the first camera and we know the camera centres c_0, c_1 and the extrinsic parameters as well, it is possible to project the camera centre c_0 into the image plane of the second camera. The projection point e_1 is called *epipole*. The back projection of the second camera centre c_1 in the first image is the correspondent epipole e_0. Connecting the epipoles with the projection points x_0, x_1 and extending these lines to infinity results in a pair of corresponding *epipolar lines*. The epipolar lines are the intersections of the *epipolar plane* with the image planes (see Fig. (2.3) b). The epipolar plane is defined through the camera centres c_0, c_1 and the point p. [90]

The epipolar geometry can now be used to find corresponding points in both images. The *epipolar constraint* [90] defines that each corresponding point has to project onto the accordant epipolar lines. This constraint limits the search for corresponding points to the epipolar lines. The step of rectification uses the knowledge about the camera set-up to horizontally align the epipolar lines, which restricts the search to the same horizontal line. This process uses an image warping, which rotates, translates and scales one camera image into the other in a way that the camera centres are the same and both cameras look perpendicular at the same scene, see Fig. (2.4). The warping information is encoded in the *essential* matrix E and the *fundamental* matrix F (further details can be found in [90]). The essential matrix is a pure physical transformation of the image centres and the fundamental matrix additionally encodes the camera parameters (E operates in physical and F in image pixel coordinates).

The resulting rectified geometry allows to write the inverse relationship between 3D

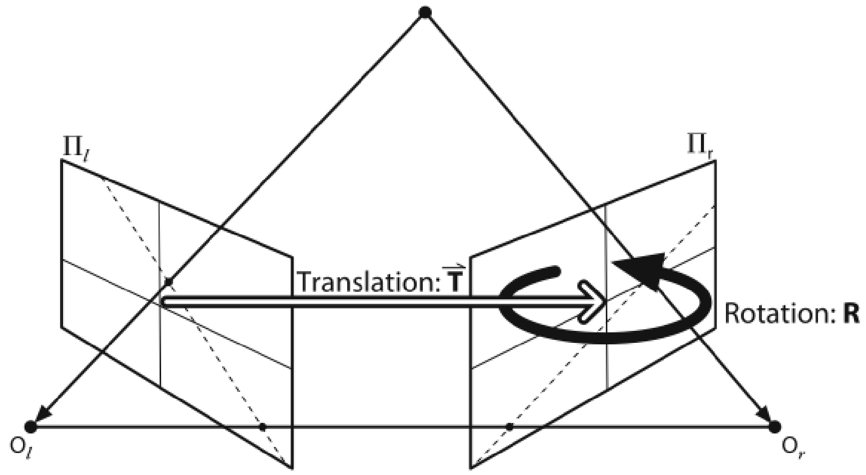


Figure 2.4: *Transformation of one camera centre into another camera centre.* Utilizing a rotation matrix R and a translation T, the camera centre O_l can be transformed into O_r.((Image found in [36]))

distance Z and disparity d

$$d = \frac{fb}{Z} \tag{2.7}$$

with f focal length and b distance between the camera centres (also called *baseline*). The corresponding points in the images (l,r) can be found through

$$u_r = u_l + d(u_r, v_r), \quad v_r = v_l \tag{2.8}$$

with u, v pixel coordinates in the left and right image. The resulting values from the found correspondences can be stored in the disparity map $d(u, v)$.

The process of finding correspondences can be solved with sparse feature-based algorithms like optical flow [95] or dense stereo algorithms, whereupon most of the stereo algorithms in literature can be divided in two main categories, the *local* and *global* methods.

The local methods use a sliding-window based approach, where the disparity value is calculated through the best match of the intensities of a target window and search windows. Many algorithms try to optimize the search with the sum-of-squared-distances (SSD) algorithm.

Global methods minimize a global cost function with explicit smoothness assumptions to seek the disparity values. The main distinction between these methods is the minimization procedure. Expectation minimization [29], graph cuts [35] or simulated annealing [144] are some example methods, which are used in literature.

Finally, after the disparity map has been applied, it is possible to re-project the image points onto 3D world coordinates using formula 2.7 and the inverse projection of formula 2.5 (using rectified images).

$$Q \begin{bmatrix} u & v & d & 1 \end{bmatrix}^T = \begin{bmatrix} X & Y & Z & W \end{bmatrix}^T \tag{2.9}$$

with scaling factor W and projection matrix Q

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c'_x)/T_x \end{bmatrix} \tag{2.10}$$

with T translation between the cameras and all parameters from the left camera except c'_x, which is the image centre of the right camera (if the principal rays intersect at infinity, then $c_x = c'_x$ and the lower right term is equal to 0). Stereo vision has been in the focus of research for a long time, because several cameras provide the possibility to recalculate the depth information of the projective scene geometry. In the next section, the extension with more than two cameras is described.

2.1.3 Multi-Camera Set-up

The use of more than two cameras offers two advantages. First, taking more than two images results in several disparity maps, which can be used to enhance the result of the consolidated disparity map. One possibility is the sum of summed-squared-difference (SSSD) [156]. Second, more cameras can deal with different views around the scene, which can remove occlusion artefacts and deliver a complete 3D scene [114]. *Scene Flow* is a closely related topic, where the optical flow is extended to 3D scene flow. The scene flow is calculated through multiple cameras similar to the stereo correspondence problem [203].

Stereo or multi camera vision offers the calculation of depth data of the scene. But, the calculation is erroneous and restricted to structured areas. Plain walls or e.g. the sky are hard to process for stereo algorithms. Additionally, a multi-camera set-up requires a solid calibration in order to deliver reliable results. Here, the use of other sensors than usual intensity cameras offers high potential to get superior results more easily. The calculation of the correspondences of the intensity values consumes a lot of processing time, which additionally pushes the use of other vision sensors on a mobile robot.

2.1.4 Active Cameras for 3D Vision

Active vision names the use of active sensors. This could be a controlled movement or the active emission of light. As the interest lies on the active range finding, time-of-flight sensors are another possibility to measure the distance of a scene. One of the famous active cameras is the Swissranger sensor.

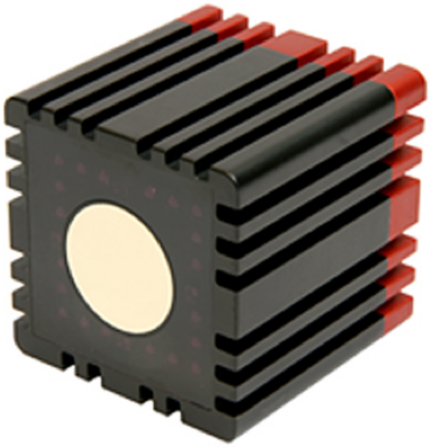


Figure 2.5: *Mesa Swissranger.* Time-of-flight sensor for active vision

The Swissranger SR4000 (see Fig. (2.5)) provided by Swiss Center for Electronics and Microtechnology (CSEM) [209] delivers a matrix of distance measurements independent from texture and lighting conditions. It consists of 176 × 144 CMOS pixel sensors which are able to determine actively the distance between the optical centre of the camera and the real 3D world point via measuring the time-of-flight of a near-infra-red signal. Besides a distance value matrix, the camera provides per frame a matrix containing amplitude values. The amplitude value indicates the amplitude of the reflected near-infra-red signal received by the sensor and implies therefore the amount of light reflected by a world point. A small amplitude corresponds to a small amount of light reflected and therefore indicates a weak signal.

Several researchers have already developed preprocessing and calibration techniques dealing with noise arising from the different reflectance properties and characteristics of the ToF cameras, like additional infra-red light in the scene, and measurement errors at edges (so-called "flying pixels"). Schiller [174] proposed an automatic calibration of the entire 3D ToF signals using a bunch of different cameras. Colour information is also used by Huhle [101] for outlier detection and smoothing using Non-local Means filter [41]. Smoothing techniques relying only on the ToF data are amplitude thresholding with a fix value [145], removing of "flying pixels" at edges via detecting iteratively geometric outliers taking into account the 2D neighbourhood [100], and correcting the amplitude values using distance values and vice versa [158].

One disadvantage of the 3D time-of-flight sensor is the reflectance characteristic. Glasses or black regions do not mirror the light and hence, the run time of the light beam can not be measured for these areas. But, the advantages of fast and reliable 3D information outweigh mostly the bad reflecting characteristics. The sensor does not provide colour information. The lack of colour information is often compensated through an additional calibrated colour camera, yielding a colour image with connected distance information [101]. But, the distance information is sparse due to the small resolution of the sensor. Hence, the use of different sensors in one calibrated set-up, providing colour and depth information for each pixel, would reach the optimum. In the next section, such a sensor is introduced, which offers high potential for mobile robotics.

2.1.5 Sensor Ensemble

Generally, a sensor ensemble offers broader possibilities for the detection and tracking of objects, because different sensors can represent the characteristics of objects more explicit. A thermo-graphic camera e.g. could detect creatures, human beings [207] or running cars [93] more easily. The combination with a range camera delivers the 3D position of the object. Additionally, the sensors could be distributed, what removes the effects of occlusions [47]. This benefit comes with a high cost, as the different sensors need a calibration to provide their information in a common representation. Usually, a pattern with a known size is used, which is placed such that it is viewed in every sensor. Then, a calibration technique known from stereo vision could be used from sensor to sensor. If 3D sensors are used, one possibility is the use of point cloud registration techniques like iterative closest points (ICP) or more advanced approaches like the use of surfaces, to iteratively converge the sensor data [97].

One upcoming sensor, which incorporates multiple sensors in one casing, originates originally from the gaming sector. Microsoft offers an external device for its gaming console XBox, which contains a bundle of sensors and a pan-tilt unit (see Fig. (2.6)). The sensors consist of a RGB colour camera, an infra-red light source, an infra-red camera, 3D microphones and a three axis accelerometer. The infra-red emitter projects a light pattern into the scene, which is used by the infra-red camera to calculate depth information through triangulation. The colour camera and the depth sensor run at 30 Hz or 15 Hz with 320x240 resolution and 640x320 respectively. The colour camera can additionally provide a resolution of 1024x768 with about 10 Hz. The depth calculation is able to provide a depth resolution up to 10 meter with a measurement error about 10 centimetre at 3.5 meter and 50 centimetre at 10 meter. The device is provided by PrimeSense, who developed the sensor for the project Natal by Microsoft to play games without controllers. Therefore, the device has an included computing board, which has e.g. the ability to detect human players and to track hands [185]. But, the detection is restricted to a an attached camera, while the implemented algorithms are not able to handle ego-motion.

Figure 2.6: *Kinect Sensor provided by Microsoft and developed from PrimeSense.* The Kinect sensor is equipped with a RGB camera, an infra-red emitter and sensor, 3D microphone and Accelerometer

The driver software provides the possibility to calibrate the internal sensors and to trigger the capture time, which adjusts the different data onto each other. The light emitter projects a point pattern, which is encoded by the camera through the infra-red image and the triangulation of the distances of the projected points. Thus, it is possible to get a colour image, a depth image and a colour encoded 3D point cloud from the sensor for further usage. All data is provided in real-time at about 15 Hz.

Comparing all different sensor types, it becomes evident that the usage of an active 3D camera enhances the possibilities on a mobile robot. Hence, the introduced solutions apply both, the Swissranger and the Microsoft Kinect camera. All parts in this thesis are able to handle any sensory input as long as it provides a colour image and 3D data.

2.2 Object Detection

After introducing possible sensor set-ups, it is essential to describe further processing steps for the incoming data. Thereby, the essential step for an autonomous tracking system is the detection of interesting objects. The human is capable of detecting many complex objects or humans in a scene at once, but using machine vision, the capabilities are even nowadays limited. The problem is based on the variability and complexity of a scene. Complex classes of objects are hard to detect, because they are mostly non-rigid and they have extreme variations in their shape. The matching against a database is e.g. possible only for classes with low variability within the class (e.g. street signs), but it is hard to accomplish for objects with a high variability (e.g. humans). The class of humans is particularly challenging for a number of reasons:

- Humans can be located at every image position with various poses, clothing and different articulations of body parts.
- Humans are mostly found in strongly cluttered background, whereas the clutter covers appearance and depth.
- Humans are hard to detect as they can be very small in the image due to their distance to the observer and thus, look very similar to background objects like trees, poles or narrow openings.
- As humans are dynamic objects they are often detected by their movement. Anyway, motion is barely usable on a moving platform.

If one is not trying to analyse the complete scene in order to look for each possible object, the problem can be reduced and defined. Hence, most detection algorithms are looking for a specific type of object. If the interesting class is known and each image is scanned for its occurrence, the problem is called *object detection*.

Definition 1 (Object detection) *:*
Object detection is the process of determining regions, which contain a specific object.

Generally, a detector can be based on the *sliding-window* approach, which moves a search window over the image and analyses any possible sub-window. These approaches are likely to be slow and error-prone. More advanced approaches try to find likely regions, which could contain the searched object. The problem in object detection are regions, which look similar to an object, which leads to *false-positives*. On the other hand, a desired object can be partially occluded or moved into shadows, which would lead to a miss-detection or *false-negative*. A reliable detector tries to minimize the false-positive rate and to maximise the true-positive rate.

In the following, I give an introduction into object detection algorithms with static cameras and subsequent, with moving cameras to give an overview of the actual literature in this field.

2.2.1 Object Detection ideal for Static Cameras

The use of static cameras is mostly found in surveillance scenarios, where some kind of area is observed for security reasons. Especially in the united states or in the united kingdom is a big interest in surveillance to prevent terrorist acts or crime. Using video surveillance, it is possible to detect moving blobs, e.g. humans, and to track them through the entire observed area. Static cameras abet the use of background subtraction (Sec. (4.1)), where it is possible to subtract the background to segment such foreground blobs.

Background Subtraction

Background subtraction is a powerful tool to extract moving objects, but it also has to take the following points into consideration:

- Illumination changes (e.g. clouds)
- Motion changes (e.g. camera oscillation, tree branches)
- Changes in the background (e.g. parked cars, removed chairs)

The basic method of background subtraction is *frame differencing*

$$|frame_t - frame_{t-1}| > Thresh \tag{2.11}$$

The subtraction only relies on the previous frame, which is sensitive to slow motion, the frame rate and the threshold *Thresh*. Another approach is to model the background as an average of the last frames

$$Bg_{t+1} = \alpha * frame_t + (1 - \alpha) * Bg_t \tag{2.12}$$

Both simple approaches can be further adapted by selecting each pixel as fore- or background and adaptively update it as background or skip it as foreground.

$$\begin{aligned} Bg_{t+1}(u,v) &= \alpha * frame_t(u,v) + (1-\alpha) * Bg_t(u,v) && (2.13) \\ & \quad \text{if } frame_t(u,v) \text{ background} \\ Bg_{t+1}(u,v) &= Bg_t(u,v) && (2.14) \\ & \quad \text{if } frame_t(u,v) \text{ foreground} \end{aligned}$$

It is also possible to fit a Gaussian [217] or a mixture of Gaussian [188] over the histogram of background values. More extended algorithms make use of codebooks [117] modelling the pixels either separately from each other or incorporating nearby pixels using subspaces [150]. For a lot of approaches a static background is mandatory. However, Sheikh and Shah introduced an approach that is able to cope with uniformly moving background like a river or a tree [183].

Optical Flow

Different to subtracting the static parts the moving parts can be calculated in order to reveal the moving objects in the scene. A widely used technique is to compute the dense *Optical Flow* using each 2D image pixel. The optical flow is the distribution of apparent velocity of moving brightness patterns in an image and arises both from the relative objects' and the viewer's motion [82]. The flow of a constant brightness profile

$$\begin{aligned} I(x,y,t) &= I(x+\mathrm{d}x, y+\mathrm{d}y, t+\mathrm{d}t) \\ &= I(x+v_x\cdot\mathrm{d}t, y+v_y\cdot\mathrm{d}t, t+\mathrm{d}t) \qquad (2.15) \\ &\Rightarrow \frac{\partial I}{\partial x}\cdot v_x + \frac{\partial I}{\partial y}\cdot v_y = -\frac{\partial I}{\partial t} \qquad (2.16) \end{aligned}$$

is described by the constant velocity vector $\vec{v}_{\mathrm{2D}} = (v_x, v_y)^T$. Usually, the estimation of optical flow is founded on differential methods. They can be classified into global strategies which attempt to minimize a global energy functional [95] and local methods, that optimize some local energy-like expression. A prominent algorithm developed by Lucas and Kanade [136] uses the spatial intensity gradient of the images to find a good match using a type of Newton-Raphson iteration. They assume the optical flow to be constant within a certain neighbourhood $\mathcal{N}$ which allows to solve the Optical Flow Constraint Eq. 2.16 via least square minimization.

3D Background Subtraction

If 3D data is available, z-keying is a simple algorithm to segment foreground data from the background. The image is cropped by depth data, where only those pixel rest in the foreground, whose depth is shorter than a specific cut-off [113]. Z-keying was first used for video conference applications and to exchange the background.

The other mentioned background algorithms could also be extended to work with 3D data, which enhances the possibilities of the background subtraction. Care should be taken for noise, which occurs more often in specific depth data than in intensity image data.

2.2.2 Object Detection for Static and Moving Cameras

The following algorithms are not restricted to static cameras, but work also on moving cameras. The most important differences in these cases are that the algorithms do not rely on previous frames or on motion of the object. Instead, the techniques search for the desired object only in the actual frame, which enables the use of the algorithms on a mobile platform. For the following algorithms I assume that the image does not contain motion blur, which could violate the detection process.

Many detection algorithms are based on the *sliding window* approach [159] [204] [205] [52] [180] [181] or make use of *evidence aggregation* to build hypotheses [64] [149] [131] [64] [6] [7].

Sliding window detection systems perform an exhaustive scan over the image for each possible location or scale of the object. In each window a *feature* component extracts essential features, which encodes the visual appearance of the object. A successive *classifier* component decides independently for each window if it contains the searched object or not. This two part principle of a feature component and a successive classifier is valid for most detection algorithms. Hence, the following algorithms are additionally ordered to first describe the possible feature components and adjacent, the classifiers used.

Evidence aggregation uses the possibility to segment an object in different parts, where each part calculates a vote for the object. Hypotheses are build out of the joint assembly of each particular vote.

Feature Component

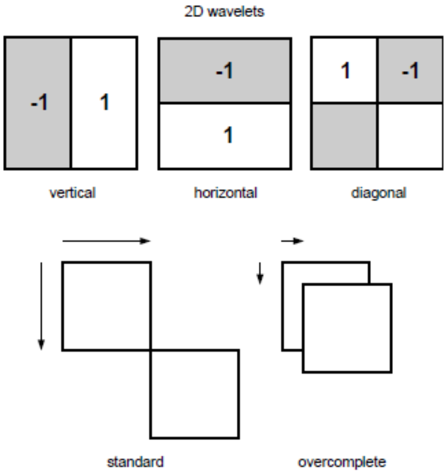


Figure 2.7: *Haar-wavelets.* The top row shows the three orientations - vertical, horizontal, and diagonal - of the 2D wavelets. The bottom row illustrates the difference between the standard wavelet shift and our quadruple density transform. (Image found in [159])

First, some sliding window detection systems are presented. One early approach used overlapping **Haar-Wavelets** to train a polynomial Support vector machine (SVM) [159]. The feature patterns consist of three types of rectangle patterns (Fig. (2.7)). The patterns are horizontally or vertically adjacent and they have the same size and shape. Each feature corresponds to the difference between the sum of the pixels in the black and white rectangular regions. In contrast to the traditional wavelet transform, where the wavelets do not overlap, the authors propose to achieve a better spatial resolution by shifting only $\frac{1}{4}$ of the the wavelet size, yielding an over complete dictionary of wavelet

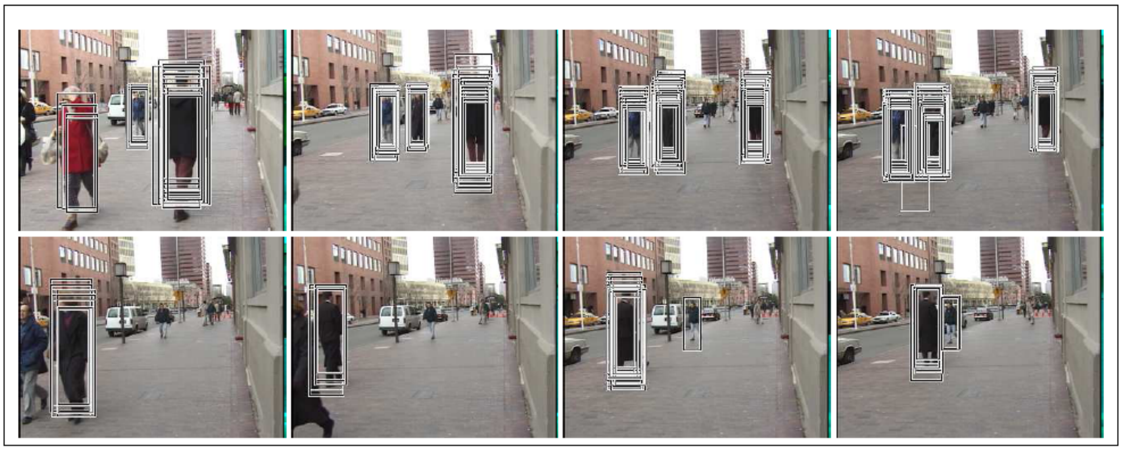


Figure 2.8: *Detection Results.* Some sample results using frontal, rear and side training images and a Haar-Wavelet pedestrian detector. (Image found in [159])

features. In order to search for the object with their classifier, they achieved scaling by incrementally resizing the image and running the sliding window detector over each scaled image version, instead of the usual sliding window approach. In their paper, they also describe an on-line version of their algorithm running with 10 Hz in combination with the *Daimler Chrysler Urban Traffic Assistant*. To achieve the fast processing rate they reduced their set of Haar-Wavelets to a small set containing only the most important features. They additionally reduced the set of support vectors and finally, they use Gray-level images instead of colour images. An important speed up arises in the combination with the Daimler Assistant, which delivers interest regions what narrows the required search windows a lot. Some example detection results of the described classifier can be seen in Fig. (2.8).

Another use of wavelets is presented in [204] from Viola and Jones. They propose the use of Haar-like Wavelets in addition with a cascade of **AdaBoost** classifiers. Initially invented as a face detector, the classifier can be trained on each object type due to its simple feature patterns. The first key for their fast detector is based on the computation of the features. Viola and Jones proposed to use an *Integral Image*, which allows to rapidly compute the sum of the pixels in one rectangular. The second key of their algorithm is found in a learned cascade of feature detectors. The idea is to first segment the interesting regions with a small set of features, which are rechecked by additional and more accurate classifiers. The cascade quickly removes any region, which does not have a particular similarity to the searched object (a more detailed explanation can be found in Sec. (2.2.2)).

An extension to the previously described algorithm can be found in [205]. The authors added motion information to the spatial information of one single detector. The detector has a low false positive rate and can detect humans at very small scales (as small as $20x15$ pixels). The system is trained on full human bodies and thus, can not detect partially occluded persons. Additionally, the system requires a static camera, which disables it for the requirements of this work.

Templates are also feasible for detecting objects. A template K with m, n pixel can be matched against each possible sub window I

$$corr = \frac{1}{mn}\sum_{u,v} K_{u,v} * I_{u,v} \quad with(u,v) \in [1\ldots m] \times [1\ldots n] \tag{2.17}$$

The correlation *corr* is equal to 1, if the template perfectly matches and 0, if it is a total mismatch. The template matching is expensive due to its comparison at every single image position. Gavrila employs a hierarchical Chamfer matching strategy to overcome this problem [78]. The templates and the image are binarised to employ a distance transformation, where the value corresponds to the distance to a pixel with a value > 0. This leads to a continuous similarity measure, where a coarse-to-fine search grid is sufficient. Additionally, a hierarchical template structure saves time through the reduction of template equations. Utilizing about 1000 training examples, Gavrila achieves a detection rate about 75% - 85% per frame with maximum 2 false positives.

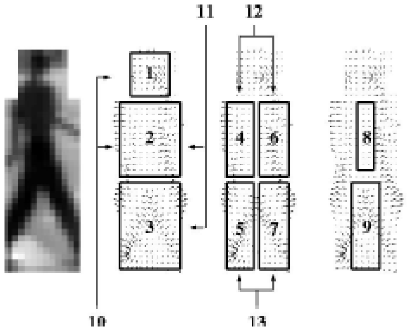


Figure 2.9: *Multi region training.* Utilizing a gradient image, Shashua et al. propose to divide the region into 9 subregions and additionally, 4 pair combinations (10,11,12,13) to train multiple classifiers with AdaBoost. (Image found in [181])

In recent work many authors propose to employ statistics on image gradients for people detection. Shashua [181] uses edge orientation histograms in conjunction with AdaBoost, which shows reliable results in conjunction with a pre-selection of interest regions and clutter removement strategies (see Fig. (2.9)).

One prominent approach to detect objects are the **Histograms of Oriented Gradients** from Dalal and Trigs [52]. Inspired from the *Scale Invariant Feature Transformation* approach, the authors show that the use of local orientation histograms combined with local normalisation schemes works very well in the application of human detection. The algorithm works as follows: The input image is normalised in gamma and colour. Afterwards, gradients are computed utilising simple 1-D masks without smoothing. The authors show that smoothing decreases the detection rate as well as complex derivative masks. Next, the image window is divided into small spatial regions (*"cells"*). Each spatial region accumulates a local 1-D histogram of edge orientations over the pixels of the cell. The vote of each pixel is weighted by a function of the gradient magnitude of the pixel. The authors propose to use fine orientation coding and coarse spatial binning, which in numbers is about 9 orientation bins spaced over $0° - 180°$. For better invariance to illumination or shadowing, the intensity values of neighbouring cells are accumulated over larger spatial regions ("blocks") in order to normalize all cells in each block using the specific results. These normalised descriptor blocks are called Histograms of Oriented Gradients descriptors. These blocks densely overlap in

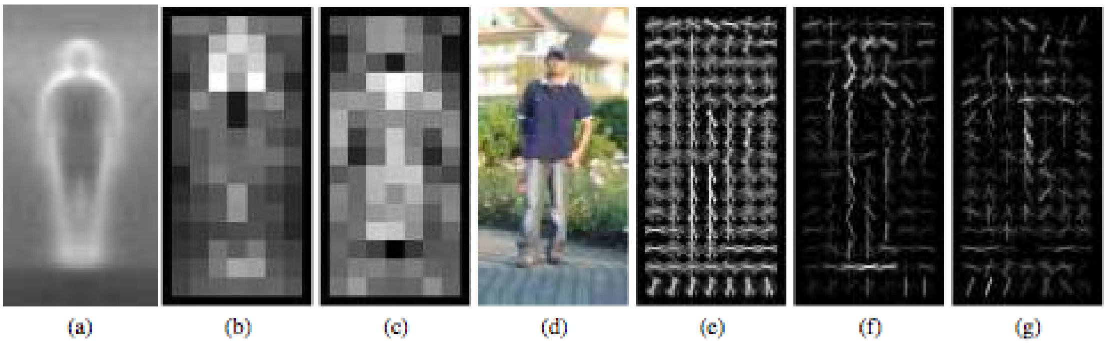


Figure 2.10: *Histograms of Oriented Gradients.* (a) Average training gradient over all training examples. (b) Maximum positive weight of the SVM in each block. (c) Likewise for the negative SVM weights. (d) Original image. (e) Computed R-HOG descriptor. (f) R-HOG descriptor weighted with positive SVM weights. (g) Weighted with negative SVM weights respectively. (Image found in [52])

the detection window. Using the combination of all blocks as a feature vector in a linear SVM results in the proposed human classifier. The proposed calculation has the advantage of capturing very characteristic local gradient or shape structure using local representations with an easily controllable degree of invariance to local geometric and photometric transformations. Of course, this is only applicable for translations or rotations much smaller than the local spatial or orientation bin size. The strong normalisation over each block ensures photometric transformations. The final detector and some other example images are shown in Fig. (2.10).

The calculation of these big amount of low level features for statistical analysis of the image makes it affordable to use many training samples in order to train properly the SVM. Especially, if even more features are used the high-dimensionality of the feature space becomes nearly intractable. Consequently, other authors propose to reduce the feature space with dimensionality reduction techniques. Schwartz et al. propose to use HoG-features with additional colour and texture information and to reduce the feature space with **Partial Least Squares** (PLS) analysis [180]. The idea of PLS is to construct predictor variables ("latent variables"), which are a linear combination of the original variables summarized in a matrix. Additionally, PLS provides a class label as output and hence, it provides a vector with response variables (one for each class). The dimensionality reduction is performed by projecting the feature vector onto some weight vectors, obtaining a latent vector as result, which is used in classification.

In the following, I give a broad overview over further state of the arts techniques, which can be studied in detail in the given literature. Like Schwartz et al. Mu et al. propose to use colour information, but instead of extending the HoG features they use a variation of local binary patterns to overcome the lack of colour information in HoG. Zhu et al. further improved the work of Dalal and Triggs by using different block sizes in a rejection cascade using HoG features [223]. The work from Tuzel et al. also improved the results from Dalal and Triggs by using low-level features such as intensity, gradient, and spatial location combined by a covariance matrix [202]. The covariance matrices are not feasible with SVM's since they do not lie in a vector space. Hence, the authors propose a classification by LogitBoost classifiers combined with a rejection cascade designed to

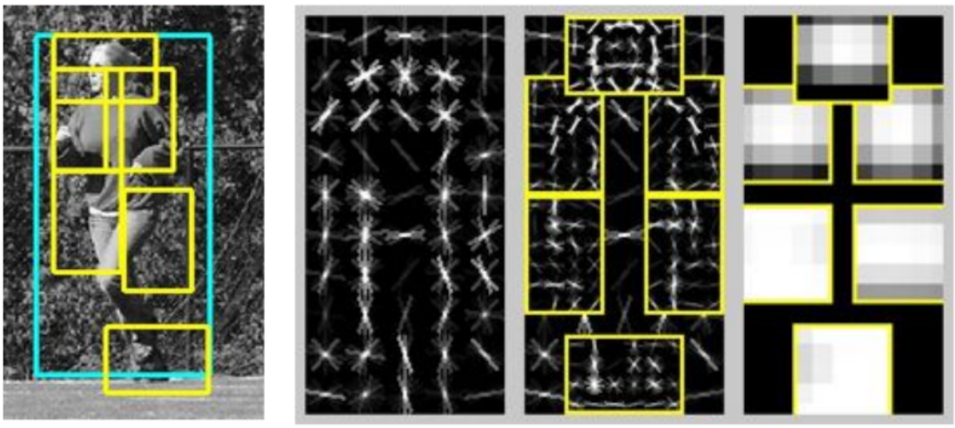


Figure 2.11: *Part-based-model.* From left to right: Example detection using a part-based-model. The model is defined by a coarse template and a spatial model for sub parts, which have a higher resolution. The features are based on HoG features, which are trained using each individual part of the detector. (Image found in [65])

accommodate points lying on a Riemannian manifold. Chen and Chen [45] combine intensity-based rectangle features and gradient-based features using a cascaded structure for detecting humans. Wu and Nevatia [218] describe a cascade-based approach where each weak classifier corresponds to a subregion within the detection window from which different types of features are extracted. The features are a combination of edge-lets [33], HOG descriptors [52], and covariance descriptors [202]. Maji et al. [141] apply HoG features as well, but in a multi-level version and a histogram intersection kernel SVM based on the spatial pyramid match kernel [127].

The other feature component part is evidence aggregation. Here, the features are not segmented from one sub window, but they are extracted from several parts, which are again combined in one **part-based model**. The part-based models have a long history in computer vision as object detectors and as human detectors as well. Here, the illustrated approaches are all utilized in human detection. Again, the part-based models can be divided in two major directions. The first one uses low-level features to model individual parts of the object and the second one models the topology of the human body. In the following, some recent part-based human detectors are briefly described (cf.[32] [131] [7]).

Mikolajczyk et al. [149] [148] divide the human body into different parts and utilise a cascade of detectors for each part. Based on deformable parts, Felzenszwalb et al. [64] [65] simultaneously learn part and object models and apply them to person detection, among other applications (see Fig. (2.11)). Applying logical reasoning, Shet and Davis exploit contextual information, augmenting the output of low-level detectors [184]. Tran and Forsyth [200] propose a mixture of two stages of part-based methods and windowing approaches. First, a window including the person is detected and a possible configuration of the person is estimated. Second, features for each part are extracted from the estimation. In a similar way, Lin and Davis [133] extract pose-invariant features in order to simultaneously detect and segment humans, while descriptors are calculated based on human poses. In [83] different people are tracked in crowded scenes by a learned torso classifier. A codebook representation is used as recognition technique, where appearance clusters are built from edge based features, which are shared among several object classes. Especially, the basic edge based features are shared by several object classes, while the features of the image are arranged in an efficient tree type hierarchical design. The human torsos are thereby represented by clusters of features. The hierarchical clustering detection offers minor occlusion of edge features of the torso. In [6] [173] [7] the authors detect the approximate articulations of humans through local features that model the appearance of individual body parts. Using a hierarchical Gaussian process latent variable (hGPLVM) they incorporate prior knowledge on possible articulations and temporal coherency within a walking cycle.

Classifier Component

After extracting one or several features a classifier has to decide, whether the actual window contains the object or not. Two popular choices are Support vector machines (SVM) or decision trees in conjunction with the AdaBoost framework. Other choices are relying on biologically inspired neural networks or different machine learning approaches, which excess the focus of this work and which are consequently not mentioned here. The interested reader is referred to [58] for further reading.

The **Support vector machines** provide a set of supervised learning methods, which predict, for a given input of two classes, to which class the input the belongs. A support vector machine learns a classifier margin out of labelled examples, which is a process of finding a separating hyperplane or set of hyperplanes with the largest margin to each class. The maximum of the distance to the corresponding classes assumes to build a better generalisation. Here, I give a short introduction in support vector machines, for further reading take a look in [58] or [42].

The name for support vector machines arise from the important parts of the training data. Some examples are closer to training examples from the other class and hence, are more important in the decision process. The training patterns, which are (equally) close to the separating hyperplane are called *support vectors*. The original formulation of SVM's stated the problem in a finite dimensional case, which can be resolved using a linear classification margin. But, it can happen that the problem is not linearly separable (e.g. the XOR-problem). This leads to a mapping from lower dimensional data to higher dimensions using Kernels. The needed cross product from the linear separable case can be defined through a kernel function, which permits to classify non-linear separable data. Generally, the data can be of any type, i.e. scalar, vector or intensity features.

Adaptive Boosting or AdaBoost is a machine learning algorithm invented by Freund and Schapire [72]. The first usage as a classifier was presented by Viola and Jones [204]. The algorithm is based on boosting, which improves the accuracy of any given learning algorithm by adding new component classifiers to form an ensemble whose joint decision rule has arbitrarily high accuracy on the training set [58]. In short, the idea is to train several weak classifiers, which work in conjunction better than one classifier on its own. AdaBoost is a variation on boosting as weak learners can be added as long as a desired training error is achieved. Additionally, the training patterns are weighted due to their “difficulty”. If a chosen pattern is misclassified, the weight is increased and if a pattern is correct classified, the weight is decreased. A higher weight ensures that a pattern will be used again and conversely, a pattern with a low weight will be skipped in further training iterations. This constrains the classifier to learn the important features out of the database, which separate the classes most. Summing up, the cascades yield a speed optimisation, AdaBoost has few parameters to tune and can be combined with any classifier to find weak rules.

The idea of **Multi-Layer-Neural-Networks** is to learn the non-linearity in the data at the same time as the linear discriminant, which divides the data in different classes. The area of neural networks is very large and most of the introduced algorithms do not rely on networks. Hence, I present in short the idea of the most famous approach of the multi-layer network trained through the back-propagation algorithm. Different layers of neurons provide a non-linear decision boundary. The problem is to find a good configuration of layers and neurons and a fitting parameter set, which can learn the accordant boundary of the provided data. The advantage of neural networks is the handling of raw data, i.e. no explicit feature extraction process is needed. Instead, the network learns the important features directly out of the data.

Here, I presented a lot of approaches for detecting humans, which could be utilized to segment a hypotheses out of image data. Histograms of oriented gradients and part-based models showed high potential in the detection rate, while background extraction or moving foreground detection are not promising on a mobile platform.

In the following, systems have to keep track of found hypothesis in order to build a knowledge base of each movement of a human in the scene. This can be done by detecting each hypotheses from frame to frame and to correlate all detections with each other (*tracking by detection*) or through a particular tracking algorithm, which is not based on global category features, but on individual features for each hypotheses (*object tracking*). The object tracking approach offers a more stable tracking, as the features more distinguish each object from the background than the global detection. In the following section, I give a general introduction in the possibilities of tracking algorithms.

2.3 Visual Tracking

Visual tracking has got the function to recover the position of a target over time. This can be e.g. the 2D image position or the placement of the object in real world. Tracking is very important for this work, because the gathered knowledge has to be put in a temporal continuity. Hence, tracking is defined by the following statement:

Definition 2 (Object Tracking) *:*
Tracking is the process of creating temporal links of the occurrences of a moving object, which recovers the object’s path.

In this section the most important tracking algorithms are described without guarantee of completeness, because of the broadness of this discipline. Tracking is studied by many researchers with elusive methods over many years, which has to be restricted to fit into this work. Hence, the section first defines the possible compositions of a tracker and subsequent, it provides the most important configurations in literature.

> *"Two major components can be distinguished in a typical visual tracker. Target Representation and Localisation is mostly a bottom-up process which has also to cope with the changes in the appearance of the target. Filtering and Data Association is mostly a top-down process dealing with the dynamics of the tracked object, learning of scene prior, and evaluation of different hypotheses."*
> Comaniciu 2003, [49, p. 1]

In each tracking problem the combination and weighting of bottom-up and top-down parts have to be reconsidered in order to track a target robust and efficient. In some applications it is an advantage to rely more on target representation than on dynamics, while in other scenarios it could be more robust to consider the motion of a target. Target representation is e.g. better in face tracking in crowded scenes and considering the dynamics would benefit in aerial surveillance, where target and camera motion are more important. Additional requirements are necessary, if the system should run in real-time, because only a small amount of the processing power can be assigned to the tracking itself. Preprocessing stages or high-level tasks such as detection, recognition or reasoning consume much available processing time. Therefore, the computational complexity of a tracker should be kept as low as possible.

The first step in a tracker is to define a target model. This model can reach from simple point representations to complex feature models. Generally, existing feature tracking techniques fall into two areas: *Correspondence* based techniques and *texture correlation* based techniques. Correspondence based techniques extract a set of features from frame to frame and try to establish a connection between each corresponding feature in two subsequent sets. On the other side, texture correlation based techniques extract features from a window and try to find globally a best fit in the subsequent image.

Tracking is done most powerful, if the chosen features discriminate most between the object and the actual background. In [48] it is proposed how the features could be chosen by computing a log likelihood ratio of class conditional sample densities from object and background. The feature selection algorithm is embedded in a mean shift tracking algorithm that adaptively selects the top-ranked features. Of course, more complex features require often more computation time, which also has to be considered. Hence, the features should be chosen problem dependant in a way that the tracking is also fast and most robust.

2.3.1 Feature Tracking

One typical correspondence based technique is point tracking. Point tracking is known from optical flow (Sec. (2.2.1)), where points like corners or other descriptive points are temporal related. More sophisticated approaches like SIFT (Scale-invariant feature transform) [135] or SURF (Speeded up robust features) [18] incorporate a small neighbourhood of the point, which ensures that the feature is more stable for matching and recognition. The drawback of such simple features are the impracticality to extract more stable region features and the correspondence complexity, which point belongs to which in the ongoing frame.

To overcome these drawbacks, most trackers use a complete region, where features can be directly extracted. Blob tracking is one the first attempts to track an image region. In [69] they describe a way to establish a temporal relationship between different blobs. They use a coarse to fine resolution of the image to build up a graph of correspondences to track not only large and slow objects, but also small and fast ones. They show interesting results over some video-sequences (see Fig. (2.12)).

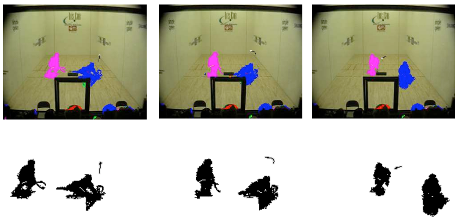


Figure 2.12: *Multi-Resolution Blob tracking.* In each frame two players and the ball of a racquetball game are tracked. At the bottom the ground truth is manually labelled. (Image found in [69])

More about Blob tracking could be found in [105]. They use a cylinder model to track multiple persons with a multi-blob likelihood function.

2.3.2 Multiple Feature Tracking

If one feature is not sufficient to describe and track the object, a mixture of several features could produce relief. The easiest approach consists of calculating several features and trying to find correspondences between them in time. The multiple feature tracking could consist of different types of features. In [12] the authors propose to track objects through a colour based particle filter and an adaptive template tracker, where the priority between both cues could be dynamically switched. Here, it is important to mention that each additional feature increases the computation time, which should be avoided for real-time systems. Ensemble Tracking e.g. is a collection of weak classifiers combined to a strong classifier using Adaboost. The strong classifier is trained to separate back- and foreground. The maximum or best position of the object is found using mean shift. Coherence is achieved through updates with new weak classifiers matching the actual object condition [11].

2.3.3 Template Tracking

One easy way of target tracking is to use *templates*. The algorithm uses a snapshot or image snippets from a database in order to compare them to each sub-window of the search-image. The best match is computed by the difference of the intensity of each point from the template to each point from the search images. The bast match is taken as actual position, if the difference is above a certain threshold. The tracking could be speeded up, if only the sub-windows in a specific area around the object are taken into account. An efficient and robust version of the Lucas-Kanade template matching algorithm is presented from [178], which estimates robust parameter over many frames and corrects the template drift.

2.3.4 Kernel Tracking

Kernel tracking is an important method to regularise target representations by spatial masking with kernels, which creates spatially-smooth similarity functions suitable for gradient-based optimisation [49] [8] [88]. The optimisation is often induced by similarity between target model and target candidates measured using some metric like the Bhattacharyya coefficient [49]. Comaniciu et al. propose background weighted m-bin colour histograms, with a target model $\hat{q} = \{\hat{q}_u\}_{u=1...m}$ and target candidate $\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1...m}$. Thereby, the target model is represented by an ellipsoidal region in the image, which weights pixel farther away from the centre less by using again an isotropic kernel. The similarity function defines a distance between the target and the candidate model. It has the form $\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}]$ and is masked with an isotropic kernel in the spatial domain in order to transform $\hat{\rho}$ to a smooth function in y. The used metric is the Bhattacharyya coefficient $\hat{\rho}(y) = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y), \hat{q}_u}$. The current location of the target is then found by minimising the distance as a function of y. The localisation starts from the last known position and searches the neighbourhood. Through the usage of the kernel, the gradient information provided by mean shift can be used. Han et al. combine kernel density tracking with the mixture of Gaussians approach in order to get a non-parametric method with variable components [88]. The outcome is a tracking system, which can handle multi-modal densities while modelling the target appearance online. Thereby, the update rate of the density representations is critical for the tracking process and the system is not ideal for full-occlusion sequences.

2.3.5 Tracking using Filters

One of the most common formulations of the filtering and data association approaches is through the state space approach in order to model discrete-time dynamic systems [15]. The information about an object is modelled by the state sequence $\{x_k\}_k = 0, 1, ...$ and its evolution over time by the dynamic equation $x_k = f_k(x_{k-1}, v_k)$, with v_k noise. The dynamic equation is typically specified by a motion or transition model, which transfers the state x_{t-1} from the previous step to the current moment x_t. For each time step there are measurements $\{z_k\}_k = 1, ...,$ which are related to each state through the measurement function $z_k = h_k(x_k, n_k)$, with n_k as noise. Both noise values $\{v_k\}_{k=1,...}, \{n_k\}_{k=1,...}$ are expected to be independent and identically distributed.

Tracking using filters is defined as the estimation of the state x_k given the actual measurements $z_{1:k}$ and the dynamic equation. This is equivalent to the calculation of the probability density function (pdf) $p(x_k|z_{1:k})$. The recursive Bayes Filter is the theoretically optimal solution for this problem. It possesses two essential steps. The first step is called the control update or *prediction*. In this step the actual state x_k is calculated from the prior state x_{k-1} and the probability that the dynamic equation induces a transition from x_{k-1} to x_k. Next, the *update* step computes the posterior pdf $p(x_k|z_{1:k})$, which means the probability that the current state is x_t given the actual measurement z_k. The algorithm is recursive as the posterior is calculated by the knowledge about previous states. This requires an initial state $\{x_0\}$, which should centre all probability mass on a correct value $\{x_0\}$ and assign zero probability anywhere else. It is possible to start with an unknown value $\{x_0\}$ as well, resulting in an uniform distribution or to start with a partially knowledge expressed by non-uniform distributions. In the field of tracking, the first case is the most common, as we have a target model, which we try to track over the time.

In literature exists quite a variety of techniques and algorithms that are all derived from the Bayes Filter, where each technique relies on a different assumptions regarding the state transition probabilities, the initial belief and the measurement [199]. The different filters model the approximation of the posterior distribution in a different way, which has an important impact on the complexity and the approximation of the algorithm. Therefore, if one is choosing an approximation, following properties should be considered.

- Computational efficiency: How time consuming is the finding of a solution
- Accuracy of the approximation: Which distributions can be approximated with the selected algorithm
- Ease of implementation: Code maintenance and implementation time should be kept in mind.

If the noises $\{v_k\}_{k=1,...}, \{n_k\}_{k=1,...}$ are Gaussian and the functions f_k, h_k are linear, the *Kalman Filter* (KF) [210] provides the ideal solution [15]. The posterior is also Gaussian in this case. The presentation of the posterior as a Gaussian is characteristic for many tracking problems as the Gaussian is uni modal and has only one maximum with a lower probability around it and the searched object has one true state and a small margin of uncertainty. A Gaussian has the following form:

$$p(x) = det(2\pi \sum)^{-\frac{1}{2}} exp\{-\frac{1}{2}(x-\mu)^T \sum -1(x-\mu)\} \tag{2.18}$$

The Gaussian density $p(x)$ is described by the mean μ and the covariance $\sum$, which is symmetric and positive-semi definite.

If the functions f_k, h_h are not linear, there is one possible solution using the *Extended Kalman Filter* (EKF) [15], which uses linearisation to model the posterior density still as a Gaussian. An alternative is the *Unscented Kalman Filter* (UKF) [110]. Another possibility is the use of sequential monte-carlo methods or respectively particle filters [104].

Kalman Filter

The Kalman Filter was invented in the 1950s by Rudolph Emil Kalman [112]. The discrete Kalman Filter can only deal with continuous state spaces and it is not applicable to discrete or hybrid state spaces [210].

Kalman Filters are beliefs at time t, represented by the mean μ_t and the covariance $\sum_t$. Three properties must hold in order to represent the posteriors as Gaussian.

(i) The probability for the next state $p(x_t|u_t, x_{t-1})$ must be *linear* in its arguments with added Gaussian and white noise ε_t, which is expressed in the following equation.

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \tag{2.19}$$

x_t and x_{t-1} are vertical state vectors and u_t is the control vector. A_t and B_t are matrices with according size, such that the state transition function becomes linear in its arguments.

(ii) The measurement probability $p(z_t|x_t)$ must also be linear in its arguments, with added Gaussian noise

$$z_t = C_t x_t + \delta_t \tag{2.20}$$

C_t is a matrix, with the dimension of the measurement vector. δ_t corresponds to the measurement noise, which is a Gaussian with zero mean.

(iii) The initial belief $bel(x_0)$ has to be normal distributed.

Considering the three assumptions in addition to the Markov assumptions, the posterior $bel(x_t)$ is always a Gaussian, for any time t. The assumption that the noise is both white and Gaussian means that the noise is not correlated in time and that its amplitude can be modelled by an average and a covariance [36]. For the mathematical proof take a look in [199]. Using these assumptions it is possible to build a model for the state of the system that maximises the a posteriori probability of previous measurements.

The Kalman-Filter estimates the process state by two equations, the *time update equation* and the *measurement update equation* (see Fig. (2.13)). The time update equation projects the current state and error covariance forward in time to obtain the a priori estimates for the subsequent time step. The measurement step incorporates the measurement into the a priori estimate to obtain an improved a posteriori estimate. Both equations are often called *predictor* and *corrector* equations.

If the process to be estimated and (or) the measurement relationship to the process is non-linear, the original Kalman-Filter can not be applied. One possible solution is provided by the *extended Kalman-Filter* (EKF) or the *unscented Kalman-Filter* (UKF).

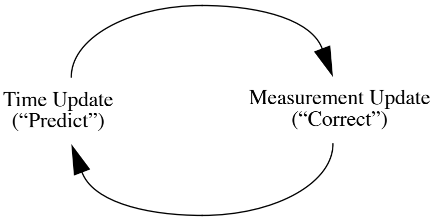


Figure 2.13: *Kalman-Filter cycle.* The *time update* predicts the current state and error covariance to the next time step. The *measurement update* corrects the initial estimate by incorporating the measurement (Image found in [210])

The EKF overcomes the linearity assumption and represents the state probability and measurement probabilities by non-linear functions g and h:

$$x_t = g(u_t x_{t-1}) + \varepsilon_t \tag{2.21}$$

$$z_t = h(x_t) + \delta_t \tag{2.22}$$

The functions Eqn. (2.19) and Eqn. (2.20) are generalised through this model by replacing the matrices A_t and B_t with g in Eqn. (2.21) and the matrix C_t with h in Eqn. (2.22). But the Bayes filter does not possess a closed-form solution for the non-linear functions. Hence, performing a belief update exactly is usually impossible. Instead, the extended Kalman Filter approximates the true belief by a Gaussian [199]. The difference of the EKF and UKF are explained with the following citation.

> *"A central and vital operation performed in the Kalman Filter is the propagation of a Gaussian random variable (GRV) through the system dynamics. In the EKF, the state distribution is approximated by a GRV, which is then propagated analytically through the first-order linearisation of the non-linear system. This can introduce large errors in the true posterior mean and covariance of the transformed GRV, which may lead to sub-optimal performance and sometimes divergence of the filter. The UKF addresses this problem by using a deterministic sampling approach. The state distribution is again approximated by a GRV, but is now represented using a minimal set of carefully chosen sample points. These sample points completely capture the true mean and covariance of the GRV, and when propagated through the true non-linear system, captures the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any non-linearity. The EKF, in contrast, only achieves first-order accuracy. Remarkably, the computational complexity of the UKF is the same order as that of the EKF."*
> Wan 2002, [206, p. 1]

In general the use of EKF or UKF in the area of object tracking is advantageous compared to the standard KF, because of the better approximation of the underlying posterior. A successful proposal of contour tracking using unscented Kalman-Filter in conjunction with HMM's is shown in [96].

Particle Filter

Another technique for tracking is the *particle filter* (PF), also known as *Sequential Monte Carlo methods* (SMC). The particle filter is especially useful to represent multiple hypotheses simultaneously. This is e.g. needed, if the tracked object is occluded and the tracker has no measurement for a specific amount of time. The particle filter is a non-parametric implementation based on the Bayes-Filter. The idea of particle filters is to represent the posterior $bel(x_t)$ by a set of random state samples drawn from this posterior, which approximate the final state. Due to the non-parametric design the particle filter can represent much broader space of distributions than e.g. the Kalman-Filter based on Gaussians.

The samples of the posterior distribution are called particles

$$\Phi_t = x_{i,t} \quad \text{with} \quad i = 1 \dots M \tag{2.23}$$

Each particle $x_{i,t}$ is a complete instantiation of the state at time t. The number of particles M in the set Φ_t is often large to approximate reliably the belief $bel(x_t)$.

$$x_{i,t} \quad \text{with} \quad i = 1 \dots M \tag{2.24}$$

If the number of particles is very large $M \Rightarrow \infty$ the particle filter is proportional to the Bayes filter posterior.

Like the Kalman-Filter the particle filter constructs the belief $bel(x_t)$ recursively from the previous belief $bel(x_{t-1})$. The particle set Φ_t is constructed from the set Φ_{t-1} one time step before. Thereby, each particle has got an *importance factor*

$$w_{i,t} = p(z_t | x_{i,t}) w_{i,t-1} \tag{2.25}$$

which incorporates the measurement z_t into the particle set. The importance factors can be interpreted as *weight* of a particle, where the complete set represents the Bayes filter posterior $bel(x_t)$. The weights are initialised with 1 in the first step and recalculated dependent on the accordant measurements. If the particles are all kept and only re-weighted, many particles would end up in regions with low probability. Accordingly, the particles are re-sampled corresponding to their importance called *Sequential Importance Resampling* (SIR) [84]. If an importance weight is low, the particle is replaced by a particle copy of a particle with high importance. This forces particles back to the posterior $bel(x_t)$. Fig. (2.14) clarifies the sequential importance re-sampling.

The typical procedure of a particle filter is as follows:

(i) Generate a hypothetical state $x_{i,t}$ for time t based on the particle $x_{i,t-1}$ and the control u_t (cf.Kalman Filter) for each particle $i = 1 \dots M$. This includes sampling from the next state distribution

$$p(x_t | u_t, x_{t-1}) \tag{2.26}$$

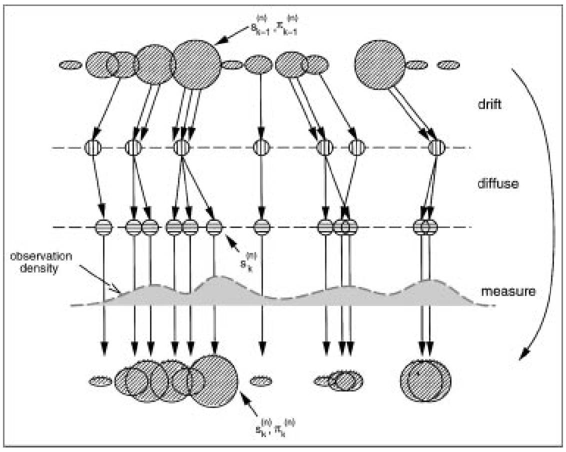


Figure 2.14: *Particle filter importance re-sampling.* One time stamp in the particle filter process. The first step (drift) keeps only the important particles with high weight. The next step (diffuse) re-samples from the best particles and the final step (measure) incorporates the new measurement in order to estimate the posterior of the actual state. (Image found in [104])

(ii) Calculate the importance factor $w_{i,t}$ for each particle $x_{i,t}$ in order to incorporate the measurement z_t.

(iii) Use Sequential Importance Re-sampling to better approximate the true posterior $bel(x_t)$

The first step arranges the sampling of the particles, where the transition probability, see Eqn. (2.26), models the drawing of the particles.

One particle filter technique, which uses SIR with transition prior as importance function, is the *Condensation* algorithm [104]. The authors applied the Condensation algorithm to the problem of tracking curves in dense visual clutter. They use learned dynamical models for their filter, which propagate together with visual observations the particle set over time. The Fig. (2.15) shows three exemplary results for the Condensation algorithm.

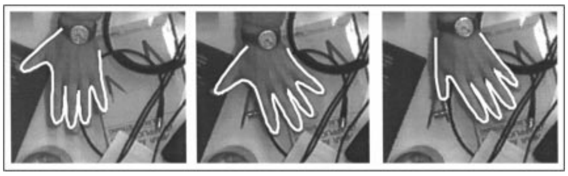


Figure 2.15: *Tracking result from Condensation.* The Condensation algorithm is applied for curve tracking. The images show three results for the hand model, where the algorithm uses 500-1500 samples per time stamp.(Image found in [104])

Many other particle filters exist in literature [109] [153] [116] [140] [132]. *Rao-Blackwellized particle filter* are e.g. used for motion estimation as alternative to Structure from Motion (SfM) [2]. *Variational particle filter* are applied for multi object tracking [109]. They use a mixture of a non-parametric contour model and a non-parametric edge model to represent the object using kernel density estimation. The *Auxiliary particle filter* (APF) tries to deal with tailed observation densities by using auxiliary variables and reference points [163].

> *"The APF is a lookahead method, where at time n we try to predict, which samples will be in region s of high probability masses at time $n+1$."*
> Doucet 2008, [57, p. 23]

Hence, the auxiliary particle filter uses an intermediate step, which measures the likelihood at some points beforehand. Thereafter, the conditional samples are drawn.

2.3.6 Tracking by Multiple Models for Adaptive Estimation

Interacting multiple models (IMM) [30] have shown good results in the case of motion uncertainties [146] [108]. The idea is to use more than one motion model in order to represent different possible motions. Each motion model is incorporated in an elemental filter (e.g. Kalman), which are merged to calculate the final posterior state. The probability that the object changes its displacement mode is encoded in a transition probability matrix (TPM). One cycle of an IMM is composed of several parallel steps [43]. The new data z is used with the previous state $P(x_{t-1})$ to update each filter. The TPM is also updated in this step according to the likelihood of observation with filter internal prediction. The final state is computed by a fusion of each filter and the TPM. IMM's have also been successfully applied in the area of human tracking [62] [43]. An IMM estimator performs significantly better than a Kalman filter, if the manoeuvring index λ of a target is above a certain threshold.

$$\lambda \triangleq \frac{\sigma T^2}{\sigma_w} \tag{2.27}$$

$\sigma T^2/2$ relates to the motion uncertainty and σ_w to the observation uncertainty. Intuitively, the higher the uncertainty the more the tracking benefits from a versatile tracker. (cf.[16, p. 281])

Actual work in literature is about automatic online estimation of the transition probability matrix [43] in order to create a parameter set for the IMM fitting on the real data.

2.3.7 Tracking by Joint Probabilistic Data Association Filter

In the case of diverse measurements, whose origin is uncertain, data association is of main interest. Especially in the case of noisy data or multiple objects in the scene data association cares for selecting measurements, which update the state of the target of interest. Additionally, it is important to determine, if the filter has to be modified in order to account for the data association uncertainty [14]. As filter mostly Kalman or Extended Kalman filter are applied. The difference of data association compared to a Kalman filter is located in a few additional steps (Valid for single object tracking. Multi object tracking acquires additional steps):

(i) The measurement has to be validated at each time

(ii) For each validated measurement, an association probability has to be computed to weight the measurement in the combined innovation.

(iii) "The final updated state covariance accounts for the measurement origin uncertainty."[14, p. 90]

In literature, there exist different data association algorithms, which are namely the probabilistic data association filter (PDAF), joint PDAF (JPDAF) for multiple objects [165], mixture reduction PDAF (MXPDAF), particle filter (PF) and multi-hypothesis tracker (MHT).

There exist also the approach of connecting IMM and JPDAF into one system, which simultaneously avoids track coalescence through JPDAF and tracks multiple manoeuvring targets through IMM [31].

2.4 Summary

In this chapter I described the visual basis for the subsequent algorithms and methods. The sensory part described the data acquisition process, especially how three dimensional data and two dimensional projective images are created. I stated actual human detection algorithms, where Histograms of oriented Gradients showed a high classification rate and good stability. The detection process is generally time consuming due to the sliding window approach or the feature calculation complexity. Here, the thesis proposes solutions to circumvent this overhead. Finally, I presented the most important tracking algorithms, where each tracker has particular strength in certain areas. The most promising capabilities are shown by the particle filter, because the particle distribution accounts best for the uncertainties arising in a mobile robot scenario. Ego motion, fast changing movements and occlusions have to be considered in order to keep reliably track of each entity. In the following Chapter (3) a solution to realise the first category of situation awareness is presented. The system is based on 3D data and uses a particle filter for tracking in order to build a coherent model of the environment. In Chapter (4) I state a solution for the second category of situation awareness on a mobile robot, where the good detection characteristics of the Histograms of oriented Gradients approach are further refined and the particle filter tracking process is extended to 2D and 3D data simultaneously. In both parts data association is assured due to a hypotheses management, where all incoming measurements are correctly associated to each present entity. Chapter (5) does not rely on detection and tracking, but is shows how the outcome can be utilised to direct further perception.

3 Acquiring 3D Scene Models in Vista Spaces

Embodied agents, both humans and mobile robots, have to perceive, to analyse and to segment an observed scenery into meaningful parts to deal with and communicate about the unknown and dynamic environment. Here, I want to present a 3D scene analysis approach, which enables mobile robots to solve such problems by gathering broad knowledge about their environment only by observation of the scenery. This implements the first part of a situation awareness described in the first chapter.

The following research questions arise:

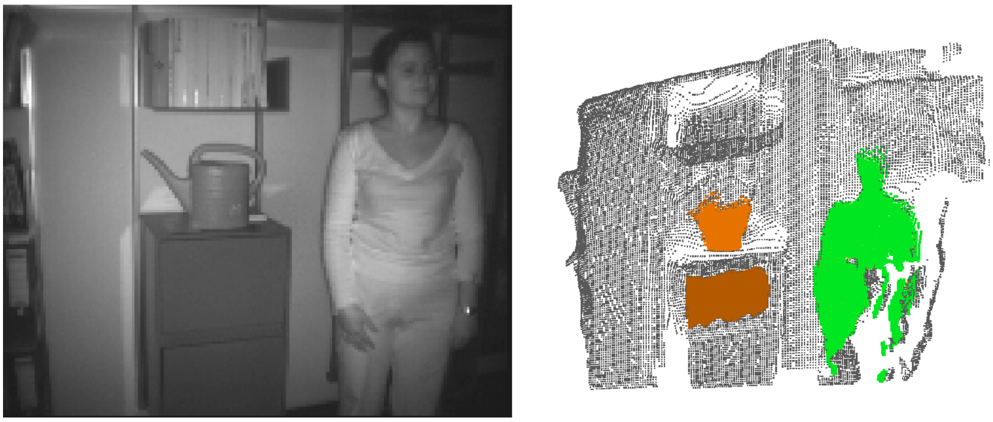
- *How should the environment be modelled in order to represent all information in it?*
- *Which parts are the most important for a mobile robot?*
- *How can the model be updated with new information?*
- *How can the model support the perception of every single part?*

In general, the robot needs information about different parts of the world. First, the robot has to detect and track humans as possible interaction partners or to learn their typical movement pathways. Second, the static scene parts like walls, cupboards or tables have to be segmented to give a broad knowledge about the room structure for e.g. navigation purposes [221] or room classification [197]. In contrast to other typical background modelling approaches [189] [117] [150], the suggestion is to distinguish as well between static objects and objects like chairs, teddy bears or other smaller objects that can be moved by an agent. Instead of building a complex ontology of human environments to describe which parts may be moving or could belong to the static background and equipping the robot with strong detectors for every possibility, it is obviously better to learn an *articulated scene model* on the basis of scene observation. This bottom up learning of a spatial awareness enables a mobile robot to extract essential knowledge about the environment which can only be achieved by observation. The articulated scene model is composed of the following three scene parts.

Definition 3 (Articulated Scene Model) *:*

- *Static scene (never changing parts)*
- *Moving entities (e.g. humans or robots)*
- *Movable objects (e.g. chairs, doors)*

This model is updated in one single and simultaneous computation. The figure 3.1 is meant to give an example. On the left the accordant frame of the scene is presented and on the right an example of a 3D articulated scene model is shown. Coloured in black is the static background, in orange and brown are two articulated objects and in green the actually tracked human is displayed.



(a) 2D amplitude image

(b) 3D point cloud

Figure 3.1: *Articulated Scene Model.* In the left image the frame of an example sequence is shown. In (b) two detected articulated scene parts are shown (cupboard door, water can) in red and orange, which emerge after a few seconds of observation, if the specific object is moved by an agent. The Gray 3D points belong to the background and the green points to the currently tracked person.

Usually, the observation of an environment refers to the large-scale-space [124] [151], where a main property is the necessity of locomotion to perceive the space, which could be, e.g. , a complete flat or apartment. In the proposed system the observation is applied on the so called *vista space*, which describes the visual field only by slightly moving the gaze.

Definition 4 (Vista Space) *:*
The vista space is a part of the world, which can be viewed at the same moment only be slightly moving the gaze.

This means that the system relies on the perception of a single room or parts of a room and that the robot does not move during the perception of one vista space. As the robot should not analyse externally one vista space, the short observation time limits the number of available frames. By the use of the vista space one can derive the assumption that the farthest measurement in the scene describes the background. If an object appears in front of previously seen static parts, one can assume a moved object, while upcoming observations of more distant points indicate a removed object.

Assumption 1 (Vista space assumption) *:*
The farthest measurement in the scene describes the background.

As vista space models deliver complementary information to large-scale space models the combination of both model types into a common representation e.g. using the Hybrid Spatial Semantic Hierarchy (HSSH) of [19] will form the foundation for modelling spatial knowledge of the entire environment an agent interacts with. However, in the following the focus relies on the analysis of the vista space as the system could be used in the home-tour scenario [46], in which a user guides the robot around his flat and particular vista space situations arise.

The robot needs a meaningful sensory input to perceive the environment, which in the following case is achieved by using a time-of-flight 3D camera. The 3D data is extended with additional 3D velocities using optical flow. The use of a 3D sensor translates the problem to an inherent 3D interpretation task.

The proposed system builds the articulated scene model only by observing the 3D scenery for a few seconds, thereby segmenting the environment into different parts and incorporating the already gained knowledge. The humans in the observed scene are detected by consideration of velocity information and a weak object model suitable for many different kinds of objects. The human is tracked by a hybrid particle filter with mean shift, which enables the robot to keep track of the movements of the human. The calculated trajectories supply a broad knowledge about the typical movement areas in the scene and additionally, the robot gets the required positions of possible interaction partners.

In contrast to other background modelling strategies, the articulated parts of the scene are separated from the static scene. Usually, the articulated parts are incorporated again into the background model after the objects become static again. Even with a strong detector the articulated objects are hard to detect as they could have any shape or size. Here, the articulated parts are detected through the vista space assumption revealing the objects by an intelligent modelling process through observation.

The static scene is composed of the remaining parts after excluding the persons and the movable objects. Through the exclusion of dynamic parts the static scene is very reliable for navigation or scene classification as many potentially changing parts have been already removed.

However, the main advantages of the proposed system are based on the parallelism and the generality of the detection of the different parts of the articulated scene model. Through the detection and exclusion of moving persons and movable objects the building of the static scene is much more robust. On the other hand the knowledge about the static scene enhances the detection of humans as the static background could be subtracted and the detection can be limited to dynamic parts of the current observation. The static scene again is used in the assumption of the vista space to detect the articulated scene parts. In contrast to the existing approaches, movable objects can be detected without the explicit detection of a particular object's movement but through the knowledge about the static scene and the information from observation.

The contribution of the proposed model is a solid basement of information for situation awareness, which could be used by the robot as input for further processing. In the following, I want to present some ideas or possible applications for the articulated scene model. The possibilities are comprehensive as the model is a good starting point for several learning or interaction scenarios. As mentioned before, the tracked persons or moving objects could be directly associated as interaction partners. On the other hand, the information about their movements can be used as data for typical movement areas or pathways, which could be used for navigation purposes of the robot. The articulated parts apparently enable the robot to recognize objects, which are handled by the human or more simply, which objects are movable. This knowledge could be utilized in a tabletop learning scenario, where each object put onto the table could be easily recognized as a new object independent from its topology or appearance. Again, using the whole information about the recognized objects and the appearance and disappearance areas the robot gets an idea about the action spaces of these objects. If it is a door, the robot could see the articulation or the opening range of the door as an action area. Several other scenarios are imaginable, but as the main contribution is a solid basement of knowledge for a mobile robot I skip further suggestions how to use the articulated scene model in a specific application.

The presented thesis has been developed in cooperation with my colleagues Agnes Swadzba and Joachim Schmidt. The work has been previously published in [194] [26] [27] [195] [25]. My particular focus in this thesis lies on the human detection and tracking in combination with the articulated scene model.

The chapter is structured as follows. First, related work is presented in Sec. (3.1) to give an overview of other research in the field of scene analysis. The proposed system in general is described in the subsequent Sec. (3.2). The preprocessing of the sensor data is explained in Sec. (3.3) and the computation of the scene flow in Sec. (3.4). The detection and tracking of moving entities is described in Sec. (3.5), followed by the description how to build the static parts and how to detect the movable parts of the articulated scene model (Sec. (3.6)). In the end, I shall explain the experiments and present the results of the algorithm on several self-created data sets in Sec. (3.7).

3.1 Introduction in Scene Analysis

Research on dealing with dynamic scenes has become more and more important since the manual analysis of the huge amount of video data provided by video surveillance is not suitable any more. Diverse methods have been developed to model the background that can be subtracted from the current image to extract the moving foreground (Sec. (2.2.1)). The problem of a moving camera has to be considered, if the approaches for detecting moving regions developed in a surveillance scenario are transferred to a robotic scenario. This can be done directly by an ego-motion compensation [177], by visual navigation [130] or by detecting moving objects through inconsistencies in a scene motion field arising from a optical flow computation [119]. Another problem in robotics scenarios is the short observation time and the unknown environment so that a previous training of the background is not possible. Therefore, Hayman and Eklundh [91] developed a Bayesian model for incorporating the possibility that the background has not yet been uncovered.

Besides from moving persons also movable objects are interesting for a robot. Movable objects are characterized by occasional relocation and longer static periods. In classical background subtraction approaches such objects will be integrated into the background model after relocation thus cannot be detected any more [162]. Sanders et al. [169] try to solve this problem by integrating pixel information over time. The pixel history is clustered to temporal coherent clusters, the so-called temporal signatures, which allows to detect quasi-static objects under the condition of these objects having arrived and departed from the scene. As movable objects belonging to the class of scene structuring elements like a chair are of special interest for a robot some approaches try to find such scene elements through analysing the human activity instead of detecting them directly. For example, trajectories can be segmented to actions using Hidden Markov Models (HMMs) [162] concluding that the location of an action point to an object associated with an action like, e.g. , "sitting down" is coupled with a chair. Alternatively, clustering of motion histograms computed per scene cell allows an image segmentation providing interesting indoor scene regions like a sofa [53]. The analysis of trajectories of moving objects can reveal – besides image regions that correspond to scene elements – general semantic regions like junctions or paths that do not match a specific movable object. Analysing person trajectories in indoor rooms could reveal semantic regions like a grouping of table and chairs [122]. Analysing person activities and car trajectories in outdoor environments could provide models of roads and paths [208], "walkable" ground surfaces [39], or routes, paths, and junctions [142]. A detailed review of further methods for understanding scene activity is given in [44].

In the case of detecting movable objects, e.g. , a door, which motion is caused by a human manipulation [169], trajectories of such objects reveal their possible articulation. Inspired by articulated body models, Sturm and colleagues [191] developed techniques for learning kinematic models of scene elements like table or drawer. As tracking of such objects is a challenging problem they bypass it in their paper through attaching markers to test objects. In their last paper [190] they have presented an automatic tracking of a planar surface from a cupboard door or a drawer front for observation situations restricted to a close-up view of the surface.

The proposed articulated scene model aims to combine background modelling with detection of semantic scene elements. As the focus relies on the modelling of dynamic 3D scenes the assumption that static measurements which are furthest away determine the scene background allows an elegant way to model the background especially in robotic scenarios where observation times are short. Subtracting the background in 3D reveals directly quasi-static/articulated objects without special requirements like an object has to arrive and depart [169] and independent from their shape or size or the human activity connected to them. Detecting arbitrary articulated scene elements using human activity requires recognition abilities of a lot of different daily-life activities which means that a huge database of all possible actions is needed for training. The approach provides for 3D data a bypass to this exhaustive learning problem.

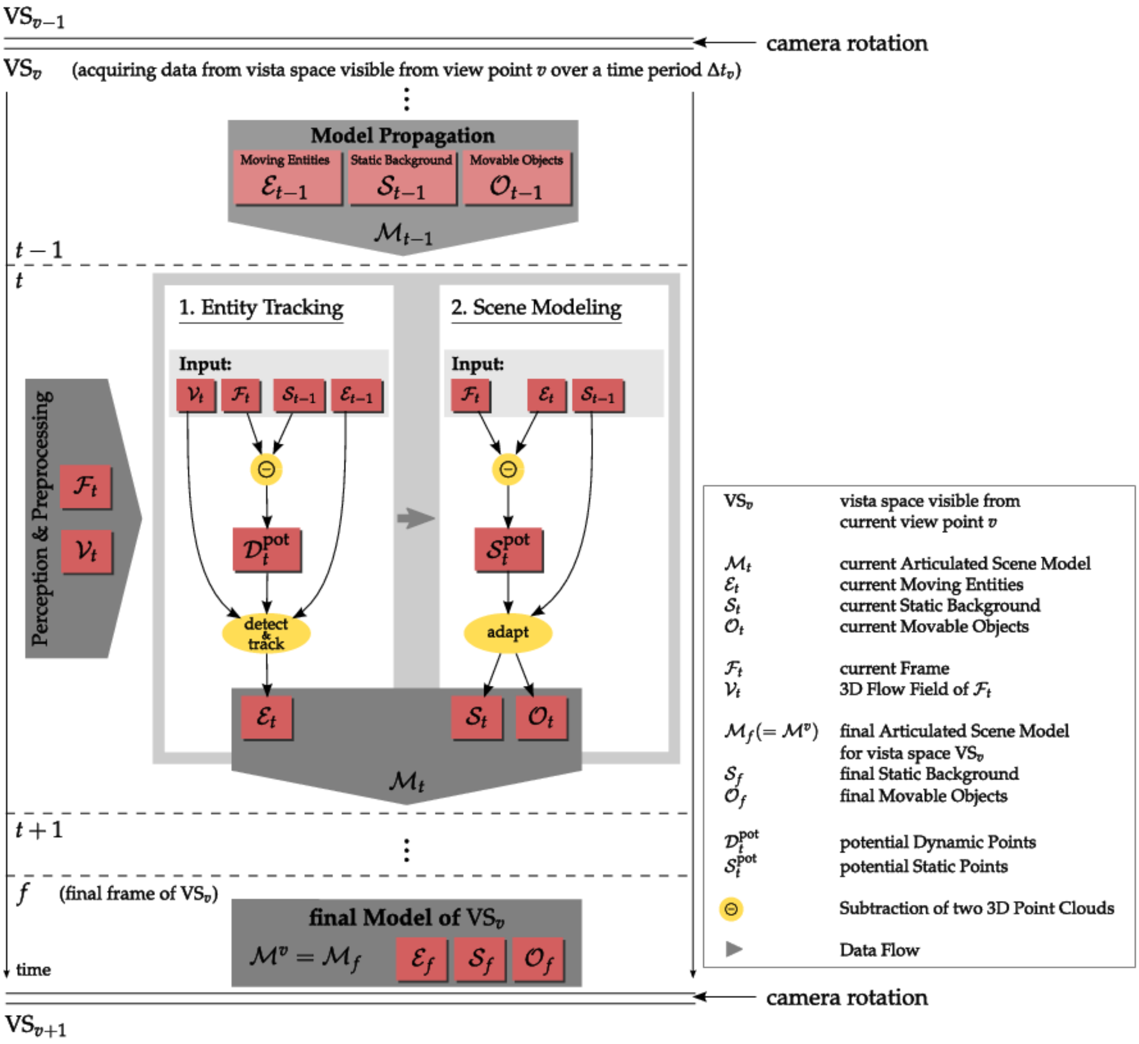

VS_{v-1}
camera rotation
VS_v (acquiring data from vista space visible from view point v over a time period Δt_v)
Model Propagation
Moving Entities $\mathcal{E}_{t-1}$
Static Background $\mathcal{S}_{t-1}$
Movable Objects $\mathcal{O}_{t-1}$
$\mathcal{M}_{t-1}$
$t-1$
t
1. Entity Tracking
Input:
$\mathcal{V}_t$ $\mathcal{F}_t$ $\mathcal{S}_{t-1}$ $\mathcal{E}_{t-1}$
$\ominus$
$\mathcal{D}_t^{pot}$
detect & track
$\mathcal{E}_t$
2. Scene Modeling
Input:
$\mathcal{F}_t$ $\mathcal{E}_t$ $\mathcal{S}_{t-1}$
$\ominus$
$\mathcal{S}_t^{pot}$
adapt
$\mathcal{S}_t$ $\mathcal{O}_t$
$\mathcal{M}_t$
Perception & Preprocessing
$\mathcal{F}_t$
$\mathcal{V}_t$
$t+1$
f (final frame of VS_v)
time
final Model of VS_v
$\mathcal{M}^v = \mathcal{M}_f$ $\mathcal{E}_f$ $\mathcal{S}_f$ $\mathcal{O}_f$
camera rotation
VS_{v+1}
VS_v vista space visible from current view point v
$\mathcal{M}_t$ current Articulated Scene Model
$\mathcal{E}_t$ current Moving Entities
$\mathcal{S}_t$ current Static Background
$\mathcal{O}_t$ current Movable Objects
$\mathcal{F}_t$ current Frame
$\mathcal{V}_t$ 3D Flow Field of $\mathcal{F}_t$
$\mathcal{M}_f (= \mathcal{M}^v)$ final Articulated Scene Model for vista space VS_v
$\mathcal{S}_f$ final Static Background
$\mathcal{O}_f$ final Movable Objects
$\mathcal{D}_t^{pot}$ potential Dynamic Points
$\mathcal{S}_t^{pot}$ potential Static Points
$\ominus$ Subtraction of two 3D Point Clouds
▶ Data Flow

Figure 3.2: *System overview articulated scene model.* The articulated scene model is calculated for each vista space. The model is updated from frame to frame by observing the scenery. Utilizing the model from the previous frame and the sensor data from the current frame the updated model can be calculated by two steps. First, the entity tracking detects and tracks moving objects by shifting an elliptical model through the potential dynamic points D_t^{pot}. The potential points are all points, which are not conform to the known static background. Second, the static scene and the articulated objects are adapted. Therefore, all found moving objects are subtracted and the produced potential static points S_t^{pot} are analysed with the vista space assumption to separate movable objects from the updated static scene.

3.2 Proposed System Overview

The robots purpose is to interact with the human and to work with him in the same environment, but the environment is naturally not static and the human moving in front of the robot is inhibiting the background modelling process. Therefore, the robot should acquire knowledge about its surrounding by detecting and tracking moving objects, modelling the static background without these persons and perceiving scene changes in the vista space. In the process the robot observes its environment passively, which means the robot camera stays static for a few seconds to gather information before the robot changes its view and observes the next vista space.

The algorithm is designed to calculate an articulated scene model M for each of the vista spaces (see fig. 3.2). The model consists of the dynamic parts D, the static background S and the observed articulated scene parts O. The model for one vista space is updated as long as the robot does not change its view. The model $\mathrm{M}_{\mathrm{t}-1}$ is updated by propagating it to the next frame at time *t*. In each frame the following processes are accomplished to update the model:

(i) **Model propagation:** The model $\mathrm{M}_{\mathrm{t}-1}$ from the previous frame is propagated to the current frame

(ii) **Perception & Preprocessing:** The actual sensor input is preprocessed and annotated with velocities

(iii) **Entity Tracking:** Moving objects are detected and tracked to exclude them from the static scene

(iv) **Scene Modelling:** The background and the movable objects are adapted

The *preprocessing* cares for the 3D data smoothing and velocity computation V_{t} based on optical flow resulting in 6D data as sensor input for frame t. The next step is to detect and track the moving parts, named as *Entity Tracking*. Thereby, the detection and tracking of moving persons is supported by the knowledge of the actual static scene $\mathrm{s}_{\mathrm{t}-1}$ generated out of all previous frames and vice versa.

In a first step, the known static scene points n from the previous frame

$$\mathrm{S}_{t-1} = \{\vec{s}_t^{\,i}\}_{i=1\ldots n} \tag{3.1}$$

are subtracted from the current scene

$$\mathrm{F}_t = \{\vec{f}_t^{\,i}\}_{i=1\ldots n}. \tag{3.2}$$

The remaining potential dynamic points

$$\mathrm{D}_t^{pot} = \mathrm{F}_t - \mathrm{S}_{t-1} \tag{3.3}$$

are annotated with the velocity data V_{t}. Based on the potential dynamic points D_t^{pot} new objects are detected. Using a clustering algorithm and a simple elliptical object model, the moving objects are found and subsequently tracked with a hybrid particle filter with mean shift. The potential points

$$\varepsilon_t \subset \mathrm{D}_t^{pot} \tag{3.4}$$

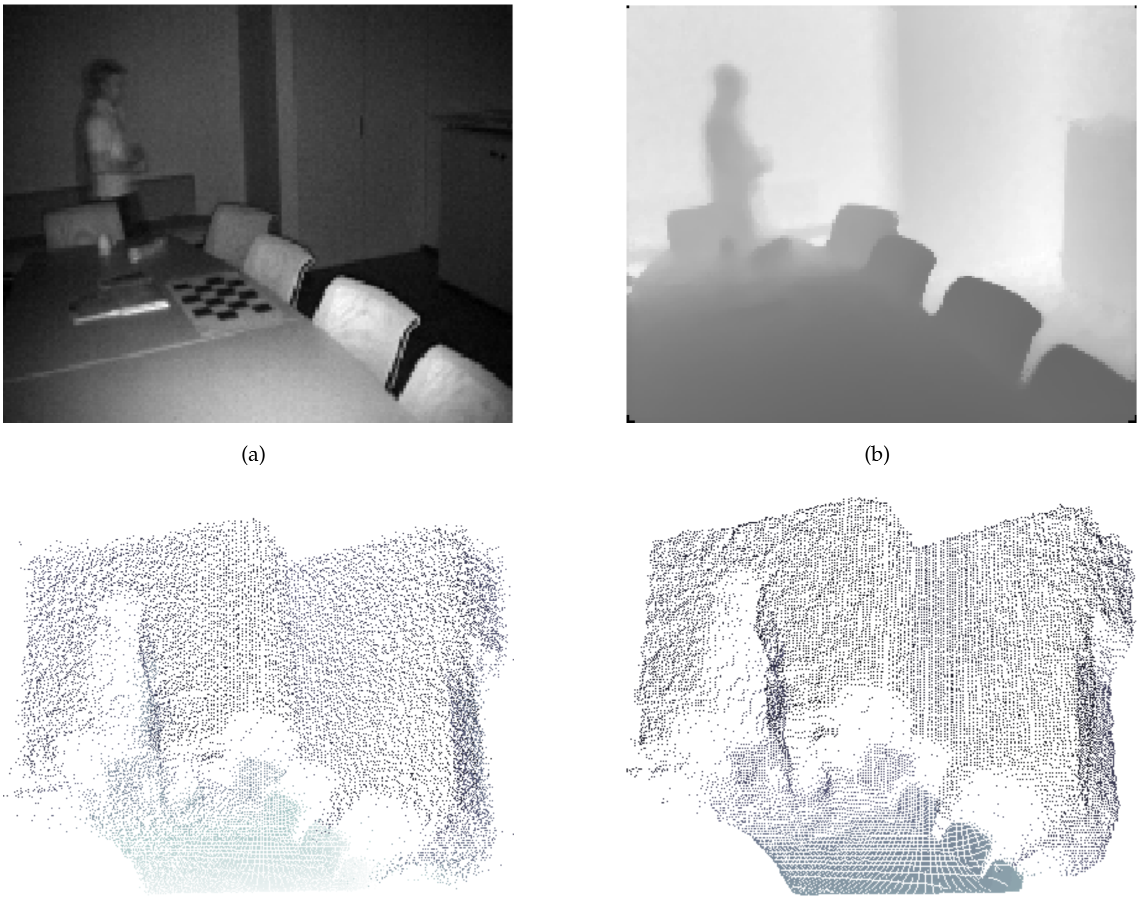
which belong to a dynamic object are passed to the current articulated scene model M_{t}.

In the *scene modelling* step these points ε_t are subtracted from the actual frame F_t to identify the potential static points

$$\mathrm{S}_t^{pot} = \mathrm{F}_t - \varepsilon_t \tag{3.5}$$

in the current frame. By applying the vista space assumption and utilizing the knowledge S_{t-1} from the last frame the movable objects O_t that form the articulated scene parts can be detected and the static background S_t can be updated, simultaneously. Both are passed to the current articulated scene model M_t, which is propagated again to the next frame.

The updating of the vista space ends if the robot changes its view and during the motion of the camera from one vista space to an other the model computation is stopped. At this moment the outcome from the articulated scene parts O_t are all the areas where a movable object is newly detected by the vista space assumption. From the moment the robot observes a new vista space the building of the next articulated scene model begins. By incorporating the motion of the robot the vista spaces can be merged to build a global knowledge base. Here, the motion information from a laser-based SLAM approach [221] is utilised.



(a)

(b)

(c)

(d)

Figure 3.3: *Raw data acquired of the Time-of-Flight sensor.* (a) amplitude image, (b) distance image, (c) Not preprocessed 3D point cloud, and (d) preprocessed 3D point cloud.

3.3 Preprocessing of the Input Data

The system uses the Swissranger SR4000 provided by Swiss Center for Electronics and Microtechnology (CSEM) [209]. Besides the distance value matrix (Fig. (3.3(b))), the camera provides per frame a matrix containing amplitude values (Fig. (3.3(a))). The sensor and several preprocessing techniques are described in Sec. (2.1.4). The applied preprocessing techniques are proposed in [196].

The distance image is smoothed with a distance-adaptive median filter, which uses for each pixel a different mask size (e.g. 3×3, 5×5, or 7×7) depending on the distance value of the pixel. Generally, pixels with larger distance value are filtered with smaller filter masks, and vice versa, so that significant structures at large distances are not blurred, and at the same time, noisy surfaces at small distances can be smoothed. As the amplitude value refers to the quality of the distance measurement, points with a small amplitude value are removed from the final 3D point cloud. The threshold needed adapts automatically to different reflectance properties in different scenes as it is a fraction of the mean amplitude value per frame. Further, edge points (so-called "flying pixels") arising in the case where light from the fore- and the background hits the same pixel simultaneously are rejected if the amount of near neighbours in the 2D neighbourhood is insufficient. Last, 3D coordinates are generated out of the distances with regard to a 3D camera coordinate system. With the assumption of ideal perspective projection, the known position of the principal point, pixel sizes, and focal length, the 3D coordinates can be computed from the distances via ray proportions in triangles. As a result the computed 3D points are organized regularly in a 2D matrix. Figures (3.3(a)) to Fig. (3.3(c)) show a frame of the 3D ToF camera consisting of an amplitude image, an distance image, and the raw 3D point cloud. Fig. (3.3(d)) presents the resulting 3D point cloud after applying the described preprocessing techniques.

In order to distinguish between static parts of the scene and moving persons or objects the motion in the 3D point cloud has to be determined. In the following an image based method for motion computation is presented which can be applied here easily by treating the point cloud as planar depth maps or images [194].

3.4 3D Motion Computing using Optical Flow

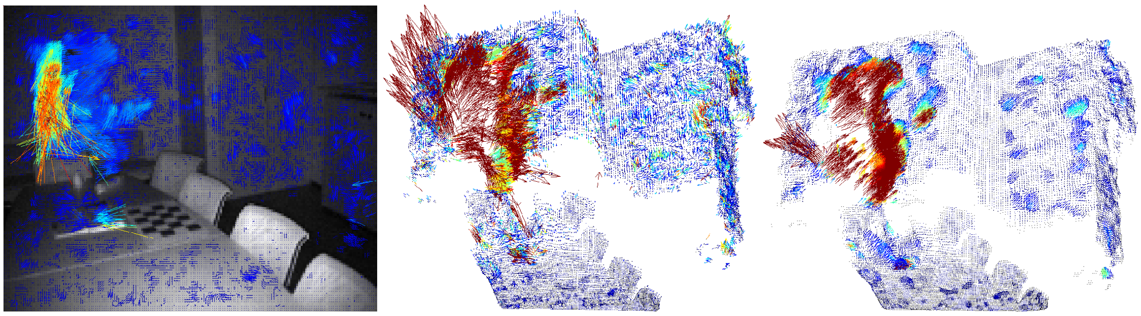
Optical flow (Sec. (2.2.1)) is used to initially detect moving objects in a scene. Therefore, the usual 2D optical flow is extended to 3D optical flow through the use of a 3D sensor. Here, a hierarchical implementation of Lucas's and Kanade's 2D optical flow algorithm written by Sohaib Khan [1] [2] is taken as basis.

As the Swissranger camera provides normal 2D intensity images based on the amplitude values it is possible to reduce the 3D correspondence problem to a 2D correspondence problem and to compute the optical flow for each frame $\mathcal{F}_i$ of a sequence of ToF images

[1] http://www.cs.ucf.edu/~khan/

[2] http://server.cs.ucf.edu/~vision/source.html

$(\mathcal{F}_1, \mathcal{F}_2, ...)$ based on data of two consecutive frames $(\mathcal{F}_i, \mathcal{F}_{i-1})$. Each pixel of frame $\mathcal{F}_i$ is annotated with a 2D velocity vector $\vec{v}_{2\text{D}} = (v_x, v_y)^T$ as shown in Figure 3.4(a) which results into pixel correspondences between frame $\mathcal{F}_i$ and frame $\mathcal{F}_{i-1}$. As 3D information is available for each pixel these pixel correspondences can be directly transformed into 3D point correspondences $(\vec{p}_k^i, \vec{p}_l^{i-1})$ which can be used to compute 3D velocities $\vec{v}_{3\text{D}} = (v_x, v_y, v_z)^T = \vec{p}_k^i - \vec{p}_l^{i-1}$. Figure 3.4(b) presents a 3D point cloud of one frame of a test sequence annotated with 3D velocity vectors. The processing from 2D optical flow on 2D images to real 3D velocities is supported by the used hardware. As the Swissranger camera provides good distance measurements velocities with reliable values especially in the z component can be determined. This is usually not suitable for many other camera set-ups like stereo rigs or multi-camera systems. The velocity annotated 3D point cloud results in 6D data.



(a) (b) (c)

Figure 3.4: *Velocity processing with the optical flow method.* (a) 2D velocity vectors (b) 3D velocity vectors from combining 2D velocities and point correspondences in consecutive images, (c) the latter smoothed component wise by a median filter. Each 3rd velocity vector is displayed and colour coded with respect to its length: red denoting a big motion vector and blue a small one.

Due to the low resolution of the camera and inaccuracies of the optical flow erroneous velocity vectors at changing depth steps are computed. To get rid of those outliers a 5×5 median filter is applied separately to the three components v_x, v_y, v_z of the flow vector $\vec{v}_{3\text{D}}$. In Figure 3.4(c) the smoothed result of the 3D velocity field of Figure 3.4(b) can be seen.

3.5 Detection and Tracking of Dynamic Objects

The dynamic scene analysis involves the detection and tracking of moving objects, which on the one hand enhances the segmentation of the different scene parts and which is on the other hand useful for the understanding of the scene as the trajectories of the objects give a broad picture of the movements in the actual vista space.

The basic algorithm is based on the implementation of [176]. The algorithm is extended with several further improvements like a new elliptical model, a more sophisticated motion model and several improvements in the hypotheses management like a more stable tracking over several frames.

Using the 3D point cloud and the annotated 3D velocities, the scene is simplified by applying a 6D hierarchical clustering technique. The segmentation is enhanced through the incorporation of the velocity information in the early clustering stage, because it enables the segmentation of neighbouring objects, like a person walking in front of a wall. The first step of the algorithm is to span small contiguous regions in the cloud of the 6D points, based on features for spatial proximity and homogeneity of the velocities. A hierarchical clustering technique is applied using the complete linkage algorithm [21], which, choosing a small branching factor in the hierarchical tree, deliberately over-segments the scene, generating many small motion-attributed clusters (see figure 3.5(a)). Each calculated cluster is annotated with the 2D position of its centroid projected on the ground plane, a weight factor accordant to the number of included points and the mean velocity of all these points.

From here on, persons and objects are represented by an upright ellipse of variable radius, which is a suitable model for the moving entities in the presented scenarios (here, humans). The object hypothesis $h(a)$ is characterized by a six dimensional parameter vector

$$a = [x, y, v_\theta, v_r, r_x, r_y] \tag{3.6}$$

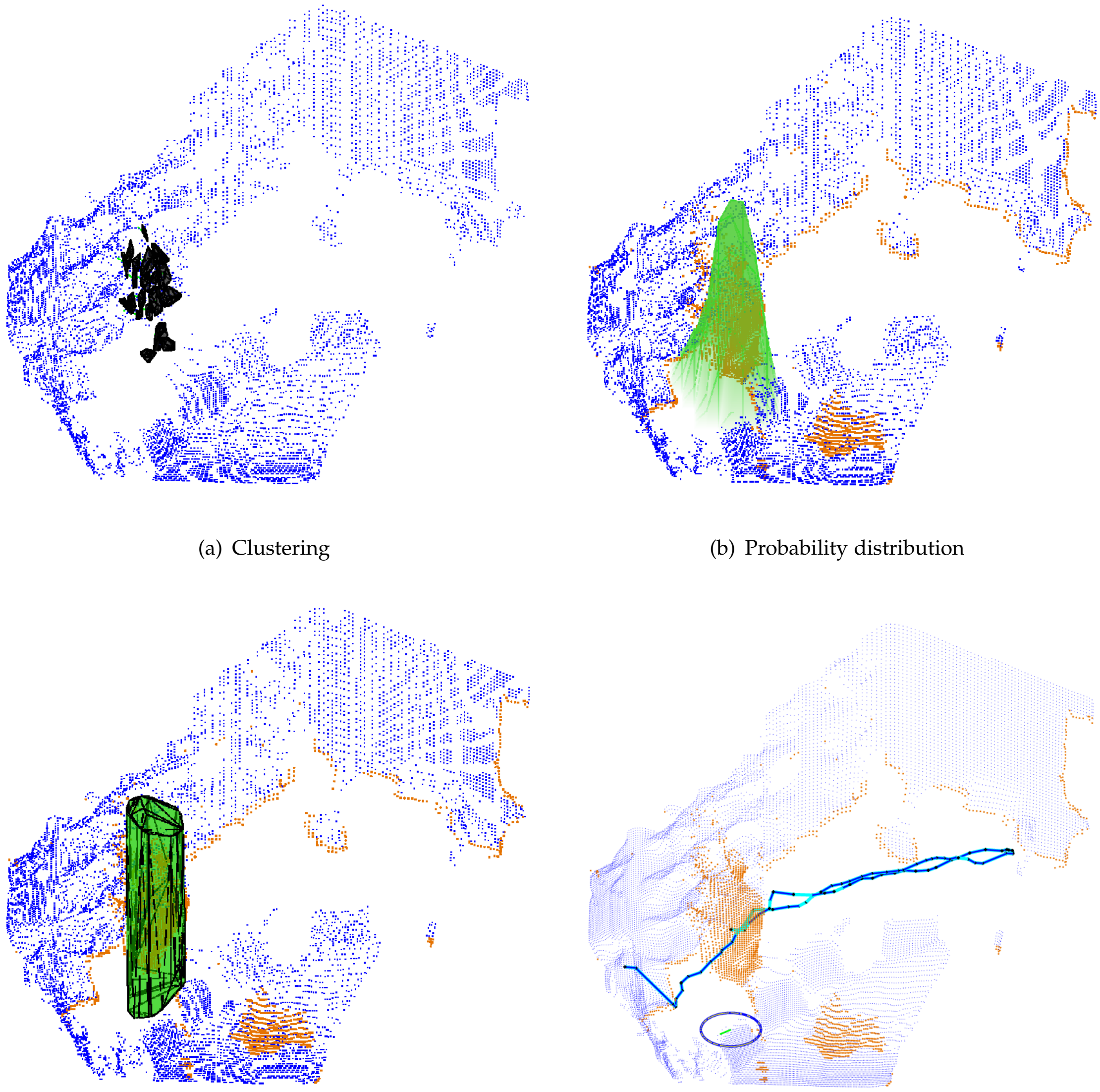
where x and y are the centroid on the ground plane, r_x, r_y the radii of the main axes and v_r the magnitude and v_θ the direction of the velocity of the ellipse.

The next step is the detection of the moving objects. Here, the elliptical model is advantageous for the detection as the velocity computation is noisy and many points between the found clusters have different velocities. The detection only needs a few clusters denoting a moving object and afterwards, using the elliptical model, all cluster which are lying in the ellipse are added to the moving object. This means, that the hypothesis covers mostly the full moving object even if the velocity data is noisy. To generate a hypothesis, an ellipse is shifted through the small clusters searching for meaningful collections of clusters with similar velocities. Grouping close clusters together, a hypothesis is found if the weight of all clusters together is higher than a certain threshold. Here, 20 close points moving in a similar direction are sufficient. Afterwards, each cluster integrated into a hypothesis is marked as an already found object.

All found hypothesis are additionally annotated with an *id* to identify them over the observation time. The detection by moving needs the object to move at least one time and afterwards, it is capable to track the object even if it is not moving any more.

All extracted hypotheses from the current frame are merged with the ones tracked from the previous frame resulting in one hypotheses matrix for each frame.

The tracking of the hypotheses is calculated like follows. The K hypotheses in the current frame t are tracked based on the position, velocity and size of each hypothesis in the previous frame $h_k^{t-1}(a)$, utilized in a hybrid kernel particle filter with mean shift [176]. The particle filter creates a set of new hypotheses $s_k^t(h)$ for each tracked object, called particles, and distributes them with a first order motion model (Υ) mixed with a random Gaussian noise (Ω) (see figure 3.6(a)). The mixture is particular reasonable for



(a) Clustering

(b) Probability distribution

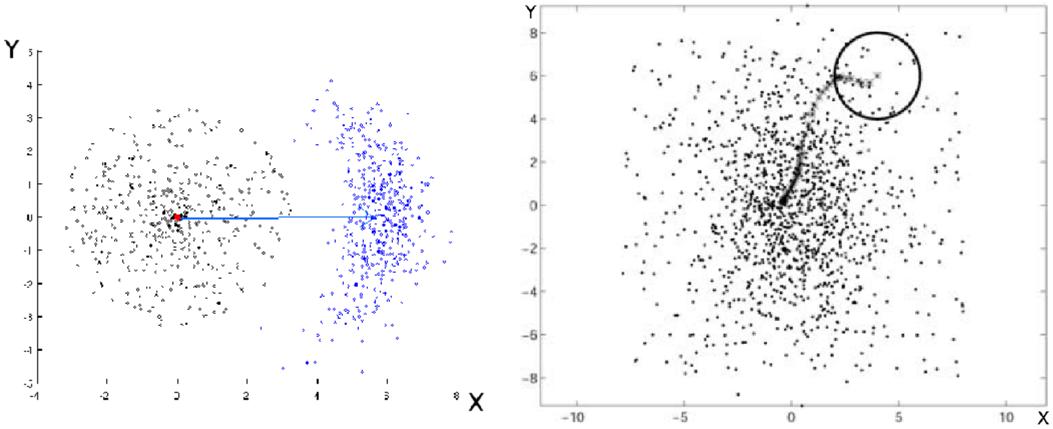
(c) Found object

(d) Trajectory

Figure 3.5: *Human detection and tracking steps.* The images explain the tracking algorithm. The blue points belong to the static scene S_{t-1}. The dynamic pixels P_t are plotted in orange. (a) At a first stage the dynamic points are clustered, generating small motion attributed regions. (b) The objects are detected and tracked using the observation function (see Eq. 3.23). The probability of the particle distribution is plotted in green. (c) The maximum of the observation function denotes the found object (shown as green box). (d) The resulting object trajectory is plotted in cyan. The blue ellipse contains the object at the actual position.

the movements of a human, because it covers straight movements in a certain direction as well as rapid movement changes.

The linear movement is represented by a prediction of particles with the following motion equation ($a = [x, y, v_\theta, v_r, r_x, r_y]$ is the current state and $a' = [x', y', v'_\theta, v'_r, r'_x, r'_y]$ represents the state at $t - 1$).



(a) Particle distribution

(b) Mean shift

Figure 3.6: *Particle distribution.* (a) The particle distribution follows a motion model and a random distribution to cover all possible motions of a human. The figure shows the distribution of the particles in the XY plane. The movement of the object is in positive X direction. The particles are distributed in the accordant direction and for random movements as well. (b) The distributed particles are weighted and then shifted with mean shift to recover the best possible object position. (Here, shown for a random distribution)

$$x = N_m \Upsilon_x + N_r \Omega_x \tag{3.7}$$

$$y = N_m \Upsilon_y + N_r \Omega_y \tag{3.8}$$

$$v_\theta = N_m \Upsilon_{v_\theta} + N_r \Omega_{v_\theta} \tag{3.9}$$

$$v_r = N_m \Upsilon_{v_r} + N_r \Omega_{v_r} \tag{3.10}$$

$$r_x = r'_x + rnd(r'_x, A_1) \tag{3.11}$$

$$r_y = r'_y + rnd(r'_y, A_2) \tag{3.12}$$

The variables (N_m, N_r), with $N_r = 1 - N_m$, control the amount of linear movement model and random Gaussian noise. The variables $A1 - A2$ define the magnitude of a Gaussian process noise $rnd(a, b)$ with a, b as mean and variance. The random Gaussian motion model is defined as

$$\Omega_x = x' + rnd(x', A_3) \tag{3.13}$$

$$\Omega_y = y' + rnd(y', A_4) \tag{3.14}$$

$$\Omega_{v_\theta} = v'_\theta + rnd(v_\theta, A_5) \tag{3.15}$$

$$\Omega_{v_r} = v'_r + rnd(v'_r, A_6) \tag{3.16}$$

with $A3 - A6$ controlling the magnitude of a Gaussian process noise. The first order motion model is calculated like following

$$\Upsilon_x = x' - v'_r sin(v'_\theta) + v'_r sin(v'_\theta + \tilde{v}_{rot}) \tag{3.17}$$
$$\Upsilon_y = y' + v'_r cos(v'_\theta) - v'_r cos(v'_\theta + \tilde{v}_{rot}) \tag{3.18}$$
$$\Upsilon_{v_\theta} = v'_\theta + \tilde{v}_{rot} + \tilde{\gamma} \tag{3.19}$$
$$\Upsilon_{v_r} = v'_r + rnd(v'_r, A_7) \tag{3.20}$$

with

$$\tilde{v}_{rot} = rnd(0, A_8) \tag{3.21}$$
$$\tilde{\gamma} = rnd(0, A_9) \tag{3.22}$$

, where $A_8 - A_9$ are defining again the magnitude of a Gaussian random variable. This mixture distribution of particles covers the potential movement of most moving objects as it follows linear and random movement.

In order to identify the new position of each hypothesis the particles are rated with an underlying observation. The observation is based on the relative position, relative velocity and weight of all clusters within the ellipse of each hypothesis weighted with Gaussian kernels.

$$\rho(s_k) = K_r(s_k) \sum_{l \in s_k} K_d(l, s_k) K_v(l, s_k) \tag{3.23}$$

$$K_r(s_k) = \exp\left(-\frac{r(s_k)^2}{2H^2_{r,\min}}\right) - \exp\left(-\frac{r(s_k)^2}{2H^2_{r,\max}}\right) \tag{3.24}$$
$$K_d(l) = \exp\left(-\frac{\|d(l) - d(s_k)\|^2}{2\ H_d^2}\right) \tag{3.25}$$
$$K_v(l, s_k) = \exp\left(-\frac{\|v(l) - v(s_k)\|^2}{2\ H_v^2}\right) \tag{3.26}$$

with (3.24) keeping the radius in a realistic range, (3.25) reducing the importance of clusters further away from the cylinder centre, and (3.26) masking out clusters having differing velocities. The functions $r(\cdot)$, $d(\cdot)$, and $v(\cdot)$ extract the radius, the 2D position on the ground plane and the velocity of a cluster l or a hypothesis s_k. The kernel widths H are determined empirically. Eq. 3.23 is also called the observation function $\rho(s_k)$ of the particle filter. The outcome is a density approximation based on the object hypothesis and the attributes of the associated clusters, with the maxima corresponding to the actual objects (Fig. 3.5(b)).

Several mean shift iterations refine the particles to concentrate at the local maxima of the distribution, which decreases the needed amount of particles (see figure 3.6(b)).

```
**Input:**
{- $F_t = \{\vec{f}_t^i\}$ (current frame)}
{- $S_{t-1} = \{\vec{s}_{t-1}^i\}$ (current background)}
{- $\varepsilon_t$ (current dynamic clusters)}
{**Output:**}
{- $S_t = \{\vec{s}_t^i\}$ (new background)}
{- $O_t$ (movable objects)}

for $i = 1$ to $n$ do
  if $\vec{f}_t^i \notin \varepsilon_t \wedge |\vec{v}_t^i| < \theta_v$ then
    if $|\vec{s}_{t-1}^i - \vec{f}_t^i| < \theta_d$ then
      $\vec{s}_t^i = \vec{s}_{t-1}^i + \frac{1}{w}(\vec{f}_t^i - \vec{s}_{t-1}^i)$ ;
      {$w$: # accumulated values}
    else
      if $|\vec{f}_t^i| > |\vec{s}_{t-1}^i|$ then
        $\vec{s}_t^i = \vec{f}_t^i$;
      else
        $\vec{s}_t^i = \vec{s}_{t-1}^i$;
        $O_t = O_t \cup \vec{f}_t^i$;
      end if
    end if
  end if
end for
```

Figure 3.7: Algorithm per time step t for background adaptation and movable object detection.

Individual particles selected from these best modes of the distribution represent the objects found in the current frame (Fig. 3.5(c)).

For each tracked object hypothesis, all 3D points associated with this object are back projected in the 2D amplitude image and used for computing a 2D convex hull of the tracked object. All points within this 2D polygon are marked as non static points and are finally excluded from the reconstruction step. The convex hull inherits also points, which were potentially not incorporated in the clustering process due to bad reflectance properties or other circumstances, which do not allow a valid 3D value for this point.

3.6 Adaptive Background Modelling

So far I proposed methods to distinguish between static and moving parts in a scene. In the following, the calculated moving objects are extracted and the static parts of the observation are analysed. By applying the vista space assumption and utilizing the knowledge from the last frame the movable objects that form the articulated scene parts can be detected and the static background can be updated, simultaneously. The basis of the vista space assumption that the most distant measurement in the current view describes the background has to be expanded due to noise of the the 3D sensor. Therefore, I introduce a threshold θ_d above which a change in the distance is significant and does not arise from noise (here, $\theta_d = 10$cm given by the noise level of the camera).

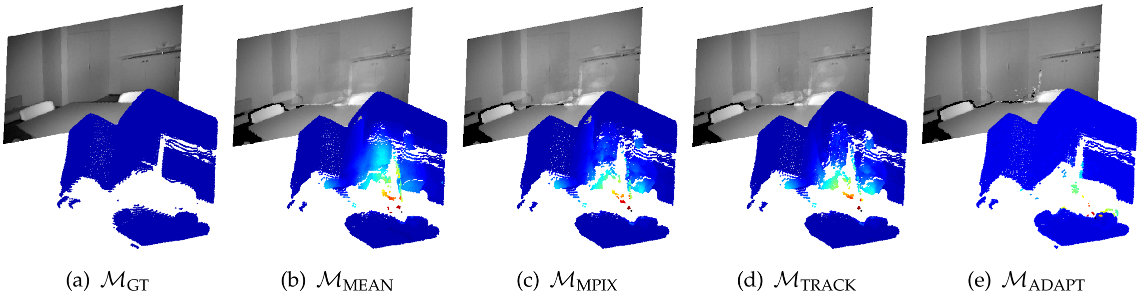
The algorithm presented in 3.7 is applied to each time step of the observation to calculate the updated static scene $S_t = \{\vec{s}_t^i\}$ and the movable objects O_t. Therefore, the algorithm uses as input the current frame $F_t = \{\vec{f}_t^i\}$ and the last known static scene $S_{t-1} = \{\vec{s}_{t-1}^i\}$ and the dynamic clusters ε_t from the previous frame. The dynamic clusters contain the

3D points from the moving objects of the tracking module. These points are removed, before the update process takes place.

The static scene is updated in line 12, if the difference of a known static point s_{t-1}^i to the actual frame point $\vec{f}_t^i$ is below the sensor noise level θ_d. Then, the static point and the current point are accumulated to a new static point $\vec{s}_t^i$ with improved reliability. Otherwise, it has to be determined if a new static scene point is detected in line 16 or the point belongs to a movable object in line 19. The vista space assumption is used to identify the matching case. All points belonging to movable objects are saved in a separate list, where the time of detection and the number of times the points has been seen are considered. Clustering these points in space and time the different objects can be separated. Consequently, objects can only be separated if they appear at a different point in time or at least at different places.

3.7 Experiments and Results

The evaluation of an articulated scene model does not follow typical standard reports as it is not feasible to build a complete ground truth model. Hence, I split up the different parts of the model and I compared the static scene to a ground truth model and to some simple background modelling techniques to give quantitative results. In the following, the proposed system $\mathcal{M}_{\text{ADAPT}}$ is evaluated by comparing the results to the naive approach of only summing up the images and building the mean for each pixel ($\mathcal{M}_{\text{MEAN}}$). It is also compared to the neglecting of moving pixels $\mathcal{M}_{\text{MPIX}}$ and last, to $\mathcal{M}_{\text{TRACK}}$ [194] where only dynamic objects are determined through tracking without background model feedback and no distinction is made between static background and static movable objects. All methods are checked against a ground truth static scene model $\mathcal{M}_{\text{GT}}$, which has been taken without any movable or moving object for each sequence. The articulated parts and the trajectories of the moving objects are presented qualitatively in illustrations.



(a) $\mathcal{M}_{\text{GT}}$ (b) $\mathcal{M}_{\text{MEAN}}$ (c) $\mathcal{M}_{\text{MPIX}}$ (d) $\mathcal{M}_{\text{TRACK}}$ (e) $\mathcal{M}_{\text{ADAPT}}$

Figure 3.8: *Results of scene $\mathcal{S}_{\text{s2,r1}}$ for the evaluated algorithms.* In the front the reconstructed 3D static scenes and in the back the accordant 2D images can be seen. (a) shows the ground truth. In (b) the reconstruction by simple averaging, in (c) the reconstruction by excluding moving pixels, and in (d) the reconstruction by tracking objects is shown. In the 2D image the wrong reconstruction can be seen as a ghost of the person moving in the scene. (e) shows the result using the here proposed method. The colours encode the error of the model if compared to the ground truth – blue means small and red means big error.

	$\mathcal{S}_{s1,r1}$	$\mathcal{S}_{s1,r2}$	$\mathcal{S}_{s1,r3}$	$\mathcal{S}_{s1,r4}$	$\mathcal{S}_{s1,r5}$	$\mathcal{S}_{s1,r6}$
$\mathcal{M}_{\text{MEAN}}$	103 ± 177	106 ± 204	124 ± 222	157 ± 284	142 ± 278	147 ± 262
$\mathcal{M}_{\text{MPIX}}$	64 ± 121	74 ± 184	79 ± 185	111 ± 216	99 ± 230	95 ± 193
$\mathcal{M}_{\text{MTRACK}}$	71 ± 166	108 ± 209	75 ± 189	97 ± 212	79 ± 308	98 ± 219
$\mathcal{M}_{\text{ADAPT}}$	18 ± 59	19 ± 47	21 ± 61	24 ± 78	24 ± 68	21 ± 55

	$\mathcal{S}_{s2,r1}$	$\mathcal{S}_{s2,r2}$	$\mathcal{S}_{s3,r1}$	$\mathcal{S}_{s3,r2}$	$\mathcal{S}_{s4,r1}$	$\mathcal{S}_{s4,r2}$
$\mathcal{M}_{\text{MEAN}}$	95 ± 187	108 ± 147	89 ± 105	85 ± 183	219 ± 403	321 ± 639
$\mathcal{M}_{\text{MPIX}}$	71 ± 155	80 ± 118	63 ± 145	61 ± 125	163 ± 328	299 ± 635
$\mathcal{M}_{\text{MTRACK}}$	84 ± 182	85 ± 140	71 ± 141	134 ± 712	51 ± 165	74 ± 218
$\mathcal{M}_{\text{ADAPT}}$	20 ± 96	16 ± 37	20 ± 58	22 ± 52	14 ± 26	75 ± 319

	$\mathcal{S}_{s4,r3}$	$\mathcal{S}_{s4,r4}$	$\mathcal{S}_{s5,r1}$	$\mathcal{S}_{s5,r2}$	$\mathcal{S}_{s6,r1}$
$\mathcal{M}_{\text{MEAN}}$	234 ± 451	246 ± 594	117 ± 161	713 ± 1013	182 ± 284
$\mathcal{M}_{\text{MPIX}}$	229 ± 588	229 ± 588	70 ± 132	654 ± 1016	182 ± 284
$\mathcal{M}_{\text{MTRACK}}$	356 ± 677	246 ± 601	71 ± 152	674 ± 1014	207 ± 317
$\mathcal{M}_{\text{ADAPT}}$	18 ± 64	98 ± 404	19 ± 90	471 ± 1008	55 ± 146

Table 3.1: Evaluation of four reconstruction methods on 17 sequences (mean error ± mean variance). The error shown in the table is computed as mean Euclidean distance over all model points to the corresponding ground truth points. The mean error is given in mm as well as the mean variance. The high error in $\mathcal{S}_{s5,r2}$ results from a wide range view, where the sensor produced a high amount of noise.

The underlying data sets $\mathcal{S}$ are self-created and they show different challenging dynamic scenes. The human shows different moving behaviours or stops moving, which makes it difficult to detect him as not static. Furthermore, the human interacts with the environment as he cleans up $\mathcal{S}_{s3}$, moves chairs, searches a teddy bear $\mathcal{S}_{s2}$, opens and closes doors $\mathcal{S}_{s4}$ and rearranges teddy bears $\mathcal{S}_{s1}$, water cans $\mathcal{S}_{s5}$ and plants$\mathcal{S}_{s6}$. Each run i of a sequence belonging to one scenario j is labelled with $\mathcal{S}_{sj,ri}$.

In Fig. (3.8) the resulting static scenes for one example vista space are presented. The figure shows the resulting 3D static scene from the different background modelling techniques and the 2D image created from this model. The colours encode the error of the models compared to the GT, where blue denotes a small error and red a big error. The naive background modelling strategies failed in removing the person correctly in all frames, which results in a big error at those positions of the 3D point cloud, where the person is still visible. This gets apparent as a ghost appears at the same positions in the 2D image. The approach presented in this paper reliably removes the person, which provides a sound background model. Tab. (3.1) shows an analysis of the arising errors from the background modelling strategies. The first value is the mean euclidean distance in mm over all pixel compared to the ground truth and the second value denotes the corresponding standard deviation. The presented errors affirm the viewable impression from Fig. (3.8) as $\mathcal{M}_{\text{ADAPT}}$ results in the lowest error rates. The rates are promising with an error mostly at 2 cm and never above 10 cm. Even in scene $\mathcal{S}_{s4,r4}$, where sparse static points in the door can be detected, the result of the proposed method is much more robust than the naive approaches, where the mean error is always above 20 cm. A mostly low standard deviation stands for good results in each point as well. Higher error rates ($\mathcal{S}_{s5,r2}$) could occur due to noise arising from the 3D sensor, if the observed scene has some disadvantageous characteristics. The sensor has increasing noise per distance and it is sensitive to reflecting and black surface. You can see this in the mentioned figure in the open door in frame 1 and 26.

Fig. (3.9) gives some examples of the detected articulated scene parts. The found objects are colour-coded in the image and they are separated from the background to show the variability in detecting diverse objects due to the model independent approach.

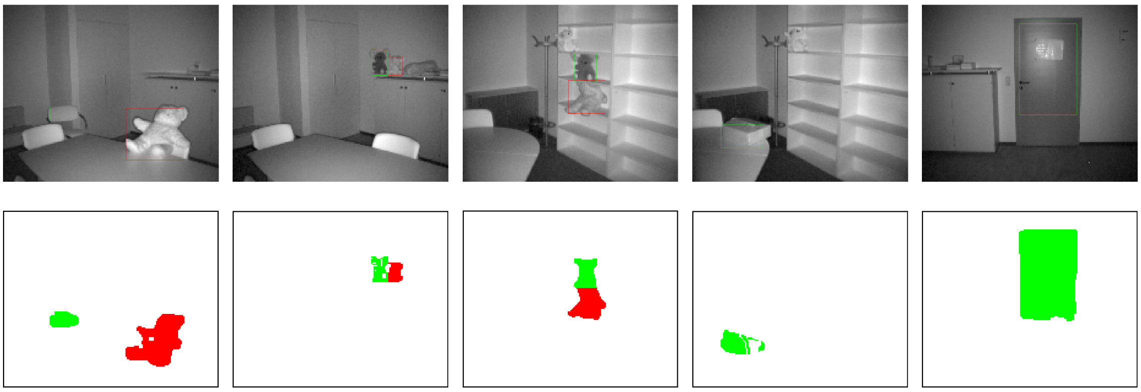


Figure 3.9: *The images show diverse objects detected by the method.* All presented objects have been moved around by the human in the scene. Different colours encode different objects. The pictures show nicely the huge variability in detecting movable objects due to the model independent approach.

The objects can be marked directly in the 2D image (see Fig. (3.10)), which could be used by further processes to calculate more precise information like the shape or texture of the objects.

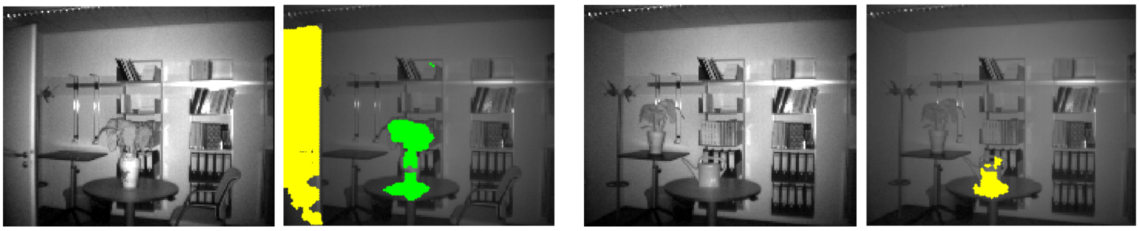
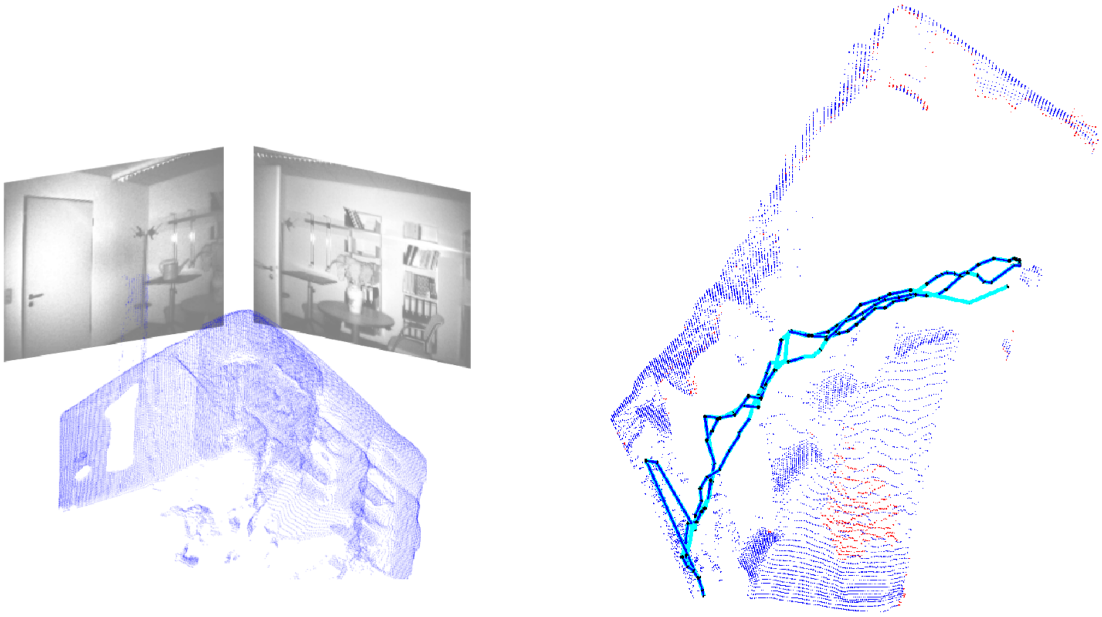


Figure 3.10: *Two examples showing the segmented movable objects in a 2D image.* The first and the third image are the original images and the second and fourth show the marked object. The coloured areas belong to different recognized objects, which have been moved at least one time.

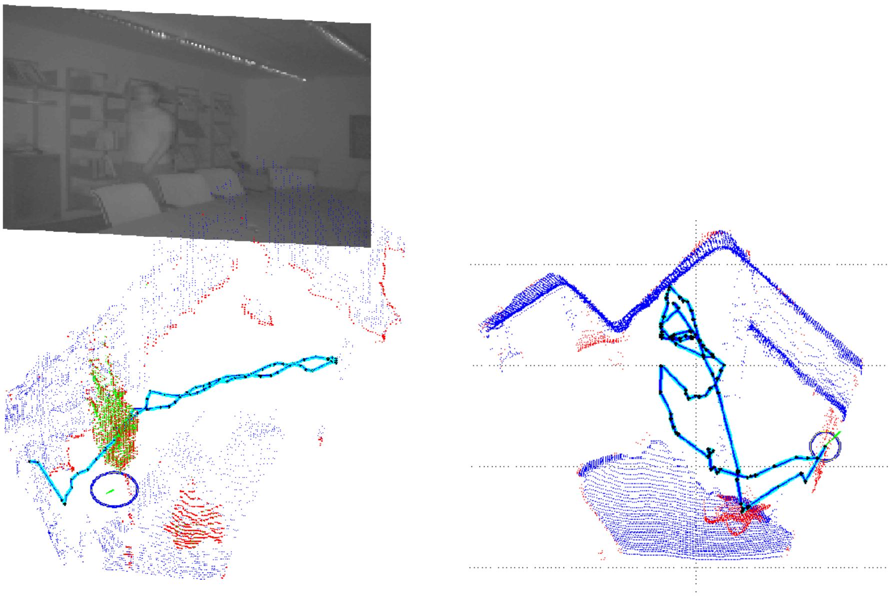
In Fig. (3.11) an example of a combination of two subsequent vista spaces is presented. Here, I transform the vista spaces into the same world coordinate system by incorporating the movement of the robot. The two images in the back belong to the different vista spaces. In the second image all trajectories from one observation of a vista space are plotted. One human walked three times back and forth. His movements are consistently tracked. Two other example vista spaces and their resulting trajectories are shown in Fig. (3.12(a))- 3.12(b) using two different views.



(a) Combined vista spaces

(b) Tracks from human movements

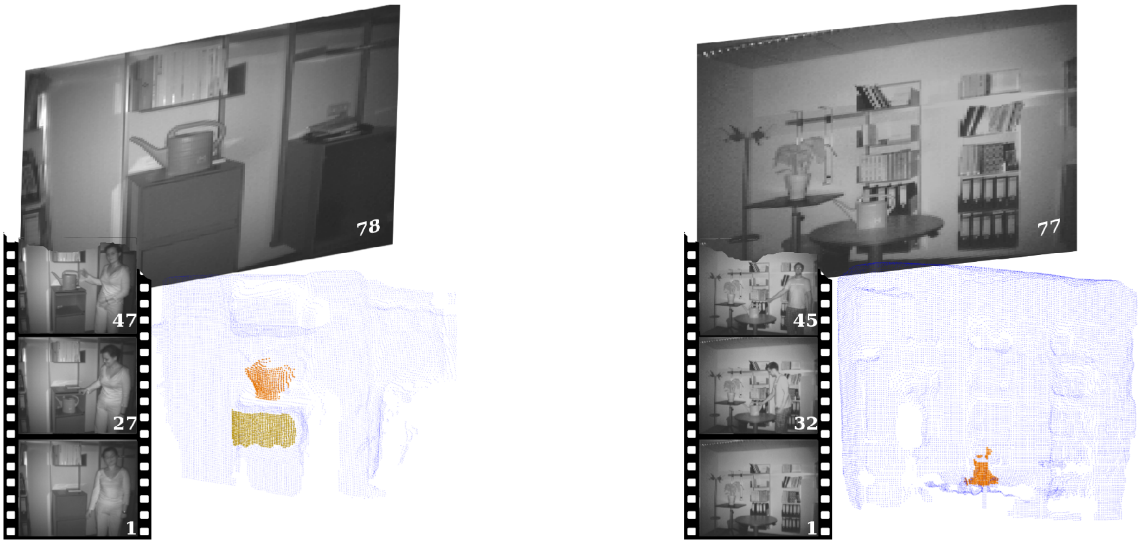
Figure 3.11: *Combination of vista spaces and human trakcs.* (a) Subsequent vista spaces can be combined by a transformation of the particular spaces in one world coordinate system. The transformation is extracted out of the motion of the robot. (b) All tracks of a human walking behind a table (in cyan). The human walked three times back and forth.



(a) Trajectory

(b) Trajectory

Figure 3.12: *Trajectories of humans.* In (a)-(b) tracking results of the proposed system are shown (in cyan). In both views the red pixel denote the dynamic and the blue ones the static parts of the scene. The right scene is taken from $\mathcal{S}_{\mathrm{s2,r1}}$ (see figure A.1(a))



(a) $\mathcal{S}_{s6,r1}$

(b) $\mathcal{S}_{s7,r1}$

Figure 3.13: *Articulated scene model results.* (a)-(b): For all recorded sequences the learnt background model (blue points) and the detected movable objects (orange points) are shown. In the bottom left three selected images of the sequence characterize the tide of events from bottom to top finishing with the last frame in the background.

Finally, the articulated scene model for each of the sequences is plotted in Fig. (3.13). In the bottom left a film-strip gives an idea of the presented sequence, starting at the bottom and ending with the big picture in the background. The corresponding frame numbers are shown in the bottom right in each image. The static background model relates to the blue 3D points and the found articulated parts correspond to the coloured areas, whereas different colours encode different objects. The results of the other sequences are shown in the appendix Chapter (A).

3.8 Conclusion

In this chapter I presented an efficient approach to analyse dynamic scenes directly in 3D. The vista space assumption enables a mobile robot to segment knowledge about the static background, the moving entities and which objects are movable combined in one articulated scene model out of its observations. The gathered knowledge builds a good basement for many following research areas like object learning, navigation or just as an attention on human action spaces. In the future, it is possible to integrate the static 3D background model in a SLAM approach to realize a better and safer navigation. It is also imaginable to investigate more work in the detection of the articulations of several objects, like the opening range of a door or the typical movement areas of humans to develop an understanding of safe movement areas or where to pay attention.

4 A 3D Tracking System on a Moving Platform

In the following chapter I am going to describe a modular software system, which enables a mobile robot to implement the second category of situation awareness. The second category is characterised by a temporal linking of information, which relates actually gathered information with previous knowledge.

The second category is most important for a mobile robot, because other moving entities can only be safely avoided, if the robot has knowledge about their previous movement. The temporal linking generates a trajectory, which denotes the past movement of an entity. Using the past positions it is possible to predict a future movement and hence, avoiding collisions by incorporating this movements in the own planning strategy.

Here, the temporal linking is of main interest in order to provide the information for other subsequent planing strategies. General questions arise concerning the implementation of a system for the second category of situation awareness:

- *What data is needed for the robot in order to get information about the movement of present humans?*
- *How could the humans be detected fast enough to ensure an accurately timed reaction of the robot?*
- *How could a stable temporal linking be achieved even during occlusions?*
- *How could the entering or leaving of a person be detected?*
- *How is it possible to deal with ego movements of the robot?*

In this thesis a solution to the described problems of detecting and tracking humans on a mobile platform is presented.

I present a complete system approach, which uses multi-dimensional data in order to provide as much information as possible about the present humans. Like mentioned above, the real-world trajectories of the humans are the most important information for a mobile robot. Hence, 3D information is of essential significance to get the real world movement. In order to deal with the special need for a real-time calculation on a mobile robot, several pre-processing steps are proposed. The different steps restrict the search space for the detection of the humans efficiently (see Fig. (4.1)). This facilitates a fast and reliable detection. If a person is detected he/she has to be recognised in every frame in order to get his/her position at every time step. This is realised by a tracking module, which cares for the temporal linking of each person. An additional intermediate layer merges all gathered information in order to get a more stable result. This layer also analyses, if a person enters or leaves the scene and triggers the creation and deletion of hypotheses. Generally, it is assumed that a person has left, if he/she leaves the apparent area for at least a few frames.

The proposed system approach is designed to deal with ego motions of the robot by detecting the persons frame-wise and tracking the persons with a dynamic particle filter. Thereby, the system detects persons and keeps track of them even during own movements.

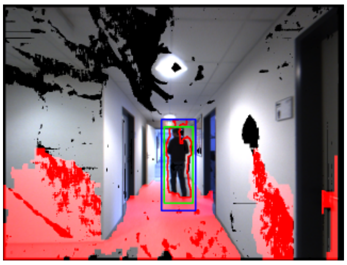


Figure 4.1: *Detection and Tracking of a human.* The scene is simplified through a floor elimination (in red). Afterwards, the human is detected and tracked (green and blue rectangle). In red the person is segmented from the background. The information about humans present in the scene and their movements is an essential knowledge for mobile robots in order to interact safely with their environment.

The system implements state of the art algorithms, which are enriched with new features and combined in a new and efficient way. Additionally, the whole system is designed to work fast enough to process enough frames for an efficient tracking on real data on the robot.

The system generally adheres to the system approach presented in Chapter (2). This means, that the proposed system includes a perception module (cf. Sec. (2.1)), a detection module (cf. Sec. (2.2)) and a tracking module (cf. Sec. (2.3)). Additionally, the system deploys a self-developed hypotheses management.

In contrast to the previous chapter, the sensors are now moving during the action, which entails a new chapter of problems (described in Sec. (4.1)). Despite occurring motion blur and fast changing lightning conditions a lot of simplifications could not be applied any more. Background subtraction is not feasible with a moving camera and thus, the scene can not be simplified to the foreground. Motion calculation of image features through optical flow delivers indeterminate motion information as the whole image is moving instead of single parts in the image. In opposition to the system described in chapter 3, the following system uses target-oriented algorithms to handle the described problems due to a moving camera. In detail, the detection is based on single frames instead of subsequent frames and the tracking is able to deal with the movement of the robot. Again, depth information is used to enrich the feature set. Despite more informative trajectories in 3D, the additional gain of depth information is an enhanced stability for tracking.

In the following an outline of the chapter is presented. In the next section I am going to describe the essential differences of stationary computer vision systems compared to moving vision systems Sec. (4.1). Afterwards, current relative systems are introduced, which implement all steps of a complete tracking system (Sec. (4.2)). The systems give a picture of state-of-the-art in the area of person tracking on a mobile platform. The subsequent section introduces my system proposal and gives an overview of the functionality of each module. Afterwards, the different modules are explained in Sec. (4.4), Sec. (4.6) and Sec. (4.5). Finally, I give a detailed analysis of the algorithm in the results (Sec. (4.7)).

4.1 Request of Static and Moving Cameras

In this thesis I am presenting several ways of gathering visual information for situation awareness on a mobile robot. In the previous chapter, the aspects of a moving camera were circumvented by the use of the Vista space. In this chapter, the robot is moving during the action. Hence, the system has to deal with the demands of a moving camera. Here, I define the differences of a static and a moving camera, what clarifies the necessary usage of specific algorithms and techniques applied.

In the image of a static camera the intensity values of the background pixels usually underlie only small changes, because the background is not changing. On the other hand, the intensity values of foreground pixels, which here means moving objects, vary in their intensity values. Therefore, it is possible to establish simple threshold filters, which filter all points with a low difference in their intensity values (cf.Sec. (2.2.1)). This leads to remaining blobs in the image, where only moving objects retain. Accordingly, it is possible to detect motion in the image, which represents an easy detector for dynamic objects.

In the case of a moving camera the intensity values of all pixels change due to the motion of the camera. Consequently, movements in the scene can not easily be detected. Only an error prone preprocessing step of aligning the images and compensating the ego motion can enable the frame differencing for moving cameras. Visual odometry [118] [147] [3] is used to find features in both images and to calculate a homography or projection matrix, which warps the images onto each other. The projective warping induces small errors, which is reflected in the frame differencing process.

The next distinction between static and moving cameras is located in the placement of objects. An object can only be detected relative to the camera. If the camera is static, the relative coordinate system is equal to the world coordinate system. This fact is not true for a moving camera. The detection at each time step is relative to the camera, but a tracking process needs to know where to find the object in the next frame. The prediction of the object's position is dependent on the movement of the camera. If the camera is e.g. rotating, the prediction of the tracker has to incorporate this rotation in order to know that the object exhibits an additional rotational movement.

In a nutshell, three important facts have to be considered in the case of a moving camera:

- The scene and objects therein rapidly change their appearance due to lightening changes
- Frame differencing without further preprocessing is not possible for moving cameras
- The motion of the camera has to be incorporated in the tracking of objects

4.2 Tracking Systems on a Mobile Platform

In this section I state mobile tracking systems published in literature. Different to Sec. (2.3) not only the tracking part is of importance, but rather the complete process of detection and tracking of humans running on a mobile platform. The systems are explicitly described to give an overview of the state of the art in this area. All systems are build on the presented system approach of Chapter (2), which consists of sensor input, a detection module and a subsequent tracking part.

The early approaches of mobile tracking systems were often based on laser data [128] [179]. Laser data provides an easy approach to detect obstacles in a specific height of a scan line. Although restricted to the scan line, some algorithms achieve feasible results. Combined with additional sensors, laser information can be utilised to detect and track humans in the scene.

Many systems, which are mentioned in the literature, are modelled to detect and track only one specific person in order to follow this person to an arbitrary area. The ideas and algorithms have some parts in common with multiple target tracking, but they underlie less requests like detecting objects in different depths or they often do not handle occlusions. This special subject is also addressed, but with a minor focus.

Tracking multiple objects in clean highway or engineered situations has been studied quite successful in literature [23]. But, multiple person or object tracking on a mobile platform in a dynamic environment still poses considerable challenges for all state-of-the-art approaches. Accordingly, there exist only a few systems, which attempt to solve this high complex problem. Here, I state two different approaches, which are based on motion detection on the one hand and object specific classifiers on the other hand.

Additionally to the robot systems, I present the most important driver assistance systems, which try to analyse the environment in order to detect other road users. The intelligent driver assistance area is very related to the object detection and tracking on a mobile robot. Hence, the presented approaches in literature achieve fruitful results, adaptable also for mobile robots.

4.2.1 Laser-Based Tracking Systems

Schulz et al. presented in 2001 a first version of their human detection and tracking system based on laser data. The authors equip the mobile robot *Rhino* with two laser scanners at the height of 40 cm to sense the surrounding. The person detection is done in two steps. First, two subsequent scans are aligned in order to segment moving and non-moving parts of the scan. Second, the remaining parts are analysed if they fit one of two different patterns describing typical leg positions of humans walking. To track the determined objects the authors apply a variant of Joint Probabilistic Data Association Filters (JPDAF) [51]. Instead of using a Gaussian assumption of Kalman-Filter based JPDAFs the authors propose to use a sample-based version. The use of the JPDAF is meant to track different humans with one tracker and additional occlusion handling. In their experiments they show practical results, where the robot could track up to four

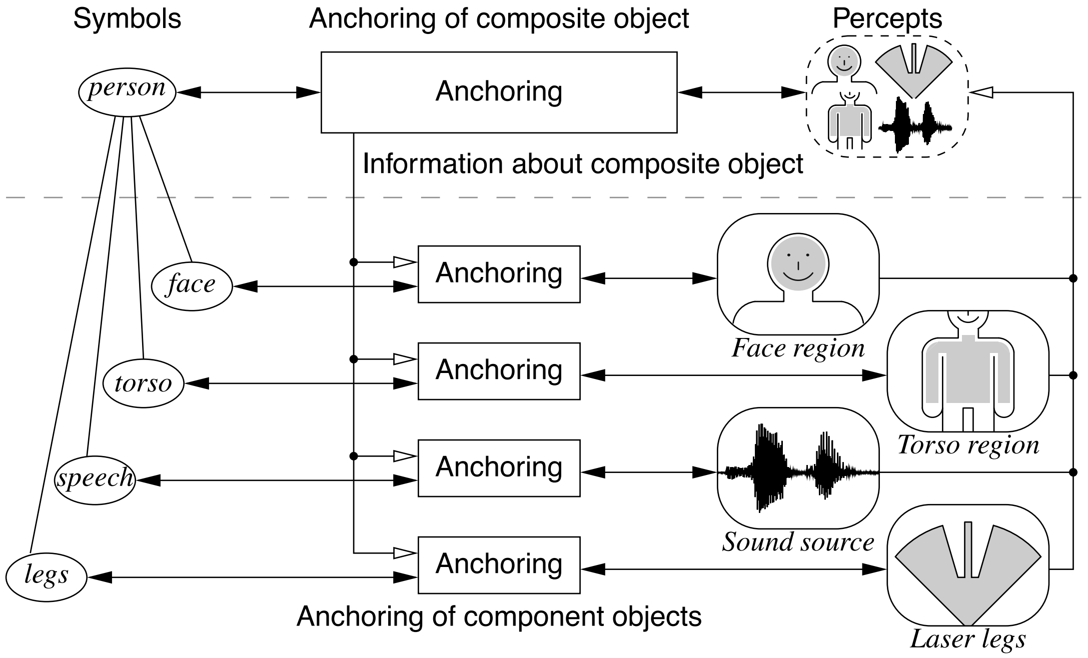


Figure 4.2: *Perceptual anchoring.* Different percepts describe the presence of a human. The torso colour is used to track a specific human. (Image found in [76])

people in an office environment. There are no comments about the computation speed of the algorithm.

Another approach of detecting people is the combination of laser data with other sensors like cameras. A common method is the combination of different cues coming from each sensor, which are integrated to a person hypothesis. Like above, a laser is designated to detect legs in the near field of the robot. As the purpose of a mobile robot is the interaction with humans, many researchers make use of face detection in addition to leg detection [67] [172] [20].

The robot Biron used in this work also has a human detection system, which uses several percepts to detect humans around him [76]. Leg detection, face detection, torso colour and a speaking analysis from a 3D microphone are fused in an anchoring process, where each percept adds a probability that a human is present. The torso colour is used to track a specific human during the interaction.

Typical laser data has the disadvantage that objects farther away are only detected with a low amount of laser points and hence, objects and background or even noise can not be distinguished. Objects, that are not seen in the scan line can not be detected, which is often a problem with tables. In the following, the detection and tracking with cameras is deeply investigated, because it promises better results due to the richer information of the sensor.

4.2.2 Person Following Tracking Systems

Many mobile robots are build to track one specific human in order to follow this person to an arbitrary area. One of the first human tracking systems on a mobile robot was presented in 1995 from Huber et al. [99]. This early system used an in-artificial model of people to detect them in the depth image. In the mentioned work the tracking was simply done by finding the nearest position of a candidate in the next stereo image.

Simple, but efficient approaches are often based on colour or shape. [175] offers a mobile robot platform, which tracks a human target based on the colour of its shirt. As colour model the normalised colour components (NCC) are used.

$$r_{norm} = \lfloor \frac{R}{R+G+B} \rfloor, g_{norm} \lfloor \frac{G}{R+G+B} \rfloor \tag{4.1}$$

For each colour they calculate the mean μ and the variance σ in the specified search region of the human torso. This forms a rectangular area in the NCC colour space, which meets the colour of the tracking region. In order to track the human body, every pixel is analysed, if it fits the searched colour. After some noise deletion the largest blob is chosen as the actual body area. The mean and the variance of the colour components are adapted over time through a simple adaptive filter $v_{t+1} = (1-\alpha)\tilde{v} + \alpha v_t$, with v_{t+1} as new value and $\tilde{v}$ as actual measurement. This model constrains the system to track humans with a uniform coloured shirt. To circumvent this fact, the authors combine their colour based approach with contours. Therefore, a canny edge filter is applied, which results in binary edge image. The rectangular image region of the person is taken as input and all edges within this region are taken as contour model. The contour model may evolve up to a certain threshold over time. In order to find the best fit of the model, all transformations of actual contour models are compared to the person model using the generalised Hausdorff-distance [102]. Both, colour and contour are merged by restricting the contour areas to similar colour regions.

The person following behaviour is based on disparity from stereo images and the displacement of the person relative to the image centre. The robot tries to keep the distance to the person equal and steers left or right if the angle α changes.

$$\alpha_x = \frac{\frac{\sharp_c}{2} - x}{\sharp_c} * 2 * arctan(\frac{ccdsize(x)}{2 * f}) \tag{4.2}$$

where $\sharp_c$ is the number of columns, $ccdsize(x)$ the size of the CCD-chip and f the focal length. The system provides a simple tracking approach, which uses no detection and which is only valid for really simple scenes. The work of [67] added a detection based on faces combined with colour, motion and contour information, which overcomes the restriction to manually label the initial person. In their work they additionally build a background map, which is used to avoid collisions of the robot with the environment. The work of Beymer [28] used a similar approach, which builds occupancy grids to find people and to build a background model. The tracking is done with low resolution stereo to hold the distance to the tracked person.

It is shown that a multi-modal approach is very efficient for the detection and tracking of people [76] [171] [86], especially if the person is interacting with the robot. Face, voice or movement detection are possible features to represent the presence of a human.

All mentioned systems often rely on the detection of faces or legs. These features are only existent, if the human is in front of the robot in a near distance. For the face, the person has to gaze at the robot and the leg-detector needs the human in a frontal- or rear-view to see both legs. Furthermore, the systems are only able to track one person at a time. For a following scenario this is sufficient, but for a situation awareness of the surrounding, the robot has to perceive all present humans and their movements from a greatest possible distance.

Summing up, the tracking of one specific person is not as difficult as the tracking of several different persons due to the lack of occlusions, less computational effort and the absence of a need for a hypotheses management. Additionally, the size of the person does not change very much, because the distance to the robot does not vary. The detection of the specific person is limited to the near-field and mostly, the person directly interacts with the robot, which makes it possible to use features as faces or legs.

4.2.3 Motion Detection Tracking Systems

Some interesting works try to compensate the ego motion of the robot or to learn a specific flow field of the movement to detect other ways of movements in the scene. Ego motion compensation is needed, if the camera moves during the data acquisition process. Perrone [160] e.g. learns the motion flow of the ego motion of the robot and separates target motion from background motion through the learned neural network. The optical flow is characterised, if it belongs to the own motion or to a target (Fig. (4.3)).

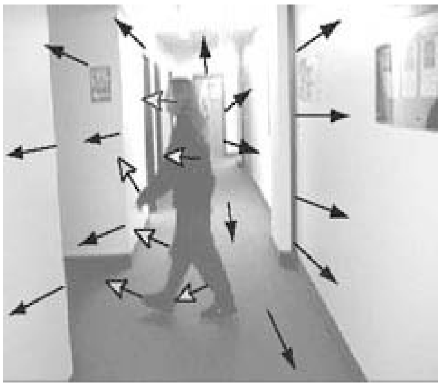


Figure 4.3: *Motion on a moving platform.* Motion different from the learned ego motion flow field (white arrows) indicates other moving objects. (Image found in [160])

Thereby, Perrone uses the characteristics of the human motion. This is only possible for lateral walking humans and a straightforward moving robot, because of the learned robot and human motion model.

Another way of compensating the ego motion is presented by Jung [111]. Assume a robot is located at (x, y, α) at time t. The data D is acquired at the same time. After some movement of the robot, the sensor is located at $(x + \Delta x, y + \Delta y, \alpha + \Delta\alpha)$ with data D' at time $t + \Delta t$. Both data D and D' can not be compared directly, because they are in a different coordinate system. Hence, data D should be transformed in the coordinate system from D' with transformation T. Jung et al. propose to estimate the transformation T in a direct matter by corresponding image features. In most cases the scene is not static, which causes errors in the estimation process, because some part of scenery has been moved differently compared to the background. Therefore, the estimation process needs an outlier detection algorithm, which filters all features from moving image objects. Jung uses the point tracker from Kanade-Lucas-Tomasi (KLT) [136], which is a promising standard technique in computer vision. The algorithm computes feature correspondences F^t and F^{t+1} for all subsequent image frames. The search window for each feature is chosen very small, assuming slow robot motion. Using the correspondence set $S =< F^t, F^{t+1} >$, the ego motion of the camera is estimated by a bilinear transformation model.

$$\begin{bmatrix} f_x^{t+1} \\ f_y^{t+1} \end{bmatrix} = \begin{bmatrix} a_0 f_x^t + a_1 f_y^t + a_2 + a_3 f_x^t f_y^t \\ a_4 f_x^t + a_5 f_y^t + a_6 + a_7 f_x^t f_y^t \end{bmatrix} \tag{4.3}$$

The transformation for image I^t to the image I^{t+1} is defined as T_t^{t+1}. The cost function for least square optimisation is given by

$$J = \frac{1}{2} \sum_{i=0}^{N} (f_i^{t+1} - T_t^{t+1}(f_i^t))^2 \tag{4.4}$$

with N number of features. The feature set has to be separated in moving and non-moving features, before the final transformation is calculated. The first step is the computing of an initial estimate T_0 using the whole feature set. Second, the feature set is partitioned into two subsets

$$\begin{cases} f_i \in F_{in} & \text{if} |f_i^{t+1} - T_{0,t}^{t+1}(f_i^t)| < \epsilon \\ f_i \in F_{out} & \text{otherwise} \end{cases} \tag{4.5}$$

Finally, the transformation T is calculated with the subset F_{in} only. The building of subsets and the transformation can work correctly, as shown in Fig. (4.4). The problem with this calculation is the assumption that only a small amount of features is moving in the picture. If there is too much motion in the image or there are not enough static background features, the main motion can not be estimated and thus, the separation of the features fails. Of course, the parameter ϵ has to be chosen correctly to filter out the moving points.

If the assumption is working, the frames can be handled as if the robot did not move. Hence, frame differencing can be applied for the transformed image I^t and the image I^{t+1}. The difference in the frames results in moved particles and noise, which separates from the background. The authors propose to use a probabilistic detection algorithm

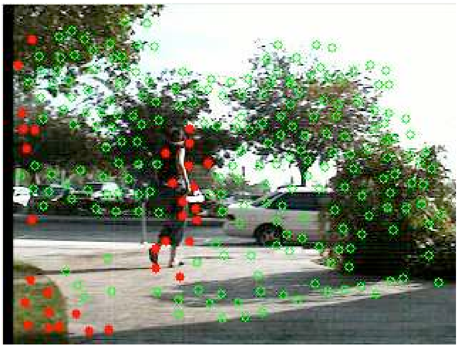


Figure 4.4: *Motion subsets.* The features are divided in two subsets. The one set belongs to self-moving objects (filled red circles) and the other to the background (green empty circles). (Image found in [111])

to get rid of the noise. The posterior probability distribution $P_m(x^{t+1})$ over the state $x = [x, y, \vec{x}, \vec{y}]$ is calculated as follows.

$$P_m(x^{t+1}) = \eta^{t+1} P(I_d^{t+1}|x^{t+1}) \int P(x^{t+1}|x^t) P_m(x^t) dx^t \tag{4.6}$$

$P(I_d^{t+1}|x^{t+1})$ is a perception model and $P(x^{t+1}|x^t)$ a motion model, which have to be used for updating the probability. The perception model is implemented in a generic form as step function with limited evaluation range. The motion model is assumed to be a constant velocity model, because the object's motion is not known a priori. The authors propose to estimate the posterior probability distribution recursively utilizing an adaptive particle filter. The weight of each particle $s_i^t = [x_i^t, y_i^t, \vec{x}_i^t, \vec{y}_i^t]$ is determined by

$$w_i^t = \frac{1}{m^2} \sum_{j=-m/2}^{m/2} \sum_{k=-m/2}^{m/2} I_d(x_i^t - j, y_i^t - k) \tag{4.7}$$

The motion model is used to update the position of the particles by the motion of each specific tracked object. The particle filter is designed to change its number of used particles. This is implemented by a kd-tree, which influences the number of particles by its size. The kd-tree is additionally used to cluster the particles by transforming them into a lower-resolution, uniform-sized grid. Each grid with a particle density higher than a certain threshold is selected as a candidate. All candidates are clustered by their connectivity, where each cluster represents an object. Finally, the statistics of the particles in the cluster are calculated by summing up the characteristic of each particle.

Instead of using one particle filter for tracking all objects, the authors implement multiple particle filter in their system. They create one additional particle filter, if the last particle filter converges to a new object. If a particle filter diverges, the particle filter is destroyed. This shelters the risk of false associations or wrong deletions of hypotheses. A hypotheses management would provide a better way to regulate the creation and deletion of hypotheses (cf.Sec. (4.5)).

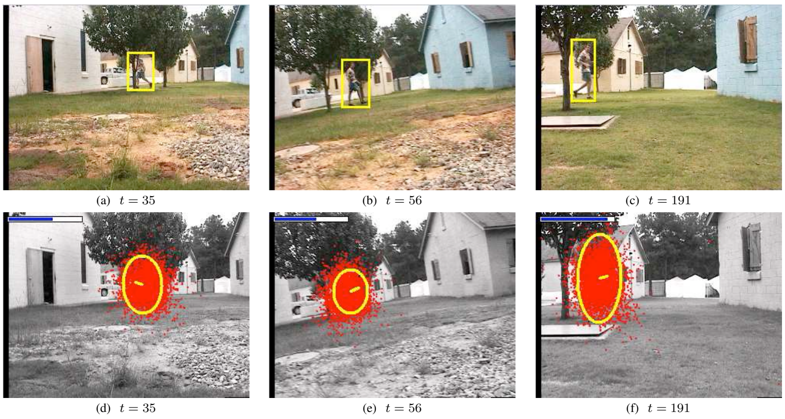
To track objects in 3D, a laser range finder is added to their system. The laser data is projected into the camera image using the standard pinhole model. The depth of tracked objects is acquired through partial information from the laser data. In general, this procedure is very expensive, because the different sensors have to be calibrated and

Platform	Frames	Motions	Detected	True +	False +	Detection Rate	Avg. Error
Robotic helicopter	43	35	28	28	0	80.00 %	11.90
Segway RMP	230	220	215	211	4	95.90 %	21.31
Pioneer2 AT	195	172	158	146	12	84.88 %	15.87

Figure 4.5: *Results from [111].* The table shows the overall performance of the system presented from Jung and Sukhatme. (Table found in [111])

synchronised. In this thesis sensors are used, which circumvent the overload through the use of active cameras, which directly provide additional distance information.

The authors show their system performance on different platforms, which are a robotic helicopter, a segway and a pioneer2 robot. The system reaches 5 frames per second on a low resolution image of 320*x*240 pixels. The tracking results are compared to manual labelled tracks of the objects. The exact results of their evaluation are shown in Fig. (4.5). *Frames* is the number of total frames, *motion* is the number of moving objects, *detected* the count of detected objects, *True+* is the number of correct detected objects and *False+* the number of false-positive objects. *Detection Rate* shows the percentage of moving objects correctly detected, and *Avg. Error* is the average Euclidean distance in pixels between the ground truth and the output of tracking algorithm. The qualitative system results are shown in Fig. (4.6).


(a) $t = 35$ (b) $t = 56$ (c) $t = 191$

(d) $t = 35$ (e) $t = 56$ (f) $t = 191$

Figure 4.6: *Experimental results from [111].* The upper row shows the manual labelled moving objects. The lower row shows the system result. The accordant frame number is printed below each image. (Image found in [111])

The approach of estimating the ego motion and detecting humans by motion is limited to specific motions (cf.Perrone) or the scene has to consist mostly of static parts, which can be used to detect the ego motion (cf.Jung). Wrong estimates of the ego motion result from wrong feature correspondences or frames, where the specific requests are not fulfilled. Other systems [59] try to compensate the risk of estimating a wrong ego motion by adding an additional Kalman filter. But, this also compensates only a few frames and can not avoid some failures due to wrong estimates, if e.g. a person is walking directly in front of the camera and there is no static data available. Hence, the detection of persons by motion is hazardous or unstable in some cases and should be avoided.

4.2.4 Mobile Robot Tracking Systems

In the following, I present related work to mobile robot tracking systems, which fulfil all requirements of a mobile tracking system. This includes a person detection, the handling of occlusions and the tracking of several targets in the scene.

I do not consider systems, which use a static human detection and tracking on a mobile platform like in the previous chapter or like [17]. In e.g. [17] an autonomous mobile robot is exploring the city in order to reach a certain target. The authors use human detection and tracking in order to achieve a conversation with pedestrians. But, the detection and tracking is based on background subtraction and needs the robot to be static. In their case, the robot is standing still, waiting for pedestrians walking by. Here, I only describe systems, which are able to detect and track humans during ego motion.

One popular mobile tracking system is presented from Bastian Leibe, Andreas Ess et al. [61] [59] [60] [130] [131], which evolved over many years to the actual reference system. The system is designed to track multiple persons, or additionally, cars in outdoor environments. Here, the tracking of dynamic obstacles is meant to deliver important information for a path planing algorithm. The system uses camera images from a mounted stereo cam on a mobile platform. The authors propose an approach, which uses camera position, stereo depth data, ground-plane estimation, object detections and trajectories based on visual information. The tracking information is used to predict future motions from dynamic objects in order to incorporate this information in a static occupancy map. Fig. (4.7) shows the system proposal.

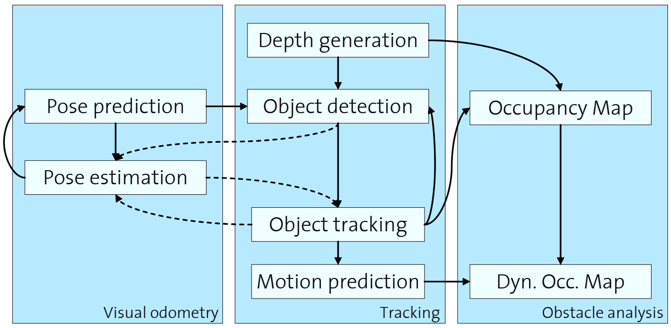
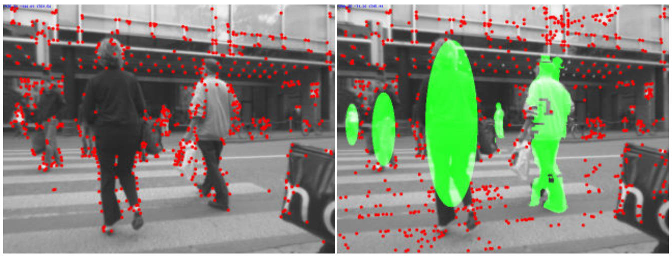


Figure 4.7: *System set-up from [59].* Flow diagram of the system presented from Leibe, Ess et al. . (Image found in [59])

First of all, they calculate a 3D map through structure from motion [50]. Utilizing the depth map the pose of the camera is estimated and updated through time by visual odometry. Therefore, the image is divided in a grid of $10x10$ bins. In each bin a similar amount of Harris Corners is calculated. The information from the tracker is incorporated to mask out bins, which belong to a possible moving target (see Fig. (4.8)). All remaining features are analysed with RANSAC to estimate the camera movement. The trajectory is smoothed with the last 15-18 frames utilizing Bundle Adjustment. Additionally, the authors implement a failure detection mechanism, which uses a Kalman-Filter estimate instead of the visual odometry. It is not mentioned, when this mechanism applies, but one could assume that it is used, if the estimates are too different.

The next step is the simultaneous detection of the ground surface and possible human targets. The system uses the presumption that every interesting object has to reside on

Figure 4.8: *Visual Odometry Masking.* The moving of the robot is calculated through visual odometry. Harris Corners are used as corresponding features (left). Corners on known targets are masked out (right). (Image found in [59])



a common ground plane in the scene. Thereby, the scene is modelled with a Bayesian Network. This network is constructed for each frame and it models the dependencies between object hypothesis, their corresponding depth and the ground plane. The ground plane is estimated from previous estimates and the actual depth measurement. The probability to detect a searched object at some place is computed through its position, size and depth and additionally through a classifier and the candidate trajectories from the last time steps. Thereby, the size is modelled as a Gaussian and as classifier the authors use the Histogram of Oriented Gradients approach.

The tracking of the detected objects is applied through tracking by detection. All detected objects are marked with their global position in a space-time volume. At each time step Kalman-filters are started from each detection resulting in an over complete set of trajectories. The trajectories are finally resolved through a global optimisation step, which tries to maximise joint probability. Some results of the system are presented in Fig. (4.9).

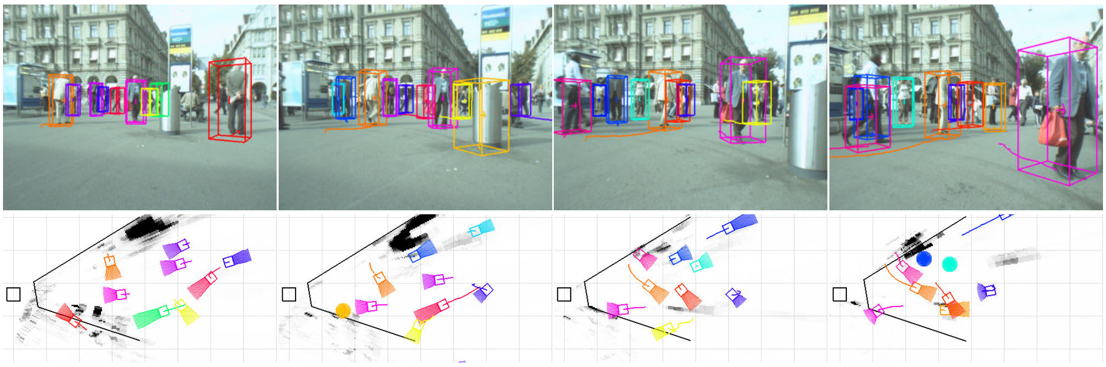


Figure 4.9: *People detection and tracking.* In the upper row the 2D images of the particular frames are shown. In the bottom row the birds-eye-view of the found tracks is plotted. (Image found in [59])

In [77] they show an extension to their system, which adds articulation information to the tracks of the single persons. They implement their tracking with a Gaussian Process articulated tracking approach based on global pedestrian silhouettes learned from a data set. This restricts their detection to humans with a low walking speed and with a known articulation. They show in their results, that the system is able to detect also the articulation of persons at a reliable detection rate if the camera is static. In the case of a moving camera they show only qualitative results, which permits the assumption that the system is not as stable as in the static case. The quantitative results of their current system are presented in [61]. The tracking rate of persons is at 73 % (shown for one sequence with 999 frames length) with 1 false positive per image. The rate slightly increases, if the detection range is restricted to 15 meter. The authors report detection improvements through a ground plane assumption and by using better stereo data. Additionally, other authors report good results, but on the cost of computation time. The computational efficiency of the system in [61] is reported as not fully real-time capable. They spot the Histograms of oriented Gradients detector as bottleneck for their system. As a solution they propose an implementation on a graphic card, which speeds up the computation by parallelisation. In my thesis I show another possibility to speed up the calculation of the Histograms of oriented Gradients detector by a fast and reliable pre-detection step.

The system from Leibe, Ess et al. delivers promising results and it represents the current state of the art reference system in the area of mobile tracking systems. In my thesis, I do not claim to achieve better results than their system, which is not possible to compare anyway. I focus on a system, which is able to detect and track humans in near and far positions and even during occlusion, while running in real-time on a mobile robot. Hence, the focus is slightly different, but emerging in similar results.

Few other systems exist in literature, like the system from N. Belotto, which uses multi sensor fusion to detect legs through a laser and to identify persons by their colour histograms from a video camera [20]. The system from Abd-Almageed et al. [1] uses partial similar approaches like the proposed system in this thesis with some variation in single modules. Their detection is based on the v-disparity, which reasonably restricts the image search space. The v-disparity is also used in this thesis and thus, will be discussed in Sec. (4.4.1). In short, the v-disparity is utmost useful to detect both the floor and possible ceiling. In their work, the remaining image parts after removing the floor and ceiling are clustered through mean shift. On each region the authors utilise a combined human detector of Histograms of Oriented Gradients and a cascade of boosted features (based on the work of Zhu [223]). Instead of SVM, they use a simple Fisher linear discriminant.

4.2.5 Vehicle Tracking Systems

Finally, I indicate related work to advanced driver assistance systems. They provide similar solutions like the algorithms running on mobile robots, but they have to deal with the specific requirements of outdoor and traffic scenarios. Hence, the algorithms are comparable and I want to give an overview of this related area. Driver assistance systems aim at warning and assisting the driver in dangerous situations, where the usual driver would cause an accident. This includes appropriate protective measures, which should turn on, if the driver is inattentive. Initially, these systems were based on simple mechanisms like seatbelts. Afterwards, new and more complex systems were developed, like the adaptive cruise control, which holds the car in its lane. Most important is the progress in pedestrian protection systems to avoid injuries in vehicle-to-pedestrian accidents.

Daimler research e.g. is working on pedestrian detection since the late 90's [78] [212] [213]. The quite actual system PROTECTOR [80] is able to detect pedestrians based on stereo vision. In their system they use several closely coupled modules, which narrow down the image search space for each subsequent module. The first important step is the reduction of regions of interest (ROI), which reduces computation time a lot. Gavrila et al. propose the use of a calibrated pair of cameras, which delivers a sparse disparity map. In this disparity map a shape-based pedestrian detection [78] delivers person hypotheses. The shape-based detection is generally a template matching based on the chamfer distance transform. The templates are arranged in a hierarchy, where whole sets of templates can be efficiently matched. Each hypothesis is classified in pedestrian and non-pedestrian by a neural network with local receptive fields [212]. A verification step uses the flat world assumption in order to fit a second order polynomial over the dense disparity values in each hypothesis area. The parameters are chosen according to the measured depth at the corresponding distance. If the area contains more background or other depth values than the expected, the area is discarded. To overcome gaps in detection the authors use an $\alpha - \beta$ tracker on a 2.5-D bounding box around the object. The system showed a good performance of slightly above 60% detection rate with up to 5 false detections per minute and fast processing time. Fig. (4.10) shows an example detection of the proposed system.

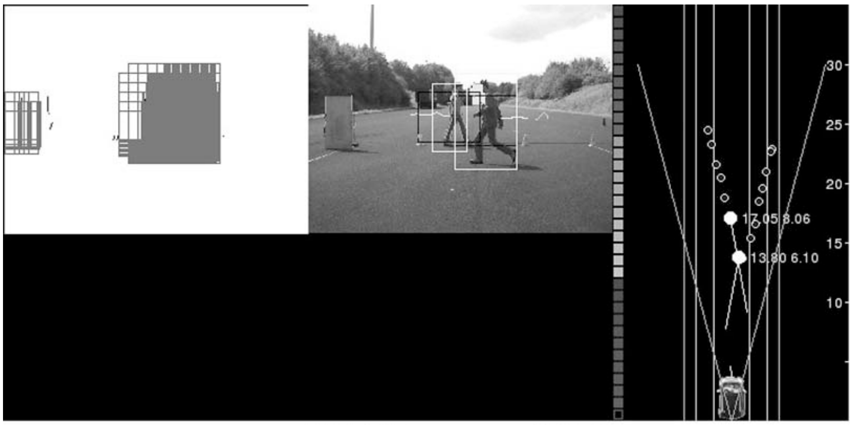


Figure 4.10: *People detection and tracking.* (Image found in [80])

Research on pedestrian detection has employed different areas of monocular vision[181] [138] [9], stereo vision [80] [129] [187], LIDAR [164] or thermal imagery [40] in the last years. A detailed overview can be found in [81].

In 2004, Grubb et al. published their work on *3D vision sensing for improved pedestrian safety* [87]. They developed a system, which uses stereo vision to detect and track pedestrians on a moving vehicle. From stereo vision they build a disparity map, which is used to detect the objects in the v-disparity image. The hypotheses are classified in pedestrian and non-pedestrian by shape classification with SVM's. The used shape models were taken from front, rear and side poses. The hypotheses are tracked by a Kalman filter. Their average detection rates are about 83% with average false detection rate at 0.4%.

One recent system from Bajracharya et al. describes an interesting approach to detect humans on a moving vehicle [13]. The authors propose to use geometrical features to detect upright standing people in depth data. First the stereo data is projected on a two-dimensional polar-perspective grid (The authors state that the polar-perspective grid better preserves the coherency of the stereo range data than a traditional Cartesian map). Afterwards, the data is preprocessed by a 3D box of 1 m x 2 m x 4 m to reduce the possible number of region candidates. Each possible region is classified by further geometric features in human and non-human. Tracking is implemented as region of interest association by the 3D position and the Bhattacharyya distance of a colour (RGB) histogram. Fig. (4.11) shows the detection performance of the system. The omitting of a texture based classifier results in some false detections due to the similarity of the geometric dimensions of some objects to humans (see the bus in the shown Fig. (4.11)).

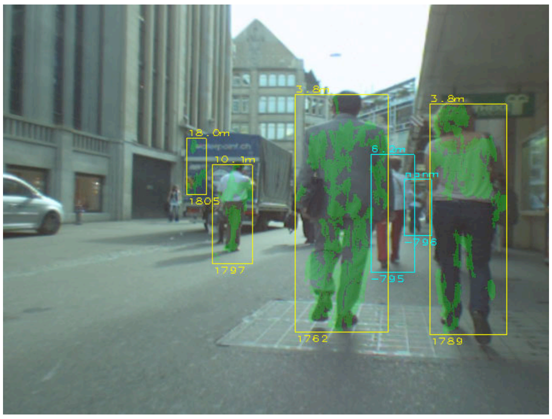


Figure 4.11: *People detection and tracking.* Yellow boxes with a green overlay show the detections of the system. Cyan boxes are missed detections. There is a false detection on the bus. (Image found in [13])

Summing up, most actual systems running on robots or other platforms rely on depth data, which reduces the amount of false positives compared to pure 2D images [28] [80] [13] [81] [60] [173]. In literature, there exist quite a lot of mono approaches, which deal only with a 2D image frame. However compared to the correspondent 3D approaches, all authors report an enhancement using depth data. Consequently, the approaches in this work rely on depth data. All presented approaches are not able to detect only partially visible humans and they do not cope with near and far persons. Additionally, the related systems are partially only able to track one person in order to follow him/her. In this thesis a tracking system is presented, which accounts for partially seen humans as well as near and far persons. Additionally, the proposed algorithms are able to handle multiple persons simultaneously. In the results it is shown that my system approach is able to deal with all mentioned requirements of human detection and tracking on a mobile robot platform also considering the computation speed.

4.3 A Modular Person Tracking System

In order to achieve the second category of SA on a mobile platform a system has to incorporate different abilities in order to detect and track correctly all present persons. Despite the correct detection and association of humans, the system has to deal with short occlusions, different lightning conditions and the movement of the robot. If only one aspect of these conditions fails, the tracking is not guaranteed. Consequently, the proposed system takes these conditions into consideration and implements different methods, which solve these requirements. The occlusions are handled by the tracking

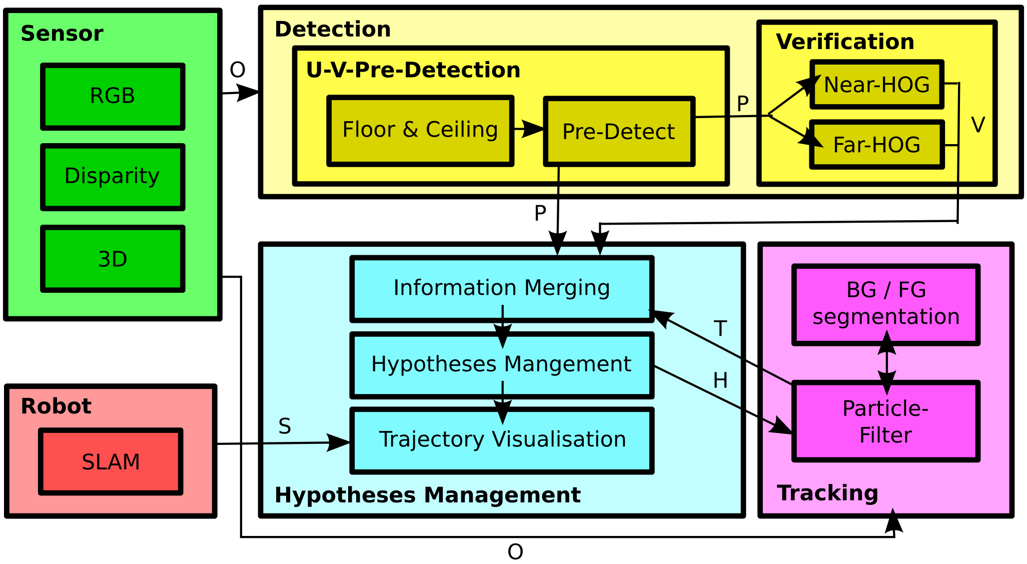


Figure 4.12: *Tracking System Design.* The sensory input delivers RGB, distance and 3D information as observation O. Additionally, the robot provides SLAM information S about the global positioning of itself. All sensory data O is used to detect and track humans in the scene. After removing floor and ceiling a pre-detection based on u-v-disparity delivers efficient hypothetical windows $P = p_i$ with $i = 1 \dots N$, where an object with similar 3D dimensions is present. Verification of these areas is done with an Histograms of oriented Gradients (HoG) based support vector machine. If the human is very close a Near-HOG is applied, which is trained only on the upper body. Else a Far-HOG full human body detector is used. All verified detections V are handed over to the hypotheses management (HM), where all incoming information is meaningful merged. The HM cares for creating and deleting hypotheses. If a new person is detected, the HM informs the tracking module to create a new particle filter tracker for each new person. Additionally, the information about all current hypotheses H is delivered in order to adapt the particle filter target model to the current size of the object. The tracker itself informs the HM about his actual tracks T. The HM also incorporates the mapping information S from the robot in order to provide global consistent hypotheses. Finally, the HM provides the tracking information for a visual output.

module, where the tracker preserves the specific object, even it is shortly occluded. Additionally, the tracker updates dynamically its observation model in order to refresh its picture of the object and to keep track of it even under varying lightning conditions. The tracking itself incorporates the movement of the robot and updates its prediction model accordingly. Summing up, the system is designed to detect humans both fast and independent from the ego motion of the robot.

4.3.1 System Overview

The proposed system architecture is presented in Fig. (4.12). It shows the different engineered modules and their connections. The system runs on the mobile robot Biron , which is presented in section 1.1. The robot needs a meaningful sensory input, which is achieved by a multi-sensor set up. The sensor set up consists of a RGB camera, an infra-red camera and an infra-red emitter. All sensors are devised in the Microsoft Kinect camera for the XBox Sec. (2.1.5). The infra-red camera measures depth data from a light pattern. The depth image is calibrated onto the RGB image and synchronised with at most 16 milliseconds difference. Hence, the robot is able to use colour, depth and 3D information as observation O. As the robot is moving and the tracked positions of the humans are related to the robot position at the certain time step, it is required to transfer the positions into world coordinates. Therefore, the actual world position of the robot is acquired through a simultaneous localisation and mapping (SLAM) approach [221]. The SLAM is working on the data of a Laser-range-finder, which is installed on the basis of the robot. At each time step the actual position $S_{x,y}$ and view angle S_θ of the robot are incorporated into the tracking results in order to save the world position of each entity as well.

The next step consists of processing the data O to segment and detect all humans currently in the scene. Humans are a very dynamic category with many different poses, clothes and appearances. Humans also look differently depending on the view from the front, back or side. Hence, it is complicated to detect humans in all different situations. In Sec. (2.2) several methods to detect humans are presented. Nearly all of these approaches only detect humans using frontal or back views. One possibility is to train several classifiers in order to detect humans from different poses. But, the execution of a classifier is expensive and the execution of several classifiers would slow the system down. Here, the detection of the side, back and front views is applied through an additional pre-detection algorithm.

The pre-detection algorithm has two advantages. First, the heavy computation time of the classifier can be drastically reduced, because only windows from the pre-detection have to be classified instead of the whole image. The following citation originates from the driver assistance area, but explains the importance of the search window reduction.

> *"For example, a typical exhaustive scan on a* 640*x*480 *image can provide from 200,000 to 1,000,000 ROIs, depending on the sampling step and the minimum ROI size. In contrast, sampling just the estimated road can reduce this number to 20,000-40,000, again depending on the density of the scan. Furthermore, stereo-based segmentation could further reduce this number by at least a factor of 10, depending on the content of the scene."*
> GERONIMO 2010, [81, P. 1243]

Hence, the search space for a subsequent classifier is immensely reduced by a 3D based pre-detection. The second benefit of the pre-detection is due to the reduction of false-positive detections. This derives from the additional use of three-dimensional features. The 3D features constrict the classifier to reasonable regions and prevent false detections due to similarity of the background to the pure 2D image features from a human.

Pre-detection can be accomplished through several methods. The ground plane can be estimated and only objects on this plane considered [60] [79] or geometrical features can be calculated and only objects with accordant size retained [13], to name two examples. Both presumptions speed up the detection process and are consequently used in this work. A preprocessing step measures the ground plane Γ_{ground} and ceiling plane $\Gamma_{ceiling}$ and removes these parts from the processing by defining a mask, which is applied from each process module. Here, the ground plane assumption can not be applied as adoption for a possible hypothesis location like proposed in some literature, because the humans are often directly in front of the robot and are only viewed from hip to face. But the removal of the accordant areas reduces the amount of features to inspect. The geometrical features can be used to examine and classify only areas, where an object has similar 3D dimensions like the searched object. It is important to mention, that the dispose of geometrical features requires an additional classifier. Otherwise, the detection includes many false positives (cf.[13]). I consider this fact by applying an additional verification step, which stabilises the process.

Using the depth data and the width and height (see Sec. (2.1.2)) the dimension of the object can be calculated and compared to the search dimensions. Here, an approach based on u-v-disparity [1] [22] [125] [126] [192] [152] with a probabilistic detection equation is applied.

All pre-detected regions p_i with $i = 1 \dots M$ with adequate size and good probability of being a searched object are passed to the classifying step. The windows are scaled to the classifier size and the window is verified, if the detection can be passed to the hypothesis generation. If the probability of the sub-window is over a certain threshold, the detection p_i is verified as v_j with $j = 1 \dots M$. In this thesis, the state-of-the-art *Histograms of oriented Gradients* (HoG) classifier in conjunction with a linear SVM is applied. The HoG classifier shows very good results compared to the actual literature (see Sec. (2.2.2)) and it is the best human detector in the near range [55]. The low computation speed, which is often described in literature, is compensated by the pre-detection step. The hypotheses management also associates all incoming information from each module to the known hypotheses. As the pre-detection and tracker are not bound to a frontal or back view of the human, the system is able to keep track of the human independent of the view of the human. Hence, it also circumvents the need for different classifiers for each perspective.

The hypotheses management has got the important function to manage the construction and removing of hypotheses h_i with $i = 1 \dots N$. During the process, the hypotheses management merges all new detections and tracks to update the known hypotheses. It also triggers the initialisation of a new tracker for each newly created hypothesis. A hypothesis is generated, if an object is newly detected and verified and which could not be associated to an already known hypothesis. The deletion of a hypothesis is initiated, if a person is not verified for a specific amount of time (here, at least once all 30 frames).

Getting a signal from the hypothesis management, the tracking module starts a new tracker t_i with $i = 1 \dots N$. In this thesis, a particle filter with an adaptive multi-dimensional observation model is applied, which also implements a human alike transition model. The tracker offers the possibility to track objects even during short occlusions and it makes the tracking process more robust, if e.g. the object should not be detected.

The tight binding of all modules facilitates a very effective tracking system, where all relevant information is incorporated into the detection and tracking process. This produces essential tracking information for the situation awareness of a mobile robot.

[1] u and v are the coordinates of a pixel in the (u,v) image coordinate system

4.3.2 System Integration on a Mobile Robot

All system modules are implemented in C++ using the vision framework *icewing* and the open computer vision library *opencv*. In general, the system is able to run on several platforms because of its modularity. Here, the experiments are accomplished on the mobile robot BIRON (see Sec. (1.1)). The complete robot system is running on two standard laptops. All components are communicating via the XML Based Framework for Cognitive Vision Architectures (XCF) [215]. The overall performance and results of the system are shown in Sec. (4.7).

4.4 Object Detection Module

The first step to a meaningful picture of the environment around the robot is to detect the humans in the scene. Here, a two stage detection algorithm is applied, which is both reliable and fast. This is achieved through the efficient combination of a fast pre-detection step with an accurate classifier.

4.4.1 Pre-Detection through U-V-Disparity

The pre-detection step provides a speed up of the subsequent classifier by restricting the hypothesis windows to a minor number. But, all unknown hypotheses have to be detected already by the pre-detection step in order to classify correctly all humans in the scene. If the pre-detection misses a new detection, the accordant human can not be verified any more. Thus, to really speed up the process without corrupting the results, the pre-detection has to be on the hand fast and on the other hand reliable.

My proposition here is the usage of the u-v-disparity [22] [125] [126] [152] [192], which offers two enhancements. First, the floor and the ceiling can easily be removed and second, possible hypotheses can be detected fast and reliable through their geometrical properties.

The u-v-disparity can be computed from an available disparity map I_Δ (see Sec. (2.1.2), cf.[125]), computed from e.g. a stereo image pair or a depth sensor. Let H be the function, which transforms the image variable I_Δ to the value in the v-disparity map $I_{v\Delta}$

$$H(I_\Delta) = I_{v\Delta} \tag{4.8}$$

The function H accumulates the points with the same disparity value that occur on an image line i. For the image line i in the v-disparity map $I_{v\Delta}$, the abscissa u_M of a point M corresponds to the disparity Δ_M and its grey level i_M to the number of points with the same disparity Δ_M. A value P in the line i maps to the position i_M as follows

$$i : i_M = \sum_{P \epsilon I_\Delta} \delta_{v_P, i} \delta_{\Delta_P, \Delta_M} \tag{4.9}$$

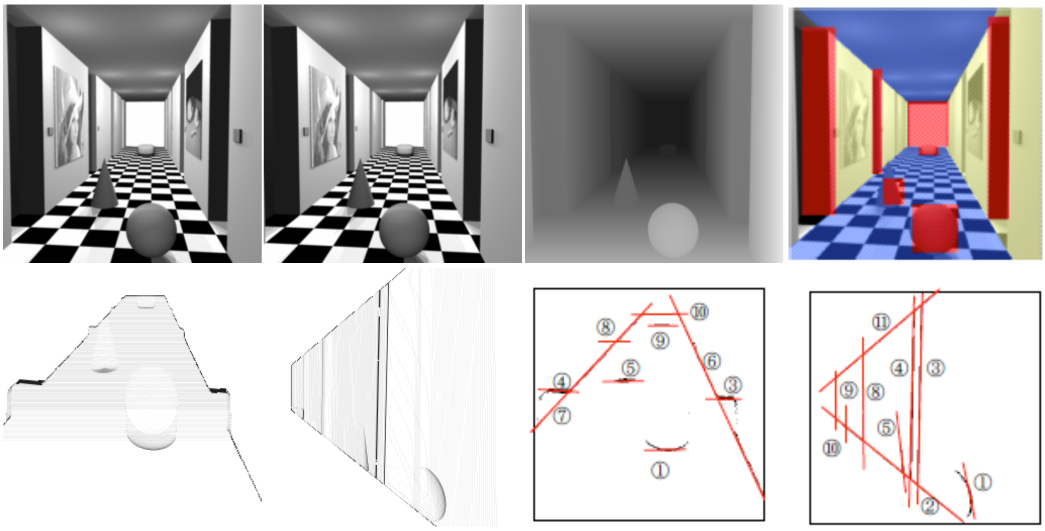


Figure 4.13: *Virtual example for u-v-disparity.* (From left to right) In the upper row the left and right camera image and the disparity image are shown. Additional, the image in color in presented. In the bottom row the u-disparity and the v-disparity are calculated. Applying a threshold and hough transform, the strongest lines can be extracted. Using the depth value Δ_M of a line in the u-disparity image a corresponding line with the same Δ_M in the v-disparity can be found. The line in the u-disparity image directly denotes the width and the v-disparity the height of the object. (Images found in [126])

with $\delta_{i,j}$ as Kronecker delta.

This mapping can be done in both rows and columns, where u-disparity corresponds to the accumulation in $\vec{u}$ and v-disparity to accumulation in $\vec{v}$. Both disparity mappings are useful for different purposes. The images in Fig. (4.13) clarify this statement. In the upper row, the original left and right image are shown. On the right is the disparity image and the original image in colour. The accumulation of disparity values per column is presented at the bottom left, which represents the u-disparity. The next image corresponds to the accumulation of disparity values per row, called v-disparity. Applying a threshold on the u-v-disparities, only the strong lines remain. Hough transform reveals lines, which denote the width (u-disparity) and height (v-disparity) of all distinctive objects in the scene.

A nice characteristic of the v-disparity lies in the presentation of the floor and ceiling. They are presented as oblique lines (2, 11 in the bottom right image). Using these lines, it is possible to remove the floor Γ_{floor} and ceiling $\Gamma_{ceiling}$ from the image in order to reduce the possible hypotheses locations. Both planes can be removed by applying a height-threshold to the (x, y, z) data of the scene. The height threshold is defined by two straight lines, which represent the height of floor and ceiling at each depth. Both lines are defined by the following line equation with gradient m and the point of intersection n with the axis of ordinates.

$$y = m * x + n \tag{4.10}$$

This assumption is only valid for a horizontal aligned camera, which is assumed to be true in the presented robot scenario. The error due to non-alignment is smaller than the error from the re-projection of the 2D image points to 3D. Otherwise the floor and ceiling have to be removed by two planes instead of two lines.

Let g be a straight line that is running through two points $P_1(x_1|y_1)$ and $P_2(x_2|y_2)$, in which x_1 and x_2 are different. Then, the gradient m of the line g can be calculated through the theorem of intersecting lines

$$m = \frac{\Delta_y}{\Delta_x} = \frac{y_2 - y_1}{x_2 - x_1} \tag{4.11}$$

Afterwards, the point of intersection n can be broken down by inserting the known variables. The values are taken from the found lines in the v-disparity image. Using the lines, every 3D point is checked, whether it lies above or below the ceiling and floor line. If the point resides above the ceiling line or below the floor line it is marked as unnecessary. Fig. (4.14) displays all marked points in red. All points, which are not correctly re-projected into 3D, are additionally added to the mask.

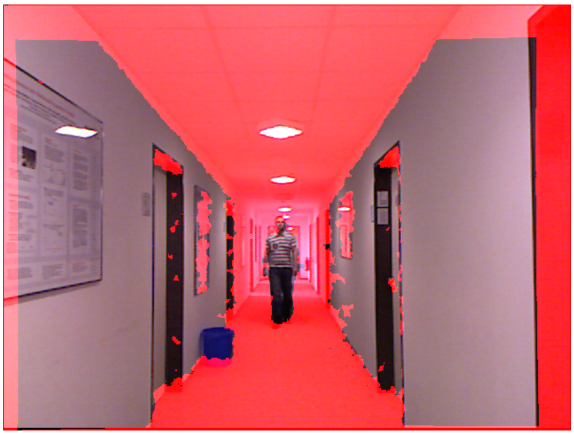


Figure 4.14: *Floor and ceiling removal.* The floor and ceiling are removed by a height threshold, which is applied for each 3D point. The height threshold is calculated by two straight lines. Here, the floor, ceiling and irregular 3D points are removed (in red).

After removing floor Γ_{floor} and ceiling $\Gamma_{ceiling}$, the objects have to be extracted. In literature, they propose to utilise Hough transform to find lines in the u-v-disparity images. Here, I propose to use a fast version of the connected components algorithm [94]. This approach is advantageous compared to Hough lines, because the depth of the object can be directly calculated and also dispersive objects can be detected. The speed-up of the usual connected components algorithm is achieved through an undirected graph in conjunction with disjunct datasets. The usage of disjunct datasets postulates a pure incremental graph without the possibility to remove edges. The time complexity for the whole process is $O(V + E\alpha(E, V))$, where E is the total number of edges in the graph and V is the number of vertices. α is the inverse of the Ackermann function [2].

The connected components algorithm is generally used with two stages. First, the connected components are labelled line by line. An additional id image is created, where the background is zero and each connected component gets its own id. The id's are allocated with the following neighbourhood (① corresponds to the current pixel position. The image borders are separately handled):

[2] The Ackermann function has explosive recursively exponential growth. Therefore its inverse function grows very slowly.

3	4	5
2	①	

If the pixel value of ① is above a threshold (~10, for removing noise) it is taken into consideration. The neighbours are analysed from 2-5, if already an id exists. If an id different from zero is found, the current position gets the same id. Additionally, it is checked, if other id's around are different from zero and different from the actual id. Thereby, the different id's are inserted as children in the graph. If no neighbour is different from zero, but the actual position has got an object, a new id is assigned.

If the whole image is passed through, the second stage of breaking down the id's to the real connected components is started. Each id is substituted with its representative, which corresponds to the smallest id from each branch of the graph. The outcome is an image, where all connected components have the same id and the background has the value zero.

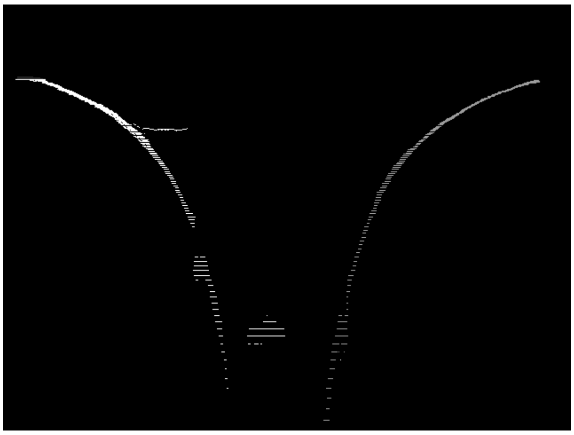


Figure 4.15: *Distance adaptive ID-connector.* Due to the sensor noise, objects in far distance collapse into several unconnected lines with different distance. The distance adaptive ID-connector connects these lines dependent on their global and relative distance.

For this thesis, the Microsoft Kinect camera is used, which directly measures the depth in centimetres instead of disparity values. Hence, the measurement error is growing with rising distance due to the detach in resolution of the structured light pattern. This results in lines, which are possibly not connected, but belong to the same object (see Fig. (4.15)). In respect to this fact, a distance adaptive id connector stage is proposed. To avoid multiple detections for the same object, an additional processing stage connects the already found connected components although they are not directly connected. The optional stage runs before the second stage of substituting the id's and iterates through all found components. Dependent on the distance of the object to the camera, an upright search beam looks at a few pixel with increased distance in the same column. If several lines are behind each other, it is assumed that the lines belong to the same object. Of course, the detection of an object close to a wall is hindered, but the affect of the lower amount of detection windows on the speed of the subsequent classifier weights definitely higher. Here, 4 different depth areas are used, where objects with a gap of 2, 4, 6 or 10 lines still belong together.

After the id image is calculated and all connected objects are detected, a pre-detection p_i is created for each connected object. So far, only the width and depth of the object can be extracted out of the u-disparity image. The width directly corresponds to the maximum distance in the x-dimension of the connected component. The depth calculates from the maximum offset in the y-dimension.

The height can be calculated from the v-disparity image. But, the lines are not always unique assignable to each other, if e.g. two objects have the same depth. The v-disparity is meaningful, if the scenery is on a plate without walls. Here, corridor data is used, where the walls are present at each depth step (see Fig. (4.16)). Thus, the height of persons can not be directly extracted.

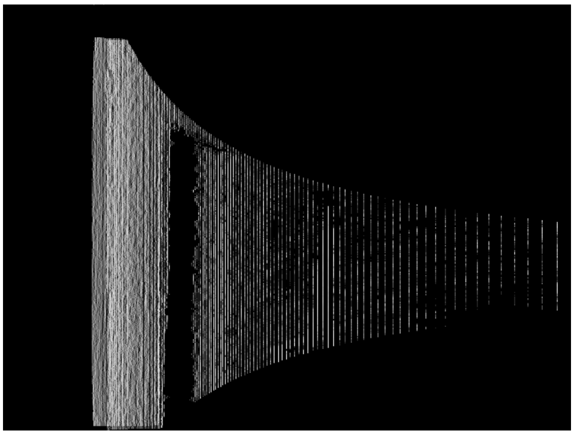


Figure 4.16: *V-Disparity.* The v-disparity can not be used for the calculation of the object's height, if the scenery takes place in a corridor. The walls are visible as upright lines. This hinders the association of an object line with a found object in the u-disparity image.

Instead, I propose to calculate the height directly in the original disparity image. The height is estimated by using the width $u_{w,i}$ and position (u_i, v_i) of each object. A proofing algorithm tests each row from bottom v_{max} to top v_{min}, if in the corresponding line a pixel with the depth of the object is present. v_{max} corresponds to the image height and v_{min} is zero in the brute force approach. Using the maximum length of the correspondent area in the v-disparity image fastens the search. The proofing algorithm starts from u_i and runs until $u_i + u_{w,i}$ for each line. The bottom is found, if a pixel has a similar depth value like the object. Similar means, the object's value plus minus a threshold (here, about 0.1 meter). The height of the object is reached, if no pixel with similar depth value can be found in a line.

Finally, a bounding box (u, v, u_w, v_h) and the real world values x, y, z, x_s, y_s, z_s are determined for each pre-detection p_i from the gathered information. These pre-detections are probabilistically matched against the geometrical dimensions of the human w_H, h_H, d_H. The width x_s, depth z_s and height y_s of the object have to be similar to those of a human (The human size is chosen as $0.7x1.65x0.7$ metre).

$$width_{prob} = \frac{1}{\sigma_w\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{x_s - w_H}{\sigma_w}\right)^2\right) \tag{4.12}$$

$$height_{prob} = \frac{1}{\sigma_h\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{y_s - h_H}{\sigma_h}\right)^2\right) \tag{4.13}$$

$$depth_{prob} = \frac{1}{\sigma_d\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{z_s - d_H}{\sigma_d}\right)^2\right) \tag{4.14}$$

$\sigma_w, \sigma_h, \sigma_d$ are the standard deviations for the correspondent probability distributions. If the object's dimensions are similar to those of a human, the final probability is calculated as follows:

$$p_{prob,i} = width_{prob} * height_{prob} * depth_{prob} \tag{4.15}$$

If the probability $p_{prob,i}$ is satisfactory, the pre-detection is preserved. Else, the pre-detection is removed from the set. The remaining values are handed over to the classifier, which makes a final determination whether the object is a human or not. In Fig. (4.17) an example of the pre-detection is shown. All objects with similar dimensions to a human are marked with a red rectangle. In the upper left corner the minimum depth of the object is denoted, while in the lower right the maximum distance of the object is shown.

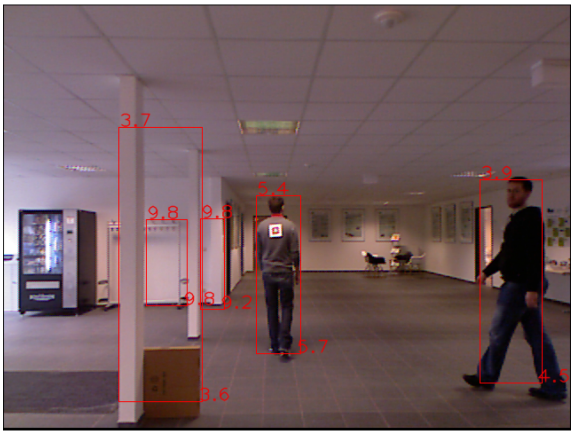


Figure 4.17: *u-v-disparity human detection.* The pre-detection searches for objects with similar size dimension compared to a typical upright standing human. The size is calculated through the u-v-disparity. All found objects are marked with a red rectangle. In the upper left and lower right, the minimum and maximum depth of an object is denoted. Areas with similar dimensions are also found by the pre-detection.

4.4.2 Detection Verification

The pre-detections p_i with $i = 1 \ldots M$ are each validated by a classifier. The classifier is meant to filter out false positive from the reduced set of possible windows. Here, the Histograms of oriented Gradients are used in conjunction with a linear SVM, because it shows best detection results in near distance human classification (cf.Sec. (2.2.2)).

The input windows are first checked for their size, because the width and height have to be reasonable to inherit the searched object. If the window is too small or too low, the region is classified as non-human. Else, the windows are resized to fit the detector window size and subsequently classified.

The scenario requires some more effort to recognise humans in all instants. As the purpose of the robot is to interact and communicate with humans, the robot often has little space between itself and the human. Furthermore, if the robot moves along a corridor the space is limited and thus, the crossing with humans happens in close distance. Therefore, a distance adaptive human classifier stage is proposed. Is the human pre-detected in close distance only the upper part of the body is classified. Should the human be farther away the standard full human body classifier will be applied. Accordingly, two classifiers are trained, one for the close distance and one

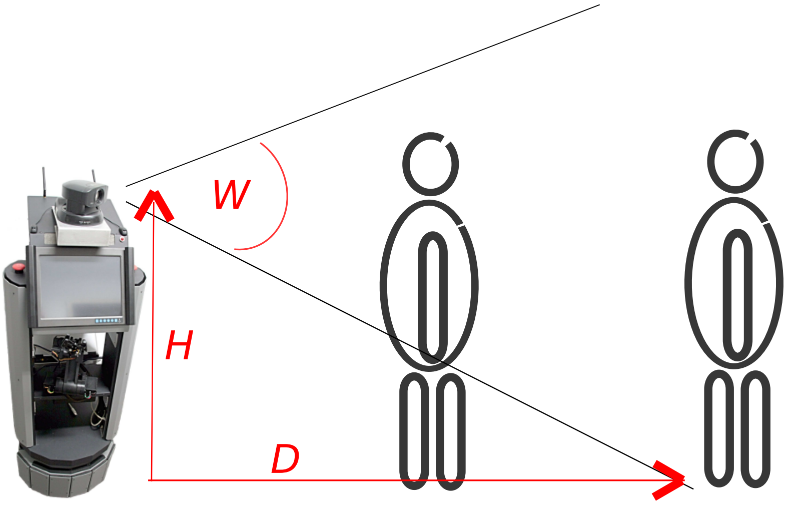


Figure 4.18: *Distance-adaptive classifier verification.* The classifier is chosen distance dependent due to the opening angle of the camera W and the height of the sensor H. Is the person is only partially visible the near classifier is activated and should the person be fully visible the far classifier is chosen. Hence, D resolves, which classifier is used for the verification step.

for the increased distance. The classifiers switch depending on the opening angle of the camera W and the height of the sensor H (see Fig. (4.18)). The far classifier is chosen when the distance z of the object is greater than D.

$$D = tan(90 - W/2) * H \tag{4.16}$$

D is calculated under the assumption that the camera is horizontally and vertically aligned. Small aberrations to this assumptions do not effect the calculation, because a small security value of a few centimetres is added to D.

If a window is classified as human (close or far), the pre-detection p_i is marked as verified and copied to the current verified detections v (see Fig. (4.19)). Finally, all pre-detections p and verifications v are committed to the hypotheses management.

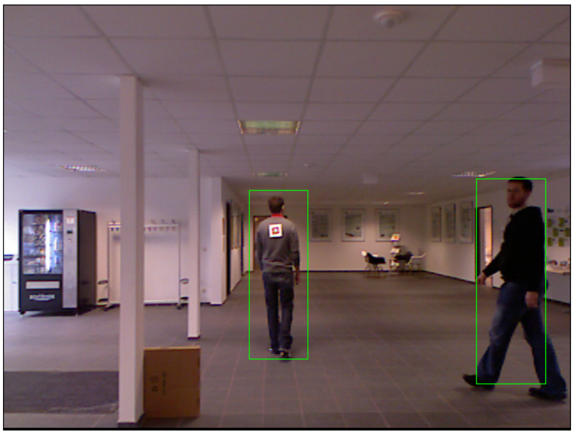


Figure 4.19: *Detection verification.* All hypotheses from the pre-detection (see Fig. (4.17)) are verified through the Histograms of oriented Gradients classifier. All verified detections are marked with a green rectangle.

Additionally, if the pre-detection detects an object, it separates the foreground object from the background. This is done by a k-means algorithm with 2 centres of mass. One for the back- and one for the foreground. All pixels in the detected sub-window are segmented due to their distance information. The resulting foreground is additionally marked with a connected component algorithm in order to provide an easy contour describing only the foreground. Utilising this information, the tracking is able to use only pixel information from the foreground without mixing it with the background.

4.5 Hypotheses Management

The hypotheses management has got the important function to manage the construction and removing of hypotheses h_i with $i = 1 \dots N$. A hypothesis h is constructed, if a pre-detection p is verified a specific number of times as v. Here, the pre-detection and classifying are robust and the number can be set to one or two.

A hypothesis consists of the following information

$$h_i = [u, v, x, y, z, u_w, v_h, x_s, y_s, z_s, \vec{\delta}, \vec{\delta}_{world}] \tag{4.17}$$

with (u, v) image and (x, y, z) relative world position, (u_w, v_h) rectangle size in image, (x_s, y_s, z_s) world size in each dimension and the known positions $\vec{\delta}, \vec{\delta}_{world}$ as trajectory. The positions for the trajectory are saved as world positions $\vec{\delta}_{world}$ and relative positions to the robot $\vec{\delta}$. Then, the trajectory can be analysed related to the ground truth and each track can be shown in a world map. All relative hypothesis positions are transformed into world coordinates using the following equations:

$$x_{world} = z * sin\varphi + x * cos\varphi \tag{4.18}$$

$$z_{world} = z * cos\varphi - x * sin\varphi \tag{4.19}$$

where φ corresponds to the viewing direction of the robot or the camera, respectively. The height y of the hypothesis is kept, because it does not change during the transformation from relative to world coordinates. x_{world} and z_{world} are still relative to the robot, but in world coordinates. Hence, the current position of the robot has to be added to the positions in order to get the world position relative to the origin. The position of the robot (x_{slam}, z_{slam}) is acquired through the global mapping, running on the robot.

$$x_{origin} = x_{world} + x_{slam} \tag{4.20}$$

$$z_{origin} = z_{world} + z_{slam} \tag{4.21}$$

During the tracking process the hypotheses management assigns new pre-detections p, verifications v and tracking results t to the existing hypotheses. A known hypothesis is merged with the new information, if it has the closest distance from all hypotheses to the new information and if the distance is below a threshold. The threshold is chosen as 0.5 meter, which has shown to be sufficient and stable for merging the information. The size and positions are merged using the update formula

$$h_i = \alpha * h_i + (1 - \alpha) h'_i \tag{4.22}$$

with h'_i as known hypothesis and h_i as actual measurement. The update rate α is chosen dependent on the reliability of the input. The pre-detection has low confidence, which is marked with a low value for α, while the verification has high information input, which is rated with a high α.

The tracking information t is incorporated by using only the position information without the size, because the actual tracker does not adapt the size on its own. The size of the tracker is changed by the hypotheses management, which informs the tracker about the current known size of the object. This is reasonable, because the pre-detections and verifications determine the size already. This saves computation time on the side of the tracker.

If a new hypothesis h_i is initialised, the hypotheses management informs the tracking module, that a tracker t_i has to be started.

Does a person leave the scene, the hypothesis has to be deleted after a short time. The hypothesis is kept for a few frames, because the person could return into the scene or be revealed after an occlusion. Should a hypothesis h_i be not detected or verified for a specific amount of time, the hypothesis is removed and the tracker is informed that the tracker t_i has to be deleted. Here, the value of 30 frames or two seconds shows a stable performance. Hypotheses are not deleted during occlusions in this time, but are successfully removed, if the person leaves the view space. Thereby, the tracker does not update its state, if its confidence is too low. Thus, the tracking process does not create indeterminable results, even if the person is already gone.

4.6 Tracking Module

If a new hypothesis h_i is created, the hypotheses management informs the tracker about the presence of a new person. Then, the tracker initializes a new tracking process t_i for the accordant hypothesis. Thereby, the tracker has got a tracking process for each hypothesis on its own. Here, I propose the use of a dynamic particle filter with multidimensional observation model. The particle filter is advantageous compared to other tracking processes, as it can deal with occlusions and non-linear state spaces (cf.Sec. (2.3.5)). This way, a stable tracking process is achieved even in the occurrence of ego motion.

In contrast to Sec. (3.5), the state of each hypothesis $t(a)_i$ does not describe an elliptical model. Here, it is represented by a point, which consists of the actual relative position and velocity.

$$a = [x, z, v_x, v_z] \tag{4.23}$$

The coordinates (x, z) represent the position of the object on the ground plane and (v_x, v_z) corresponds to the velocity of the object. Position and velocity values are relative to the robot. The state a does not contain any information about the size of the object, because the size is dynamically adapted by the simultaneous detection process. The tracker incorporates the current size into its calculations by adapting the target model. This reduces the dimensionality of the state space and offers a faster calculation procedure.

The target model σ of a hypothesis should characterize best the tracked object. Here, I propose the use of both 2D image features and 3D position in order to handle occlusions and similar object appearances. The hypotheses management provides all necessary information included in the hypothesis h_i. The particle state is initialised using the included 3D information. Applying the 2D image rectangle of the hypothesis, the target model is calculated. The 2D image features consist of a colour histogram, where the *HSV* colour space showed more stable results than the *RGB* space in the presence of illumination changes [170]. To remove the intensity changes, only trimmed H-channel and S-channel are included in the histogram representation. Chromatic information is not reliable, if the component is too small or too big and hence, pixels on this situation are not included in the histogram. Each target model σ is comprised by b_h, b_s bins for hue and saturation.

Let $c_{i=1...n}$ be the positions of pixels, which are used to create the target model σ. The pixels are taken from the foreground, which originates from the contour information of the detection process. Each location of a pixel is associated by $f(\{c_i\})$ to a bin of the histogram corresponding to the colour of the pixel, with the function $f : \mathbb{R}^2 \to \{1 \dots m\}$. Each histogram bin calculates like follows

$$\sigma_{bin} = \frac{1}{n} \sum_{i=1}^{n} f(c_i) \tag{4.24}$$

Once the target model is calculated and the state from the hypothesis is saved, the particle filter is created in order to track the human in the subsequent frames. In each frame the target model σ is compared to each measured colour model π using the *Bhattacharyya* coefficient [4]

$$\rho(\sigma, \pi) = \sum_{bin=1}^{m} \sqrt{\sigma_{bin}, \pi_{bin}} \tag{4.25}$$

The coefficient ρ gives a similarity measure of the colour models in the range of $[0, 1]$, where 1 relates to a perfect fit and the similarity decreases with dropping value. The resulting similarity is again weighted with the similarity of the estimated object position $(\hat{x}, \hat{z})$ and the measured object position $(\tilde{x}, \tilde{z})$. The similarity is weighted by the euclidean distance

$$P = C * \rho * K((\tilde{x}, \tilde{z}), (\hat{x}, \hat{z})) \tag{4.26}$$

with

$$K((\tilde{x},\tilde{z}),(\hat{x},\hat{z})) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{\sqrt{(\tilde{x}-\hat{x})^2+(\tilde{z}-\hat{z})^2}}{\sigma}\right)^2\right) \tag{4.27}$$

as Gaussian weighting kernel and C as a normalisation constant (to get a probability between$[0, 1]$). This calculation aims to achieve an appearance and localization dependent weighting where both the position and the colour model have to match the accordant track. One of the weighting factors reduce the similarity in the case of a bad estimation in order to avoid mismatches. The concatenation is especially useful, if an object is occluded by an object with similar appearance. The distance weighting resolves the wrong match where the appearance would join the occluding hypothesis. If two objects are very near, the appearance gives the key evidence to separate both objects.

The particle filter represents each hypothesis by a set of particles Φ_t at time t. Each particle $x_{j,t}$ with $j = 1 \ldots P$ relates to a complete copy of the hypothesis h_i (cf.Sec. (2.3.5)). The particle sampling follows the same sampling strategy like in Sec. (3.5). The particles are redistributed according to a random movement and a forward movement motion model. This covers most possible movements of a person including the movement of the robot.

In order to calculate the probability distribution, each particle is back projected into the 2D image. The measurement Z_t is incorporated through the image information. A rectangular region around the projected position can be determined using the known size of the object. For each particle the underlying colour histogram and the world position of the rectangular region are calculated. The world position is determined as average in the middle of the rectangular region. The weight of each particle is estimated with Eqn. (4.26). The final position of the current particle state $t(a)_i$ is estimated as weighted mean of all its particles.

$$t(a)_{i,t+1} = \frac{1}{P}\sum_{j=1}^{P} P_j * x_{j,t}(a) \tag{4.28}$$

Afterwards, the tracker updates its internal target model using the final position. Thereby, the colour histogram σ is updated using the standard update formula (with σ' as new measurement).

$$\sigma = \alpha * \sigma' + (1-\alpha)\sigma \tag{4.29}$$

If the overall probability of all particles is very low, the update step is skipped in order to prevent false incorporation of wrong target features.

Finally, the outcome of each tracker t_i including its current probability are postulated to the hypotheses management. Fig. (4.20) shows an example output of the predicted particles as purple rectangles back-projected into the image. The green rectangle represents the actual state of each hypothesis.

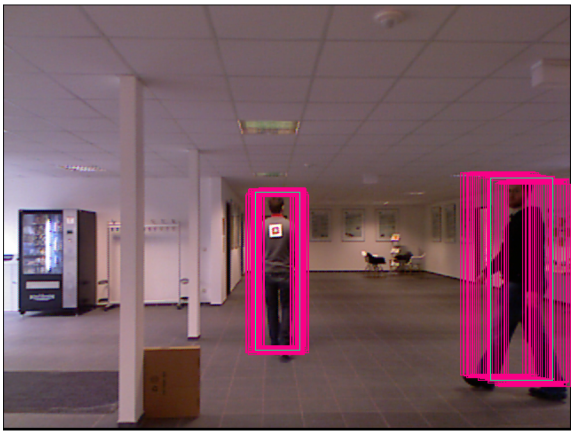


Figure 4.20: *Particle Tracking of each hypothesis.* For each verified hypothesis a particle filter is created. Each sampled 3D position is back-projected into the image and shown as purple rectangle. The green rectangle represents the actual state.

4.7 Experiments and Results

In the following a deep and comprehensive analysis and evaluation of the proposed mobile tracking system and each of its components is presented. First, the self-recorded datasets are introduced comprehending difficult and dynamic scenes. Thereafter, a qualitative and a quantitative analysis of the complete system is conducted in order to show the capabilities and performance of the system. Here, it is not possible to compare the outcome to other systems, because on the one hand other systems are not publicly available and on the other hand the robotic platform and its sensors are very specific. Additionally, the focus of this thesis relies on the realisation of an awareness system running in real-time on a mobile robot, which can not be compared to offline systems which are not restricted to small computational resources. Instead, the detection and the tracking modules are individually investigated in multiple subsections due to their enhancement compared to other approaches in literature. The focus of this thesis lies on the speed and reliability of the complete system whereby each module on its own has become reduced importance.

4.7.1 Evaluated Datasets

The datasets used for evaluation should represent best the aspired scenario in Sec. (1.1) and concern the specific requirements of narrow rooms or corridors and persons near and far from the robot. In order to show the capabilities of the system to track persons even during the occurrence of ego-motion I generated 12 data sets in the corridors and the entrance hall of our research laboratory. Each dataset consists of 360-2140 frames recorded with the Microsoft Kinect Camera directly on the robot with ~15 frames per second and $640x480$ pixel resolution. The datasets include one or several persons, which enter and/or leave the scene randomly. Hence, the system has to cope with the creation and deletion of hypotheses all over the scene. All characteristics from a real interaction scenario are incorporated in the scenes. The scenes show persons, which are near and far (see Fig. (4.21(a)) and Fig. (4.21(b))), moving and non-moving and they could be occluded or partially seen. The persons show different articulations and different poses, which enhances the need of a strong detection and tracking algorithm. Additionally, the scenes involve clutter and very challenging lightning conditions. The light sources are spots on the ceiling creating strong intensity gradients and shadows. The brightness of the scenes change rapidly from very dark to bright (see Fig. (4.21(b))). The robot itself is moving in all sequences. The movement contains fast forward movements and many rotations. Especially the rotations are very difficult to handle, which is taken into account by the presented system.

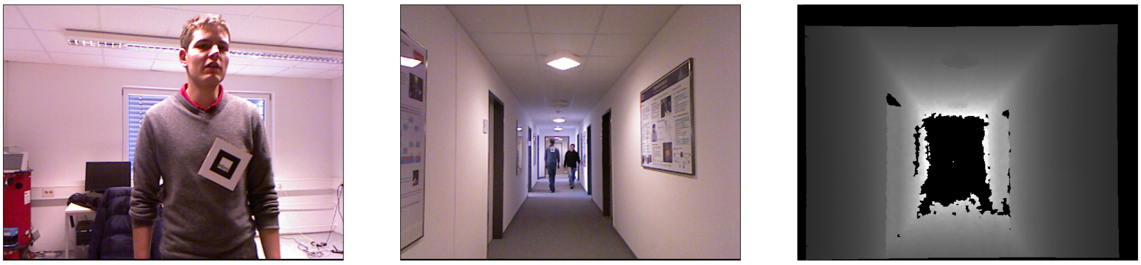

(a) (b) (c)

Figure 4.21: *Difficult scene conditions.* Persons can be near (a) or far (b). In the case of (b) the person is too far for the distance sensor (c). In (b) spot lights on the ceiling create strong intensity gradients and dark and bright areas. The gradient illumination on the wall clearly pops out.

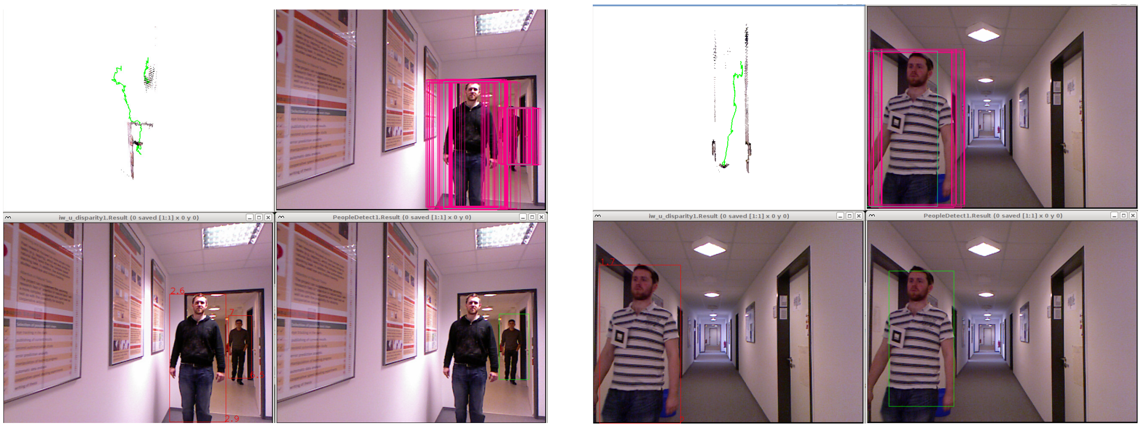
In order to compare the reliability of the vision tracking system with another independent sensor, the laser data is recorded. Depending on the laser data, the SLAM data for navigation purposes is also backed up. Using the SLAM data it is possible to create world coordinates from the detections, which transforms all data in a common coordination system.

A ground truth of the trajectories of each human in the scenes is created in order to provide a comparison for the system results. The ground truth is generated by hand, where each fifth frame is manually labelled and the intermediate frames are interpolated. The ground truth consists of a bounding box around each person and the corresponding 3D position of the centre of the bounding box, assuming that the centre represents the 3D position of the object. The ground truth is only created for persons up to 10 meter, because the sensor capabilities are reached at this distance (see Fig. (4.21(c))). Some example pictures of the datasets are shown in the appendix B.

4.7.2 Qualitative Results of the Proposed System Approach

The qualitative results visually demonstrate the system results through images of the trajectories. The trajectories are connected points of detection, which represent the pathway of each entity over time. Therefore, the trajectories are consistent with the temporal linking of information, which describes the second category of situation awareness. Hence, if the trajectories meet the real movements of the persons in the scene, the presented system satisfies the aspired function of creating temporal links between already known information.

The following images (Fig. (4.22) to Fig. (4.24)) show a snippet of the tracking of some sequences (the remaining sequences are shown in appendix B). The images illustrate



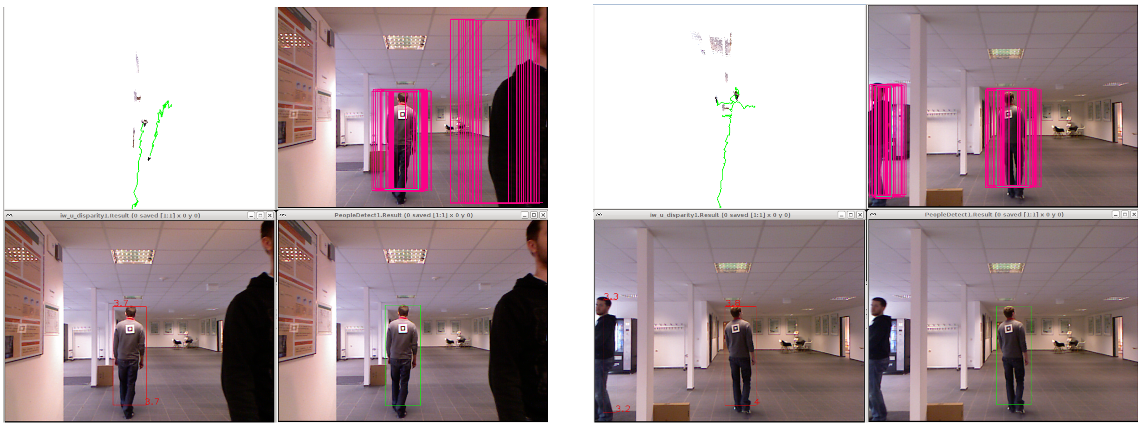
(a) Set 1

(b) Set 3

Figure 4.22: *Qualitative tracking analysis.* The pre-detection is presented in the lower right of each set. All detected areas are frames with a red rectangle. In the lower right, the verified persons have a green rectangle. If a person is correctly detected, a particle filter is initialised (purple rectangles, seen in the upper right). The green rectangles represent the state of each particle filter. In the upper left, the trajectory is plotted in a birds eye view. The 3D points of the actual camera view are also plotted.

the performance of the complete system by representing each step in an own image. The pre-detection starts in the lower left where each pre-detected area is denoted with a red rectangle. The hypotheses are handed over to the verification step in the lower right where true objects have to be verified. If they are already verified, they have to be approved in at least every 30th frame in order to monitor that the hypothesis is still valid. If a region is verified as a human, a particle filter is started, which is shown in the upper right as purple rectangles. The green rectangles represent the actual state of each hypothesis. The arising trajectory is painted in the upper left as green line in a three dimensional plot viewed from above. The already detected positions of each entity are transformed into world coordinates in order to save the global movement of each person. Thereby, it is possible to transform the elapsed points of detection into the current coordination system of the robot. This is important, as the robot coordination system permanently changes due to ego motion. Hence, the green trajectory is relative to the position of the robot in every frame, which means that the trajectory does not directly reflect the world movement of the tracked person, but the position relative to the robot at the specific time. Because the robot is moving and rotating itself, the trajectory can inherit rapid movement and direction changes. The 3D plot also shows the 3D scene of the actual frame relative to the robot, where the 3D points are superimposed with the associated colour information from the calibrated colour camera.

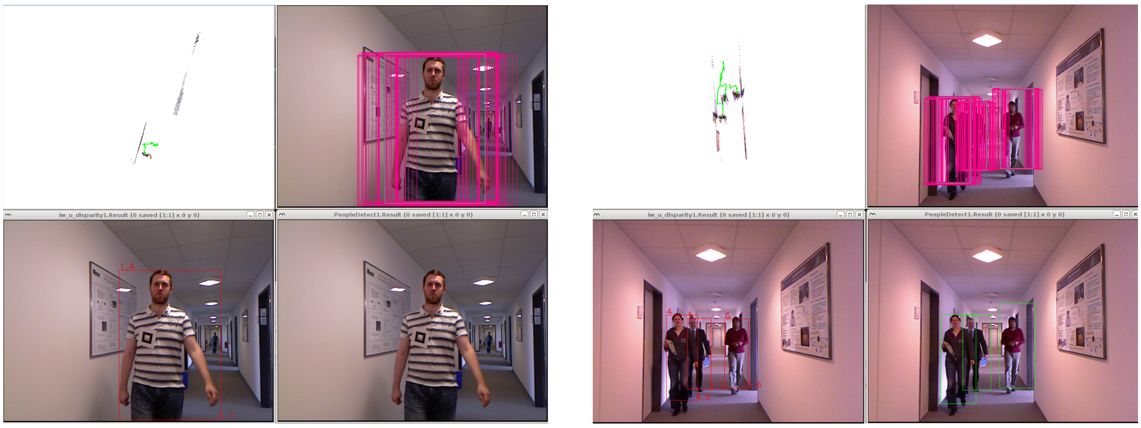
The first Fig. (4.22) shows two sets where the first set includes many rotations of the robot, but the trajectory appropriate represents the position of the person. Even the second person in the background is correctly detected and tracked. The second set shows that the system is able to detect and track a person reliably coming from far away to near to the robot. During the movement the person changed his appearance from fully visible to partially seen and walked beneath some spot lights at the ceiling, which changed the colour representation of the person a lot. The marker partially seen on the clothes of some persons are not used in the whole processing.



(a) Set 6 part 1

(b) Set 6 part 2

Figure 4.23: *Qualitative tracking analysis.* The pre-detection is presented in the lower right of each set. All detected areas are frames with a red rectangle. In the lower right, the verified persons have a green rectangle. If a person is correctly detected, a particle filter is initialised (purple rectangles, seen in the upper right). The green rectangles represent the state of each particle filter. In the upper left, the trajectory is plotted in a birds eye view. The 3D points of the actual camera view are also plotted.



(a) Set 11

(b) Set 12

Figure 4.24: *Qualitative tracking analysis.* The pre-detection is presented in the lower right of each set. All detected areas are frames with a red rectangle. In the lower right, the verified persons have a green rectangle. If a person is correctly detected, a particle filter is initialised (purple rectangles, seen in the upper right). The green rectangles represent the state of each particle filter. In the upper left, the trajectory is plotted in a birds eye view. The 3D points of the actual camera view are also plotted.

The second Fig. (4.23) shows two parts of the same scene. This scene represents twofold important actions. In the left image, two persons walk in different directions, while the robot is moving forward. Each track from far to near or from near to far is correctly created. The second image shows two tracks where one person occluded the other, but both tracks are accurately revealed. This shows the system's capability of tracking persons even under occlusions.

Fig. (4.24) shows the detection and tracking of a short sequence, where the robot rotates itself and the person is moving directly in front of the robot. Finally, the image of set 12 shows the detection and tracking of several persons walking side by side. In this sequence the two persons on the left first walked directly in contact far from the robot. This leads to only one detection of a present human, because of the noisy depth in the far distance and the connection of both persons. At the moment, where one person walked a bit in front of the other, the second track is started as both persons are detected independent from each other.

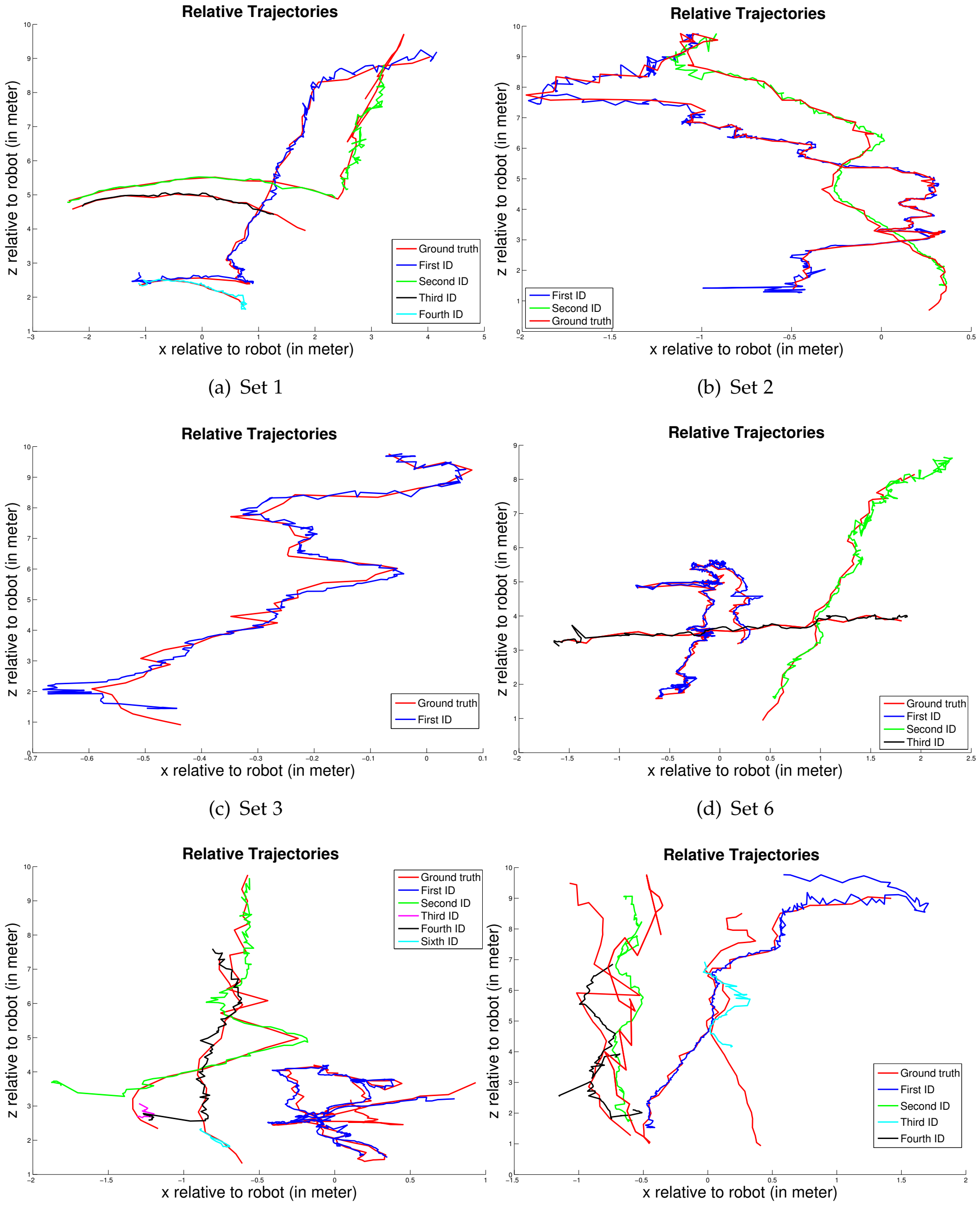
The following images show the trajectories of the persons compared to the ground truth labelled trajectories. In each figure the current relative distance from robot to person is denoted on the y-axis, while the relative difference in lateral position corresponds to the x-axis. The different colours equate to the different id's, while the red colour always represents the ground truth. Comparing the calculated trajectories to the ground truth, they show very similar characteristics (cf.Fig. (4.25)).

Fig. (4.25) presents the trajectory results from set 1, 2, 3, 6, 11 & 12. In set 1 a person is walking in front of the robot with increasing distance (blue). Above 10 meter the tracking system shortly removes the hypothesis due to missing measurements. At the moment the person returns into the sensor range it is again tracked, but with a new id (green). The track very well fits the ground truth. In set 2, the robot moves and rotates, but all persons are correctly tracked. Other examples are given with set 3,6,11 & 12. Even during occlusion the tracking performs very well (cf.set 6). In set 12 all but one person are tracked over the whole pathway (in cyan). The detection and tracking fails partially, because of the pre-detection.

The resulting world trajectories are also shown in an example in Fig. (4.26). It shows only trajectories with a length of at least 30 frames or two seconds in order to remove all false positives. The robot trajectory is shown in dashed red with blue arrows denoting the viewing direction. The different persons are shown in different colours representing their Id's. All trajectories are complete, even during occlusions and show the full paths of the persons.

The qualitative analysis reveals that the system performs very well in detecting and tracking persons from a mobile platform. The persons are fast detected and stable tracked in nearly all cases. Even if a person overtakes the robot and changes rapidly his/her appearance and size due to the visibility of the body the system performs successfully. Anyway, a few errors are related to false positives, which pass the classifier without being a real person. However, all false detections only persist a short period of time, which makes it easy to filter them out afterwards. In two cases persons are only partially tracked, because the detection fails. This is related to the pre-detection, which is based on the geometrical dimensions of the objects. Here, one person pushes himself on the wall in order to make room for the robot or other persons. Thus, the person fuses with the wall for the eye of the pre-detection (see Fig. (4.27)). In the most right image, the resulting ids are marked in terms of colour. The person and a part of the wall are merged as one object (shown in green).

Summing up, the detection and tracking is visually very stable with a few false detec-



(a) Set 1

(b) Set 2

(c) Set 3

(d) Set 6

(e) Set 11

(f) Set 12

Figure 4.25: *Trajectory set 1, 2, 3, 6, 11 & 12.* The ground truth is denoted in red, the persons in different colours. Each colour represents a different ID. In the upper right, the ID change is due to the restriction of the sensor, because the person is shortly further away than 10 meter. In the upper left image, 5 persons are correctly tracked. In set 3 and 6 are all persons correctly tracked even during occlusions. In set 11 and 12 are small errors visible and in set 12 is one person only partially tracked (cyan), because the person walks in contact with the wall, which hinders a pre-detection by the system. In general, most persons are reliably tracked.

tions and some small errors. The detailed analysis of the reliability of the tracks and the correct number of false detections is fulfilled in the subsequent section.

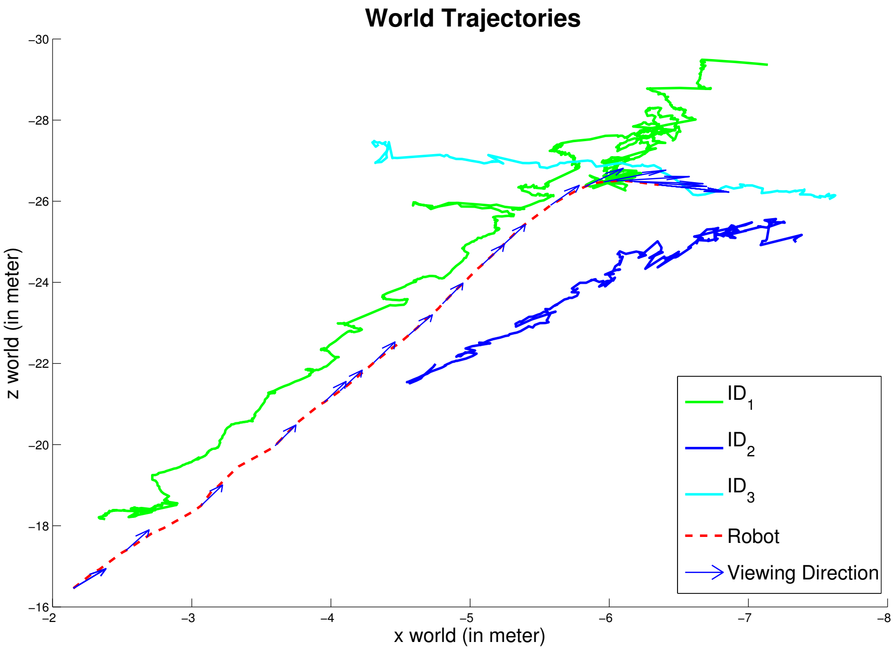
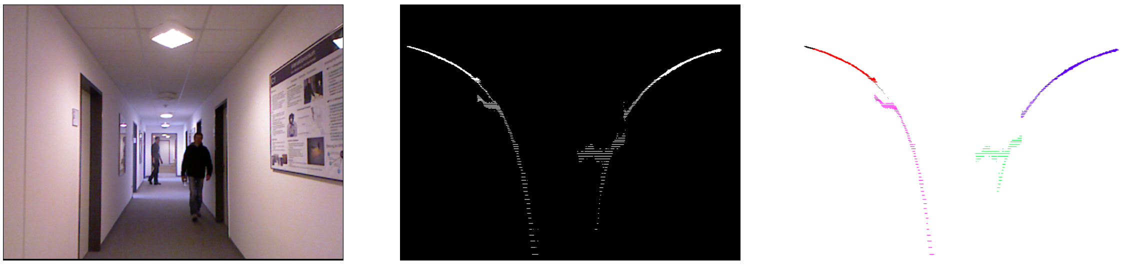


Figure 4.26: *World Trajectories set 6.* The pathway of the robot is shown in dashed red, with blue arrows denoting the viewing direction of the robot. The persons are shown in different colours, where each colour represents a different Id.



(a) Colour image (b) u-disparity (c) detected id's in colour

Figure 4.27: *Detection error.* The person walks in touch with the neighbouring wall, which leads to a miss in the pre-detection due to the wrong geometrical dimensions. (a) Colour image of the person at the wall (b) U-disparity image of the same scene (c) The detected id's in colour. The person is coloured green. The wrong assignment leads to a too big elongation in depth, which strongly reduces the possibility of being a person.

4.7.3 Quantitative Analysis of the Proposed System Approach

The quantitative analysis provides detailed information about the measurable parts of the system. Here, the trajectories have to be compared to the ground truth with respect to the distance error and variance. Additionally, the false positives (FP), the true positives (TP) and the missed humans (false negatives, FN) are enumerated in order to calculate the detection rate in the presence of false positives. The detection rate is calculated as

$$\text{Detection rate} = \frac{TP}{TP + FN} \tag{4.30}$$

A detection or track is voted as TP, if a hypothesis is created and kept. Additionally, the distance of the track to the ground truth (distance in all dimensions x, y, z) has to be in an acceptable range. Here, the maximum distance is 1 meter to be robust against inaccuracies in the ground truth (the current state of the art system [61] also uses 1 meter). If a hypothesis is farther away, it is rated as FN and FP as well. All detections and tracks, which are not associated to a ground truth track are counted as FP. In the analyses I do not account for identity switches, which occur if a person is occluded longer than a predefined period. Here, I compare the ground truth to all tracking segments belonging to the particular person. This is applicable, as an identification of persons not observed for a specific amount of time is not implemented in the system.

The following Tab. (4.1) reveals the detection rate of the system for each data set. The detection rate is between 76% and 99.68% and at 90% in average, which is a very good result considering the difficult sequences and the presence of ego-motion. The false positive rate, which is at 0.0733 FP per frame in average, is in a definitely acceptable scale for all data sets. Like mentioned in Sec. (4.7.1), the spot lights at the ceiling produce a light pattern, which looks very similar to the human in the eye of the classification and which is responsible for most false detections. Some false detections also occur due to the near-detector, which is trained with only a few training examples (35 positive and 70 negative examples).

Dataset	**Frames**	**FP**	**TP**	**FN**	**Detection rate**
1	700	0,068	708	114	0.8613
2	1200	0,122	879	33	0.9638
3	360	0	215	6	0.9729
4	1257	0,103	949	74	0.9728
5	1700	0,161	2112	107	0.9518
6	2140	0,061	1260	4	0.9968
7	1400	0,026	999	56	0.9469
8	1280	0,016	741	222	0.7695
9	1350	0,112	557	102	0.8452
10	1320	0,055	624	21	0.9674
11	700	0,021	745	155	0.8278
12	625	0,134	506	155	0.7655
avg	**1170**	**0,073**	**858**	**87**	**0.9035**

Table 4.1: *Detection rate for each data set.* For each data set the number of frames, the false positives per frame (FP), true positives (TP) and false negatives (FN) are enumerated in order to calculate the detection rate for each specific data set. The detection rate is very good and the false positive rate is definitely acceptable.

Next to the detection rate, the distance of the tracked human to the labelled ground truth is of main interest. The following plot demonstrates the error for each person in each frame (all underlying data can be found in the appendix B).

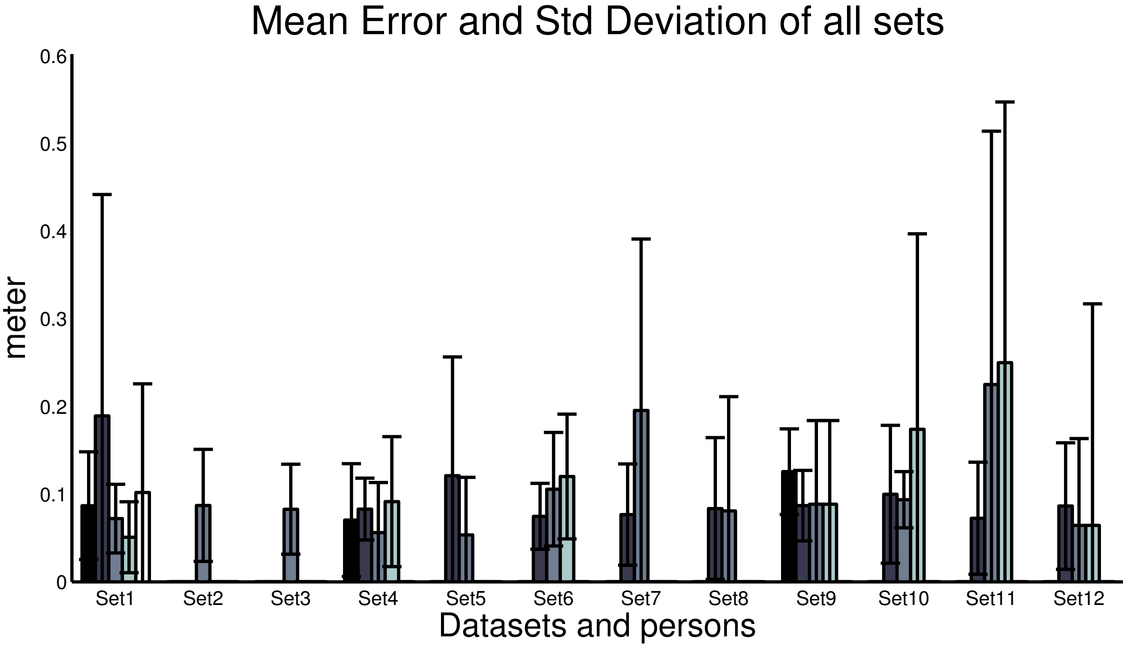


Figure 4.28: *Mean error and standard deviation for each person.* For each set and each person therein the mean error and standard deviation are shown. The error calculates through the distance from the measurement in each dimension to the ground truth.

The shown mean error and standard deviation for each single person in Fig. (4.28) are calculated through the squared distance of each single dimension error.

$$Mean = \sqrt{x_m^2 + y_m^2 + z_m^2} \tag{4.31}$$

$$Std = \sqrt{x_{std}^2 + y_{std}^2 + z_{std}^2} \tag{4.32}$$

In average the mean error is at 0.04 meter in x-dimension (x_m), 0.07 meter in y-dimension (y_m) and 0.07 meter in z-dimension (z_m). The standard deviation is also very low. It is at 0.04 meter in x (x_{std}), 0.05 meter in y (y_{std}) and 0.09 meter in z (z_{std}) (cf.Chapter (B)). The combined mean error is averaged at about 0.1 meter and the standard deviation at 0.11 meter. Only five of 33 persons have a slightly bigger mean error and standard deviation due to failures in their tracking. Generally, the failures origin from attractions of similar objects. But, the error also results from objects far away, because the sensor noise rises with increasing distance. The following plot (Fig. (4.29)) shows the difference of the tracked object position compared to the ground truth in relation to the distance of the person to the robot. In the beginning at frame 90 to frame 100 is a small error visible, where the tracking moved slightly into another object. From frame 500 to 700 the person is far away from the robot (> 5 meter), which results in a noisy measurement. The imprecise measurement itself produces a light noise in the data, visible as little jumps (see Z-error in Fig. (4.29)). This error curves match the trajectory presented in Sec. (4.7.2). There, the tracked object positions vary more from the ground truth for a farther distance. Although the error is higher for humans farther away the outcome is still precise enough to keep track of the persons.

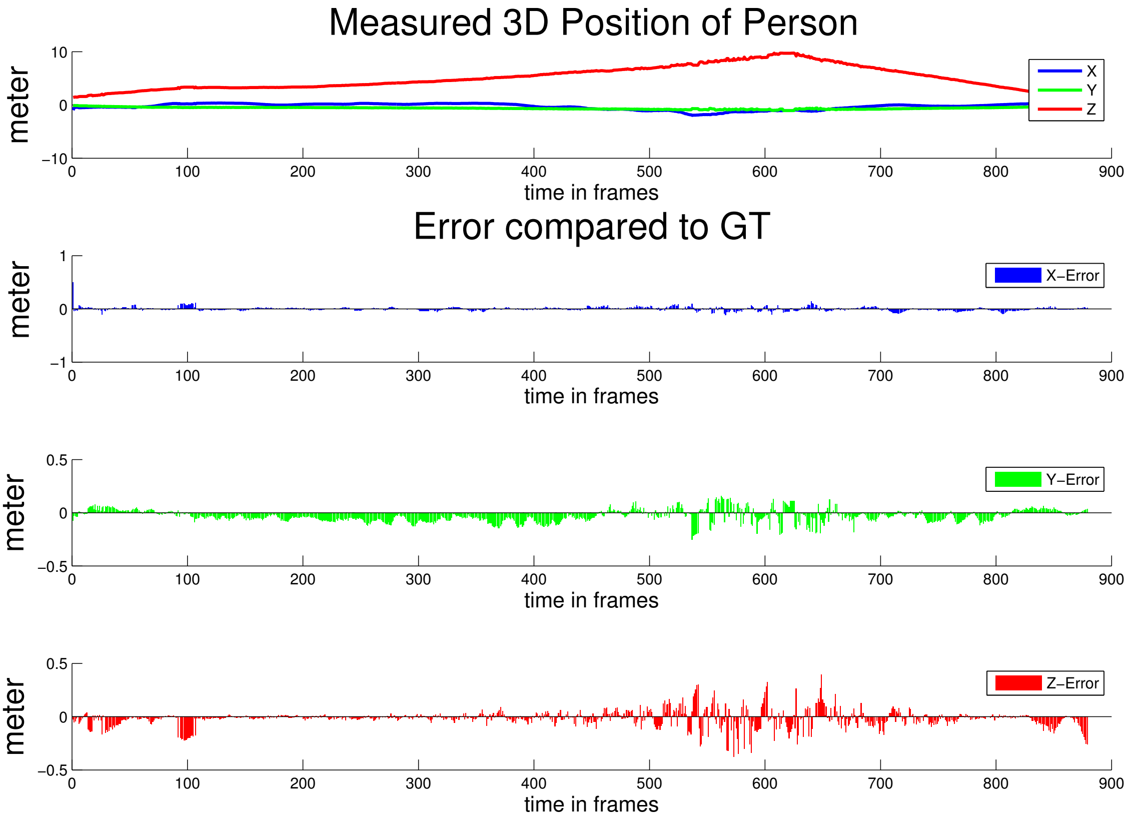


Figure 4.29: *Distance error per frame dependent on the distance, Set 2 Person 1.* The x-dimension represents the accordant frames, while the y-dimension relates to the relative distance to the robot. In the upper plot the distance of the person to the robot is shown. In the lower plots the colours represent the x (blue, lateral distance), y (green, height distance) and z (red, distance) distance errors. Between frame 500 and 700 the person is far away (>5 meter) which results in a bigger error noise for the depth component (z, red).

Unfortunately, the system proposal is not directly comparable to the current state of the art system from Ess, Leibe et al. [61], but it shows similar results. The quantitative results of their current system are presented in [61]. The tracking rate of persons is at 73 % (shown for one sequence with 999 frames length) with 1 false positive per image. The rate slightly increases, if the detection range is restricted to 15 meter. The authors report detection improvements through a ground plane assumption and by using better stereo data. Their computational efficiency is reported as not fully real-time capable. In contrast, my system proposal reaches 74 % to 99 % on a different data set with strong ego motion and operates in real-time. Like introduced, the results arise from different data under different scene conditions and hence, it only corroborates the belief that my system proposal achieves comparable results to the current state of the art system, but with faster computation time. Additionally, they assume that the exit and enter zones are always at the image borders, which is not needed for my system proposal. Here, people could enter the scene through doors in the middle of the scene, which requires a stronger hypotheses management. As a last enhancement of my system the possibility to detect humans at a very short as well as far distance to the robot should be mentioned. Summing up, the tracking is very precise for most of the persons in all sets. Some persons show partially a bigger error due to distractors in the background or failed tracks. But in general, the difference to the ground truth is very low and the tracking is very reliable. The false positive rate is mostly below one false detection every 6th frame. In order to analyse also each part individually, the detection and tracking are each inspected in the following.

4.7.4 Enhancement through Pre-Detection

The pre-detection step has been included in order to speed up the overall system performance. A time analysis of each module revealed that the detection and validation step consumes most of the system time (see Tab. (4.2)) This meets the declarations made in literature [81].

Module	Time (in ~ms)
Detection	195
Hypo. management	2
Tracking	23

Table 4.2: *System time without pre-detection.* The table presents the time (in ~ms) per module. The detection is without pre-detection and only based on a HoG-classifier.

The presented system uses two classifiers in order to detect humans far away and humans only partially seen in front of the robot. A detailed time analysis of the far detection algorithm is presented in Tab. (4.3) and of the near detection algorithm in Tab. (4.4) (all times are an average over a complete sequence).

Window shift	Window scaling	time (in ~ms)	detection rate	false positives
4	1,05	1089	0.83 %	228
8	1,05	369	0.625 %	144
8	1,1	195	0.65 %	54
8	1,2	122	0.52 %	18

Table 4.3: *Far-HoG detection results.* The detection results are dependent on different parameters. To get a reliable detection the window shift and scaling have to search most of the image. If the parameters are too broad, the detection misses many hypotheses.

All runs are performed on the same machine (Pentium IV, dual core 2.8 Ghz). The window shift corresponds to the difference in pixels, which is added to each new detection window. After a complete run of the detection window over the image, the window is scaled by the window scaling parameter. Then the detection is started again over the whole image with the new scaled window. This process iterates until the window scales greater than the original image. In Tab. (4.3) and Tab. (4.4) it is shown that the detection can be done fast to some degree, but at the cost of fewer detections.

Window shift	Window scaling	time (in ~ms)	detection rate	false positives
4	1,10	709	0.76 %	8235
8	1,05	410	0.74 %	6562
8	1,10	207	0.74 %	5760

Table 4.4: *Near-HoG detection results.* The detection results are dependent on different parameters. To get a reliable detection the window shift and scaling have to search most of the image. The detector is not well trained, which leads to a very high false positive detection rate and a good detection rate.

If the parameters are chosen best for the detection, the overall time is infeasible for the detection on a mobile robot. It is getting even worse, if more than one classifier is used like proposed in this thesis.

To speed up the detection process a pre-detection step is proposed. Thereby, the pre-detection has to be fast and reliable in order to detect all present humans. The reliability is shown in the presented results above (cf.Sec. (4.7.3)) and in Tab. (4.5). If the person resides in the range of the sensor and does not merge with other objects, all persons in each frame are correctly detected. The false positive rate originates from the near detector, which sometimes produces a false validation. But, the focus of this analysis relies on the speed-up of the detection process.

Pre-detection (in ~ms)	HoG on window (in ~ms)	Overall time (in ~ms)	detection rate	false positives
22	25	47	0.92 %	66

Table 4.5: *Pre-detection + HoG verification.* The overall detection time results from the pre-detection and the HoG based verification on the selected windows. The complete detection time is much faster than the original sliding window based detection (cf.Tab. (4.3)). Additionally, the detection rate is much higher due to the information from the pre-detection.

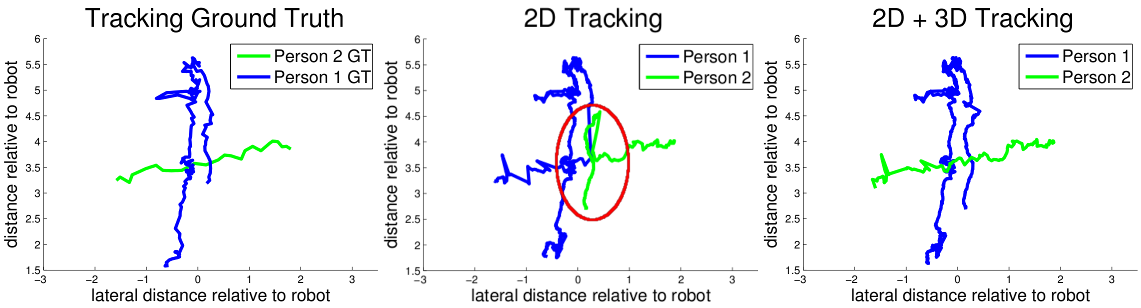
To compare the speed of the usual sliding window detector with the pre-detection based version, the time of the complete detection process has to be considered. The time is composed of the time of the pre-detection and the subsequent window verification. Both times are given in Tab. (4.5) (all times are an average over a complete sequence).

The results show that the pre-detection speeds-up the complete detection process without a loss of accuracy. Rather, if the window is verified the pre-detection enhances the subsequent detection rate (If a window is verified, all subsequent pre-detections without verification are counted as detection for the same object) by reducing the false-positive rate and increasing the true-positive rate. The detection rate rises up to 92% for the analysed sequence.

4.7.5 Analysis of the Proposed Tracking Algorithm

In this section the design of the tracker is questioned. The proposal of a combined 2D and 3D approach is thought to be more stable and robust against occlusions or simple

3D point associations. In order to show the enhancement of additional 3D information, the following sequence is taken as an example (Fig. (4.30)). The figure shows three images, with each two trajectories. The left image corresponds to the ground truth (GT) of each person's movement. The image in the middle shows pure 2D tracking and the right image my combination of 2D and 3D tracking. The blue line in each image corresponds to the first person. The green line describes the second person coming from the right. In the red circle the occlusion takes place. It is clearly seen that the pure 2D tracker attaches to the wrong person and all subsequent tracking goes wrong. In the right image the tracking correctly follows the right persons. Even during the occlusion the additional 3D data avoids a wrong association and leads to a successful tracking over the whole sequence.



(a) GT trajectories (b) 2D tracking (c) 2D & 3D tracking

Figure 4.30: *Enhancement through 3D information.* The left image shows the ground truth trajectories of two persons (blue & green). In the sequence an occlusion takes place and the pure 2D tracking fails which is shown in the middle. In the right, the additional 3D information correctly associates each track to the right person.

This example shows the importance of additional 3D data, which reduces errors due to wrong assignments by comparing also the 3D position of each hypothesis and the known tracks. The subsequent section gives a further comparison of the proposed tracking algorithm compared to other tracking mechanisms, which also shows the good quality of the proposed tracking algorithm.

4.7.6 Comparison with State-Of-The-Art Tracking Algorithms

In the following the tracking part is compared to actual tracking algorithms published in literature in order to show the good design and capabilities of the tracker. Unfortunately, it is hard to find public available datasets including 2D image data as well as 3D data with additional tracking results from an actual tracking system. Hence, the tracker is compared without the 3D component to another 2D tracking system. The datasets and the tracking results from the reference system are public available at the homepage of the university of Bonn [3] (see Fig. (4.31)).

The authors present an adaptive real-time particle filter with an ensemble classifier based observation model [120]. Generally, they use a condensation particle filter with a

[3] http://www.iai.uni-bonn.de/ kleind/tracking/

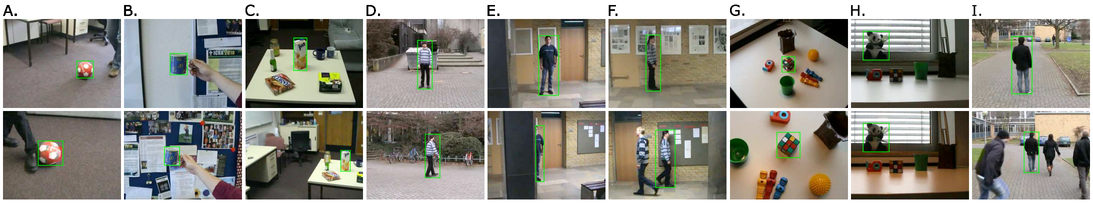


Figure 4.31: *Bonn data set.* The data set includes different scene conditions. They inherit a moving camera, object scaling, clutter, partial and full occlusions, viewpoint changes and rapid changing lightning conditions. (Images found in [120])

first order autoregressive motion model. The state x is described by a vector

$$x = (x, y, w, h, v_x, v_y, C)^T \tag{4.33}$$

with x, y position of a rectangle with width and height w, h, v_x, v_y the velocity of the rectangle and C as the particle's object classifier. The object classifier decides between background and foreground. The particles are weighted accordant to a continuous exponential function of a target rectangle (x, y, w, h). The observation model is based on a rectangular model, which is divided into four sub-rectangles. In each sub-rectangle Gentle AdaBoost is used to pick out the best features based on a weighted set of training examples. As features the authors propose simple Haar-like centre-surround features varying in size, relative position and RGB colour channels (see Fig. (4.32)).

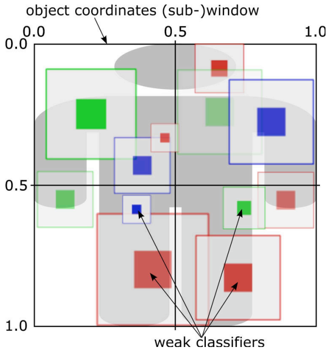


Figure 4.32: *Observation model build of an ensemble of weak-classifiers.* The object rectangle is divided into four sub regions, in which the best features are used for tracking. (Image found in [120])

The observation model is updated by each new frame (if the confidence is above a threshold), where the detected state and the background are additionally taken as positive and negative learning examples. Finally, the classifier is retrained using the updated set of examples.

The outcome of their algorithm is compared to hand-labelled ground truth data by calculating the overlap between both rectangles. If the overlap is above 33.33% the rectangle is marked as hit. The following Tab. (4.6) compares the outcome of their algorithm with the best parameter set to the proposed tracking algorithm in this chapter. Additionally, the tracker are compared to another histogram based tracker and a multi-component tracker (cf.[120]). It is important to emphasize that my proposed tracker is trimmed to fit the pure 2D data.

Seq	♯ Frames	Histogram	Mult.-Comp.	best H.-cs (in %)	My approach (in %)
A	601	70.73	63.24	65.06	81.56
B	628	67.02	50.73	79.01	94.92
C	403	47.58	63.71	91.33	23.70
D	946	63.35	76.39	75.21	99.79
E	304	78.21	77.42	86.32	99.67
F	452	44.43	40.02	68.32	81.72
G	715	46.27	49.62	77.30	91.77
H	411	62.19	86.50	95.79	99.76
I	1016	68.94	47.63	75.02	81.69
avg.		60.97	61.70	79.26	83.84

Table 4.6: *Comparison of the proposed tracking algorithm with a tracker presented in [120].* The sequences are taken from the internet in order to compare the tracking results (green > red). In one case (seq. C) the camera zooms rapidly out, which can not be handled by the proposed tracker alone. As the proposed tracker is scaled through the complete system approach, the tracking does not perform very well in this case. In all other sequences my proposed tracking algorithm performs better than the best tracker in [120].

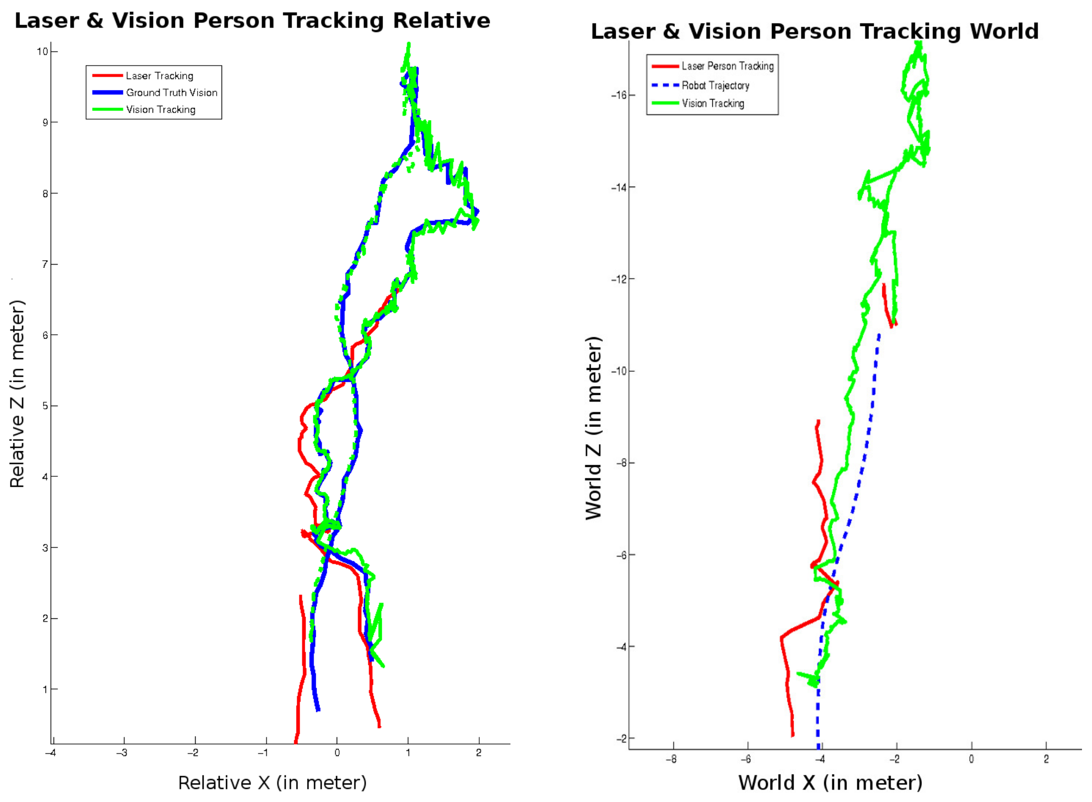
Comparing the results, my proposed algorithm performs better in 8 of 9 sequences. In the case of scaling (seq. C) my tracker does not perform very well. This is evident, because the tracker itself does not contain any scaling parameters. Instead, the scaling is achieved through the complete system approach. In all other sequences (moving camera, clutter, partial and full occlusions, viewpoint changes and rapid changing lightning conditions) the proposed particle filter works superior. Using 3D data would again improve the outcome of the tracker (cf.Sec. (4.7.5)).

4.7.7 Comparison with a laser based Person Tracking

The robot Biron is also equipped with a laser range finder. A previous attempt to detect and track humans in the scene utilised its provided scan line in order to find legs. Assigning ids to the detections it is possible to track leg pairs as human entities. This method is widely used in literature and provides meaningful data. But, the confident detection of leg pairs is only possible for humans not farther away than 3-5 meter. Additionally, the detection is error-prone due to table legs or other similar objects. The following plots (Fig. (4.33)) should demonstrate the enhancement of using 3D vision to detect and track humans.

In the left image the relative trajectories to the robot are plotted and in the right image the same trajectories are shown in world coordinates. Both sensors are not calibrated to fit the same coordination system, but the sensors are physically aligned to some degree. Here, this suffices as the analysis should only reveal the strengths of both systems. If both sensors are used in a combined system the sensors have to be calibrated.

In both images it is clearly visible that the laser based person tracking is not able to track



(a) relative trajectories

(b) world trajectories

Figure 4.33: *Comparison with laser based person tracking.* Both images show the trajectory of the person in data set 2. In the left image the trajectories are plotted relative to the robot. In blue the ground truth trajectory of the person gained from the visual data is presented. In green the result of the proposed tracking in this thesis is denoted. In red the revealed trajectories of the laser tracking are shown. In the right image the trajectories are plotted in world coordinates. The robot trajectory is plotted in dotted blue. In both images it is clearly visible that the laser based person tracking is not able to track the person in a further distance.

the human up to the same distance like the vision based tracking. This fact origins from the laser resolution, which is not appropriate for detecting legs in further distances. The detection and tracking through the laser is even worse, if the object moves towards the robot. This is best visible in the world coordinates, where the person is only shortly tracked (short red line at -11 to -12 meter in Z-distance). The laser based system has its strength in the viewing angle. The laser has an opening range of 180 degrees which allows to detect and track already next to the robot. The vision based system shows its power in further distances and better accuracy. The better accuracy is achieved through the appearance verification using image features, which provide a more details for the human detection.

4.7.8 Parameter Analysis

For a technical system it is very important to provide generality and easy usage in order to get a solution for different problems and for different scenarios. Often a solution is presented, which is either very special for one problem or very difficult to handle due to its amount of parameters. Here, a complete tracking system is presented, which has to deal with complex requirements like ego motion and difficult scenes. Therefore, it is important to make a declaration of all important parameters to show the manageability of the system. Additionally, I state the steps needed to adapt the system to another kind of tracking problem, which indicates the generality of the system.

The parameters of the system are depicted in dependence on the module, which implements it (see Tab. (4.7)). The total number of parameters in each module is declared in the first column. In the second and third column the numbers are a subset of the total amount. The second column is the most important one, as it offers the number of critical parameters in each module. Critical means that the value has to be chosen correctly or adapted to each new application area. The last column depicts, if the parameters are calculated by the system. Here, all 12 data sets are evaluated with the same set of parameters.

Module	parameters	critical	automatic
Floor removal	2	2	2
Ceiling removal	2	2	2
Pre-Detection	7	4	-
Classification	1	1	1
Hypotheses Management	4	1	-
Tracking	8	5	-

Table 4.7: *System parameters.*

The floor and ceiling removal need each 2 parameters (height, gradient), which are both important to correctly remove the floor and ceiling. Both parameters are estimated automatically from the v-disparity image (cf.Sec. (4.4)). The pre-detection implements 7 parameters, which are the typical width, height and depth of a searched object and the number of pixels to search for the distance adaptive object connector. One parameter describes the minimum probability for an object to be a human. Thereby, the width, height, depth and the minimum probability are critical as they have to represent the object dimensions and probability correctly. The classification has usually at least two critical parameters (Window-shift, window-scaling), which arrange the window search for the whole image. Here, the pre-detection delivers a window, which removes the necessity of both parameters. Hence, the classification has only one parameter, which controls the change of the classifier (Near and far distance classifier).The accordant parameter is chosen automatically dependent on the height of the robot and the data of the floor (cf.Sec. (4.4.2)). The hypothesis management has got 4 parameters, which care for the assembly of all incoming information. The first three parameters are the weighting factors for each external module (pre-detection, classification, tracking), which controls the importance of each new information. A hypothesis from e.g. the classification has more weight than a hypothesis from the pre-detection. The only critical parameter relates to the maximum distance, where a hypothesis can be merged with existing information. Above this distance, the togetherness is implausible. The tracking is defined by eight parameters, which are the number of particles, the mean variation in both hypothesis directions and the velocity and its direction, the kernel width for the distance weighting, the number of bins for the colour histogram and the update rate for the histogram over time. Here, it is difficult to decide, which parameters are critical. The number of particles has to be in a specific area in order to work correctly. The same is valid for the kernel width, but both parameters are not really critical because of their variability. The mean variations in both hypothesis directions and for the velocity are denoted critical, because a too short variation could cause a loss of the hypothesis through rapid movement, while a too broad variation looses preciseness and attracts other similar objects The number of bins for the colour histogram are not definitively critical, because the colour is merged with the distance information and it should only preserve the filter from migrating to a false hypothesis. The update rate itself is denoted critical, because it can ensure a more stable tracking, if the particle filter slightly adapts the current hypothesis.

Summing up, the system has 24 parameters, whereof 15 parameters can be seen as important or critical. 5 of these parameters are estimated by the system. In general, all parameters have to be chosen once and afterwards, they are correct for the same kind of problem. This is shown in the results, as all experiments are conducted with the same set of parameters. Utilising the system in another kind of problem is possible. If the data origins from another type of sensor, the sensor module has to be exchanged. Additionally, it could be needed to adapt the object size in the pre-detection to the desired object size. If other objects than humans are of interest the validation through HoG has to be retrained. If very fast or very slow objects are in the focus, the particle distribution of the tracking module has to be adapted.

4.8 Conclusion

In this chapter a complete system for detecting and tracking humans on a mobile robot platform is presented. The tracking of all accordant humans in the scene achieves successfully the second category of situation awareness. Utilising this information the robot has got a knowledge about dynamic objects in the scene. This information is important for the further planing of the robot.

The presented system approach deals with the requirements of a moving robot by running in real-time and managing the ego-motion of the robot through a particle filter. In order to achieve real-time, a pre-detection step is introduced. The pre-detection is based on the *u-v-disparity*, which allows to search easily for geometrical objects and to reduce the search space by subtracting the floor and ceiling. The pre-detection is again fasten through an undirected graph, which allows the calculation of connected components in short time. The reduced search set is forwarded to a verification step, which verifies the detection as a precaution. Here, I introduce a distance adaptive verification, which determines the classifier to choose. The decision is based on the set-up of the robot and the distance from the camera to the observer. Because the robot has to interact with the human in the near space and the far space as well, two classifiers are used. The upper body classifier is chosen, if the human is near and only partially visible. The complete human body verification is used, if the human is fully visible from a specific distance. The verification is based on the well performing *Histograms of oriented Gradients* approach. The chosen trained support vector machine decides, whether a human is present in the sub-window or if it belongs to the background. All verified detections are forwarded to the self-developed hypotheses management, which cares for the creation and deletion of hypotheses. Additionally, the hypotheses management associates new information with known hypotheses in order to incorporate each new measurement in an efficient way. Each hypothesis is additionally tracked by an adaptive particle filter with multidimensional observation model. The simultaneously used two and three dimensional data provide a very efficient tracking system, which is able to handle occlusions and fast movements. The results show that the system is working very well in the area of human detection and tracking. Each module on its own delivers superior results. Additionally, all modules are enhanced to provide their information for the complete system in order to consolidate all available information. Some small errors sill remain, if objects are not revealed through the pre-detection. Here, further improvements are going to be implemented like a wall detection or a combination with a mobile version of the articulated scene model.

Summing up, the presented chapter shows how to achieve the second category of situation awareness on a mobile robot by realising a complete human detection and tracking system. Fast detection and tracking modules work in conjunction with a hypotheses management directly on a mobile robot in order to provide all necessary steps for a situation awareness.

5 Attention Focus for Situation Awareness

The third category of situation awareness answers the qualification to direct the attention focus on a specific object (see Fig. (5.1)). By directing the focus on an area or a point the underlying data can be examined more sophisticated. This is an important feature for a mobile robot, because the robot has to gather more information about specific objects of interest and to regard other unimportant information. Additionally, the robot needs to focus on special interaction partners in order to keep the conversation up, even in the presence of other humans. Directing the attention focus also supports recovering a human or an object which have been shortly out of view. All these abilities support a mobile robot in the interaction with humans.

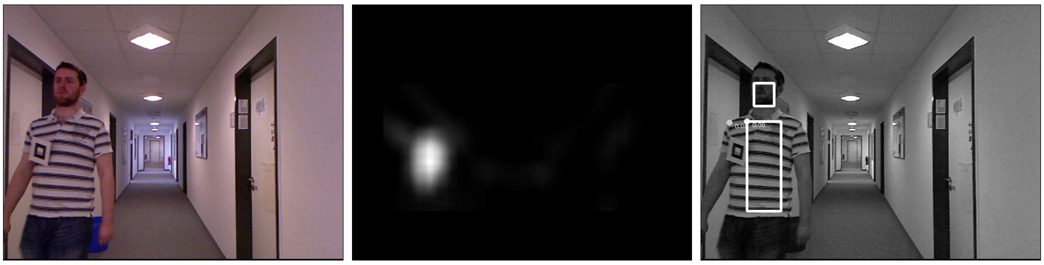


Figure 5.1: *Attention on a human in the scene.* (Left) Original image (middle) Attention map (right) Focus of attention using a human model

Directing the attention focus has a biologically origin, because each creature needs the ability to focus on a desired object or location. This ability is the result of a long evolutionary development of effectiveness in realisation, time and quality. One prominent example for this ability is the human visual system. It is optimised to direct the attention on a specific area in order to extract more detailed information out of this region. The basic principle behind this causes from the information processing bottleneck in the human brain [5]. The visual input provides too much information to handle. The human employs a *bottom-up* attention, which isolates different salient spots in the visual field due to their accentuation. Directing the attention focus on one spot concentrates the processing on this area. The directing of the attention is called *top-down* process or *guided search*.

It appears reasonable to use the biologically inspired ideas of the human visual system as a way to gather visual information for situation awareness on a mobile robot. In fact, during the recent years much work has been invested to copy the biological concepts onto computer systems. But, directing the attention focus is still a challenging task for mobile robots, because the human visual system cannot be copied one-to-one.

To realise an attention system on a mobile robot, the following research questions arise:

- *How can the attention be directed on specific areas using a technical system?*

- *What features are general and meaningful for diverse classes of objects?*
- *How could a specific object be taught in order to find it again in subsequent views?*
- *How can the attention on a human body be modelled?*

In order to realise an attention system, the human capabilities should be transferred to the technical system as far as possible. Like described above, the human has got a general bottom-up awareness about his surrounding. Interesting areas pop out due to their discriminative features. Otherwise, the top-down search looks for an object, which differs in one or several specific features. The *guided search theory*, presented by Wolfe [214], combines the bottom-up feature calculation with the top-down specific search through an activation map. The activation map weights the bottom-up features related to their importance in the top-down search. Hence, the bottom-up attention forms the basis for the attention system, which is extended by the top-down approach to weight the important features describing the searched object. Therefore, both parts have to be implemented in a technical system in order to combine and weight features to a coherent object.

The biologically bottom-up attention relies on simple features like colour, orientation, spatial frequency, or movement, because these features respond to different visual receptors. They are physiological suggested, because simple features are processed in the early stages of vision and are each encoded in a different area of the brain [222]. The simple features are ideal to describe any type of object. Because an object is mostly described by several features, the *Feature Integration Theory* (FIT) is consulted. It was introduced by Treisman and Gelade in 1980 [201]. The theory states that all simple features are processed in parallel. To combine the information for one object, a later process is used to integrate the information from the different brain areas. Therefore, the presented system uses simple features like colour, orientation and spatial frequency. Motion is not used as the system is meant to run on a mobile platform, where single object motions are difficult to calculate. The used features are integrated in a subsequent step in order to form conspicuity maps, which represent the contribution of each feature to a spatial area. Afterwards, all conspicuity maps are merged in a subsequent step to form a saliency map. The saliency map denotes the most likely place or places for the searched object. In this way, the biologically processing stages are adopted. Furthermore, the presented system uses diverse biologically motivated feature weighting strategies. The weighting strengthens the response of features, which separate most informative the fore- from the background.

Usually, an object dependent detection or awareness system has to learn the specific features of an object out of many examples. Boosting or histogram based approaches e.g. learn the appearance out of labelled data [71] (cf.Sec. (2.2.2)). In a human-robot interaction the human is often not previously known and there are only a few frames where the person is present. Obviously, the learning progress for each person should require online capability. Here, the target model for a human or an object is learned in one frame.

The target model is dynamically adapted to reliably turn the focus on the specific object. As mentioned above, for the robot the most interesting object is the human. In order to get an optimal focus on a specific interaction partner, the target model should be most discriminative. Humans have some special characteristics. On the one hand humans have a similar appearance through the body. Each human usually stands upright and has got a body and a head on top. On the other hand, the clothing is often very characteristic. Hence, the clothing could turn the essential balance to differentiate between several humans. These two characteristics are incorporated into this thesis. The target model is designed by two rectangular regions, which represent the head and the body. Both regions are joined in order to discriminate between the current interaction partner and other humans. Additionally, the features should distinguish between the object and the background. For this reason the current background information is also considered.

Besides the realisation of the third category of situation awareness, the contribution of this part lies in the combination of top-down situation awareness with a human body model. The model approach strengthens the attention focus on the desired object.

For the evaluation of the system I focus on two scenarios: First a human guides a mobile robot around and second a scenario where a robot has to learn about objects. The main idea is to show that the system is able to handle arbitrary objects and that the learned features accurately differ between the fore- and background, so that the main focus of the robot remains nearly always on the searched object. Parts of the text and the results in the following chapter have been previously published in [24] [134].

5.1 Attention Systems in Human-Robot-Interaction

For a mobile robot the human is the main object of interest. Humans are dynamic objects, which have to be detected in order to avoid collisions and to initiate conversations with them. The usual way of detecting humans is to detect the legs by laser data [68] (Sec. (2.1)), face detection [143] or window based classifiers (Sec. (2.2.2)). The legs are only visible in front or back view and the face is only visible, if the human is facing the robot. This complicates the detection using laser data or face detection a lot. As shown in both previous chapters, the detection of the full body shows promising results. But, the robot also has to differ between the persons, even if they get out of sight for a longer moment. It happens quite a large number of times that the interaction partner shortly gets out of the view of the robot, but the interaction is still not ended. Here, the detection and tracking systems reach their border. Thus, it is proposed to remember the interaction partner using a top-down attention system, which identifies the person even after longer absence. Additionally, the attention system could be used to focus on many other objects, which is shown in the results.

The most popular approach to design a computational attention system was given by Laurent Itti [106]. He used the feature integration theory of Treisman and Gelade [201] in combination with several simple features in a bottom-up attention framework. The system showed promising results, which yield to many constitutive approaches.

The first attention system on a robot was presented by Breazeal and Scassellati. They showed an attention system for the robot Kismet, which is used in a human-robot interaction scenario [38]. The system is based on the proposal of Itti. They use face, colour and motion to detect the human in front of the robot and to maintain a social interaction with this person. The robot directs his focus by turning his head in the direction of the human.

Many stationary bottom-up attention systems were developed over the years, where mostly objects are of special interest [193]. Y. Nagai presented an interesting system that uses attention to learn an action taught by a human [154]. The idea is to teach robots like in a parent-child interaction. The used basic features vary in the different attention systems in literature. In some cases motion is the most important feature [220]. In other areas even multi-modal cues like sound and vision are combined to reproduce the human attention framework [168] [167].

Contrary to bottom-up saliency top-down attention deals with information about the object and/or the background to influence a guided search for the desired object. There are several approaches, which propose an attention control to direct the focus on special objects [157] [73]. In [157] they use a model of attention guidance based on global scene configuration. This means that the knowledge about the context of the scene has to be incorporated. Studies in visual cognition showed that humans search e.g. on a table for objects or at the ground plane for other humans. The authors used that knowledge by learning the typical appearance areas of specific objects.
An extension of the system VOCUS (Visual Object detection with a CompUtational attention System), developed from Simone Frintrop [73], incorporates the background into the calculation of the foreground features. Thus, the features are weighted according to their appearance in the fore- and background. Frintrop uses the top-down attention in combination with a simple motion model, which assumes the object in a close neighbourhood. In this way the system operates like a tracker for a newly presented object. The suggestion of [166] is similar to the one from Frintrop but with another weighting strategy. Here, the weights are learned with a Neural network in order to learn typical feature weightings in advance. Similar to these approaches [155] uses bottom-up and top-down attention in an integrated system with background incorporation and calculates optimum feature weights through a signal-to-noise ratio maximisation. All these systems are based mainly on the bottom-up feature calculation from Itti [106][107], which is based on the theory of Koch and Ullman [121], and a top-down weighting inspired by the guided search theory of Wolfe [214]. The learning of an attention vector and the calculation of an object-directed attention map show promising results. But, fast learning and real-time possibility is crucial for a human-robot interaction and has to be kept in mind.

The use of 3D data is not common in the area of visual attention. To my knowledge the only work in literature is from Frintrop and Nuechter et al. [75]. They propose to use 3D data from a Laserscanner and intensity data from reflectance as pre-detection for an AdaBoost Haar-like object detector. They show some results that the attention can be directed to specific objects without being attracted from shadows or pictures from the same object. In this thesis depth data is not used as a direct feature, because the depth itself does not describe an object. Edges in the depth image could describe the object, like in [75], but the data of long corridors and halls does not provide stable depth data for this case (see Fig. (4.21(c))). Holes in the depth data produce unpredictable depth edges, which would entangle the feature calculation process. Here, depth data is used to segment the foreground from the background. The object rectangle is often not accurate, which is further refined by a depth segmentation process. Through the foreground segmentation the features can be differentiated more accurate from the background. The details of the presented bottom-up and top-down attention system are described in the following sections.

5.2 Directing the Attention Focus

In order to implement the third category of situation awareness a combination of bottom-up features and a task-directed top-down process has to be realised. The bottom-up features form the basis for the attention. Each feature is represented by a map, which denotes the local importance or occurrence of the feature. Each map has got the same size like the image and each value of a pixel corresponds to the answer of the feature in this area. All these feature maps could be merged in order to get a *saliency map*, which represents the undirected attention. Here, the attention is directed on a specific object using a top-down process. Hence, the feature maps are weighted according to their importance for the searched object. In order to get the discriminative features, the rectangle region around the object is taken as foreground, while the rest of the image is taken as background. Comparing the fore- and background, a value for each feature map is identified, which denotes the importance of this feature to distinguish between the object and the background.

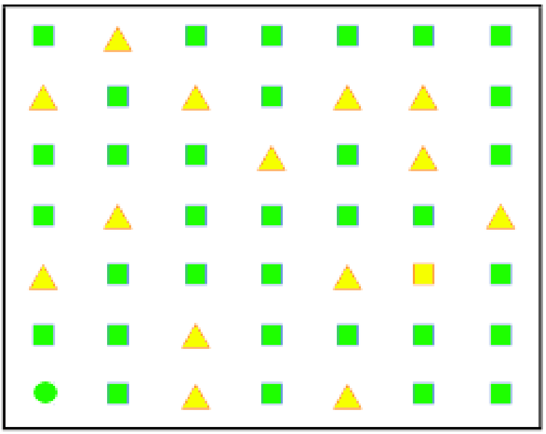


Figure 5.2: *Top-down directed search by combining features.* The yellow square is searched. To find the square the features appearance and colour have to be combined. If the green circle is in the focus of interest, only the shape is discriminative. A technical system has to reveal the discriminative features in order to search successfully the object.

The values are combined to form an attention-vector, which is used to weight the bottom-up maps in order to look for the object. The following example describes the process. In Fig. (5.2) there are rectangles and squares shown in green and yellow. In order to find the yellow square, the features appearance and colour have to be both highly weighted. Using only one feature would not lead to a result. If we are looking for the green circle, only the appearance is of importance. The colour does not discriminate between the circle and the other objects. Hence, an attention system has to find the discriminative features, which differentiate the searched object from the current background. The following attention system incorporates this fact in a combined bottom-up and top-down attention feature weighting process.

The bottom-up design is inspired by the work of Itti [106] and the top-down weighting by Frintrop [73]. Their approaches are slightly adapted in the normalisation of the feature maps and some algorithm specific calculations. In the presented work the insights of Sec. (2.2.2) are additionally used, that a human body can be described by several parts [186]. A simple model is designed, which is represented by a torso and a head region. For each region an attention-vector is calculated and they are combined to a much more definite person-specific attention map. The learning of the attention-vector is done in one frame. The target regions of the model have to be provided (through e.g. the articulated scene model or the mobile person tracking) and the system calculates the target-vectors for one frame. For the evaluation in thesis the target region is manually provided. In the subsequent frames the focus is directed on the object by using the target model. Additionally, the target vectors are updated by new frames in order to keep a reliable target model. The overall system is running in real-time, which permits the application on a mobile robot.

5.2.1 Bottom-Up Saliency

Following the guided-search theory by Wolfe [214], the bottom-up feature maps λ have to be calculated first. The bottom-up calculation is accomplished in each frame without any knowledge about the scene or the included objects.

The bottom-up basic feature maps M_i are a subset of λ, which consist of the following three dimensions. *Colour* represents the first dimension (red, yellow, green and blue colour). The second dimension is *orientation* in $0, 45, 90$ and 135 degrees and the last corresponds to the *intensity* (see figure 5.3). Thereby, the intensity is divided into the *on* and *off* contrasts, whereas *off* means the dark parts of the scene while the *on* attribute denotes the bright parts. The on and off intensity values are biologically motivated by dark and bright sensitive receptive fields in the visual system of mammals. The colour and orientation features origin from the visual system as well. The colour features are weighted on the frequency of co-occurrence of light spectra. This motivates a centre-surround computation, which will be described in the ongoing section. The orientation maps respond to orientation sensitive receptive fields in the visual system. As humans are complex classes, it is proposed to use additional complex features to build a more discriminative framework. Local binary patterns have been found to be a powerful feature for texture classification [139]. Later, it is shown that the incorporation of LBP-features increases the discrimination between specific types of objects.

The bottom-up part consequently divides the incoming picture into the 10 basic feature maps M_i (see figure 5.3). Inspired by the human visual system, the local features are first weighted using a *lateral inhibition*. The lateral inhibition fades out features with strong neighbours and emphasises the examined feature with weak neighbours (Neighbours are the most similar features. For colour the neighbours are the left and right colour in

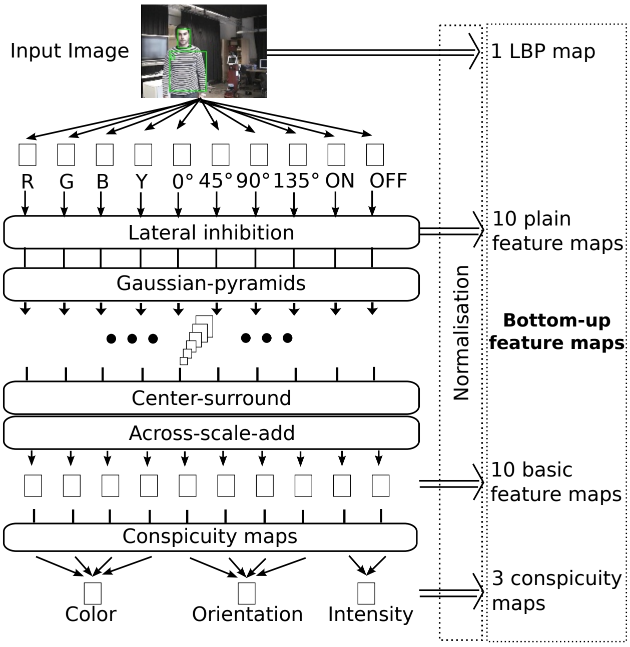


Figure 5.3: *Bottom-up feature computation.* The image is segmented into 10 feature maps, which are weighted using several biologically inspired computation steps. Additionally, a texture descriptor LBP is calculated. Finally, the bottom-up process results in 24 bottom-up feature maps

the colour circle (e.g. green has got the neighbours yellow and blue), for orientation the immediate neighbours are the ones with the closest degree value (0 degree's neighbours are 45 and 135 degrees) and intensity only has got one neighbour (on-off)). The lateral inhibition $\tilde{C(i)}$ determines the value for the pixel $C(i)$ and compares it with the value of its immediate neighbour maps $C_j(i)$ in the same dimension (Eqn. (5.1)).

$$\tilde{C}(i) = C(i) + n * C(i) - \sum_{j=1}^{n} C_j(i) \text{ with: } n = \sharp \text{ neighbours} \tag{5.1}$$

The following short example clarifies Eqn. (5.1). If we take C as green feature map with pixel i ($C(i) = 123$) the neighbour maps C_j are yellow ($C_1(i) = 15$) and blue ($C_2(i) = 81$). We calculate the lateral inhibition like following

$$\tilde{C}(i) = 123 + 2 * 123 - 96 = 273 \tag{5.2}$$

The lateral inhibition consequently emphasises the underlying pixel, because the neighbour maps are not as strong as the examined map.

To compute the local importance of the single features in different scales with less computational effort, each map M_i is transformed into 6 images using Gaussian-pyramids producing 60 feature maps M_{ij}. On each map M_{ij} a *centre surround mask* [219] is incorporated to highlight the salient point features [98]. The weighting causes a discriminative ratio for every pixel using the pop-out effect. The centre-surround weighting is described in Fig. (5.4). If the corresponding feature excites the centre of a cell, the response is high. If the surrounding is activated, the response is inhibited. The green-red contrast and the yellow-blue contrast are found in the visual system. Hence, the opposite colour bands are used for the centre-surround computation in this work.

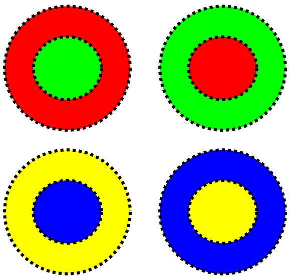


Figure 5.4: *Centre-Surround computation.* The visual receptive fields fire, if the centre is activated, but not the surrounding. If only the surrounding is activated, the field shows no response. If both areas are addressed, the field gives a mild response. For each feature (intensity, colour) exists an own cell. (Image adapted from http://camelot.mssm.edu/ ygyu/)

The weighted 60 centre-surround feature maps (6 pyramid levels for 10 feature maps) are again turned into one single map by applying the *across-scale-add* function. All submaps are scaled and added pixel-wise. Finally, each dimension (colour, orientation, intensity) is combined to a conspicuity map, describing the complete feature response.

It turned out that usual saliency approaches weight small points or edges higher than complete regions due to the centre-surround computation. This is reasonable, because a flat region does not attract the attention very well. In a human-robot-interaction the human should reside in the focus of the robot very often. There, the interaction partner is usually close to the robot and fills a large part of the picture. Therefore, plain features maps without the centre-surround computation are also considered. If the person e.g. wears a red shirt, the whole shirt could be salient contrary to the centre-surround, where only the border areas are interesting. This is a new way of calculating the saliency map specific for the human-robot-interaction. The 10 plain feature maps are equally added to the set of feature maps.

Additionally to the simple features, a Local Binary Pattern (LBP) is calculated [139]. The local descriptor is invariant to differences in the intensity and adds important local information by building up a histogram of the comparison of the pixels with their neighbours. Especially in cluttered scenes it is advantageous to add the LBP feature to the region, because the basic features like colour are shared by fore- and background simultaneously and are concludingly not always useful for the discrimination. Hence, the uniform Local Binary Pattern is used to detect interesting structural components in the image.

Finally, the process calculates 10 basic feature maps using centre-surround, 10 plain feature maps and three conspicuity maps. Additionally a LBP map is computed during the bottom-up process, resulting in 24 feature maps λ in total. Every feature map is normalised using the number of its maxima $\sharp m\vec{a}x$ to make an importance scaling for each map.

$$\lambda_i = \frac{M_{ij}}{\sqrt{\sharp m\vec{a}x}} \tag{5.3}$$

One maximum is very important, while many maxima do not discriminate and are consequently not of interest for the attention progress. If e.g. there is a lot of blue colour in the scene, the basic feature blue has got many maxima and is accordingly not important for the discrimination.

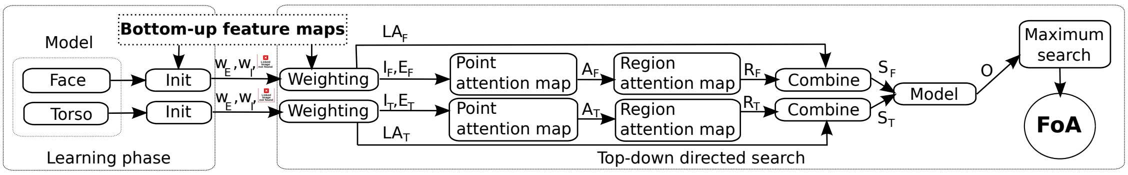


Figure 5.5: *The top-down system.* The top-down system consists of two parts. 1) The initialisation calculates an attention vector for each part of the model by analysing the bottom-up features. (The calculation of the bottom-up feature maps is equal to the system presented in fig. 5.3) 2) Through a top-down directed weighting of the bottom-up features an object attention map is determined. By applying a maximum search on the map, the focus of attention (FoA) is calculated.

The bottom-up saliency or object-independent attention is calculated by combining all normalised feature maps to one saliency map.

5.2.2 Top-Down Attention

In order to direct the attention focus, the feature maps λ_i calculated in the bottom-up process have to be weighted according to their importance in discriminating the object from the background. Therefore, the following top-down process distinguishes between the learning phase and the searching phase for an interesting object (Fig. (5.5)).

If a position x, y of a region of an object of interest with size w, h is given, the learning phase identifies the importance of each bottom-up feature. The learning is based on the measurement of one frame, because only one example is needed to calculate the object-descriptive features. At the moment the initialisation of the object region is hand-labelled, but in the future the detection and tracking system, presented in Chapter (4), could deliver an automatic presumption of the object region. In order to calculate the importance of each feature, the foreground is compared to the current background. The foreground corresponds to the object region, while the background relates to the rest of the image without the object region. Here, 3D data can be utilised to segment the foreground more accurate from the background. The depth value from a small rectangle in the middle of the region is taken as mean for the foreground. Z-keying is used (cf.Sec. (2.2.1)) to remove all pixels from the foreground which are farther away than a certain threshold. All remaining pixels are taken as foreground of the object. The comparison of foreground and background is a vital step, because a salient feature is only discriminative, if the background is different to it. If e.g. a person wears a red shirt, it is usually very salient. Is the person standing in front of a red wall, the feature is not discriminative at all. The function of the top-down process is to determine the most discriminative features between object region and background.

Two additional maps are used to weight the features accordant to their affiliation to the fore- or background. An excitation map E relates to the foreground, while an inhibition map I describes the background. For each feature map λ_i two weighting factors ω_E and ω_I are determined, which are used to weight the combination of the bottom-up features for the computation of E and I. Each factor is normalised according to the complete map with the size k, l.

$$\omega_{E,i} = \frac{\sum_{x}^{(x+w)} \sum_{y}^{(y+h)} \lambda_i(x,y)}{\sum_{1}^{k}\sum_{1}^{l}\lambda_i(k,l)} \tag{5.4}$$

$$\omega_{I,i} = \frac{\sum_{1}^{k}\sum_{1}^{l}\lambda_i(k,l) - \sum_{x}^{(x+w)} \sum_{y}^{(y+h)} \lambda_i(x,y)}{\sum_{1}^{k}\sum_{1}^{l}\lambda_i(k,l)} \tag{5.5}$$

$$= 1 - w_{E,i} \tag{5.6}$$

In equation 5.4 to 5.6 $\omega_{I,i}$ and $\omega_{E,i}$ result in values between 0 and 1, where the value appoints if the feature describes the object ($\omega_E \sim 1$) or the background ($\omega_E \sim 0$) and respectively for ω_I the other way around. To get the features only describing the foreground without the background, the weighting factors are subtracted

$$\partial = \omega_{E,i} - \omega_{I,i} \tag{5.7}$$

If ∂ is ~ 0 the value has got no discriminative power. If the value is above 0, the feature is used to describe the foreground, weighted by $\omega_{I,i}$. If the value is below 0, the feature corresponds to the background and is discarded. Consequently, the weighting considers both the object region and the background to get the strongest discriminative features.

Furthermore an uniform LBP is calculated for the foreground δ_F and for the background δ_B. The ratio δ of each LBP feature determines, if the LBP feature discriminates the foreground from the background.

$$\delta = \frac{\delta_F}{\delta_B} \tag{5.8}$$

After the initialisation the system has to search for the interaction partner in each frame. Therefore, all calculated bottom-up feature maps λ are weighted first using ω_I resulting in an inhibition map and second with ω_E leading to an excitation map. The LBP features are incorporated in a separate calculation process, following later on.

$$I = \sum_i \lambda_i * \omega_{I,i} \quad \forall\, \lambda_i \in \lambda \tag{5.9}$$

$$E = \sum_i \lambda_i * \omega_{E,i} \quad \forall\, \lambda_i \in \lambda \tag{5.10}$$

The background is subtracted from the foreground to direct the attention only on the desired object region. The negative values are cropped off to enable the most differing description of the considered human to the background.

$$A = E - I > 0 \tag{5.11}$$

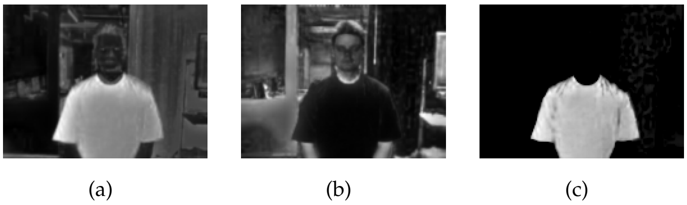


Figure 5.6: *Excitation and Inhibition maps.* In (a) the excitation map, in (b) the inhibition map and (c) the resulting attention map A are shown. In the map A, only the interesting features remain, which highlight the searched torso.

In Fig. (5.6) the inhibition and excitation maps are shown. After the subtraction the shirt clearly pops out in the attention map A. The resulting attention map A is the point feature attention map.

To search for a specific region the point attention map A is extended to a region attention map R (Eqn. (5.12)). Therefore, the initial size w, h of the region is needed. The region attention map is calculated by averaging all point values inside the initial region at the position x, y.

$$R = \sum_{i=x}^{(x+w)} \sum_{j=y}^{(y+h)} \frac{A(i,j)}{w * h} \tag{5.12}$$

Furthermore, a LBP histogram $l_{x,y,w,h}$ is calculated for each possible region at position x, y with the size w, h in the actual image. This region is compared to the initial foreground LBP histogram δ_F. Additionally, the difference is weighted by the ratio δ to highlight the discriminative patterns. To create a LBP attention map LA, the value for each point is incorporated. All values less 0 do not convey any further information and are therefore set to 0.

$$LA = 1 - \sum_x \sum_y ((l_{x,y,w,h} - \delta_{F_{x,y}})^2 * \delta) > 0 \tag{5.13}$$

To combine the region attention map R with the accordant LBP attention map LA, the maps are added and normalised.

$$S = \frac{R + LA}{2} \tag{5.14}$$

The resulting object attention map S describes the probability distribution for the best matching parts of the scene to the initial region.

Because an object varies in its appearance due to lightning changes or different articulations, the target model is slightly adapted by a learning rule. The computed

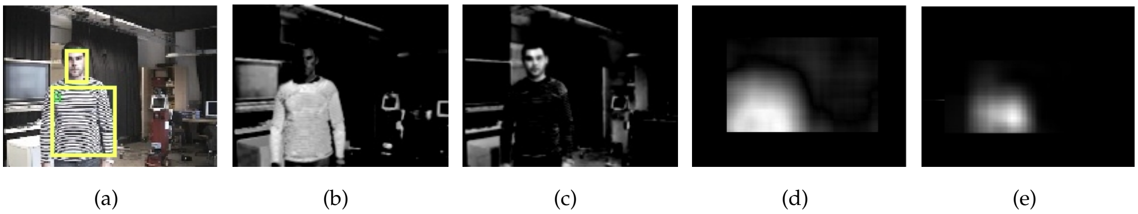


Figure 5.7: *Directed Attention.* This figure presents (a) the model consisting of torso and face, (b) the torso attention map, (c) the face attention map and (d) the LBP attention map. In the last image (e) the resulting object attention map of the top-down process is shown. The weak object model causes a black border in the images of LBP and result, because the model is restricted to be completely in the image.

inhibition and excitation weights are updated due to the newly calculated optimum, if the optimum is above a threshold and the difference to the second optimum is not too small.

$$\omega_{E,i} = \alpha\omega_{E,i} + (1-\alpha)\hat{\omega}_{E,i} \tag{5.15}$$

$$\omega_{I,i} = \alpha\omega_{I,i} + (1-\alpha)\hat{\omega}_{I,i} \tag{5.16}$$

The update rate α is chosen very small in order not to drift away from the original target model. Thus, the weighting factors always represents the actual appearance of the object related to the current background.

5.3 Weak Object Model

The human body is very complex and looks different depending on the view and on the part of the body in focus. Thereby, the face and the torso are the most discriminative areas, because the legs and the arms are similar over the class of a human. Hence, the system implements more than one region per object. A weak human model consisting of a face region F and a torso region T (see Fig. (5.7)a) is used. The face region builds the anchor at the position x_0, y_0 and the torso region is calculated in a concrete distance d_x, d_y to the face region. In order to calculate the directed attention for the whole model, each region has to be processed individually and afterwards they have to be re-combined. For each model region an own attention map is calculated through the above mentioned inhibition, excitation and LBP map (see Fig. (5.7)b-Fig. (5.7)d). To calculate the final object attention map O, all possible model positions are analysed. For each position the single region maps are multiplied with each other. If the regions are partially or fully outside the window, the saliency value is set to 0.

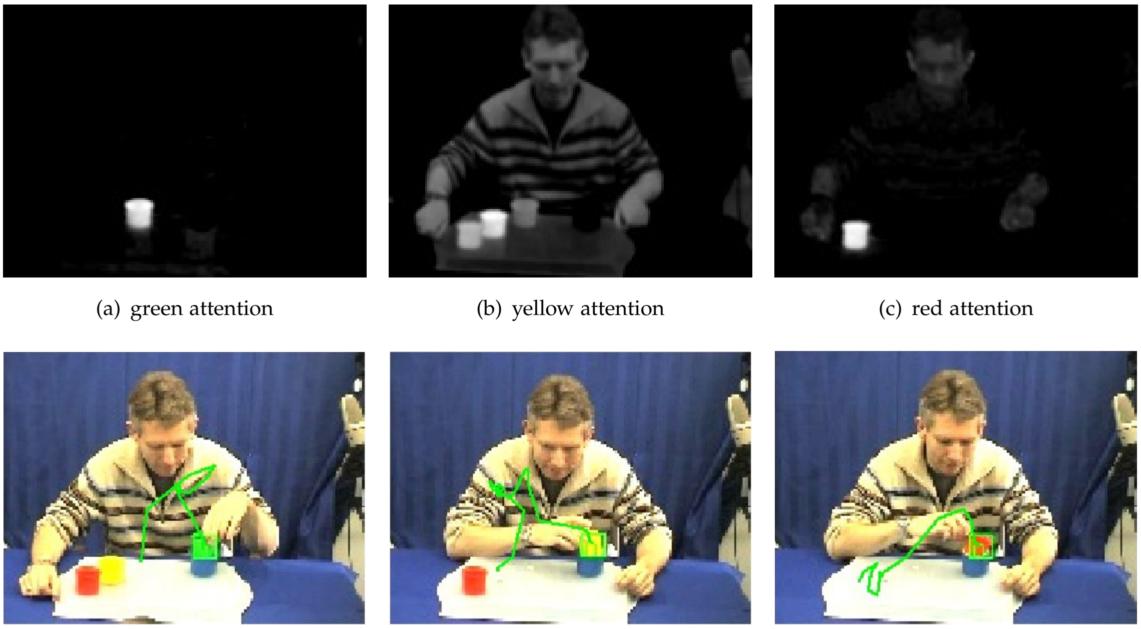
$$O(x,y) = T(x_0,y_0) * F(d_x,d_y) \tag{5.17}$$

$$\text{with} \quad d_x = (x_0 + x) \text{ and } d_y = (y_0 + y)$$

The maximum of O is determined to direct the focus of attention on the best matching model in the image.

5.4 Experiments and Results

In the experiments several scenarios are exploited, where a human and a robot interact with each other. To show the general ability of this approach the system is tested without any changes both in a human-robot interaction scenario and a human parent-infant interaction.



(a) green attention

(b) yellow attention

(c) red attention

(d) green tracking

(e) yellow tracking

(f) red tracking

Figure 5.8: *Object attention results.* In images (a)-(c), the top-down object attention map is shown for the first frame of each sequence. From left to right is the focus on (a) the green, (b) the yellow and (c) the red mug. In image (d) the attention path of the green mug over several frames of the object movement is presented. Accordant in (e) and (f) the attention paths of the yellow and red mug are shown.

In the parent-infant interaction a human subject has to show some mugs to an infant and thereby teach the child how you stack them together. In figure 5.8 one test person is shown, who handles the mentioned coloured mugs. In the top-down approach the focus has been on each mug separately and generated a trajectory for the path of the focus. In the three right images of Fig. (5.8) the plotted path is visible in green, which in every case focused on the real position of the mug.

One can argue that the coloured mugs are easy to detect, but they demonstrate the goal to develop a general attention system for a mobile robot, which needs to be robust as well for easy objects as for complex objects like humans. As shown above the attention system operates reliable and could even be used as a kind of tracker for simple objects.

To test the system for the ability to focus on human interaction partners, two sets of each nine videos recorded in the laboratories of the Bielefeld University are used. The videos contain typical scenes, occurring during typical human-robot interactions. The first set shows the upside of a person, facing the robot, then guiding the robot into another room where three persons are present (see Fig. (5.9)). This set is challenging as

Sequence	Video	Correct FoA	False FoA	%
A	1	47	0	100
	2	29	0	100
	3	57	0	100
	4	25	26	49
	5	34	0	100
	6	37	13	74
	7	43	0	100
	8	54	0	100
	9	15	51	23
Sum				83
B	1	130	0	100
	2	71	36	66
	3	25	122	17
	4	68	16 (12)	80
	5	60	18	76
	6	77	0	100
	7	79	0	100
	8	58	27 (27)	68
	9	78	28 (28)	73
Sum				85 (95)

Table 5.1: *Attention results.* In table I scenario A and B are evaluated each in nine scenes. The focus of attention (FoA) is counted for each frame, where the person is present. It is denoted how often the main focus is equal to the real object position and how often the main focus is on another position. In scenario B it is additionally mentioned in brackets, if the person is in the second attention focus. Finally the ratio of the correct focus is shown.

the rooms offer different backgrounds and lighting conditions. The second set shows again a person facing the robot, followed by different persons and ending with three persons next to each other (see Fig. (5.9)). The difficulty in this scene is caused by the similarity of the persons clothing and the cluttered background. To evaluate the system each frame is counted, where the searched person is present and if the attention system directs the main focus on it.

In Tab. (5.1) the results of the first set of sequences are shown. In most cases the FoA has been directed on the correct person, overall in 83%. Mainly in one video the person does not remain the main attention focus. In this sequence the person walks into another room, stops in front of a window and only the torso remains in the attention focus. The illumination changes from the front of the face to the back, which causes too much difference in all initially detected face features and from there on the person does not get into the focus any more. In table I_B the second sequence is analysed. The overall correct attention rate is at 85%. In this sequence the similar clothing of the persons causes 10% of the false FoA. In sequence $B3$ all three persons wear a very similar shirt and due to the initialisation, where only the main person was present, the features are not distinguishing between them. Here, the learning rate is too low to differentiate between

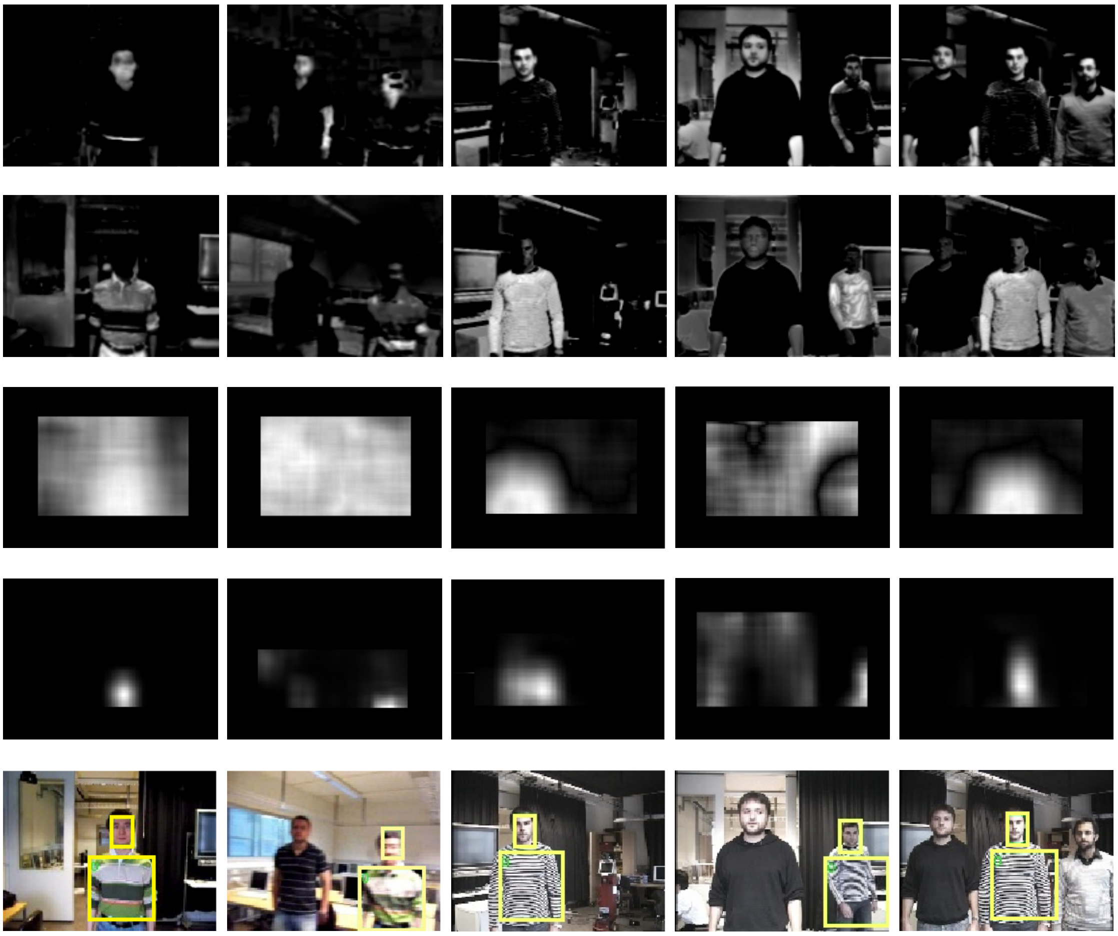


Figure 5.9: *Attention results.* Each column describes one example frame, where (from top to bottom) the face attention map, the torso attention map, the LBP attention map, the resulting object attention map and the calculated object position are shown. The first two frames belong to set A and the last three frames to set B. The second image from set A is motion blurred, which causes the confusion of the LBP. Anyway, the correct position is found, because the combination of face and torso attention map lead to the correct maxima, which indicates the robustness of the approach.

the three persons. Because the similarity leads to many maxima resulting sometimes in a false FoA, I denoted in brackets if the person had the second maxima of the attention map. Only in 5% the FoA is not located in the first or second maxima.

The results are promising as it is the goal to build an attention framework for a mobile robot. If the detection should be refined, one has to take a look at the n-highest maxima of the attention map and determine, which one is the correct person.

5.5 Conclusion

Directing the attention focus enables a mobile robot to realise the third category of situation awareness. The presented system provides a cognitive visual attention framework with a fast learning algorithm, which enables a mobile robot to focus on objects and interaction partners as well. Thereby, the system design is suitable for a mobile robot in terms of execution time and variability in object learning.

The attention system offers several advantages for a mobile robot. The one shot-learning identifies the most discriminative features of the target region and the current background. Thereby, the new idea of dividing the searched region into several sub-regions in a combined model enhances the search of especially humans. The resulting feature vector from different regions result in a more definitely attention map. The system is able to learn target models from many possible objects, like e.g. cups or even more complex objects like humans. Thereby, the features are autonomously weighted due to their information content and subsequently updated in order to represent as good as possible the searched object. If e.g. two persons wear a shirt with the same colour, the system focuses on the structure of the shirt in order to distinguish the two persons. The system is able to handle clutter, as it incorporates the feature region and the current background as well.

Although the attention system already shows promising results, some further steps have to be realised in future. The initialisation of the objects should be combined with another system in order to automatically determine the target region. Here, the presented systems in Chapter (3) and Chapter (4) could provide the initial region. The target model itself has to refined. At the moment the model is fixed, but in future the model should provide automatic scaling and a more variable connection between the attendant regions. Here, the pictorial structures provide an ideal representation for a more flexible model [64]. As a last step it could be interesting to save the features of the foreground in order to recalculate the top-down weights in a different scene with another background. Here, additional information from the robot platform are helpful to determine, if the robot has moved into a new room or more generally, if the observed scene has changed.

6 Conclusion and Outlook

Situation awareness is an important characteristic for robots, stationary and especially mobile ones. Mobile robots have to deal with different and dynamic requirements, where background changes and humans walk into and out of the scene. Particularly the humans are severely manageable. They often have different shapes and appearances and show a dynamic motion. In the beginning I pose the question:

- ***How can perceptual situation awareness be achieved on a mobile robot?***

The here presented thesis attends to this question by proposing solutions to the first three categories of SA, which enable mobile robots to have a detailed and efficient picture of their surrounding.

Chapter 3 "How can a robot system be able to sense and perceive the environment without restricting the perception onto specific objects?"
As a solution to the first category of situation awareness I propose the use of the *Vista space* in order to build an *articulated scene model*. This new model incorporates knowledge about the static background, movable objects and humans in one representation, which ideally matches to the category of extracting environmental information. Thereby, the system uses 3D observations of the scene and consolidates the new information effectively into the already known model. Due to the use of the articulated scene model the 3D background modelling of the static background can be done more reliably. This is achieved through the simultaneous detection and tracking of humans, which are removed from the background modelling process. The detection and tracking itself can be done more reliable by subtracting the known background. The detected action spaces or moved objects respectively are revealed by the modelling process and are facilitated for further processing.

Chapter 3 outlook: The articulated scene model provides a good knowledge basis of the surrounding, which on the one hand could be utilised for other subsequent processes or on the other hand further refined through the interaction with the user. Subsequent processes could use the articulated parts for object learning or the static background for navigation purposes. Also the movements of the human are interesting for the robot. They denote free spaces for the own movement and more apparently, the known positions of the humans are interesting for the interaction. Especially the last point is important for further research. The interaction manifoldly benefits from the articulated scene model. First, the robot knows about the present users. Second, the robot gets an idea about areas, that are changed or employed by the human. These areas are detected and marked by the articulated scene model. Then they should be used by the robot in order to ask the human about these areas to further improve its own knowledge. In this way, the interaction is meaningful extended and the robot improves its situation awareness. Further work could also be done in the direction of combining

Vista spaces after movements of the robot. The static background models could be merged into a global background scene model using e.g. iterative closest point (ICP) algorithms (cf.Sec. (3.7)). The background information should also be incorporated into the navigation process, which would stabilise the movements of the robot.

Chapter 4 "How can a robot system be aware of humans and their movements during ego motion?"
The detection and tracking system for mobile platforms published in this thesis presents a solution to the complex problem of the second part of situation awareness. Many map-building approaches provide a good map of the static environment, but can hardly deal with dynamic objects in the scene. The detection and tracking of moving entities is most important, as they mark potential collision objects, which could not directly be mapped in the navigation. Here, it is important to have the time sequence of the movements to be able to predict the future movement of the objects. This thesis addresses the complete process for building trajectories of all humans in the scene, where each trajectory marks all occurrences of the specific human. The process includes the measurement generation, the detection and tracking of each human and the effective coordination of information in the whole process. Thereby, I propose the combination of 2D and 3D data, which shows emending results. Additionally, a fast pre-detection step using the *u-v-disparity* is presented, which simplifies the scene through floor and ceiling removal and which serves as input for a distance adaptive version of the state-of-the-art human detector. The detector verifies the pre-detections using the *Histograms of Oriented Gradients* in conjunction with a linear support vector machine. The effective combination of the pre-detection and the window-based classifier definitively speeds-up the detection process, reduces the false-positive rate and enhances the stability of the detection (cf.Sec. (4.7.4)). A new implemented hypotheses management incorporates all detection and tracking information in order to estimate the hypotheses at the best. In addition, the hypotheses management cares for the creation and deletion of hypotheses, if a person enters or leaves the scene. The presented tracking module shows how hypotheses could be tracked fast and robust by the use of an *adaptive particle filter with a multidimensional observation model*. Again, the combined use of 2D and 3D data demonstrates an advantage over pure 2D or 3D approaches (cf.Sec. (4.7.5)).

Chapter 4 outlook: Although the presented system shows already fast processing speed and good results, future work in speeding-up the process by parallelism is planned. Especially the particle filter offers high potential to render the process on a graphic card, where each particle could be calculated on its own. It is also considered to do both stabilise the SLAM approach and help building a 3D map of the environment. More precisely, visual odometry could be a solution to combine the first system with the second system presented. If the background could be removed during the movement of the robot the problem of detecting people in connection with a wall would be obsolete due to the background removing.

Chapter 5 "How can a robot system direct its attention onto specific areas like a desired interaction partner?"
Directing the attention focus relates to the third category of situation awareness. The biologically inspired attention has got several advantages. A specific area can be analysed

more intensively or an object can be searched using its discriminative features. In my thesis I propose an attention system based on a top-down directed search, which finds the discriminative features according to the background or other objects. It weights the bottom-up features, yielding an attention map for the searched object. In contrast to other attention systems, the proposed work adds a simple model in order to strengthen the attention for a distinctive object like a human. The different parts (face and torso) usually inherit different discriminative features, which are revealed for each region independently. A combination process calculates an object attention map out of all areas. The results show the good quality of the human model approach and the possibility to direct the attention focus even on difficult objects. Hence, the third category of situation awareness can now be achieved on a mobile robot.

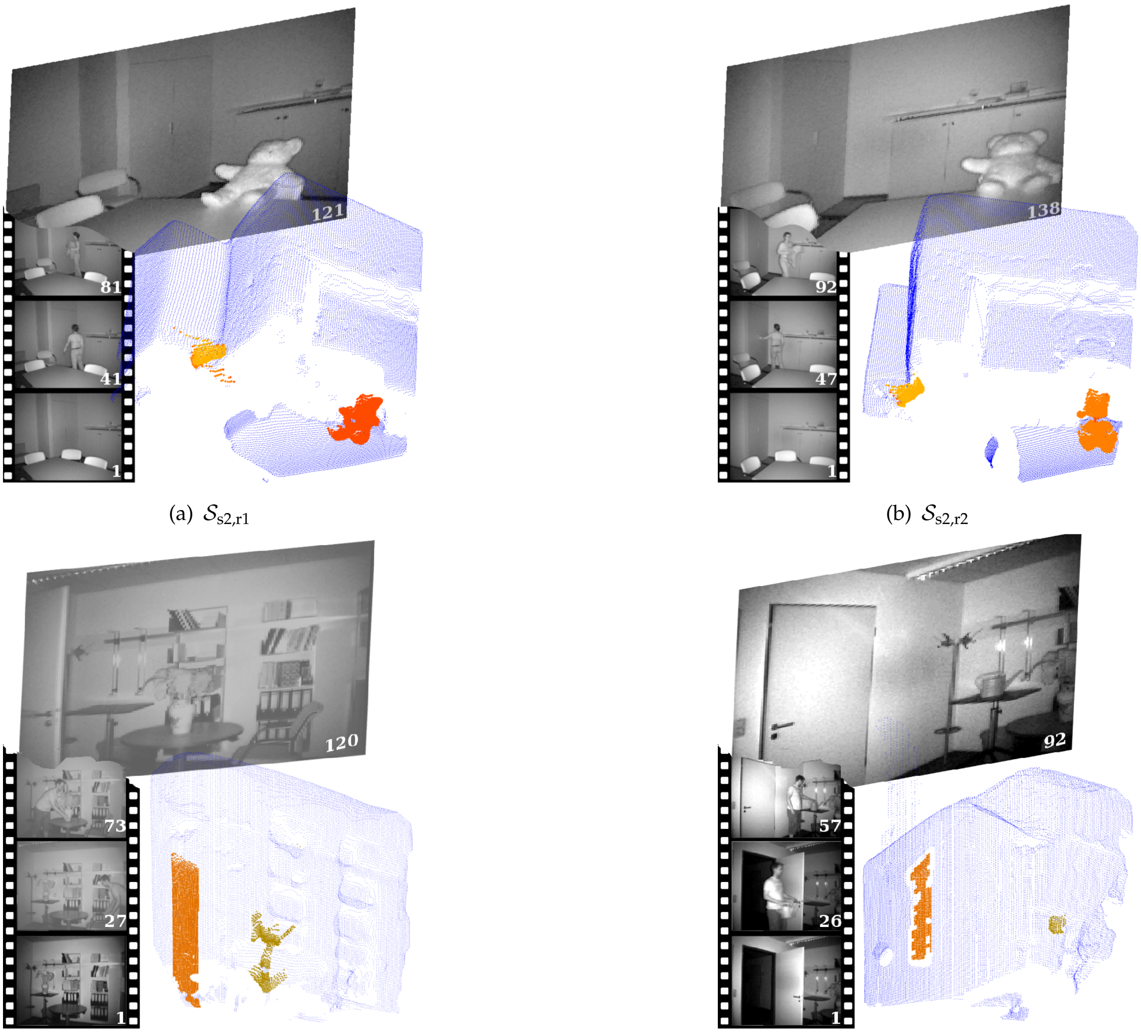
Chapter 5 outlook: The model approach relies on a simple human model consisting of face and torso. In the future, the simple model will be replaced by the pictorial structures model. It provides a better connection between the single parts, because the connection is more flexible. In connection with the attention system and the other presented systems more research has to be done. An automatic initialisation of the human model by the detection and tracking system would be ideal. Additionally, the attention system could provide an easy solution to recover a lost trajectory or to identify a known person, if he/she is out of view for a longer period.

Final statement: Of course, work has to be investigated in the combination of all three perceptual categories of situation awareness into one coherent system, where all parts gain from each other. First steps towards this direction are already done by combining the mobile tracking system with the articulated scene model. The detection and tracking based on the 6D clustering approach is exchanged with the mobile version of chapter 4. But, at the moment only partial movements are possible to gather the articulated scene model, because the background modelling process is not feasible during movement. Here, the visual odometry will be applied to overcome this restriction. The combination with the top-down attention system conforms to the subsequent step in order to keep the interaction up with special interaction partners. If a conversation is started or a target is in the focus of interest, the top-down directed learning process can be started in order to keep the target in the focus. Utilising the information from the articulated scene model, the target region could easily be applied and the subsequent search be reduced to the potential dynamic points.

Finally, work has to be done in the area of the fourth category of situation awareness, the prediction of knowledge in the future. Here, the basis for this work is presented through my solutions to the perceptual parts of situation awareness. Prospective work has to consider predictions of movements [92] [161] or to combine knowledge of articulated parts with higher knowledge, gathered from e.g. a dialogue system [137].

A Appendix

The following images represent the results of the articulated scene model, described in Chapter (3). Each image shows a small film strip, which represents the underlying sequence. In the background the current frame is displayed. In the foreground the 3D data is divided into static parts (blue) and movable parts (red and orange tones) by the articulated scene model. The articulated scene model does not differentiate between specific objects. Everything, which has been moved by a human is detected through observation and the vista-space assumption.



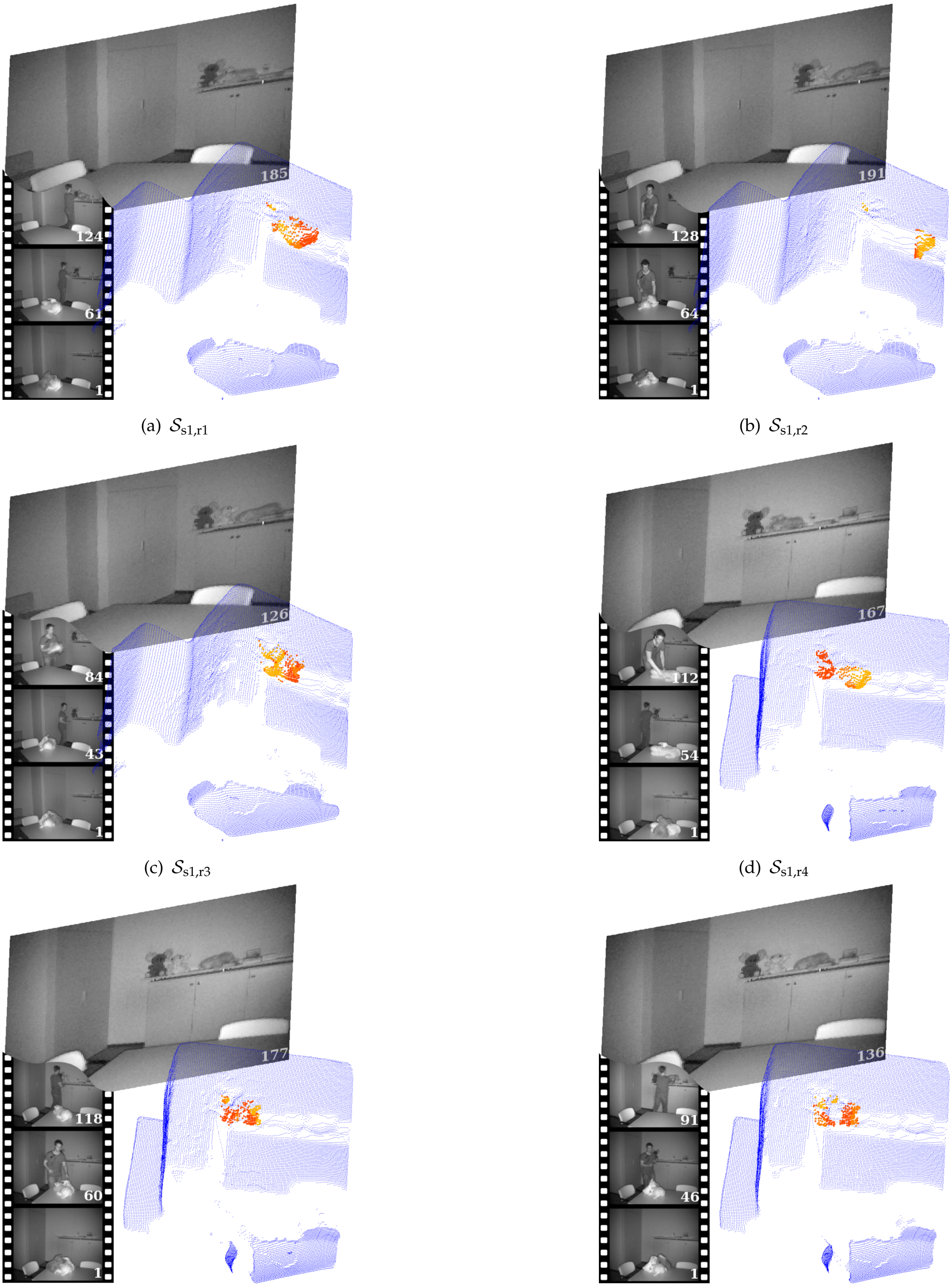
(a) $\mathcal{S}_{\mathrm{s2,r1}}$

(b) $\mathcal{S}_{\mathrm{s2,r2}}$

(c) $\mathcal{S}_{\mathrm{s5,r1}}$

(d) $\mathcal{S}_{\mathrm{s5,r2}}$

Figure A.1: *Articulated scene model results.* (a)-(d): For all recorded sequences the learnt background model (blue points) and the detected movable objects (orange points) are shown. In the bottom left three selected images of the sequence characterize the tide of events from bottom to top finishing with the last frame in the background.



(a) $\mathcal{S}_{\mathrm{s1,r1}}$

(b) $\mathcal{S}_{\mathrm{s1,r2}}$

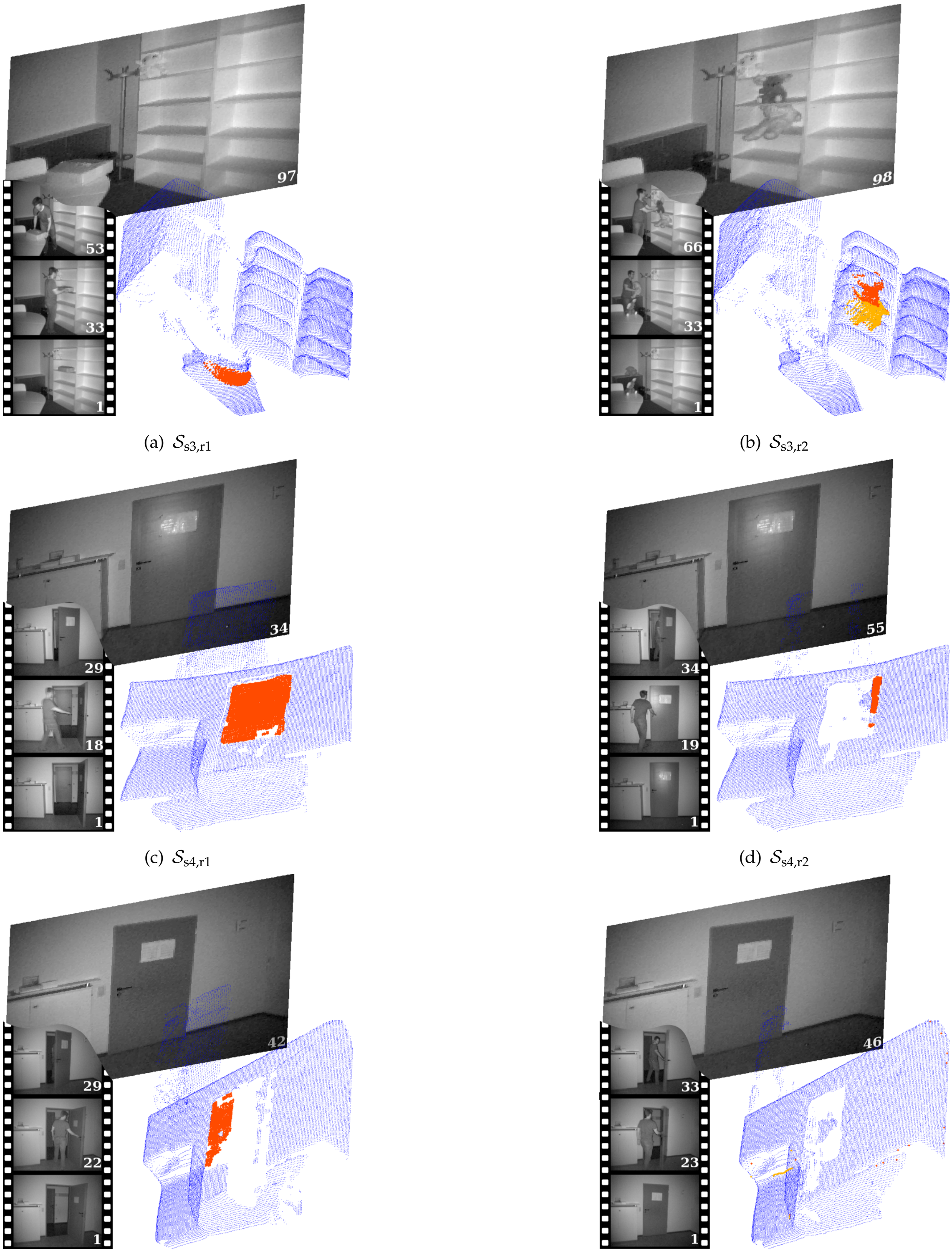
(c) $\mathcal{S}_{\mathrm{s1,r3}}$

(d) $\mathcal{S}_{\mathrm{s1,r4}}$

(e) $\mathcal{S}_{\mathrm{s1,r5}}$

(f) $\mathcal{S}_{\mathrm{s1,r6}}$

Figure A.2: *Articulated scene model results.* (a)-(f): For all recorded sequences the learnt background model (blue points) and the detected movable objects (orange points) are shown. In the bottom left three selected images of the sequence characterize the tide of events from bottom to top finishing with the last frame in the background.



(a) $\mathcal{S}_{\mathrm{s3,r1}}$

(b) $\mathcal{S}_{\mathrm{s3,r2}}$

(c) $\mathcal{S}_{\mathrm{s4,r1}}$

(d) $\mathcal{S}_{\mathrm{s4,r2}}$

(e) $\mathcal{S}_{\mathrm{s4,r3}}$

(f) $\mathcal{S}_{\mathrm{s4,r4}}$

Figure A.3: *Articulated scene model results.* (a)-(f): For all recorded sequences the learnt background model (blue points) and the detected movable objects (orange points) are shown. In the bottom left three selected images of the sequence characterize the tide of events from bottom to top finishing with the last frame in the background.

B Appendix

In the following, film strips for the evaluated data sets in Chapter (4) are presented. Each film shows different parts of the correspondent data set.

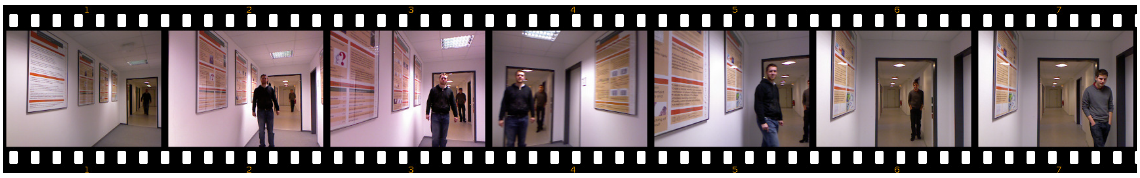


Figure B.1: *Image strip for data set 1.*

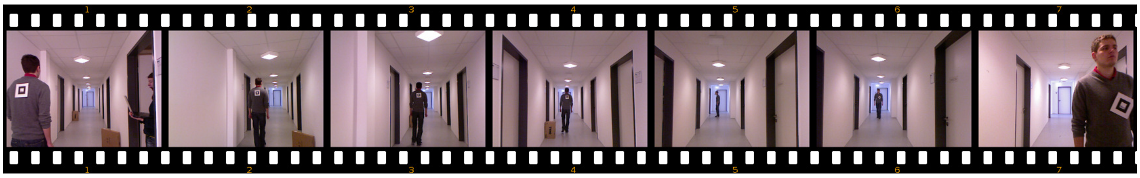


Figure B.2: *Image strip for data set 2.*

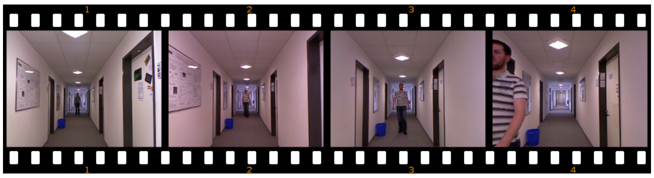
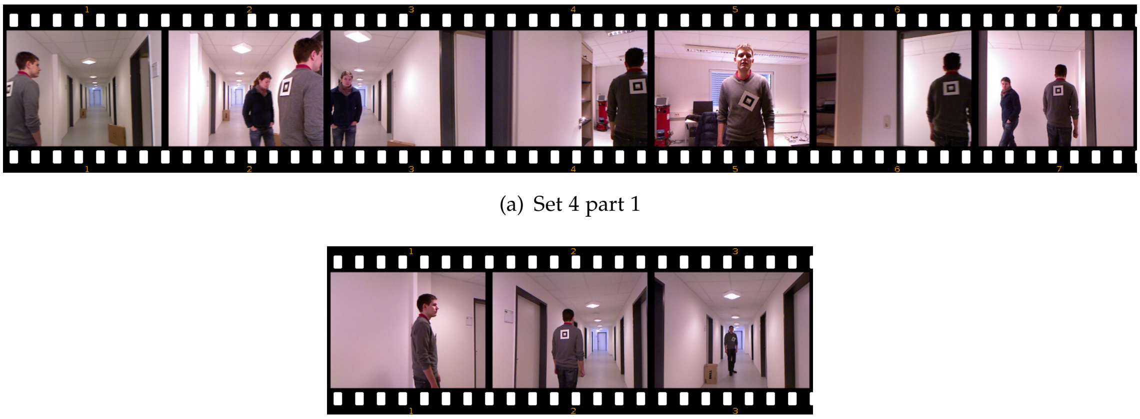


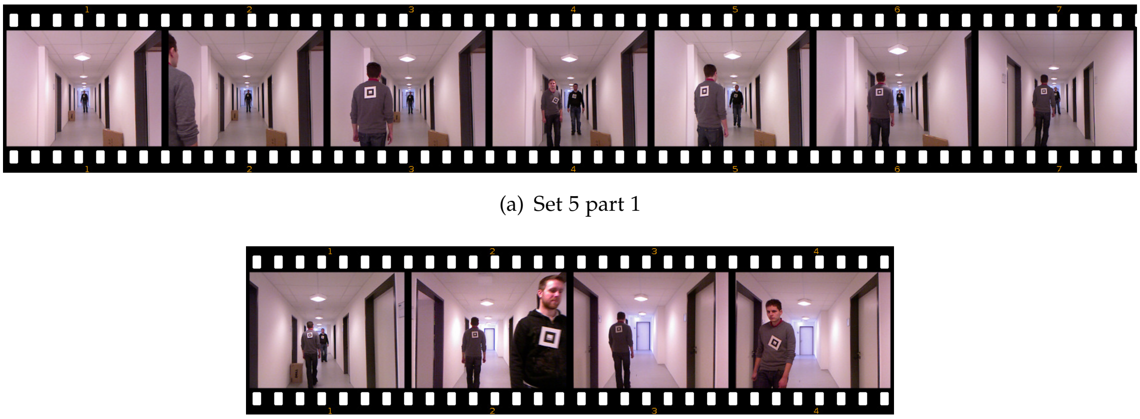
Figure B.3: *Image strip for data set 3.*



(a) Set 4 part 1

(b) Set 4 part 2

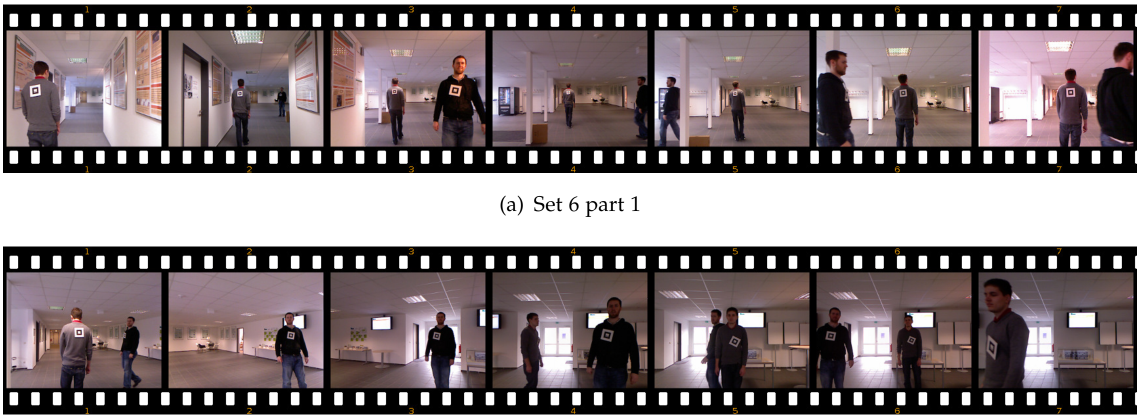
Figure B.4: *Image strip for data set 4.*



(a) Set 5 part 1

(b) Set 5 part 2

Figure B.5: *Image strip for data set 5.*



(a) Set 6 part 1

(b) Set 6 part 2

Figure B.6: *Image strip for data set 6.*

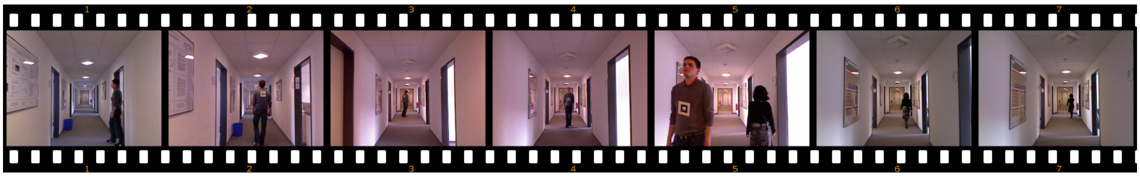


Figure B.7: *Image strip for data set 7.*

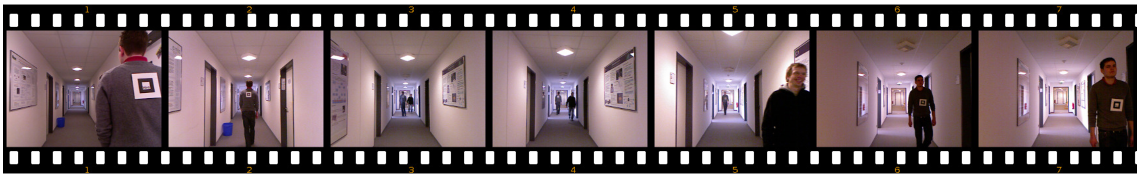
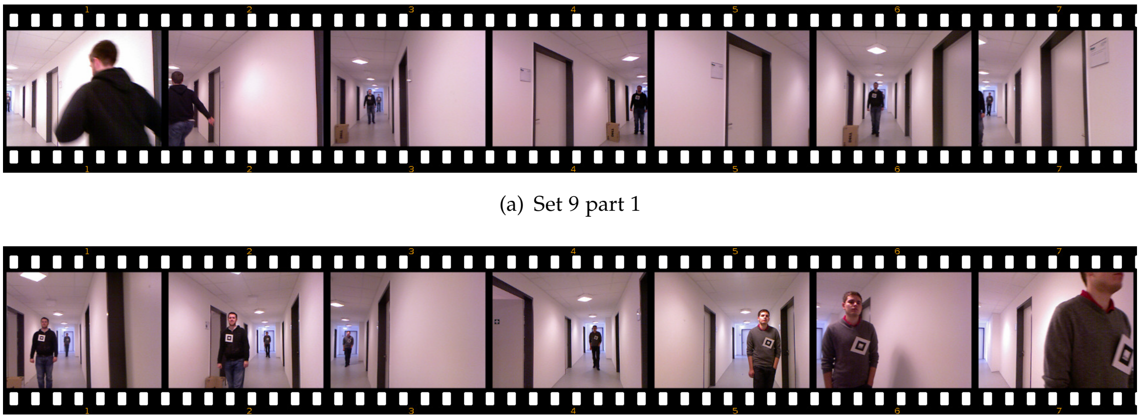


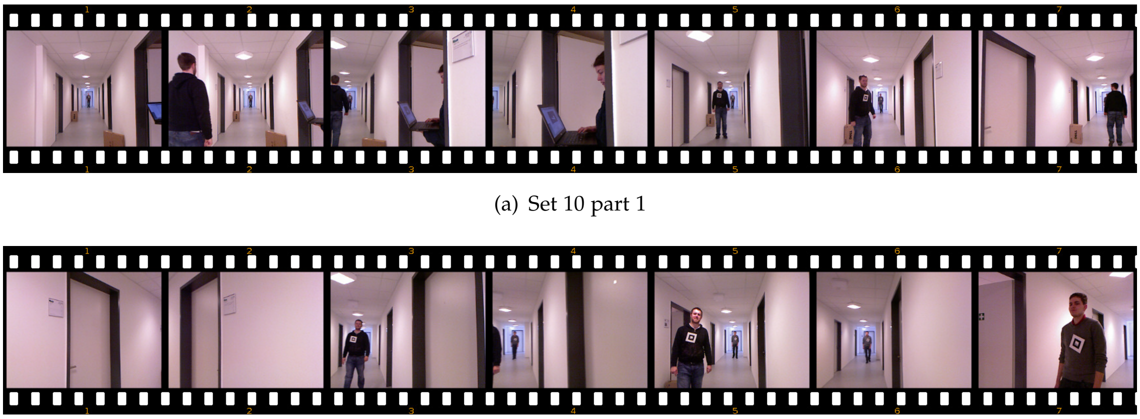
Figure B.8: *Image strip for data set 8.*



(a) Set 9 part 1

(b) Set 9 part 2

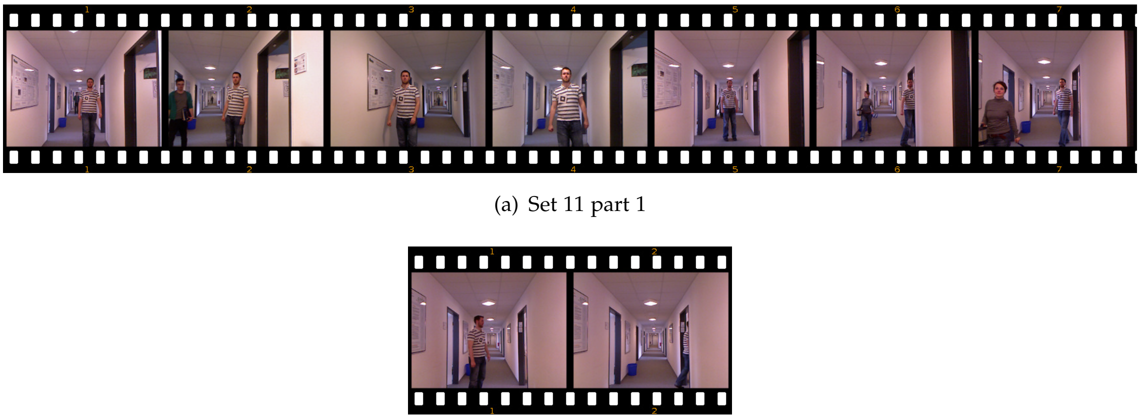
Figure B.9: *Image strip for data set 9.*



(a) Set 10 part 1

(b) Set 10 part 2

Figure B.10: *Image strip for data set 10.*



(a) Set 11 part 1

(b) Set 11 part 2

Figure B.11: *Image strip for data set 11.*

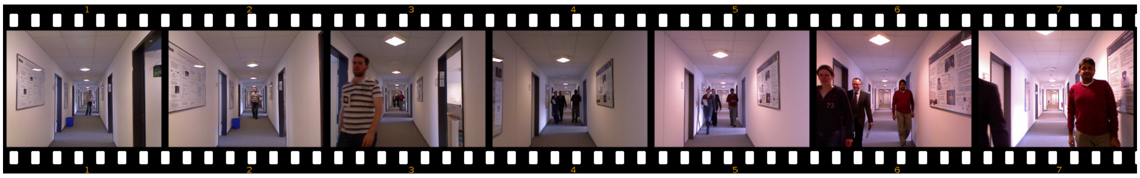
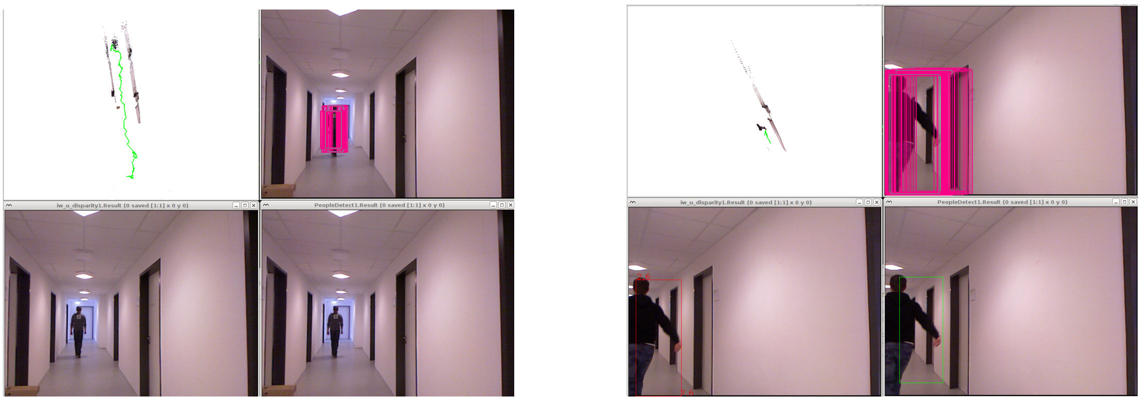


Figure B.12: *Image strip for data set 12.*

Trajectory Results of the Mobile Tracking System

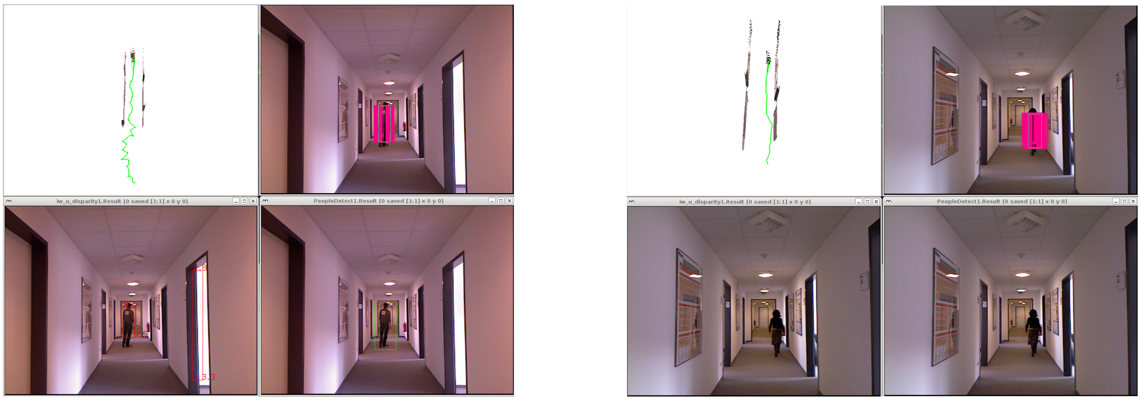
The following figures show the remaining trajectory snippets of the presented data sets in Chapter (4). The images illustrate the performance of the complete system by representing each step in an own image. The pre-detection starts in the lower left, where each pre-detected area is denoted with a red rectangle. The hypotheses are handed over to the verification step in the lower right, where true objects have to be verified. If they are already verified, they have to be approved in at least every 15th frame in order to monitor that the hypothesis is still valid. If a region is verified as a human, a particle filter is started, which is shown in the upper right as purple rectangles. The green rectangles represent the actual state of each hypothesis. The arising trajectory is painted in the upper left as green line in a three dimensional plot viewed from above.


iw_u_disparity1.Result (0 saved [1:1] x 0 y 0)
PeopleDetect1.Result (0 saved [1:1] x 0 y 0)
iw_u_disparity1.Result (0 saved [1:1] x 0 y 0)
PeopleDetect1.Result (0 saved [1:1] x 0 y 0)

(a) Set 2

(b) Set 9

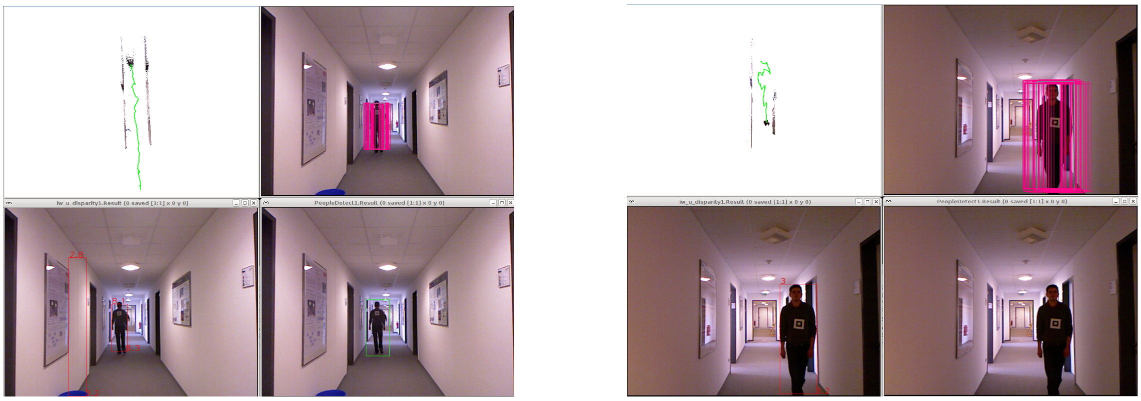
Figure B.13: *Qualitative tracking analysis.*



(a) Set 7 part 1

(b) Set 7 part 2

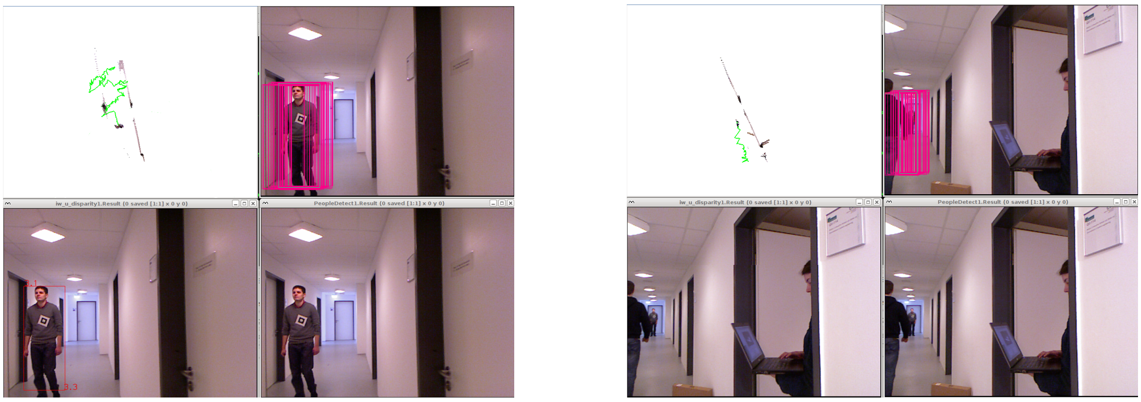
Figure B.14: *Qualitative tracking analysis.*



(a) Set 8 part 1

(b) Set 8 part 2

Figure B.15: *Qualitative tracking analysis.*

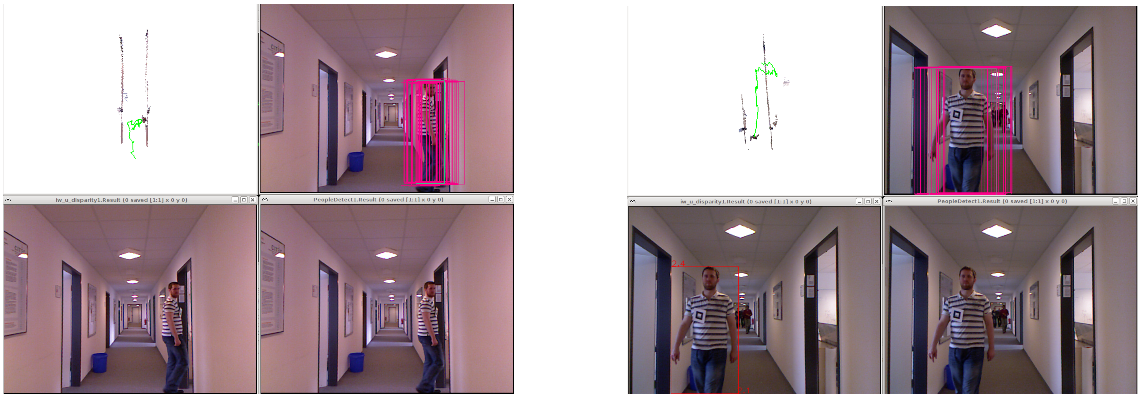


(a) Set 10 part 1

(b) Set 10 part 2

Figure B.16: *Qualitative tracking analysis.*

The trajectories for all persons and each set are presented in the following. All trajectories are in relative distance to the robot. The different colours represent the different ids

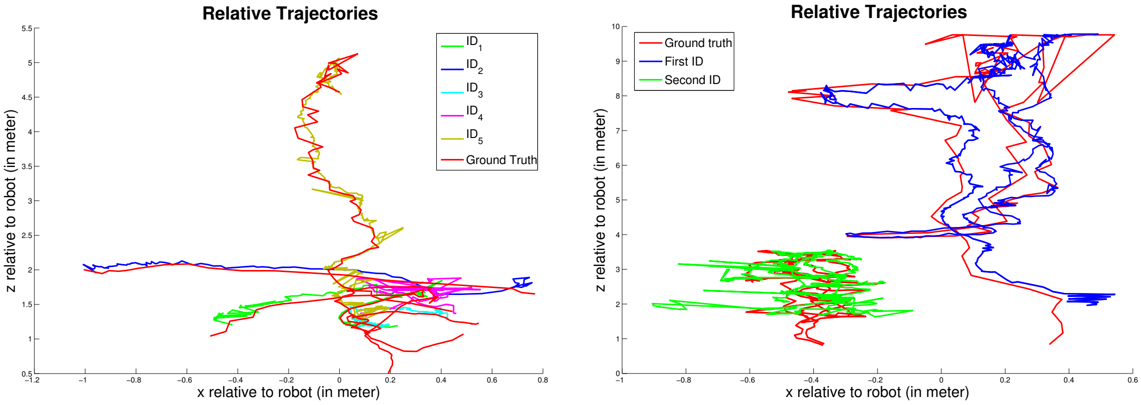

iw_u_disparity1.Result (0 saved [1:1] x 0 y 0)
PeopleDetect1.Result (0 saved [1:1] x 0 y 0)
iw_u_disparity1.Result (0 saved [1:1] x 0 y 0)
PeopleDetect1.Result (0 saved [1:1] x 0 y 0)

(a) Set 11 part 1

(b) Set 12 part 1

Figure B.17: *Qualitative tracking analysis.*

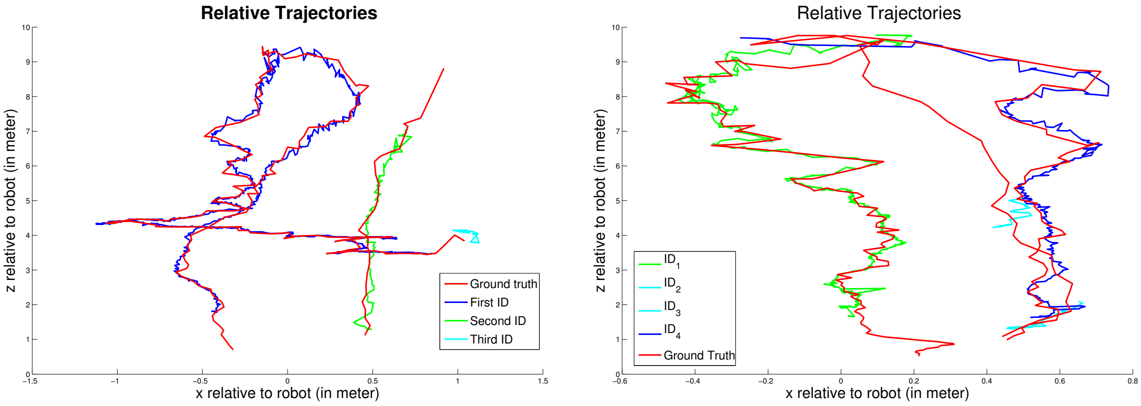
of the persons. The ground truth is also plotted in red. Most trajectories nearly fit the ground truth. The underlying errors are shown below.



(a) Set 4

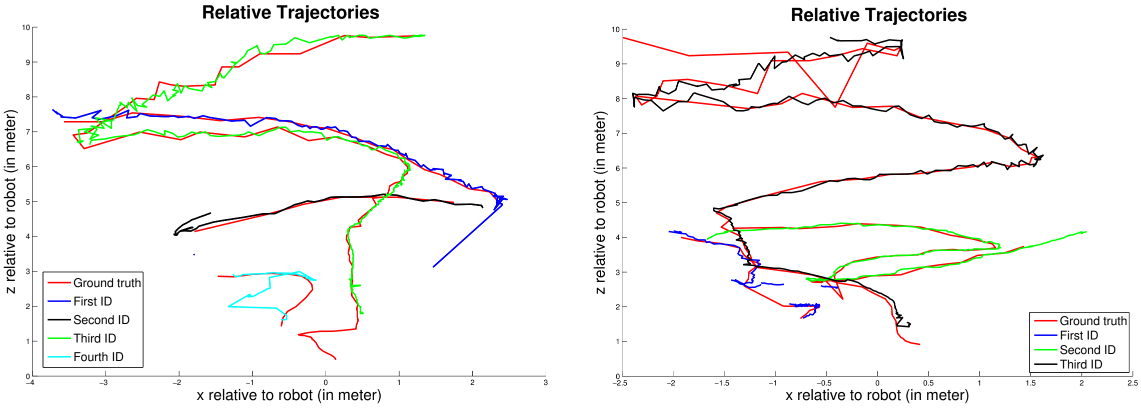
(b) Set 5

Figure B.18: *Relative Trajectories for set 4 & 5.* In red all ground truth trajectories are shown. The persons are denoted in different colours. All trajectories are in relative distance to the robot.



(a) Set 7 (b) Set 8

Figure B.19: *Relative Trajectories for set 7 & 8.* In red all ground truth trajectories are shown. The persons are denoted in different colours. All trajectories are in relative distance to the robot.



(a) Set 9 (b) Set 10

Figure B.20: *Relative Trajectories for set 9 & 10.* In red all ground truth trajectories are shown. The persons are denoted in different colours. All trajectories are in relative distance to the robot.

Tracking Errors of the Mobile Tracking System

The following table shows the errors of each person in all dimensions and in all sequences.

Set,Person	Error X	Error Y	Error Z	Std X	Std Y	Std Z
1,1	0.040513	0.049438	0.049438	0.032136	0.04097	0.051873
1,2	0.088643	0.09496	0.09496	0.10649	0.069647	0.24999
1,3	0.029564	0.060525	0.060525	0.035128	0.022627	0.025935
1,4	0.024221	0.02998	0.02998	0.020474	0.032201	0.029934
1,5	0.063252	0.12083	0.12083	0.061881	0.082595	0.10025
2,1	0.025723	0.052144	0.052144	0.027377	0.039754	0.062465
3,1	0.027284	0.047988	0.047988	0.022848	0.037269	0.050266
4,1	0.021805	0.038451	0.038451	0.026933	0.03376	0.065874
4,2	0.030417	0.017102	0.017102	0.014171	0.012102	0.037419
4,3	0.030013	0.015202	0.015202	0.019466	0.011221	0.051492
4,4	0.02852	0.045849	0.045849	0.046421	0.030588	0.069269
5,1	0.038717	0.074203	0.074203	0.034066	0.053458	0.13755
5,2	0.030683	0.047202	0.047202	0.041156	0.041775	0.052775
6,1	0.026576	0.048557	0.048557	0.017519	0.031644	0.038708
6,2	0.040384	0.045265	0.045265	0.032823	0.032331	0.067292
6,3	0.043989	0.080081	0.080081	0.036359	0.055967	0.058141
7,1	0.026733	0.050603	0.050603	0.019863	0.034011	0.057314
7,2	0.0275	0.1264	0.1264	0.019986	0.12068	0.14654
8,1	0.028995	0.054423	0.054423	0.029628	0.045385	0.07704
8,2	0.03546	0.18875	0.18875	0.025157	0.10467	0.1101
9,1	0.070379	0.068492	0.068492	0.039773	0.028993	0.054139
9,2	0.051176	0.045322	0.045322	0.048664	0.01698	0.025626
9,3	0.04128	0.068151	0.068151	0.03605	0.066763	0.081341
9,4	0.04128	0.068151	0.068151	0.03605	0.066763	0.081341
10,1	0.041672	0.061332	0.061332	0.029893	0.032375	0.08209
10,2	0.040604	0.057049	0.057049	0.026173	0.048201	0.026796
10,3	0.054658	0.073577	0.073577	0.071329	0.070325	0.22102
11,1	0.028053	0.067105	0.067105	0.021008	0.056582	0.042483
11,2	0.098065	0.15555	0.15555	0.14409	0.18802	0.20711
11,3	0.10404	0.11691	0.11691	0.15783	0.089928	0.25584
12,1	0.024167	0.04893	0.04893	0.019331	0.043435	0.068789
12,2	0.038343	0.076467	0.076467	0.040198	0.053691	0.10701
12,3	0.079664	0.05758	0.05758	0.061649	0.063785	0.26902
average	**0.041834**	**0.066252**	**0.066252**	**0.041233**	**0.051721**	**0.090083**

Table B.1: *Error and standard deviation in XYZ for each person.*

List of Figures

List of Tables

Bibliography

[1] Abd-Almageed, W., Hussein, M., Abdelkader, M., and Davis, L. Real-Time Human Detection and Tracking from Mobile Vehicles. In *2007 IEEE Intelligent Transportation Systems Conference* (Sept. 2007), IEEE, pp. 149–154.

[2] Agrawal, A., and Chellappa, R. Fusing depth and video using rao-blackwellized particle filter. *Pattern Recognition and Machine Intelligence* (2005), 521–526.

[3] Agrawal, M., Konolige, K., and Blas, M. Censure: Center surround extremas for realtime feature detection and matching. In *European Conference on Computer Vision 2008* (2008), Springer, pp. 102–115.

[4] Aherne, F., Thacker, N., and Rockett, P. The {B}hattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data. *Kybernetika 34*, 4 (1998), 363 – 368.

[5] Anderson, J. R. *Cognitive psychology and its implications*, 6 ed. Worth Publishers, 2004.

[6] Andriluka, M., Roth, S., and Schiele, B. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008* (2008), pp. 1–8.

[7] Andriluka, M., Roth, S., and Schiele, B. Monocular 3D Pose Estimation and Tracking by Detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (San Francisco, USA, 2010), no. 2, IEEE.

[8] Ansari, R., and Khokhar, a. Multiple Object Tracking with Kernel Particle Filter. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), 566–573.

[9] Arndt, R., Schweiger, R., Ritter, W., Paulus, D., and Lohlein, O. Detection and Tracking of Multiple Pedestrians in Automotive Applications. In *2007 IEEE Intelligent Vehicles Symposium* (June 2007), IEEE, pp. 13–18.

[10] Avidan, S. Support vector tracking. *IEEE transactions on pattern analysis and machine intelligence 26*, 8 (Aug. 2004), 1064–72.

[11] Avidan, S. Ensemble tracking. *IEEE transactions on pattern analysis and machine intelligence 29*, 2 (Feb. 2007), 261–71.

[12] Badrinarayanan, V., Perez, P., Le Clerc, F., and Oisel, L. Probabilistic Color and Adaptive Multi-Feature Tracking with Dynamically Switched Priority Between Cues. *2007 IEEE 11th International Conference on Computer Vision* (Oct. 2007), 1–8.

[13] Bajracharya, M., Moghaddam, B., Howard, A., Brennan, S., and Matthies, L. Results from a real-time stereo-based pedestrian detection system on a moving vehicle. In *Workshop on People Detection and Tracking, IEEE ICRA* (2009), no. May.

[14] Bar-shalom, Y., Daum, F., and Huang, J. I. M. The probabilistic data association filter. *IEEE Control Systems Magazine*, December (2009).

[15] Bar-Shalom, Y., and Fortmann, T. *Tracking and data association*, 179 ed. Academic Press Professional, San Diego, CA, USA, Oct. 1987.

[16] Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, Jan. 2002.

[17] Bauer, A., Klasing, K., Lidoris, G., Mühlbauer, Q., Rohrmüller, F., Sosnowski, S., Xu, T., Kühnlenz, K., Wollherr, D., and Buss, M. The Autonomous City Explorer: Towards Natural Human-Robot Interaction in Urban Environments. *International Journal of Social Robotics 1*, 2 (Feb. 2009), 127–140.

[18] Bay, H., Ess, a., Tuytelaars, T., and Vangool, L. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding 110*, 3 (June 2008), 346–359.

[19] Beeson, P., MacMahon, M., Modayil, J., Murarka, A., Kuipers, B., and Stankiewicz, B. Integrating multiple representations of spatial knowledge for mapping, navigation, and communication. In *Proceedings of the Symposium on Interaction Challenges for Intelligent Assistants* (Stanford, CA, 2007), AAAI Spring Symposium Series. AAAI Technical Report SS-07-04.

[20] Bellotto, N., and Hu, H. People Tracking and Identification with a Mobile Robot. In *2007 International Conference on Mechatronics and Automation* (Aug. 2007), IEEE, pp. 3565–3570.

[21] Berthold, M., and Hand, D. J. *Intelligent Data Analysis*, 2nd ed. Springer, 2003.

[22] Bertozzi, M., Broggi, A., Fascioli, A., and Nichele, S. Stereo vision-based vehicle detection. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE* (2002), no. Mi, IEEE, pp. 39–44.

[23] Betke, M., Haritaoglu, E., and Davis, L. S. Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications 12*, 2 (Aug. 2000), 69–83.

[24] Beuter, N., Lohmann, O., Schmidt, J., and Kummert, F. Directed attention - a cognitive vision system for a mobile robot. *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication* (Sept. 2009), 854–860.

[25] Beuter, N., Swadzba, A., Kummert, F., and Wachsmuth, S. Using Articulated Scene Models for Dynamic 3D Scene Analysis in Vista Spaces. *3D Research 04* (2010), 1–13.

[26] Beuter, N., Swadzba, A., and Schmidt, J. Simultaneous Tracking And Scene Reconstruction For Robot Perception. In *Workshop for Cognitive Humanoid Vision* (Daejon, Korea, 2008), IEEE, pp. 2–4.

[27] Beuter, N., Swadzba, A., Schmidt, J., and Sagerer, G. 3D-Szenenrekonstruktion in dynamischen Umgebungen. In *Oldenburger 3D Tage* (Oldenburg, Germany, 2009), Luhmann.

[28] Beymer, D., and Konolige, K. Tracking People from a Mobile Platform. *Experimental Robotics VIII 5* (June 2003), 234–244.

[29] Birchfield, S., Natarajan, B., and Tomasi, C. Correspondence as energy-based segmentation. *Image and Vision Computing 25*, 8 (Aug. 2007), 1329–1340.

[30] Blom, H., and Bar-Shalom, Y. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control 33*, 8 (1988), 780–783.

[31] Blom, H., and Bloem, E. Interacting multiple model joint probabilistic data association avoiding track coalescence. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002* (2002), vol. 3, IEEE, pp. 3408–3415.

[32] Bo, W., and Nevatia, R. 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'05) Volume 1* (Oct. 2005), IEEE, pp. 90–97 Vol. 1.

[33] Bo, W., Nevatia, R., and Wu, B. 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'05) Volume 1* (Oct. 2005), IEEE, pp. 90–97 Vol. 1.

[34] Bouguet, J.-Y. Camera Calibration Toolbox for Matlab, 2007.

[35] Boykov, Y., and Funka-Lea, G. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision 70*, 2 (Nov. 2006), 109–131.

[36] Bradski, G., and Kaehler, A. *Learning OpenCV*, 1 ed. Oreilliy, 2008.

[37] Bradski, G. R. Real Time Face and Object Tracking as a Component of a Perceptual User Interface. In *4th IEEE Workshop on Applications of Computer Vision* (Washington, DC, USA, Oct. 1998), IEEE Computer Society, p. 214.

[38] Breazeal, C., and Scassellati, B. A context-dependent attention system for a social robot. In *In 1999 International Joint Conference on Artificial Intelligence* (1999), pp. 1146–1151.

[39] Breitenstein, M. D., Sommerlade, E., Leibe, B., and van Gooland I. Reid, L. Probabilistic parameter selection for learning scene structure from video. In *Proceedings of the British Machine Vision Conference* (2008).

[40] Broggi, a., Fascioli, A., Carletti, M., Graf, T., and Meinecke, M. A multiresolution approach for infrared vision-based pedestrian detection. In *Intelligent Vehicles Symposium, 2004 IEEE* (2004), IEEE, pp. 7–12.

[41] Buades, A., Coll, B., and Morel, J.-M. A non-local algorithm for image denoising. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)* (2005).

[42] Burges, C. J. C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery 2*, 2 (June 1998), 121–167.

[43] Burlet, J., Aycard, O., Spalanzani, A., and Laugier, C. Pedestrian tracking in car parks: an adaptive interacting multiple models based filtering method. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE* (2006), IEEE, pp. 462–467.

[44] Buxton, H. Learning and understanding dynamic scene activity: A review. *Image and Vision Computing 21* (2003), 125–136.

[45] Chen, Y.-T., and Chen, C.-S. Fast human detection using a novel boosted cascading structure with meta stages. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society 17*, 8 (Aug. 2008), 1452–64.

[46] COGNIRON. The cognitive robot companion, 2004. (FP6-IST-002020), http://www.cogniron.org.

[47] Collins, R. T., Lipton, A. J., Fujiyoshi, H., and Kanade, T. Algorithms for Cooperative Multisensor Surveillance. *Proceedings of the IEEE 89* (2001), 1456–1477.

[48] Collins, R. T., Liu, Y., and Leordeanu, M. Online selection of discriminative tracking features. *IEEE transactions on pattern analysis and machine intelligence 27*, 10 (Oct. 2005), 1631–43.

[49] Comaniciu, D., Ramesh, V., and Meer, P. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 5 (2003), 564–577.

[50] Cornelis, N., Cornelis, K., and Van Gool, L. Fast Compact City Modeling for Navigation Pre-Visualization. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)* (Washington, 2006), CVPR '06, IEEE, pp. 1339–1344.

[51] Cox, I. J. A Review of Statistical Data Association Techniques for Motion Correspondence. *International Journal of Computer Vision 10* (1993), 53–66.

[52] Dalal, N., Triggs, B., Rhone-Alps, I., and Montbonnot, F. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005* (2005), In CVPR, pp. 886–893.

[53] Dee, H. M., Fraile, R., Hogg, D. C., and Cohn, A. G. Modelling scenes using the activity within them. In *Proceedings of the International Conference on Spatial Cognition VI* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 394–408.

[54] Del Bue, A., Comaniciu, D., Ramesh, V., and Regazzoni, C. Smart cameras with real-time video object generation. In *Proceedings International Conference on Image Processing* (Rochester, NY, 2002), IEEE Computer Society, pp. 429–432.

[55] Dollar, P., Wojek, C., Schiele, B., and Perona, P. Pedestrian detection: A benchmark. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (June 2009), 304–311.

[56] Dominguez, C., Vidulich, M., Vogel, E., and McMillan, G. Can SA be Defined? In *Situation awareness: Papers and annotated bibliography* (Wright-Patterson Air Force Base, OH: Air Force Systems Command, 1994), Wright-Patterson Air Force Base, OH: Air Force Systems Command, pp. 5–15.

[57] Doucet, A., and Johansen, A. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, December (2009), 4–6.

[58] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. Wiley Interscience, 2001.

[59] Ess, a., Leibe, B., Schindler, K., and van Gool, L. Moving obstacle detection in highly dynamic scenes. *2009 IEEE International Conference on Robotics and Automation* (May 2009), 56–63.

[60] Ess, A., Leibe, B., Schindler, K., and van Gool, L. Robust multiperson tracking from a mobile platform. *IEEE transactions on pattern analysis and machine intelligence 31*, 10 (Oct. 2009), 1831–46.

[61] Ess, a., Schindler, K., Leibe, B., and Van Gool, L. Object Detection and Tracking for Autonomous Navigation in Dynamic Environments. *The International Journal of Robotics Research 29*, 14 (May 2010), 1707–1725.

[62] Farmer, M., and Jain, A. Interacting multiple model (IMM) Kalman filters for robust high speed human motion tracking. In *16th International Conference on Pattern Recognition, 2002* (2002), no. Imm, IEEE Comput. Soc, pp. 20–23.

[63] Faugeras, O. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, 1993.

[64] Felzenszwalb, P., and Huttenlocher, D. Efficient matching of pictorial structures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)* (Hilton Head Island, SC , USA, 2000), IEEE Comput. Soc, pp. 66–73.

[65] Felzenszwalb, P., McAllester, D., and Ramanan, D. A Discriminatively Trained, Multiscale, Deformable Part Model. In *Conference on Computer Vision and Pattern Recognition* (2008), IEEE, pp. 1–8.

[66] Ferrari, V., Tuytelaars, T., and Van Gool, L. Real-time affine region tracking and coplanar grouping. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (2001), IEEE Computer Society, pp. II–226–II–233.

[67] Feyrer, S., and Zell, A. Detection, tracking, and pursuit of humans with an autonomous mobile robot. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on* (2002), vol. 2, IEEE, pp. 864–869.

[68] Fod, A., Howard, A., and Mataric, M. J. Laser-based people tracking. In *In Proc. of the IEEE International Conference on Robotics & Automation* (2002), pp. 3024–3029.

[69] Francois, A. Real-time multi-resolution blob tracking. In *International Conference on Intelligent Autonomous Systems* (2004), Citeseer, pp. 1–10.

[70] Freedman, S., and Adams, J. Improving robot situational awareness through commonsense: Side-stepping incompleteness and unsoundness. Tech. rep., 2009.

[71] Freund, Y., and Schapire, R. A short introduction to boosting. Japanese Society for Artificial Intelligence, 1999.

[72] Freund, Y., and Schapire, R. E. A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. In *Proceedings of the Second European Conference on Computational Learning Theory* (1995).

[73] Frintrop, S., and Kessel, M. Cognitive Data Association for Visual Person Tracking. In *Proc. of the 1st IEEE Workshop on Human Detection from Mobile Platforms (HDMP '08)* (2008), pp. 1–6.

[74] Frintrop, S., Königs, A., Hoeller, F., and Schulz, D. A Component-Based Approach to Visual Person Tracking from a Mobile Platform. *International Journal of Social Robotics 2*, 1 (Dec. 2009), 53–62.

[75] Frintrop, S., Nuchter, A., Surmann, H., and Hertzberg, J. Saliency-based object recognition in 3D data. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* (2005), vol. 3, IEEE, pp. 2167–2172.

[76] Fritsch, J., Kleinehagenbrock, M., Lang, S., Fink, G., and Sagerer, G. Audiovisual person tracking with a mobile robot. In *International Conference on Intelligent Autonomous Systems* (2004), Citeseer, pp. 898–906.

[77] Gammeter, S., Ess, A., Jaeggli, T., Schindler, K., Leibe, B., and Gool, L. Articulated multi-body tracking under egomotion. In *European Conference on Computer Vision* (2008), pp. 816–830.

[78] Gavrila, D. Multi-feature hierarchical template matching using distance transforms. In *Proceedings of the 14th International Conference on Pattern Recognition* (1998), IEEE Comput. Soc, pp. 439–444.

[79] Gavrila, D., and Giebel, J. Shape-based pedestrian detection and tracking. *Intelligent Vehicle Symposium, 2002. IEEE* (2008), 8–14.

[80] Gavrila, D. M., and Munder, S. Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle. *International Journal of Computer Vision 73*, 1 (July 2007), 41–59.

[81] Gerónimo, D., López, A. M., Sappa, A. D., and Graf, T. Survey of pedestrian detection for advanced driver assistance systems. *IEEE transactions on pattern analysis and machine intelligence 32*, 7 (July 2010), 1239–58.

[82] Gibson, J. The perception of the visual world. *Riverside Press* (1950).

[83] Gilbert, A., and Bowden, R. Multi person tracking within crowded scenes. *Lecture Notes in Computer Science 4814* (2007), 166.

[84] Gordon, N., Salmond, D., and Smith, A. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F* (1993), vol. 140, IET, pp. 107–113.

[85] Greiffenhagen, M., Comaniciu, D., Niemann, H., and Ramesh, V. Design, analysis, and engineering of video monitoring systems: an approach and a case study. *Proceedings of the IEEE 89*, 10 (2001), 1498–1517.

[86] Gross, H., Richarz, J., Mueller, S., Scheidig, A., and Martin, C. Probabilistic Multi-modal People Tracker and Monocular Pointing Pose Estimator for Visual Instruction of Mobile Robot Assistants. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings* (2006), IEEE, pp. 4209–4217.

[87] Grubb, G., Zelinsky, A., Nilsson, L., and Rilbe, M. 3D vision sensing for improved pedestrian safety. In *IEEE Intelligent Vehicles Symposium, 2004* (2004), Australien National University, IEEE, pp. 19–24.

[88] Han, B., Comaniciu, D., Zhu, Y., and Davis, L. S. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE transactions on pattern analysis and machine intelligence 30*, 7 (July 2008), 1186–97.

[89] Handmann, U., H, U., Tzomakas, C., Kalinke, T., Werner, M., and Seelen, W. V. Computer Vision for Driver Assistance Systems. In *In Proceedings of SPIE* (Orlando, 1998).

[90] Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2004.

[91] Hayman, E., and Eklundh, J.-O. Statistical background subtraction for a mobile observer. In *Proceedings of the International Conference on Computer Vision* (2003), pp. 67–74.

[92] Hermes, C., Einhaus, J., Hahn, M., Wöhler, C., and Kummert, F. Vehicle tracking and motion prediction in complex urban scenarios. In *Proc. IEEE Intelligent Vehicles Symposium* (San Diego, California, 2010).

[93] Hinz, S., and Stilla, U. Car detection in aerial thermal images by local and global evidence accumulation. *Pattern Recognition Letters 27*, 4 (Mar. 2006), 308–315.

[94] Hopcroft, J., and Tarjan, R. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM 16*, 6 (June 1973), 372–378.

[95] Horn, B. K., and Schunck, B. G. Determining optical flow. *Artificial Intelligence 17*, 1-3 (Aug. 1981), 185–203.

[96] Huang, T. Parametric contour tracking using unscented Kalman filter. In *Proceedings on International Conference on Image Processing* (2002), Ieee, pp. 613–616.

[97] Huang, Y., Qian, X., and Chen, S. Multi-sensor calibration through iterative registration and fusion. *Computer-Aided Design 41*, 4 (Apr. 2009), 240–255.

[98] Hubel, D. The visual cortex of the brain. *Scientific American 5*, 209 (1963), 54–62.

[99] Huber, E., and Kortenkamp, D. Using stereo vision to pursue moving agents with a mobile robot. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation* (1995), IEEE, pp. 2340–2346.

[100] Huhle, B., Jenke, P., and Strasser, W. On-the-fly scene acquisition with a handy multisensor-system. In *Workshop on Dynamic 3D Imaging (Dyn3D)* (2007).

[101] Huhle, B., Schairer, T., Jenke, P., and Strasser, W. Robust non-local denoising of colored depth data. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Time of Flight Camera based Computer Vision (TOF-CV)* (2008).

[102] Huttenlocher, D., Noh, J., and Rucklidge, W. Tracking non-rigid objects in complex scenes. In *1993 (4th) International Conference on Computer Vision* (1993), IEEE Computer Society Press, pp. 93–101.

[103] Intille, S., Davis, J., and Bobick, A. Real-time closed-world tracking. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (San Juan, Puerto Rico, 1997), IEEE Computer Society, pp. 697–703.

[104] Isard, M., and Blake, A. Condensation-conditional density propagation for visual tracking. *International journal of computer vision 29*, 1 (1998), 5–28.

[105] Isard, M., and MacCormick, J. BraMBLe: A Bayesian multiple-blob tracker. In *Eighth IEEE International Conference on Computer Vision* (Vancouver, BC , Canada, 2001), IEEE Computer Society.

[106] Itti, L., Koch, C., and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 11 (November 1998), 1254–1259.

[107] Itti, L., Rees, G., and Tsotsos, J. Models of bottom-up attention and saliency. *Neurobiology of attention 582*, 1980 (2005), 1–11.

[108] Jilkov, V. Survey of maneuvering target tracking. part v: multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems 41*, 4 (Oct. 2005), 1255–1321.

[109] Jin, Y., and Mokhtarian, F. Variational Particle Filter for Multi-Object Tracking. *2007 IEEE 11th International Conference on Computer Vision* (Oct. 2007), 1–8.

[110] Julier, S. J., and Uhlmann, J. K. A New Extension of the Kalman Filter to Nonlinear Systems. In *SPIE* (1997), pp. 182–193.

[111] Jung, B., and Sukhatme, G. S. Real-time Motion Tracking from a Mobile Robot. *International Journal of Social Robotics 2*, 1 (Dec. 2010), 63–78.

[112] Kalman, R. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82 (Series D) (1960), 35 – 45.

[113] Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M. A Stereo Machine for Video-rate Dense Depth Mapping and Its New Applications. In *Proceedings of the 15th Computer Vision and Pattern Recognition Conference* (1996), pp. 196–202.

[114] Kang, S. B., and Szeliski, R. Extracting View-Dependent Depth Maps from a Collection of Images. *International Journal of Computer Vision 58*, 2 (July 2004), 139–163.

[115] Kettnaker, V., and Zabih, R. Bayesian multi-camera surveillance. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)* (Fort Collins, CO , USA, 1999), IEEE Comput. Soc, pp. 253–259.

[116] Kim, D. Y., Yang, E., Jeon, M., and Shin, V. Robust Auxiliary Particle Filter with an Adaptive Appearance Model for Visual Tracking. In *Asian Conference on Computer Vision* (2010), IEEE Computer Society.

[117] Kim, K., Chalidabhongse, T. H., Harwood, D., and Davis, L. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging 11* (2005), 172–185.

[118] Kitt, B., Geiger, A., and Lategahn, H. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *IEEE Intelligent Vehicles Symposium* (2010).

[119] Klappstein, J., Vaudrey, T., Rabe, C., Wedel, A., and Klette, R. Moving object segmentation using optical flow and depth information. In *Proceedings of the Symposium on Advances in Image and Video Technology* (2009), pp. 611–623.

[120] Klein, D., Schulz, D., and Frintrop, S. Adaptive real-time video-tracking for arbitrary objects. In *International conference on Intelligent Robots and Systems* (2010), pp. 772–777.

[121] Koch, C., and Ullman, S. Shifts in selective visual attention: towards the underlying neural circuitry. *Human neurobiology 4*, 4 (Jan. 1985), 219–27.

[122] Koile, K., Tollmar, K., Demirdjian, D., Shrobe, H., and Darrell, T. Activity zones for context-aware computing. In *Proceedings of the International Conference on Ubiquitos Computing* (2003), vol. 2864 of *Lecture Notes in Computer Science*, pp. 90–106.

[123] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S. Multi-camera multi-person tracking for EasyLiving. In *Proceedings Third IEEE International Workshop on Visual Surveillance* (Dublin, Ireland, 2000), IEEE Computer Society, pp. 3–10.

[124] Kuipers, B. The spatial semantic hierarchy. *Artificial Intelligence 119* (1999), 191–233.

[125] Labayrade, R., Aubert, D., and Tarel, J. Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation. In *IEEE Intelligent Vehicle Symposium* (2002), vol. 2, pp. 646–651.

[126] Lamosa, F., and Uchimura, K. A Complete U-V-Disparity Study for Stereovision Based 3D Driving Environment Analysis. *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)* (2005), 204–211.

[127] Lazebnik, S., Schmid, C., and Ponce, J. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006), IEEE, pp. 2169–2178.

[128] Lee, J., Tsubouchi, T., Yamamoto, K., and Egawa, S. People Tracking Using a Robot in Motion with Laser Range Finder. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2006), IEEE, pp. 2936–2942.

[129] Leibe, B., Cornelis, N., Cornelis, K., and Van Gool, L. Dynamic 3D Scene Analysis from a Moving Vehicle. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), IEEE, pp. 1–8.

[130] Leibe, B., Schindler, K., Cornelis, N., and Van Gool, L. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE transactions on pattern analysis and machine intelligence 30*, 10 (Oct. 2008), 1683–98.

[131] Leibe, B., Seemann, E., and Schiele, B. Pedestrian Detection in Crowded Scenes. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), IEEE, pp. 878–885.

[132] Li, P., Zhang, T., and Pece, A. E. Visual contour tracking based on particle filters. *Image and Vision Computing 21*, 1 (Jan. 2003), 111–123.

[133] Lin, Z., and Davis, L. S. A Pose-Invariant Descriptor for Human Detection and Segmentation. In *Proceedings in 10th European Conference on Computer Vision* (Berlin, Heidelberg, Oct. 2008), D. Forsyth, P. Torr, and A. Zisserman, Eds., vol. 5305 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 423–436.

[134] Lohmann, O. Objektrepräsentation mittels visueller Salienz zur Erzeugung gerichteter Aufmerksamkeit, 2009.

[135] Lowe, D. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision* (1999), 1150–1157 vol.2.

[136] Lucas, B., and Kanade, T. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence* (1981), vol. 3, Citeseer, p. 3.

[137] Lütkebohle, I., Peltason, J., Wrede, B., and Wachsmuth, S. The task-state coordination pattern, with applications in human-robot-interaction. *Learning, Planning and Sharing Robot Knowledge for Human-Robot Interaction* (2011).

[138] Ma, G., Park, S.-B., Ioffe, A., Muller-Schneiders, S., and Kummert, A. A Real Time Object Detection Approach Applied to Reliable Pedestrian Detection. In *2007 IEEE Intelligent Vehicles Symposium* (June 2007), IEEE, pp. 755–760.

[139] Mäenpää, T., Ojala, T., Pietikäinen, M., and Maricor, S. Robust texture classification by subsets of local binary patterns. In *15th International Conference on Pattern Recognition* (Barcelona, Spain, 2000), pp. 947–950.

[140] Maggio, E., Smeraldi, F., and Cavallaro, A. Combining Colour and Orientation for Adaptive Particle Filter based Tracking. In *British Machine Vision Conference* (London, 2005).

[141] Maji, S., Berg, A. C., and Malik, J. Classification using intersection kernel support vector machines is efficient. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (June 2008), IEEE, pp. 1–8.

[142] Makris, D., and Ellis, T. Automatic learning of an activity-based semantic scene model. In *Proceedings of the Conference on Advanced Video and Signal Based Surveillance* (2003).

[143] Marc, H., Sebastian, W., Christian, L., and Gerhard, S. Who am i talking with? a face memory for social robots. IEEE, IEEE.

[144] Marroquin, J., Mitter, S., and Poggio, T. Probabilistic Solution of Ill-Posed Problems in Computational Vision. *Journal of the American Statistical Association* (Mar. 1987), 76–89.

[145] May, S., Werner, B., Surmann, H., and Pervolz, K. 3D Time-of-Flight cameras for mobile robotics. In *Intl. Conference on Intelligent Robots and Systems (IROS)* (2006), pp. 790–795.

[146] Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J. Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems 34*, 1 (1998), 103–123.

[147] Migliore, D., Rigamonti, R., Marzorati, D., and M. Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments. In *ICRA09 Workshop on Safe navigation in open and dynamic environments Application to autonomous vehicles* (2009), IEEE Comput. Soc.

[148] Mikolajczyk, K., Leibe, B., and Schiele, B. Multiple Object Class Detection with a Generative Model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1* (June 2006), IEEE, pp. 26–36.

[149] Mikolajczyk, K., Schmid, C., and Zisserman, A. Human Detection Based on a Probabilistic Assembly of Robust Part Detectors. In *Proceedings of European Conference on Computer Vision* (Berlin, Heidelberg, 2004), vol. 3021 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 69–82.

[150] Mittal, A., Monnet, A., and Paragios, N. Scene modeling and change detection in dynamic scenes: A subspace approach. *Computer Vision and Image Understanding 113*, 1 (2009), 63–79.

[151] Montello, D. R. Scale and multiple psychologies of space. In *Lecture Notes in Computer Science: Spatial Information Theory A Theoretical Basis for GIS* (1993), vol. 716, pp. 312–321.

[152] Musleh, B., Escalera, A. D. L., and Armingol, J. Obstacle Detection and Localization Using U-V disparity with Accelerated Processing Time by CUDA. In *IEEE International Conference on Robotics and Automation, 2011, ICRA* (2011).

[153] Naeem, A., Pridmore, T., and Mills, S. Managing particle spread via hybrid particle filter/kernel mean shift tracking. In *Proceedings of the British Machine Vision Conference* (University of Warwick, 2007).

[154] Nagai, Y., Muhl, C., and Rohlfing, K. J. Toward designing a robot that learns actions from parental demonstrations. In *The 2008 IEEE International Conference on Robotics and Automation* (Pasadena, CA, USA, 19/05/2008 2008), pp. 3545–3550.

[155] Navalpakkam, V., and Itti, L. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006* (2006), vol. 2, IEEE, pp. 2049–2056.

[156] Okutomi, M., and Kanade, T. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 4 (Apr. 1993), 353–363.

[157] Oliva, A., Torralba, A., Castelhano, M., and Henderson, J. Top-down control of visual attention in object detection. In *International Conference on Image Processing* (2003).

[158] Oprisescu, S., Falie, D., Ciuc, M., and Buzuloiu, V. Measurements with tof cameras and their necessary corrections. In *Intl. Symposium on Signals, Circuits & Systems (ISSCS)* (2007).

[159] Papageorgiou, C., and Poggio, T. Trainable pedestrian detection. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on* (1999), vol. 4, IEEE, pp. 35–39.

[160] Perrone, J., Voyle, T., and Jefferies, M. Towards a Human Tracking System for a Mobile Robot Using Neural-Based Motion Detectors. In *Image and Vision Computing New Zealand* (2003), Citeseer, pp. 24–29.

[161] Peters, A., Weiss, P., and Hanheide, M. Avoid me: a spatial movement concept in human-robot interaction. *Cognitive Processing 10* (09/2009 2009), S178. Abstract in doctoral colloquium.

[162] Peursum, P., Venkatesh, S., West, G., and Bui, H. H. Using interaction signatures to find and label chairs and floors. *Pervasive Computing 3*, 4 (2004), 58–65.

[163] Pitt, M. K., and Shephard, N. Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association 94*, 446 (June 1999), 590.

[164] Premebida, C., Ludwig, O., and Nunes, U. LIDAR and vision-based pedestrian detection system. *Journal of Field Robotics 26*, 9 (2009), 696–711.

[165] Rasmussen, C., and Hager, G. Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 6 (June 2001), 560–576.

[166] Rasolzadeh, B., Bjoerkman, M., and Eklundh, J.-O. An Attentional System Combining Top-Down and Bottom-Up Influences. In *WAPCV International workshop on attention in cognitive systems* (2007), vol. 1, pp. 123–140.

[167] Rolf, M., Hanheide, M., and Rohlfing, K. Attention Manipulation with Multimodal Synchrony. In *IEEE-RAS Int. Conf. on Humanoid Robots, Workshop on "Social Learning in Interactive Scenarios"* (2009).

[168] Ruesch, J., Lopes, M., Bernardino, A., Hoernstein, J., Santos-Victor, J., and Pfeifer, R. Multimodal saliency-based bottom-up attention a framework for the humanoid robot icub. In *IEEE International Conference on Robotics and Automation* (2008).

[169] Sanders, B. C. S., Nelson, T. C., and Sukthankar, R. A theory of the quasi-static world. In *Proceedings of the International Conference on Pattern Recognition* (2002), vol. 3, pp. 1–6.

[170] Scandaliaris, J., and Sanfeliu, A. Discriminant and Invariant Color Model for Tracking under Abrupt Illumination Changes. *2010 20th International Conference on Pattern Recognition* (Aug. 2010), 1840–1843.

[171] Scheidig, A., Mueller, S., Martin, C., and Gross, H. M. Generating Persons Movement Trajectories on a Mobile Robot. In *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication* (Sept. 2006), IEEE, pp. 747–752.

[172] Scheutz, M., McRaven, J., and Cserey, G. Fast, reliable, adaptive, bimodal people tracking for indoor environments. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)* (2004), IEEE, pp. 1347–1352.

[173] Schiele, B., Andriluka, M., Majer, N., Roth, S., and Wojek, C. Visual People Detection-Different Models, Comparison and Discussion. In *Workshop on People Detection and Tracking, 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan* (2009), no. May.

[174] Schiller, I., Beder, C., and Koch, R. Calibration of a pmd camera using a planar calibration object together with a multi-camera setup. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2008), vol. 37 Part B3a, pp. 297–302.

[175] Schlegel, C., Illmann, J., Jaberg, H., Schuster, M., and Worz, R. Vision based person tracking with a mobile robot. In *British Machine Vision Conference* (1998), vol. 3, pp. 418–427.

[176] Schmidt, J., Wöhler, C., Krüger, L., Gövert, T., and Hermes, C. 3D scene segmentation and object tracking in multiocular image sequences. In *Proceedings of the International Conference on Computer Vision Systems* (2007).

[177] Schmüdderich, J., Willert, V., Eggert, J., Rebhan, S., Goerick, C., Sagerer, G., and Körner, E. Estimating object proper motion using optical flow, kinematics, and depth information. *IEEE Transactions on Systems Man and Cybernetics 38* (08/2008 2008), 1139–1151.

[178] Schreiber, D. Robust template tracking with drift correction. *Pattern Recognition Letters 28*, 12 (Sept. 2007), 1483–1491.

[179] Schulz, D., Burgard, W., Fox, D., and Cremers, A. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (2001), vol. 2, IEEE, pp. 1665–1670.

[180] Schwartz, W., Kembhavi, A., Harwood, D., and Davis, L. Human Detection Using Partial Least Squares Analysis. In *International Conference on Intelligent Vision (ICCV)* (2009).

[181] Shashua, A., Gdalyahu, Y., and Hayun, G. Pedestrian Detection for Driving Assistance Systems: Single-frame Classification and System Level Performance. In *Intelligent Vehicles Symposium, 2004 IEEE* (2004), IEEE, pp. 1–6.

[182] Sheikh, Y., Javed, O., and Kanade, T. Background subtraction for freely moving cameras. In *12th International Conference on Computer Vision, 2009 IEEE* (2010), IEEE, pp. 1219–1225.

[183] Sheikh, Y., and Shah, M. Bayesian modeling of dynamic scenes for object detection. *Transactions on Pattern Analysis and Machine Intelligence 27*, 11 (2005), 1778–1792.

[184] Shet, V. D., Neumann, J., Ramesh, V., and Davis, L. S. Bilattice-based Logical Reasoning for Human Detection. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), IEEE, pp. 1–8.

[185] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. Real-time human pose recognition in parts from single depth images. In *In Proceedings of Computer Vision and Pattern Recognition* (2011).

[186] Sivic, J., Zitnick, C. L., and Szeliski, R. Finding people in repeated shots of the same scene. In *Proceedings of the British Machine Vision Conference* (2006).

[187] Sotelo, M., Parra, I., Fernandez, D., and Naranjo, E. Pedestrian Detection Using SVM and Multi-Feature Combination. In *2006 IEEE Intelligent Transportation Systems Conference* (2006), IEEE, pp. 103–108.

[188] Stauffer, C., and Grimson, E. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition* (1999), pp. 246–252.

[189] Stauffer, C., and Grimson, W. E. L. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1999), pp. 246–252.

[190] Sturm, J., Konelige, K., Stachniss, C., and Burgard, W. Vision-based detection for learning articulation models of cabinet doors and drawers in household environments. In *Proceedings of the International Conference on Robotics and Automation* (2010).

[191] Sturm, J., Predeep, V., Stachniss, C., Plagemann, C., Konolige, K., and Burgard, W. Learning kinematic models for articulated objects. In *Proceedings of the International Joint Conference on Artificial Intelligence* (2009), pp. 1851–1856.

[192] Suganuma, N. Clustering and tracking of obstacles from Virtual Disparity Image. In *2009 IEEE Intelligent Vehicles Symposium* (June 2009), IEEE, pp. 111–116.

[193] Sun, Y., and Fisher, R. Object-based visual attention for computer vision. *Artificial Intelligence 146*, 1 (May 2003), 77–123.

[194] Swadzba, A., Beuter, N., Schmidt, J., and Sagerer, G. Tracking objects in 6D for reconstructing static scenes. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (June 2008), 1–7.

[195] Swadzba, A., Beuter, N., Wachsmuth, S., and Kummert, F. Dynamic 3D scene analysis for acquiring articulated scene models. In *International Conference on Robotics and Automation (ICRA), 2010* (2010), IEEE, pp. 134–141.

[196] Swadzba, A., Liu, B., Penne, J., Jesorsky, O., and Kompe, R. A comprehensive system for 3D modeling from range images acquired from a 3d tof sensor. In *Proceedings of the International Conference on Computer Vision Systems* (2007).

[197] Swadzba, A., and Wachsmuth, S. Categorizing perceptions of indoor rooms using 3D features. In *Lecture Notes in Computer Science: Structural, Syntactic, and Statistical Pattern Recognition* (2008), vol. 5342, pp. 744–754.

[198] Szeliski, R. *Computer Vision*. Springer, 2010.

[199] Thrun, S., Burgard, W., and Fox, D. *Probabilistic robotics*. MIT Press, 2005.

[200] Tran, D., and Forsyth, D. Configuration estimates improve pedestrian finding. In *Neural Information Processing Systems (NIPS)* (2007), MIT Press, Cambridge, MA, pp. 1529–1536.

[201] Treisman, A., and Gelade, G. A feature-integration theory of attention. *Cognitive Psychology 12*, 1 (1980), 97–136.

[202] Tuzel, O., Porikli, F., and Meer, P. Human Detection via Classification on Riemannian Manifolds. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), IEEE, pp. 1–8.

[203] Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T. Three-dimensional scene flow. *IEEE transactions on pattern analysis and machine intelligence 27*, 3 (Mar. 2005), 475–80.

[204] Viola, P., and Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *IEEE Conference on Computer Vision and Pattern Recognition* (2001), IEEE, pp. 511–518.

[205] Viola, P., Jones, M. J., and Snow, D. Detecting Pedestrians Using Patterns of Motion and Appearance. In *Proceedings of the Ninth IEEE International Conference on Computer Vision* (Washington, DC, Oct. 2003), IEEE Computer Society, p. 734.

[206] Wan, E. A., and Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000* (2002), 153—-158.

[207] Wang, W., Zhang, J., and Shen, C. Improved human detection and classification in thermal images. In *2010 IEEE International Conference on Image Processing* (Sept. 2010), IEEE, pp. 2313–2316.

[208] Wang, X., Tieu, K., and Grimson, E. Learning semantic scene models by trajectory analysis. In *Proceedings of the European Conference on Computer Vision* (2006), vol. 3953 of *lncs*, pp. 110–123.

[209] Weingarten, J., Gruener, G., and Siegwart, R. A state-of-the-art 3D sensor for robot navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems* (2004), vol. 3, pp. 2155–2160.

[210] Welch, G., and Bishop, G. An introduction to the Kalman filter. *University of North Carolina at Chapel Hill, Chapel Hill, NC* (1995).

[211] Woehler, C. *3D Computer Vision: Efficient Methods and Applications*. Springer Berlin / Heidelberg, 2009.

[212] WOEHLER, C., AND ANLAUF, J. K. An adaptable time-delay neural-network algorithm for image sequence analysis. In *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* (Jan. 1999), vol. 10, pp. 1531–1536.

[213] WOEHLER, C., KRESSEL, U., AND ANLAUR, J. Pedestrian recognition by classification of image sequences - global approaches vs. local spatio-temporal processing. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000* (2000), IEEE Comput. Soc, pp. 540–544.

[214] WOLFE, J. M. Visual search in continuous, naturalistic stimuli. *Vision Research 34*, 9 (May 1994), 1187–1195.

[215] WREDE, S., FRITSCH, J., BAUCKHAGE, C., AND SAGERER, G. An XML based framework for cognitive vision architectures. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* (2004), 757–760 Vol.1.

[216] WREN, C., AZARBAYEJANI, A., DARRELL, T., AND PENTLAND, A. Pfinder: real-time tracking of the human body. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition* (1996), IEEE Computer Society, pp. 51–56.

[217] WREN, C., AZARBAYEJANI, A., DARRELL, T., AND PENTLAND, A. Pfinder : Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 7 (1997), 780 –785.

[218] WU, B., AND NEVATIA, R. Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (June 2008), IEEE, pp. 1–8.

[219] XING, J., AND HEEGER, D. J. Center-surround interactions in foveal and peripheral vision. *Vision Res 40*, 22 (2000), 3065–3072.

[220] YU, Y., MANN, G. K. I., AND GOSINE, R. G. An object-based visual attention model for robots. IEEE International Conference on Robotics and Automation, 2008, pp. 943–948.

[221] YUAN, F., SWADZBA, A., PHILIPPSEN, R., ENGIN, O., HANHEIDE, M., AND WACHSMUTH, S. Laser-based navigation enhanced with 3D time-of-flight data. In *Proceedings of the International Conference on Robotics and Automation* (2009), pp. 2844–2850.

[222] ZEKI, S. The functional organization of projections from prestriate visual cortex in rhesus monkey. In *Cold Spring Harbour Symposium on Quantitative Biology* (1976), pp. 591–600.

[223] ZHU, Q., AVIDAN, S., YEH, M., AND CHENG, K. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)* (2006), IEEE, pp. 1491–1498.