AAL-2009-2-049, ALIAS
D6.2
Collection of several approaching strategies for a robot platform in an ambient assisted living environment

Due Date of Deliverable	2011-03-31
Actual Submission Date	2011-06-09
Workpackage:	6.2
Dissemination Level:	Public
Nature:	Report
Approval Status:	Preliminary
Version:	v0.3
Total Number of Pages:	41
Filename:	D6.2-IUT-UseCases-v1.0.pdf
Keyword list:	approaches to follow and guide a person, observe a resting person, path planning

Abstract
This deliverable provides an overview of methods to follow a person, guide a person, and it presents also approaches to path planning while respecting persons inside the home environment. Additionally, it presents an short overview on how a good observation position is found inside the home environment. From all aspects, the best suited are selected to be used within ALIAS.

History

Version	Date	Reason	RevisedBy
0.1	2011-04-10	created [IUT]	Jens Kessler
0.2	2011-05-24	address comments of internal review [IUT]	Jens Kessler
0.3	2011-06-08	address comments of external review [IUT]	Jens Kessler

Authors

Partner	Name	Phone / Fax / Email
IUT	Jens Kessler	Tel: +49 36 77 691305 Fax: +49 36 77 691665 Email: jens.kessler@tu-ilmenau.de

Table of Contents

1 Executive summary

This document is one of a series of deliverables, which describes the navigational part of the ALIAS project. The navigation within ALIAS focuses on "socially acceptable navigation", which means that the human being and the robot, both as social entities, should react to each other in natural ways, and that especially the robot treats a detected person not only as an obstacle, but applies psychological rules in such cases.

While the first deliverable D6.1 covers the methods of approaching a person and the psychological background, this deliverable provides information on the state of the art for all other navigation tasks within ALIAS. We show the current state of research on four tasks, which the robot has to provide. After the investigation of the state of the art for each task, a short conclusion is drawn and a plan for the provision of the requested functionality in ALIAS is sketched.

This deliverable covers theoretical research work. Experimental results and details on implementation will be included in the next deliverables.

2 Introduction

Among the capabilities a robot has to fulfill, not only the approaching of a person is required to enable a natural behavior inside the home environment. There are a number of additional behaviors to interact in an socially acceptable way with a single person. This deliverable describes these additional approaches by observing the state of the art towards different navigation tasks and also by providing first ideas (or strategies) to adapt these approaches to our navigation framework. The approaches have to be sufficient and, when no sufficient approach is known to solve a certain task, we draw a short sketch on how we plan to solve these task.

So the current document will not deal only with another set of approaching strategies, since this was done in detail in deliverable D6.1 [19]. Here, more focus is laid on the home environment itself where the robot has to assist the person living inside the home. Assistance means not only to approach a static sitting person but to fulfill more home-centered tasks. These tasks consider, that a person is present inside the operational area. Else, socially acceptable navigation makes no sense. The navigational tasks are defined in the next section.

2.1 The robots tasks

This section covers the navigation tasks the robotic system will provide to the end user. The task of approaching a sitting or standing person will not be mentioned, since this was part of deliverable 6.1 "‘Report on different navigation strategies to approach elderly people in a polite manner"’. This deliverable reports on the ideas to solve the tasks of:

- following a walking person
- guiding a walking person to a place within a flat
- planning paths by respecting (walking) persons
- observing a sitting person from a non-intrusive distance

2.1.1 Following a walking person

There are some scenarios within the ALIAS project, which require the robot to follow the care recipient or owner. If, for example, the user wants to establish a video conference call, a chat or even a picture slide show, this would normally not happen where the robot is positioned at that particular moment, but where the user finds it a pleasure to do it. The robot cannot know this place in advance, since more than one position is possible within the home. So the robot has to follow the user until the user has reached that place.

The task of following is considered to take place in a static home environment, where the user is the only dynamic "environmental change" and person present. The main goal of this task is to keep the robots position within a personal space of 0.45 to 1.2 meters from the user(see Hall [14]), so the user recognizes that the robot is still interacting with him and the robot is still able to track the person's position. The last point is more or less a pragmatic technical issue and relies on the person tracking system capabilities. The main problem will be the modeling of the appropriate personal space configuration.

2.1.2 Guiding a walking person

The guiding of a person is more or less the opposite configuration of the following task. Here the robot should guide the person to a known location because it is dark and the person has to be protected from falling or other potential hazardous scenarios.

2.1.3 Path planning with respect to persons

Whenever the robot drives autonomously in the home environment without directly interacting with persons, i.e. when driving to the charging station or driving to an observation place, the robot should recognize and respect persons in the operational area. The task is here to find a non-disturbing path to drive to a goal, which goes beyond the goal of finding the shortest path to a target.

2.1.4 Observing a sitting person

The last task the robot should perform is the observation of a sitting person. Here the psychological goal is that the robot should select a location, where the person feels observed, but not disturbed. The technical goal is also to find a place, where the robot has a large viewing angle towards the person. So when the person moves a little bit, the robot only has to rotate. This should enable the robot to behave like a waiting butler to receive orders from the owner.

2.2 Deliverable structure

The deliverable is structured as follows: in the first two chapters the summary of the deliverable content is shown . In the next chapter 3 the topic of following a person is discussed in detail. Chapter 4 describes how to guide a person. While these chapters deal only with motion planning in a rather short surrounding of the robot the next tasks are more challenging and deal with planning tasks in a global scale. So in Chapter 5 the path planning with respect to persons is described, while in Chapter 6 the task of finding an appropriate observation position is discussed. The deliverable is finalized with a conclusions and discussions in Chapter 7.

3 Following a walking person

This chapter provides an overview of strategies and methods to enable a mobile system to follow a walking person. To keep in mind: to follow a person, the robot has to detect that person first. Issues, such as false detections, lost person hypotheses and detection noise all have an influence on the quality of following a person. Here only the approaches of the "following strategy" are shown and the person hypothesis and identity are assumed to be known and stable. The problem of how a person is detected and how the position is tracked over time is not considered here, but will be addressed in forthcoming deliverables.
This chapter focuses on the state of the art of following a person and will also describe, how the robot should follow a person within the ALIAS project.

3.1 State of the art

When searching the state of the art on the topic of following a walking person one can see that this field has been studied extensively over the past ten years. Here one major group of approaches to follow a person can be identified, which tries to handle the follow-problem by designing standard controllers on different controlling variables to steer the robot motors. A second group examines the problem of approaching a person by using environment modeling techniques and that create a set of motion actions on the base of these models. Here again, different methods to create actions are possible. Both, the controller-based approaches and the model-based approaches are described here.

3.1.1 Reactive controller approaches

Visual Servoing: A huge set of methods use the visual input queue of 2D cameras to follow a person. The main tasks of these approaches is the detection of persons in the image or image sequence. With the found person certain controlling strategies are applied.
For example in [26] the method of optical flow is used to extract the body silhouette. By using the gray values $E(x, y)$ and minimizing $e(u, v) = \sum_{x,y}(E_x(x, y) \cdot u + E_y(x, y) \cdot v + E_t(x, y))^2$ with $E_x = \frac{\partial E}{\partial x}$, $E_y = \frac{\partial E}{\partial y}$ and $E_t = \frac{\partial E}{\partial t}$ within a window inside the image (see img. 3.1), the optical flow is estimated. From the found flow-vectors a threshold image is extracted and the person silhouette is created. By measuring the mean shoulder height W_{img} of the person and knowing the focal length f of the used camera lens and the mean shoulder width W of adult persons , the distance d of a person can be estimated according to:

$$d = \frac{W \cdot f}{W_{img}} \tag{3.1}$$

With the known physical person distance, the translational- and rotational speed can be controlled using a standard proportional controller:

$$
\begin{aligned}
V_{trans} &= G \cdot (d - d_{controlled}) \\
V_{rot} &= \alpha \cdot \arctan(\frac{\epsilon}{f})
\end{aligned}
$$

Here G and α are the proportional constant for the controller and ϵ is the difference from the person center to the image center in pixels.

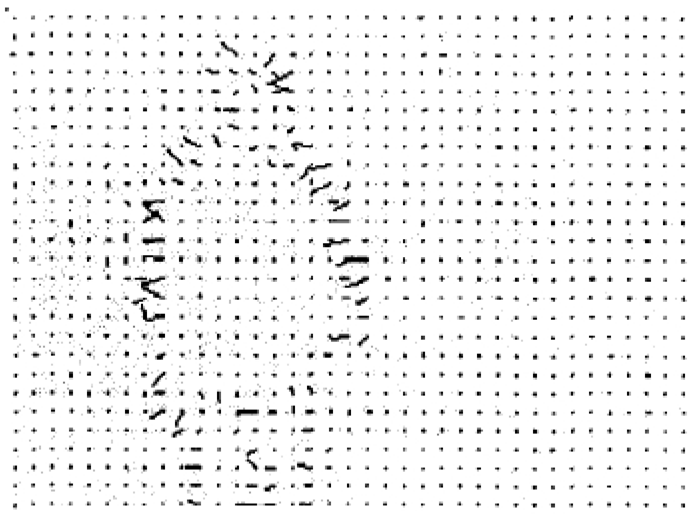


Figure 3.1: The result optical flow image of a standing robot observing a person. From the flow vectors the shoulder width is extracted and the robot is controlled towards a target shoulder width.

Another approach on visual following is presented in [22]. Here the author uses elliptical head tracking with a stereo system to estimate the head size d_{head} and position of the person relative to the robot. Also a standard proportional and differential (PD) controller is used to steer the robot. [22] notes that especially the following of a person is challenging when the person cannot be observed anymore. In such a case all controlling behaviors will fail, since the current controlling variables like d_{head} or W_{img} are not measurable. Reactive behaviors will not work, when the basic information they should react upon is not measurable.

Despite that fact, many authors refined the task of following a person by improving the person tracking task. In [7] Chen describes the selection of Lucas-Kanade[36] image features over time from a stereo vision system, using the RANSAC segmentation to select the corresponding features (see Fig. 3.2). From these features the mean disparity d and again the horizontal distance from the persons center and the image center p are used to control the robot (see equations 3.2 and3.3).

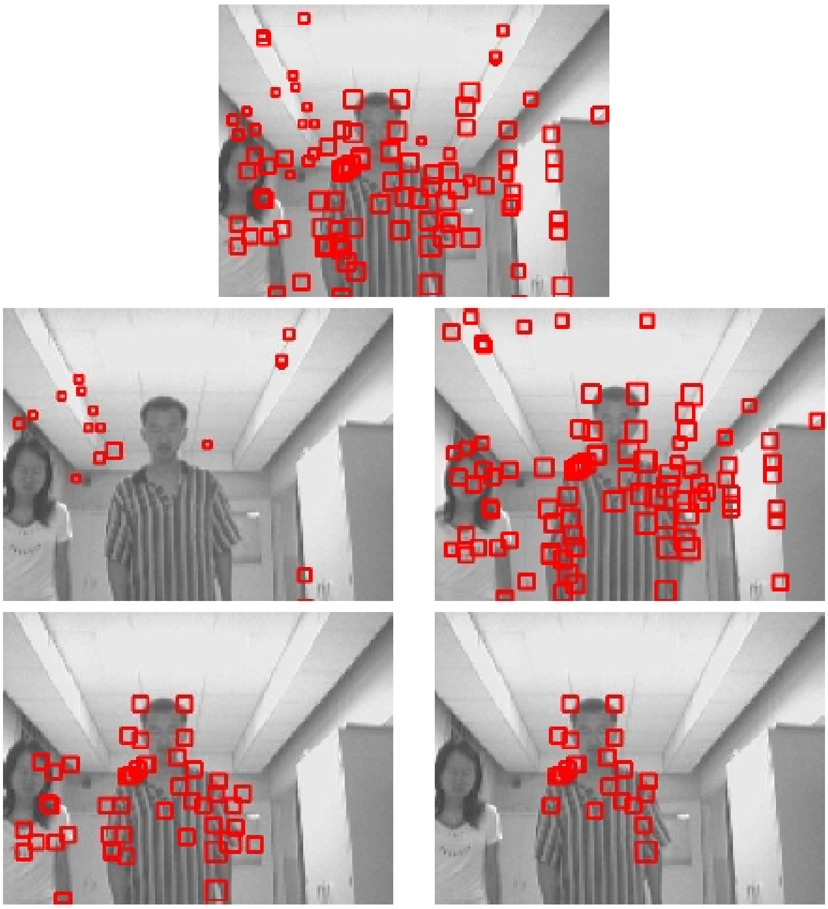


Figure 3.2: Image feature filtering over disparity, space and time. Here features, which don not fit into a motion model of the person (regarding disparity and place, projected over time) are discarded and those remaining are meant to represent the person.

$$V_{trans} = K_1 \cdot 1/d \tag{3.2}$$
$$V_{rot} = K_2 \cdot p/d \tag{3.3}$$

Apart from the standard controller approaches exist some more sophisticated controller approaches, using for example of fuzzy set [17] to control the robot. Hu proposed in [17] the use of such controllers. With a definition of fuzzy sets, fuzzy rules and inference he was able to match "human-like" expressions, what the robot should do when the difference to the controlled values is *zero, negative normal, normal, negative big* or *big*. The controlled properties are the deviations from the height difference Δh of an upper body model to 200 Pixels, created from a color histogram, and the deviation from the center of the upper body model Δh_c. The fuzzy sets are defined by using a triangular function $\mu(x)$ with a center x_0 and a width of $2 \cdot \sigma$.

$$\mu(x) = \begin{cases} 1 - |x - x_0|/\sigma & x \in \langle x - \sigma, x + \sigma \rangle \\ 0 & else \end{cases} \tag{3.4}$$

A complex set of rules are defined to map the fuzzified information in Δh and Δh_c into robot speeds. With inference, a single control command is calculated. As seen in Fig. 3.3, the parameters of the controller (not the fuzzy rules) are additionally optimized with a swarm particle filter.

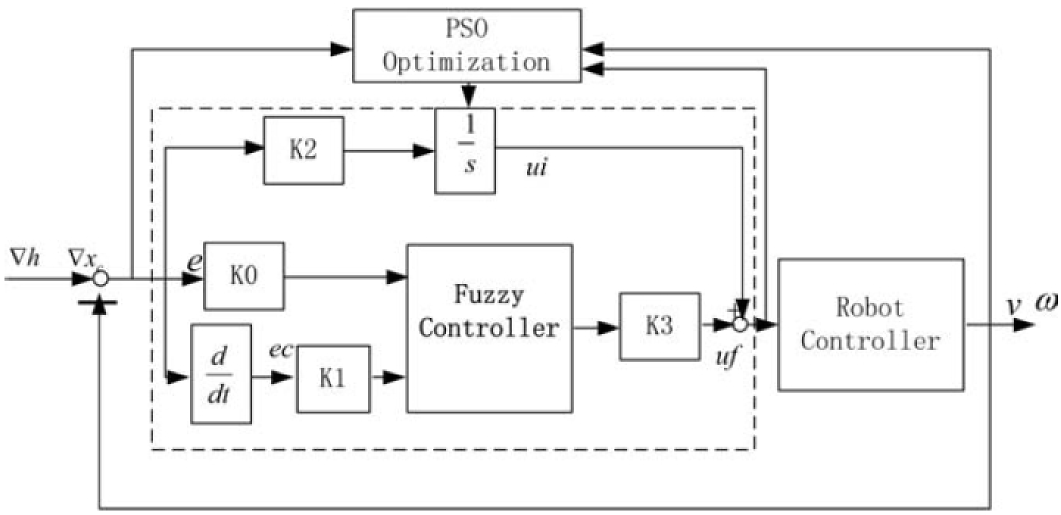


Figure 3.3: The structure of the fuzzy controller with corresponding parameter selection unit. The unit optimizes the gain of the controller with a particle swarm. The parameters K0, K1, K2, K3 are learned by the optimization process. The fuzzy rules are fixed and not changed by the swarm optimization.

The list of approaches that try to control the robot movements by controlling some parameters from person detection could be lengthened a lot. However the impression remains, that all approaches are equal each other in designing a controller. If the person is within sensor range, the robot tries simply to keep at a certain distance by acting with its controller upon several values. These values differ among the approaches. Collision avoidance or obstacle situation is never kept into account within our recherche. The main focus is always the detection of a person. By doing so, view direction is also never concerned when following a person.

Laser based controllers Laser based controllers have almost the same idea than visiual servoing. The benefit is that a laser scanner can measure the the distance of a person with very high accuracy. The methods, to detect a human (and in most cases only a pair of legs from a human), differ from the methods used by the visual queue. There are several methods in the literature to detect legs inside laser range scans [2, 29, 31]. The results are leg-pair hypotheses (see Fig. 3.4) that can directly be used to apply linear controlling strategies, such as above and in [11, 24]. In [24] the laser hypothesis is used to control the translational velocity, while a visual channel is used to control the rotational velocity, as presented in [17].

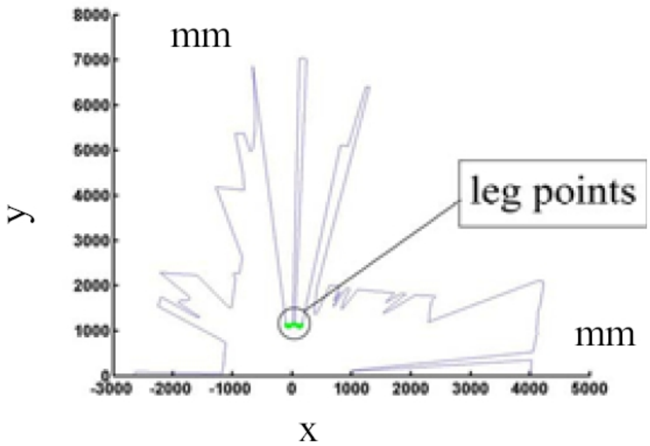


Figure 3.4: Image from [24]. Here, a laser scan, including two legs, is shown. The mean position of both legs is used to control the robot.

3.1.2 Explicit Modeling approaches

Despite the pure controller based approaches, a few more sophisticated approaches exist. Common to them is the usage of additional environment knowledge to build an explicit environment model and use this model to calculate the next driving command or a sequence of such commands. Most of these approaches are physically inspired, but there are also approaches which are rule-based or which in addition to those, use planning and optimization techniques to steer the robot. We present in this section a selection of each type of approach.

Physical approaches: One of the basic concepts to control a robot is to model the robot as a physical entity, which interacts in a physical environment by applying physical forces to the robot. The robot has to follow these forces and applies the motion command by investigating the strength and direction of the resulting force on itself. Here, we show four different approaches on modeling such forces.

The first approach is known as the "potential field" model [35]. Commonly, obstacles as well as a point-like robot, are positively charged, so forces will lead to increase the distance towards obstacles. The goal, for example a person, has a negative charge, and the robot is dragged towards it. An example is shown in Fig. 3.5 b). Here, the main force towards the goal is modeled as follows:

$$\vec{F}_{goal}(x,y) = \alpha \cdot \frac{\vec{G}(x,y) - \vec{R}(x,y)}{d^2} \tag{3.5}$$

Here, $\vec{G}(x,y)$ is the current person's position, while $\vec{R}(x,y)$ is the current robot's position. The scalar d is the Euclidean distance between $\vec{G}(x,y)$ and $\vec{R}(x,y)$. The obstacles are measured with a laser range finder. Each obstacle is a point in the environment which also creates a force towards the robot:

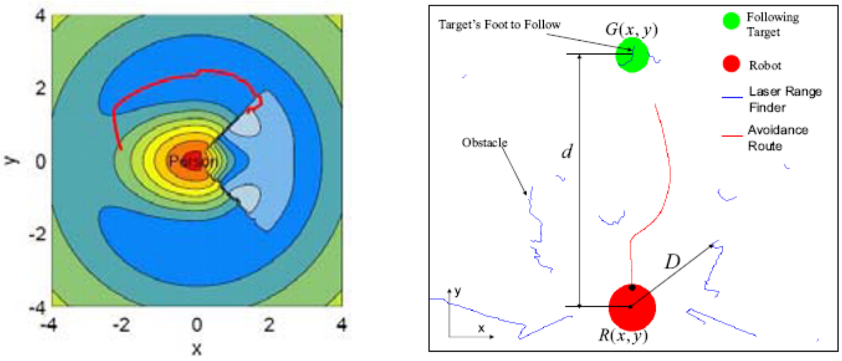
$$\vec{F}_{obst}(x,y) = \beta \cdot \sum_{i=1}^{N} \frac{\vec{R}(x,y) - \vec{O}_i(x,y)}{D_i^2} \tag{3.6}$$

The factors α and β are experimentally optimied and are critical to choose, since they can lead the robot to get stuck in local minima (at narrow spaces) or lead the robot to hit an obstacle.

Now $\vec{O}_i(x,y)$ is the position of the obstacle point and D_i the corresponding Euclidean distance between the obstacle point and the robot. The driving direction and speed of the robot can be computed by direct vector summarization of $\vec{F}_obst(x,y)$ and $F_{goal}(x,y)$. An example trajectory is shown in Fig. 3.5 b).

Svenstrup[34] and Andersen[1] created the potential field in a more complex way. Their potential field is a direct model of the personal space and they use a sum of Gaussians, modulated by the interaction interest i of the person:

$$\begin{aligned}
A(x,y) &= \alpha \cdot e^{-\left(\frac{(x-x_c)^2}{\sigma_{att_x}^2} + \frac{(y-y_c)^2}{\sigma_{att_y}^2}\right)} && (3.7)\\
B(x,y) &= \beta \cdot e^{-\left(\frac{(x-x_c)^2}{\sigma_{rear_x}^2} + \frac{(y-y_c)^2}{\sigma_{rear_y}^2}\right)} && (3.8)\\
C(x,y) &= \gamma \cdot e^{-\vec{x}^T \Sigma(\theta_1)^{-1} \vec{x}} + e^{-\vec{x}^T \Sigma(\theta_2)^{-1} \vec{x}} && (3.9)
\end{aligned}$$



a) b)

Figure 3.5: Two different methods of modeling a potential field. While in b) from [35] each point is seen as a radial positive charge (like the robot), in a) from [1, 34] the potential function is more complex.

$$\Sigma(\theta) = \begin{bmatrix} \sigma_x^2 & \left(\frac{tan(2\theta)\cdot(\sigma_x^2-\sigma_y^2)}{2}\right) \\ \left(\frac{tan(2\theta)\cdot(\sigma_x^2-\sigma_y^2)}{2}\right) & \sigma_y^2 \end{bmatrix} \tag{3.10}$$

Here, x_c and y_c denotes the center of the person, while $\sigma_{att_x}, \sigma_{att_y}$ are the variances of the attraction force, σ_{rear_x} and σ_{rear_y} are the variances of the repulsive force and $\Sigma(\theta_i)$ is a covariance matrix to model the approaching angle θ_i. The angles θ_1 and θ_2 are chosen to be perpendicular. The angle θ_1 is a function of the person interest to interact. If the person interest is high the robot should approach from the front, so $\theta_1 = 0$ and when the interest is low, the robot should approach more from the side.

In all positions $\vec{x} = (x, y)^T$ the Gaussians $A(x, y)$ and $B(x, y)$ are summed up to form a difference of Gaussian function. Only in the space of $\pm 45°$ relative to the persons view direction also the Gaussian $C(x, y)$ is added. Since the potential function is quite complex, the robots steering direction could only be derived from a numerical solution of the tangent inside this function. A resulting trajectory is shown in Fig. 3.5 a).

A third approach is presented by Luo[23], where a simple mass-spring model is applied, to follow the person. The spring issues two forces, a lateral force $\vec{F_1}$ and a perpendicular force $\vec{F_2}$. Both have linear relationship towards the distance and angle of the person to follow, defined by the spring constants k_1 and k_2 (see Fig. 3.6 a)). As seen in Fig. 3.6 a), the distance d from robot to person and the angular deviation ϕ from person and robot view direction is used to calculate the forces:

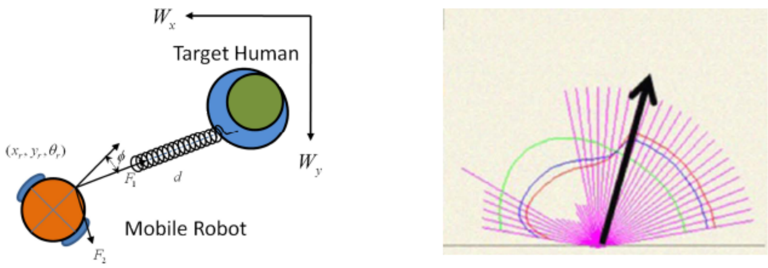
$$\vec{F_1} = k_1 \cdot d \tag{3.11}$$
$$\vec{F_2} = k_2 \cdot \phi \tag{3.12}$$

With the mass of the robot m, the moment of intertia i, the robot's wheel distance L, the translational speed $\vec{V}_{trans}$, and the rotational speed $\vec{V}_{rot}$ the following dynamic system can be used to update the robot speed:

$$m\dot{\vec{V}}_{trans} = -F_1 \cdot cos(\phi) - F_2 \cdot sin(\phi) - k_3 \cdot \vec{V}_{trans} \quad (3.13)$$

$$i\dot{\vec{V}}_{rot} = (F_1 \cdot sin(\phi) + F_2 \cdot cos(\phi) - k_4 \cdot \vec{V}_{rot})L \quad (3.14)$$

Here the constants k_3 and k_4 are chosen by the designer and denote the inertia of the robot. The fourth approach [38] describes a combination of two artificial force fields to create the current driving direction. One force field is directly created from laser range data. The distance per angle is used to create the "force" in the angular direction. A free space creates a high force, whereas an obstacle creates a low force towards this direction. These forces follow no physical rule, but are defined artificially (the author shows no details here). A second force field is generated from the direction of the person to follow, which comes from a person tracker. Here, high values are generated when the direction matches best, while low values are generated when the direction does not match. Both force fields are combined by weighted sums and the angle of the maximum force is used as next steering angle. An example situation snapshot is shown in Fig. 3.6 b).



a) b)

Figure 3.6: Model of a mass-spring system a) [23], where two forces are used to update the dynamic motion model of the robot. In b) [38] a force field model is presented, which combines angular forces to reach the goal with forces from obstacles. The obstacle force field is shown in red and is derived from the laser scanner, the steering force field is shown in green. Both force fields are combined to a resulting force field, shown in blue.

Rule-based approach: A completely different method is presented in [37]. Here three behaviors are modeled and a rule set for switching between the behaviors is discussed. The three behaviors are as follows:

- *Direction following:* The simplest behavior to follow a person. The robot always tries to drive into the direction, the person is detected at.

- *Path following:* The robot tracks the trajectory of the person and exracts way points of the trajectory, depending on the curvature. A high curvature leads to many way points, while a low curvature leads to only a few way points.
- *Parallel following:* Here the robot tries to drive parallel to the person, so the speed vector should be identical to that of the person. Obstacles should be avoided and the position to follow should be sideways behind the person.

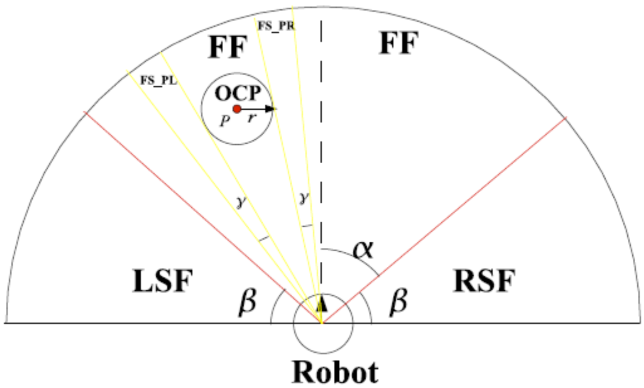


Figure 3.7: The region fields from [37]. FF means front fields, while LSF and LRF means left side field, respective right side field. The front fields are defined by the angle α while the side fields are defined by the angle β. It depends on the region the person is in, and the free space left and right from the person (FS_PL and FS_PR), defined by the angle γ, to decide which of the three behaviors should be active.

The author presents a rule set, which dictates how these three behaviors should be switched, depending on the relative position of the robot towards the person. The space before the robot is split into regions. This is shown in Fig. 3.7, where FF denotes the frontal fields, while LSF denotes the left side field and RSF the right side field. If the person is inside one of the side fields, direction following is activated. If the person is in the front field, the robot decides with a set of rather complex rules, if it has to activate the parallel following (when enough space is availible), or if it should use path following behaviour. More details are shown in [37].

Planning approach: The last group of methods to follow a person, is the group of planning and optimization approaches. Both have in common, that they find an *optimal path* or trajectory, for either a short period of time, or the complete solution. One example of short-term optimization is shown in [20]. The approach is quite unusual: the authors used visual and laser based pre-processing to extract baselines of obstacles in the area surrounding the robot. All known baselines are grouped to the left baseline group, consisting of obstacle baselines O^L, the left person baseline P^L, and the left vehicle baseline V^L. The group of right baselines also consists of O^R, V^R, P^R. Now the target is, to construct for every new steering command a quadratic function to separate both groups in an optimal fashion. This function is called the decision boundary $h(X)$ and is shown in Fig. 3.8 a). Since the person is split into two groups of baselines (the left border baseline and the right border baseline), and also the vehicle is, the boundary function has to lie between the front wheels of the robot and cross also the person. By using the optimization parameters, the tangent of $h(X)$ at the robots front wheel position can be used for the next steering angle. In this fashion obstacle avoidance and person following / homing is guaranteed.

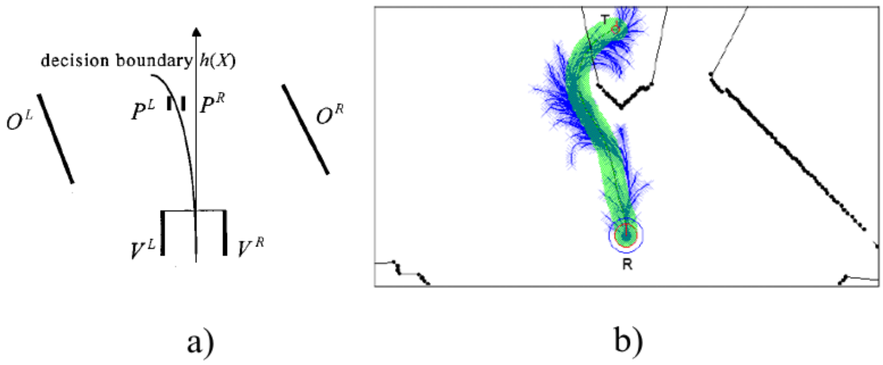


Figure 3.8: In a) the approach of [20] is shown. Here, two groups of baselines are constructed and a quadratic classifier $h(X)$ is found, that separates both groups in an optimal fashion. In part b) an search tree from [16] is shown. Here, an optimal path in 5D space has to be found. A heuristic is used to expand the graph for a directed search.

In deliverable D6.1 [19] we already investigated the planning technique of rapid expanding space trees. This belongs to the group of optimal planning techniques, and could be used also to follow a person, and not only to approach a person. Hoeller [16] used this technique to plan an optimal path in state space to a defined position relative to the person. The state is defined by the robots translational- and rotational speeds V_{trans} and V_{rot},and the pose x, y, ϕ of the robot. Each node represents a sample of this state. A set of so called "trajectorlets" can be used to expand the graph. This is usually a set of driving commands, which is applied to each node, and, given an time interval Δt, creates a trajectory snippet. The graph is expanded by defining a cost function for each trajectorlet by using a search heuristic to expand the graph with directed search. Here, the well known A* [15] graph planner is used to expand the graph towards the goal. Fig. 3.8 b) shows such a search tree.

3.1.3 Conclusions

In the previous sections we presented a set of different approaches to follow a person. In this section, the pros and cons of these approaches are shown first. Afterwards, the conclusion, what to use and why to use it is shown. But lets start with a comparison of the approaches. In Table 3.1 the approaches of a purely reactive controller-based approach are shown. Table 3.2 shows the pros and cons of the model based approaches.

Reactive controller approaches		
	pros	cons
visual servoing	-operates directly on the image -easy controller design	-person hard to detect -direct reaction on sensor noise -nothing to control, if perons lost -no obstacle avoidance
laser based	-easy controller design -more accurate person position	-direct reaction on sensor noise -nothing to control, if person lost -no obstacle avoidance

Table 3.1: Pros and cons of the approaches, dealing with reactive controllers.

It can be seen, that reactive controller based designs do not support obstacle avoidance. Because the controllers are so simple, they could not include complex obstacle situations. Another drawback is also the response to sensor noise. Some controlled values, like disparity could jitter suddenly and may lead also to jittering behaviour. When the values to control are not measurable any more, all controller based approaches will fail.

Table 3.2 shows a different picture when investigating modeling approaches. First of all, all investigated approaches support obstacle avoidance. Physical approaches could create a model of potential fields, which could lead to barriers and block the robot. But the benefit is that complex potential fields, such as the personal space, could also be modeled and used to steer the robot. Planning approaches could overcome the drawback of getting stuck in local minima, but are more complex to compute.

3.2 Planned approach to use in ALIAS

The conclusion shows a clear picture of what is possible with different approaches. While simple controller approaches work well in laboratory environments, real world applications need robust methods, which can deal with unforeseen obstacle situations and which will work in most situations. The physical approaches showed a great flexibility in modeling the environment by using potential fields. The planning approach of Hoeller [16] showed the benefit of finding even non-trivial solutions and trajectories to follow a person. We propose a mixture of both worlds, using the flexibility of modeling a potential field (and using here also a model of the personal space) and also using the ability of expanding space trees to plan even complex driving sequences to follow a person. How this is done

Modeling approaches		
	pros	cons
Physical approaches	-obstacle avoidance -plausible motion models possible -complex environment models possible -if person position is not measurable, the old environment model could still be used to control the robot	-can get stuck in local minima
Rule-based approaches	-obstacle avoidance -fast to compute	-rules based on designer -general rules often need a set of rules designed for exceptional situations -no general approach
Optimization approaches	-obstacle avoidance -plausible motion models possible -complex environment models possible -if person position is not measurable, the old environment model could still be used to control the robot	-computationally complex

Table 3.2: Pros and cons of different modeling approaches.

in detail, integrating into the Dynamic Window Approach, what practical problems have to be considered, and experimental results will be shown in the upcoming deliverable 6.3.

4 Guiding a person

In the previous chapter we described techniques used to follow a person. So why a chapter dedicated to guiding a person? Isn't that the same problem, only with the robot in front of the person? That is the problem: the roles have changed. While the robot is in following mode, the person decides, where to go to and the robot has to follow. As Cohen [9] states so well, a following task as well as a guiding task is a joint commitment between two parties, the robot and the person. But the robot does not posses its own will, which implies a following task, that the robot will stay within that task, until a different task is commanded.

This is completely different when a robot has to guide someone. This person has of course a free will and may change his/her mind upon the task. So the joint commitment can be broken and the robot has to detect such a case. The degree of freedom a person could show towards the robot without breaking the commitment, could be very high. For example a person can walk on the left or right side of the robot, but also behind the robot. All these configurations mean that the commitment is still active. The person can also change its speed. To detect all these cases, and also to tolerate that high degree of freedom of person movement, is the challenging aspect of that task. This includes questions like: What happens when the person stops to rests? Is the guiding task still active? What happens in narrow spaces, where the person may use a different path than the robot or even walk in front of the robot? Studies on state of the art leads to a devastating observation. Almost no publications deal with such an problem, although there exist today a huge amount of museum guide robotic systems. We found one publication that starts to investigate this challenging task. Due to the lack of other literature, we will investigate this publication in detail in the next section.

4.1 State of the art

There are works that deal with guiding blind persons in stores [21] like super markets or home improvement stores, but these differ to a great deal by the functionality they should fulfill from our use. The blind person has direct contact to the robot and is always in the same relative position towards the robot. The person is not free to walk beside the robot or stop to look at other persons or goods. So these robots perform a path following task at a certain speed by driving very smooth routes. Other approaches [8, 13] use a simple start-stop behavior to guide a person. They build a model to recognize the specific person and start to follow a path towards the target. When the person is lost or too far away, the robot stops until the person's position is near the robot so that guiding continues. This behavior is not socially acceptable and can lead to confusion on the user side to end the guiding process before the target could be reached.

There is one work [25], which investigates the guiding problem from a psychological

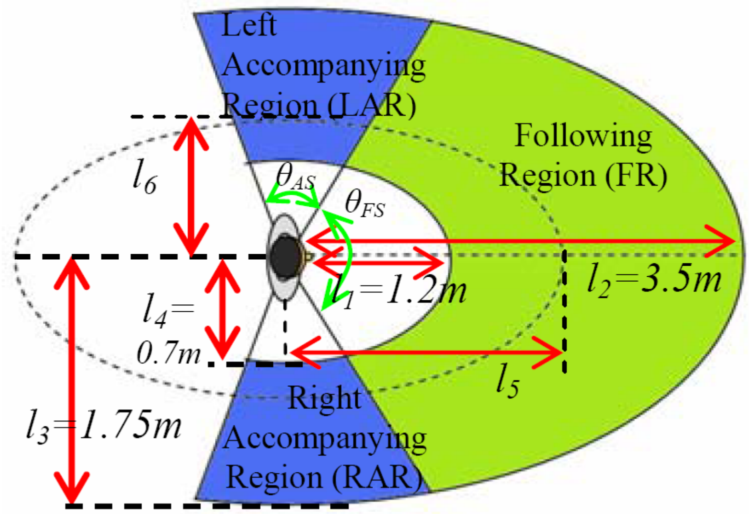


Figure 4.1: Picture taken from [25]. Here the regions are shown where the robot should drive to be in active guiding mode. These regions are modeled by a single Gaussian distribution (see equation 4.1) to give the probability of being in guide state.

perspective by using the personal space, and also by providing a sufficient model of "being in a guide situation". To our knowledge, this is the only work available now (and the authors state the same). We will describe and discuss this solution in detail here. From the proxemics theory of Hall[14] the authors construct a personal space configuration as seen in Fig. 4.1. Here the robot should be placed within a social zone of 1.2m to 3.5 meters. This zone is not circular, but elliptical. This model is not used directly, but is only a hint for a probabilistic description of the personal space. In this description the probability of "still-having-a-guiding-commitment" is modeled from the four-dimensional state vector X with the robot position x, y, the speed of the robot s_r and $\Delta\theta$, the relative angle to the persons upper body pose, by defining a multivariate Gaussian function:

$$p_{guide} = \frac{1}{(2\pi)^2 \cdot \sqrt{|\Sigma|}} \cdot e^{-1/2(X-\mu)^T(D_1+D_2)(X-\mu)} \tag{4.1}$$

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{\Delta\theta}^2 & 0 \\ 0 & 0 & 0 & \sigma_s^2 \end{bmatrix} \tag{4.2}$$

$$X = \begin{bmatrix} x_r \\ y_r \\ \Delta\theta \\ s_r \end{bmatrix} \tag{4.3}$$

$$\mu = \begin{bmatrix} x_h \\ y_h \\ 0 \\ s_h \end{bmatrix} \tag{4.4}$$

Note that equations 4.1, 4.2, 4.3, 4.4 differ from those in [25], since we believe that especially the exponential equation is wrong written! The parameters $\sigma_x^2, \sigma_y^2, \sigma_{\Delta\theta}^2, \sigma_s^2$ are chosen by hand and form the Gaussian function. The matrices D_1 and D_2 (in a normal case simply Σ^{-1}) describe two aspects of the probability distribution. The matrix D_1 models the covariance of position and the person orientation , while D_2 models the speed and relative robot orientation. These matrices are defined as follows:

$$D_1 = \begin{bmatrix} \frac{cos^2(\alpha)}{\sigma_x^2} + \frac{sin^2(\alpha)}{\sigma_y^2} & 0 & 0 & 0 \\ \frac{sin(2\alpha)}{\sigma_y^2} - \frac{sin(2\alpha)}{\sigma_x^2} & \frac{sin^2(\alpha)}{\sigma_x^2} + \frac{cos^2(\alpha)}{\sigma_y^2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

$$D_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_{\Delta\theta}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_s^2} \end{bmatrix} \tag{4.6}$$

Here, $alpha$ is the global person orientation, while $\Delta\theta$ is the robot direction, as seen from the person. They form the key components upon decision processes of the robot. They are used to compute the squared Mahalanobis distance d of X, which is simply $d = (X - \mu)^T (D_1 + D_2)(X - \mu)$. The parameters are set to $\sigma_x^2 = 3.5m^2, \sigma_y^2 = 1.75m^2, \sigma_{\Delta\theta}^2 = \pi, \sigma_s^2 = 1$, which describes an area as shown in Fig. 4.1 with an allowed tolerance of $180°$ and a speed deviation of $1m/s$. Now a few rules are constructed, how the robot should react, depending on d. The distance d is a direct measurement above which percentile within the distribution X lies. The first, and most important rule is:

- if d corresponds to $p_{guide} \geq 0.5$ the robot is still in guiding mode and follows its path

Although, there are a few sub-rules within the interval of $p_{guide} = [0.5...1.0]$.

- if $p_{guide} \geq 0.65$ the robot will not change the speed
- if $p_{guide} \in [0.55...0.65]$ the robot will slow down proportionally. $p_{guide} = 0.55$ means stop, $p_{guide} = 0.65$ means normal speed
- if $p_{guide} \in [0.5...0.55]$ the robot stands still and will enter the wait state. In the wait state the parameters are changed to $\sigma_x^2 = 3.5m^2, \sigma_y^2 = 1.75m^2, \sigma_{\Delta\theta}^2 = \pi/2, \sigma_s^2 = 0.5$, which restricts the tolerable interval concerning the speed of the person and the possible position of the robot.

The authors describe a lot more rules, i.e. how states of the state machine will change and influence the parameters, but either the state machine nor the rules are of any importance to ALIAS.

4.2 Planned approach to use in ALIAS

Since the choice of approaches is not very large, we use the key ideas of the presented approach. It uses the personal space and its tolerance upon unforseen user behavior. The rest is normal path following. All aspects could be integrated into our framework of Dynamic Window objectives, so that this task can be covered with just a configuration of the navigator module.

5 Path planning with respect to persons

The focus of the previous two chapters was the interaction with a user. This implies, that both interaction partners have a common goal, like having a video conference, reminding the user of something, or the robot providing internet services. The task of path planning relates to another problem, namely what happens if the person does not interact with the robot, but the robot has to navigate. This is the case if the robot should drive for example to the charging station or the person would like to be left alone for some reason.

In those cases the robot should respect the person's private space, and also should avoid crossing the person's predicted walking path. On the other hand, the person should notice what the robot does, to avoid any surprises. Again, the literature found during the recherches was very few, but investigates already different aspects of the path planning task. We present those aspects in the next section, discuss them and show, which methods seem to provide the most useful approaches.

5.1 State of the art

A key part of planning a path in a socially acceptable manner, is to predict the trajectory of the person (where the person will walk in the next few seconds), and to include that knowledge into the path planning. We separate approaches into two groups: one group learns trajectories with statistical methods and use them while planning driving paths, while the second group treats the trajectories as given from another module of an intelligent system. The last group differs in the way, that they do not consider probabilities when planning paths. The difference of both groups lies also in the time needed to setup such a system. While the statistical methods need a lot of training data, and also a large setup time, the methods which take the trajectories as given, are out of the box. Note that both approaches need a pre-learned environment model in the form of a map.

Statistical trajectory models: The first group of methods uses a large set of previously seen trajectories to learn trajectory models and uses them in path planning. We present two approaches: [3, 4] and [18]. The first approach of [3] uses a set of user trajectories $d = (d_1, d_2, .., d_N)$ of fixed size T and M learned models $\theta = (\theta_1, \theta_2, ..\theta_M)$, which adapt to cluster centers of the given trajectories. This is done by using the EM algorithm. Figure 5.1 shows an example training configuration of three trajectories $d_1 ... d_3$ and two models θ_1, θ_2.

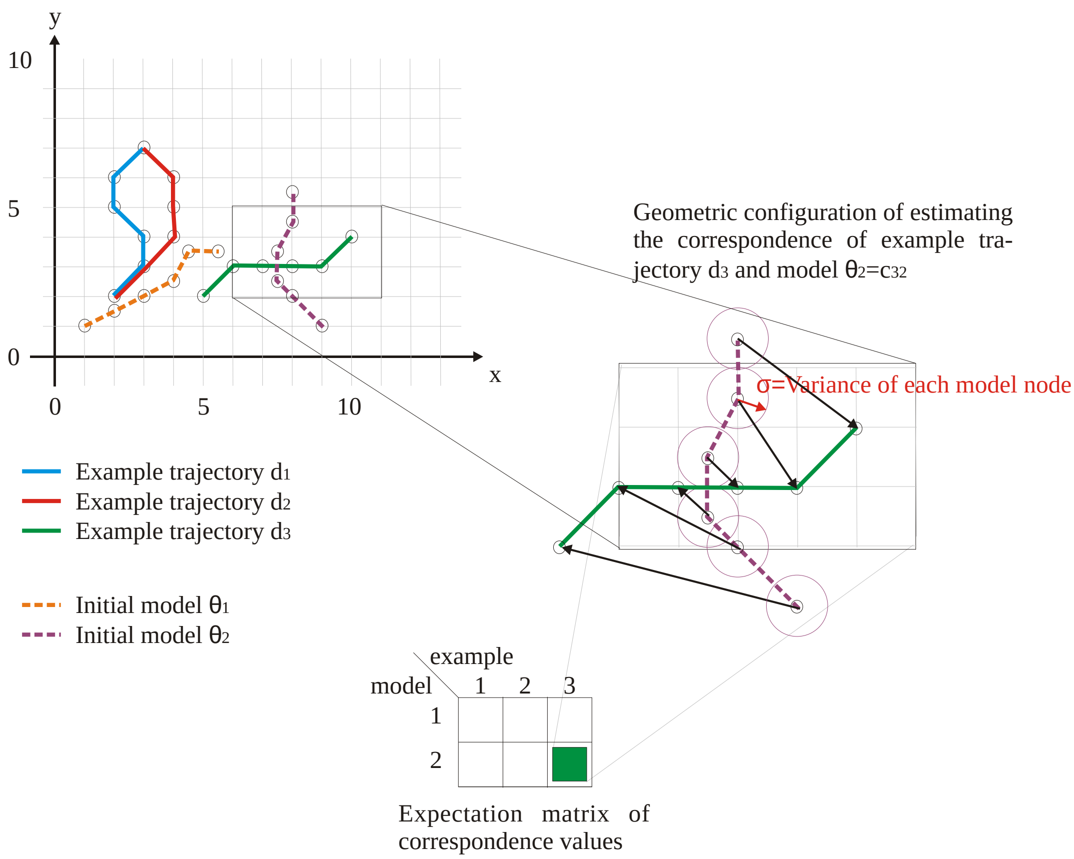


Figure 5.1: Example configuration to estimate two clusters/models of typical motion patterns of a person as described in [4, 3]. Also a geometrical configuration on how to calculate the correspondence of an example trajectory towards a model is shown.

Each training trajectory d_i consists of a set of T person positions $d_i = (x_1^i, x_2^i, ..., x_T^i)$. Each model θ_m also consists of the same number of model nodes $\theta_m = (\mu_1^m, \mu_2^m, ..., \mu_T^m)$. Now, Estimation/Maximization is used to construct a correspondence matrix in the Estimation step. This matrix denotes which example trajectory d_i belongs to which model θ_m under the assumption of the current iteration j.

$$E[c_{i,m}|\theta_m^{[j]}, d_i] = \nu \prod_{t=1}^{T} e^{-\frac{1}{2\sigma^2}\cdot \left\| x_i^t - \mu_{m[j]}^t \right\|} \tag{5.1}$$

Here, σ is constant for every node. The estimation of the correspondence matrix is updated. The correspondence matrix shows for every model how possibly a trajectory fits that model. So the sum of each row is one.

	d_1	d_2	...	d_N
θ_1				
θ_2				
...				
θ_M				

Table 5.1: Structure of the correspondence matrix, which is the core of the estimation step and updated at each iteration.

The maximization step is used to improve each model with its corresponding set of examples. Each node is updated independently and with the use of the estimation values:

$$\mu_m^{t[j+1]} = \frac{\sum_i = 1^N E[c_{i,m}|\theta_m^{[j]}, d_i] \cdot x_i^t}{\sum_i = 1^N E[c_{i,m}|\theta_m^{[j]}, d_i]} \tag{5.2}$$

There are some more rules to fuse models, when two models share equal trajectories (which means that they have equal correspondence values). In the end, unique models are created, each representing a set of example trajectories and the center of a cluster.

After training has finished, the set of models is used to classify trajectories by providing a probability, to which model the current trajectory may belong. This is described in detail in [4]. The problem on trying to classify the observed trajectory snippets is, that it is unknown, at which node of the model the current snippet of the trajectory begins, and at which node of the model it ends. This is why Bennewitz[4] suggests to assume every possible configuration of start node k and end node $\acute{k}$. This leads to the problem of interpolating trajectories an a rather complex classification task. Now, given the recognized trajectory $z = (x_1, x_2, ..., x_R)$, the probability of belonging to a model θ_m is as follows:

$$p(z|\theta_m) = \sum_{k=1}^{K} \sum_{\acute{k}=k}^{K} p(z|\theta_m, k, \acute{k}) \cdot p(k, \acute{k}) \tag{5.3}$$

$$p(z|\theta_m, k, \acute{k}) = \prod_{r=1}^{R} p(z^r|\theta_m^{[f(r,k,\acute{k}]}) \tag{5.4}$$

Note that the prior $p(k, \acute{k})$ depends on the person's position and speed and that $f(r, k, \acute{k})$ is a linear interpolation function from the trajectory point z_r to the node in the model:

$$f(r, k, \acute{k}) = \frac{\acute{k} - k}{R - 1} r + \frac{kR - \acute{k}}{R - 1} \tag{5.5}$$

Given the probability, that the trajectory z belongs to a model θ_m, and knowing the speed of the person, the motion is simply projected along the trajectory and further blurred over time. They use the A* [15] algorithm to plan over space and time and include the probabilities of cells being occupied over time.

Even more statistics are used by [18]. Here, a shopping mall is observed with laser scanners and the trajectories of thousands of customers are recorded. Kanda[18] classifies small intervals of these trajectories (5 second intervals) into four basic walking behaviors by using a support vector machine. The observed environment is split into cells and histograms are build up over time and space, to denote which basic behavior is dominant at a certain place at a certain time, called "use of space". Fig. 5.2 shows a result of this process. These histograms are important upon path planning of the robot.

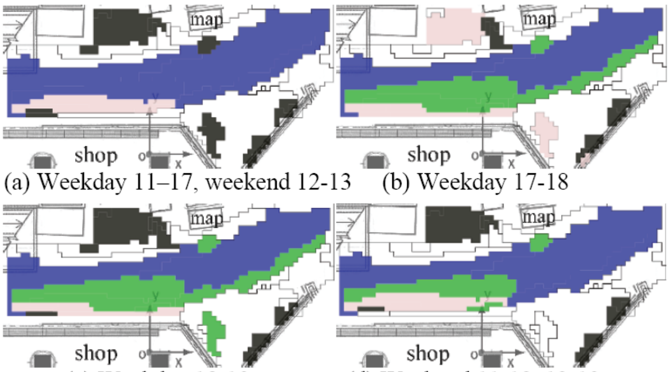


Figure 5.2: Image from [18], which shows the "use of space". This denotes which basic motion behavior is dominant in a certain time interval. Here, four time intervals are shown.

However, the local motion information is not sufficient to predict the motion trajectory over a longer distance. That is why a second model is constructed to cluster typical trajectories. Here, the k-means cluster method is used to create k cluster centers. To calculate the distance between two trajectories a dynamic programming (DP) matching algorithm is used on the precomputed trajectories. The trajectories are coded as a sequence of basic motion behaviors, and each include- and delete operation adds 1 meter to the comparison cost. The sequence is searched, where the costs of all compare-, include- and delete operations is minimal, which is the distance of the two trajectories. Essentially this is the same intention as done by Bennewitz[3]. Here, also the problem appears when observing a unclassified trajectory, to which cluster this trajectory may belong, but with the DP matching method they could also find the distance to each of the clusters.

According to the distance, weights are calculated to let each cluster vote for the next person motion. In this way, a kind of trajectory distribution is constructed, which also consists of information, where the person will likely be in the next few seconds. An example is shown in Fig. 5.3. This approach shows rather promising results.

For path planning two scenarios are shown. One scenario deals with approaching the person by driving in the persons predicted path (see [28]) in front of the person. This approach was already described in deliverable D6.1 [19]. The other approach is relatively

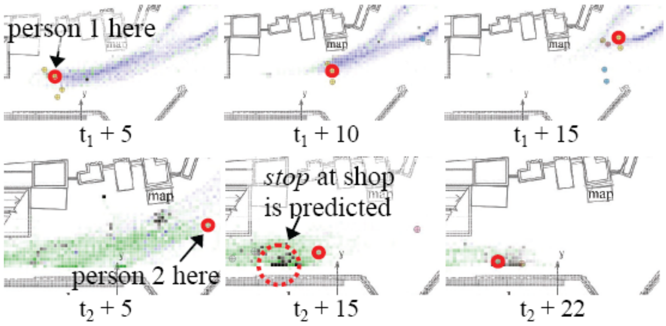


Figure 5.3: Two examples of motion prediction from [18]. The recognized trajectory is used to match against all known trajectory clusters. Depending on the match, each cluster votes for correct prediction. Over the three shown time steps, the clusters which are very unlikely could not vote strongly and fade from the prediction.

simple, since the robot tries to communicate only with persons within quiet areas, where persons normally walk slowly. Here the "use of space" maps are used to patrol the robot directly through such spaces, where idle walking and stopping dominates.

Given knowledge: This section covers approaches, that deal with knowledge given by external entities. This is interesting in the aspect that no training phase is needed or assumed. These approaches have to work out of the box! The first approach of [32] deals with different aspects of socially acceptable navigation, especially with aspects in path planning. This work is a great inspiration, albeit it does not cover the prediction of human motion within the path planning process. It was the first work, which models parts of the personal space, like the private zone, and also includes aspects like visibility of the robot and hiding behind walls. Fig. 5.4 shows all three aspects.

Sisbot[32] defines three criteria (safety, visibility and hiding), where each criterion creates a cost function. All three cost functions are merged with each other plus the obstacle map of the environment. Merging is done by taking the maximum cost value at each cell.

The safety zone (or private zone) is described as follows:

$$g(H,x,y) = \frac{(cos(2D_{Max}/\pi \cdot (x - H_x)) + 1) \cdot (cos(2D_{Max}/\pi \cdot (y - H_y) + 1))}{2} \tag{5.6}$$

The function $g(H,x,y)$ is only valid in the first half of the period, else it is set to zero. The visibility function is very equal, despite the fact that it is linearly dependent towards the deviation angle from the persons view direction $\Delta\Phi$. It is only defined outside an angle of $\pm 45°$. An example is shown in Fig. 5.4 b):

$$v(H,x,y) = g(H,x,y) \cdot \Delta\Phi \tag{5.7}$$

The hidden zone criterion focuses on free space not observable by a nearby person, because the free space is hidden behind a wall. Here the criterion is not easy to describe,

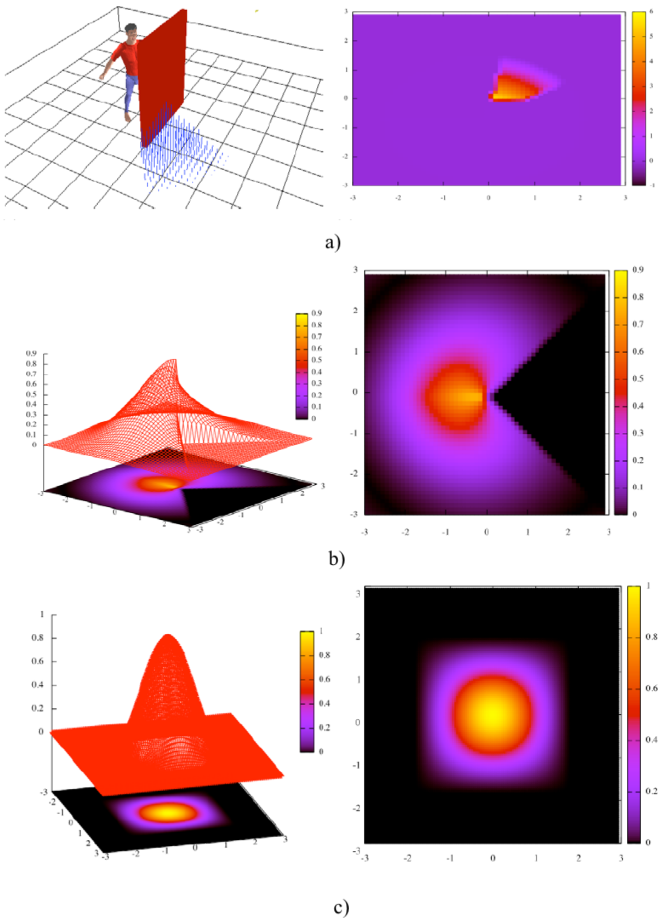


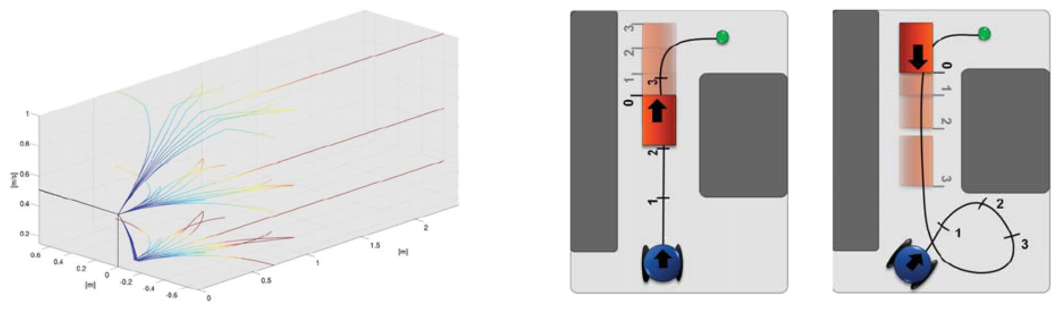
Figure 5.4: Pictures taken from [32]. Here the three elements of social navigation are shown. Element a) is the cost function of hiding, b) the cost function of visibility and c) the cost function of the private zone of the personal space.

since ray casting algorithms are involved. The basic idea is to check every free cell of the environment map, if a ray from the person's head towards this cell intersects an obstacle. If that is true the hidden condition is true. For every free cell, where the hidden criterion is true, the costs are calculated as follows:

$$h(H, x, y) = 1 - \frac{\|(x - H_x, y - H_y)\|}{D} \tag{5.8}$$

As mentioned before, all costs are merged by taking the maximum of all costs: $c(H, x, y) = max\,(g(H, x, y), v(H, x, y), h(H, x, y))$. These costs are used to apply a standard A* planner [15] to plan a path. Moving objects are not considered and another drawback is the used planner, since A* can create paths which are not smooth and sometimes look unnatural.

The last approach[27] discussed here, deals with the aspect of moving objects. It assumes that the motion of moving obstacles is known for a time interval of 4 seconds. Within that interval, planning in space and time is done with so called lattice graphs. A latttice consits of a "grid" of trajectories for a defined time interval, which stems from a certain action of accelerating, decelerating or keeping the speed while also changing the rotational speed of the robot. Fig. 5.5 a) shows an example of such a set. Each set leads to a new node of the graph, which could be expanded by another set.



a) b)

Figure 5.5: Pictures taken from [27]. In a) a lattice set to expand a graph node is shown, in b) experimental results on the benefit of time and space planing are shown.

A set of heuristics is used to expand the graph's nodes. These heuristics, like traversing time and distance to the target state, form the costs of the graph. D*[33] is used to provide fast directed planning and replanning within that graph. As experiments show (Fig. 5.5b), this approach is even able to recover from conflicts, when the robot and another person want to pass a door simultaneously. Although it is not able to perform in real time, it could prove its value under simulation conditions.

	Pros	Cons
Bennewitz[4, 3]	-mathematically advanced -only few examples needed -includes moving persons into planning	-traj. models limited to fixed size -heavy computational effort upon prediction -complete traj. observations needed -use standard A*
Kanda[18, 28]	-easy to implement -easy to understand -equal powerful as Bennewitz	-huge amount on training data needed -hand label data to classify basic behavior -have to cover full area with training data
Sisbot[32]	-covers criteria others do not have	-does not support moving persons -use standard A*
Rufli[27]	-includes moving persons into planning -lattice graph uses robot physics when planning	-no private zone around moving persons -huge computational effort

Table 5.2: Strength and weaknesses of the presented path planning approaches.

5.2 Conclusions

By investigating existing approaches, there are already powerful methods available, which support almost all needed functionality. What is missing in one approach is provided by another. While statistical approaches deal mostly with modeling the motion trajectory in a sophisticated way, they lack path planning capabilities, while approaches which need person trajectories and positions as input provide good path planning capabilities. Table 5.2 summarizes the strength and weaknesses of all presented approaches.

5.3 Planned approach to use in ALIAS

The lattice graph approach seems to have the most flexibility in planning, while also providing good modeling properties of the robot motion. Since it performs over space and time, no restrictions on moving objects are given, or the length of trajectories are truncated. Since we plan to use a similar approach when following a person (with expanding random trees), we aim to adapt the approach of lattice graphs and extend it with the criteria Sisbot has found in his work [32]. As a main question the estimation of the person's trajectory remains. We have shown here that many approaches are available and will choose one that is sufficient to show experimental results.

6 Observing a resting person

The last of the robot tasks is the observation of a resting person. This is needed if the robot is not interacting with the user, but waiting for commands. To recognize these commands the robot has to sense the userfrom a ranging distance. The task is to find a position, which is not intrusive to the user's private zone (and does not disturb the user), while also being able to observe the person when the person changes their position. The system should behave in the end like a waiting butler. This problem could be characterized as an observation problem, where three constraints should be respected. One is, to keep the observability of the person by the used sensor system (laser and camera). The second constraint, is to find a position, where the person could be observed even when changing their position within the resting area. Third, the robot should select a place, where rotation is sufficient to also recognize larger changes of position, and where the robot does not obstruct critical paths, such as line of sight to television or walking path to other resting areas. The literature gives almost no hint on solving such a problem. There is one work from ourselves which deals with position optimization and there are works which deal with optimal positioning for measurement and estimation tasks, the so called "next best view" problem. Both approaches are shown in the next section. We also present a short work, on how to estimate resting points from recorded trajectories.

6.1 State of the art

The next best view problem: In this section we show two approaches of the next-best-view problem. This problem deals with the question of where to drive next, to get either full information of 3D object structures to create maps and track objects in space.

The goal of this investigation, is to show the key idea behind all approaches. The first work, presented here, focuses on the estimation of 3D data from an object, by driving around that object with a monocular camera. The object consists of several small planes, where the position and normal vector of each plane should be estimated. The key idea is to minimize the determinant of the covariance matrix over all known planes. This is equal to a maximal information gain. To calculate the information gain, the eigenvectors and eigenvalues of the covariance matrix are used. The eigenvalue with the highest value covers the most uncertainty. For one plane, each robot motion, which observes this vector from a side view, increases the certainty and adds information to the model. This is done for every position the robot can reach and the mean over all eigenvectors and surfaces is taken. This is quite an exhaustive calculation and would need a huge amount of time, if done for every point. This is why initially two points are selected , the starting point x_0 , and a point at the opposite side of the object x_1. Now the distance between both points is split into equal lengths. Perpendicular to this mid point, a region is rastered (shown in blue in Fig. 6.1) and the minimum value is the next point to drive to. This process is

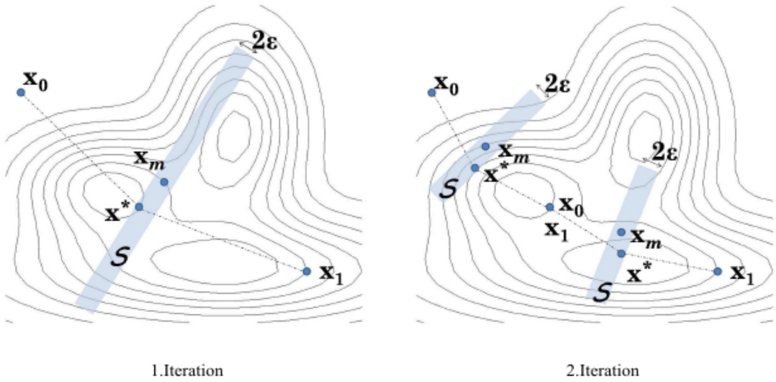


Figure 6.1: Pictures from [10]: two iteration steps to to create a path with optimal information gain. In the first iteration a region perpendicular towards the mid-point of x_0, x_1 is searched for the minimum value. The resulting two lines are also split in the middle and a region is searched for optimal information gain. Note, that the object itself is not shown, but should be places between x_0 and x_1.

done iteratively with each new line segment until a dense path is created. Two steps of the refinement process are shown in Fig. 6.1.

A second approach is shown in [6], where a particle filter is used to track an object in 3D space. Particle filters are often used as state estimators. Each particle consists of a state sample the system should estimate and a weight, how likely it is that this state is correct. All particle weights are normalized to sum up to one. The authors state that the distribution of the particles and especially the distribution of the particle weights are a measurement for the information included inside the particle filter. This is done by using the measurement of the effective particles:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_s} w_i^2} \tag{6.1}$$

Here, w_i is the normalized weight of the particles and N_s is the number of particles used within the filter. When the distribution is uniform, each particle will have $w_i = 1/N_s$ and $N_{eff} = N_s$, while in the opposite case, all particles but one have a weight of zero and one particle will have the weight of one. In the latter case $N_{eff} = 1$, so only one particle carries information on the distribution. The rate of effective particles λ_{eff} is used to update the particle filter. The rate is defined as follows:

$$\lambda_{eff} = \frac{N_{eff}}{N_s} \tag{6.2}$$

A rate of 100% is reached on equal distribution, and a rate of $1/N_s$ is reached in case of a unimodal peak distribution. Finally a random set ζ of particle sets $\zeta_1, ..., \zeta_N$ is drawn to update the filter, and the set ζ_i is used, which maximizes the effective particle rate:

$$\zeta_{opt} = \overset{\zeta_i}{argmax} \left(\lambda_{eff}(\zeta_i)\right) \tag{6.3}$$

To sum up both approaches, each one tries to maximize information. One needs to maximize information on an object, another tries to maximize information inside a particle swarm. This goal is common in all next best view approaches we read, albeit with other maximization criteria.

Observation point optimization: An approach, which deals directly with the global observation point search is presented in [30]. Here, also a particle swarm is used to find a global optimal position with different criteria. The criteria are size and position of a person inside the image, illumination background and free line of sight. All three criteria are merged by weighted sums and each particle of the swarm calculates its weight according to the criterion. By parametrizing the criteria, as well as the weights, different end position can be achieved. Fig. 6.1 shows a parameter set of passive observation, with and without including illumination information.

(a) Optimized locations for a passive observation task (green = without considering illumination, red = including illumination metrics, 80 independent runs each). Unlike in the video conference task, a larger variety of good positions exists.

(b) Image captured from the left-most pose in Fig. 10(a) (optimization result without illumination): The person is hardly visible.

(c) Image captured from the right pose in Fig. 10(a): contrast and total image quality are clearly improved.

Figure 6.2: Example from [30]: here a configuration for a passive observation optimization is used and the the results of using illumination (red) and not using illumination (green) are shown. Pictures b) and c) show the corresponding camera images.

This approach optimized directly on a quality function, and does not consider information gain. The found solution is optimal for that exact configuration and it cannot be guaranteed that it is stable upon slide changes in person position.

Estimating resting positions: Since we also want to avoid placing the robot in places covering a path between usual resting positions, we need to have a model of such resting positions and interest points like television or fish tanks. One choice is to label those points, when the robot is prepared for operating inside the home environment. Another choice is to learn such points. One approach is shown by [5], where trajectories are recorded over a long time and clustered afterwards. With the cluster prototypes, end points of the trajectories are set to be such resting points. By a Gaussian noise process and by creating a comparing function towards each trajectory of the cluster, the end point position is refined. This is quite a simple approach, but the only one we found. Another idea could be to create sums of occupancy times at all places the person is observed. If the person is not wandering constantly, a few places with very high occupancy times should appear.

6.2 Planned approach

We already have an approach to find an optimal position. This approach needs a dedicated illumination model of the environment, which is hard to do in practice and not part of the ALIAS project. We could use the particle swarm model again and test its capability in practical environments. We also have to couple the quality function of the particles to the ability of the robot to react on small pose changes, which was not provided by any known approach. But here, the knowledge of the observed information and the usage of possible resting points may help.

This task is the most unclear task of all tasks that the robot has to fulfill. So it naturally covers the most risk, and also the most work to be done. As a fallback solution, the existing solution without any changes could be used within the ALIAS project.

7 Conclusion and outlook

In this deliverable we have shown the state of the art and the variety of approaches that deal more or less with socially acceptable navigation. In the end, we do not want a set of many stand-alone behaviors the robot should switch between, but one common framework which should be configurable to fulfill all needed tasks. This has the benefit, that the core functionalities, like obstacle avoidance and motion controlling are available for every task and no double implementation has to be done. Also the project partner MetraLabs GmbH already provides a navigation framework, consisting of the dynamic window approach.

The dynamic window approach has the benefit of high configurability: by activating several objectives, different tasks and behaviors could be provided. So our goal is, to extend this approach and to adapt the shown and chosen approaches towards the dynamic window approach. In some cases (for example the expanding random trees) approaches are shown, which are even more powerful than the dynamic window, because they can plan in more than just one time step. Here we have to build a bridge between those methods, to provide the correct functionality. As a summary we state again our goals on each of the four tasks:

- Following a person: Here, we plan to combine the expanding random tree approach of Hoeller [16] with the personal space model of Andersen [1]. Since this approach is not fully compatible with the dynamic window approach we have to extend the dynamic window. Details will follow in deliverable 6.3.
- Guiding a person: Here, only one approach was found. Since this approach only describes when a normal path following could be performed, in which situation the speed should be reduced (to full stop) and when the guiding process is canceled, this could be modeled fully within the dynamic window. Details will follow in deliverable 6.3.
- Path planning with respect to persons: This is a stage not directly bound to the dynamic window approach. We have chosen to use the graph approach of Rufli[27], since the similarity to expanding space trees is huge, and we plan to extend this approach by using the critera of Sisbot[32]. Details will follow in one of the next deliverables.
- Observation of a resting person: As a starting point we use a particle filter, which incorporates knowledge about visibility, distance and reachability to find the best observation place. This filter should be extended with the ability to incorporate knowledge of a local pose distribution of persons at resting positions, to maximize observability. This task has a huge demand on research work, and so covers the most risk.

The next deliverable will include initial experimental results on the behaviors of following a person and approaching a person. It will include also methodical details on the guiding behavior.

Bibliography

[1] H. J. Andersen, T. Bak, and M. Svenstrup. Adaptive robot to person encounter by motion patterns. *Proc. EuroBot*, pages 1–11, 2008.

[2] K. Arras, O. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *Int. IEEE Conference on Intelligent Robots and Systems (IROS)*, pages 3402–3407, 2007.

[3] M. Bennewitz, W. Burgard, and S. Thrun. Learning motion patterns of persons for mobile service robots. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 2601–3606, 2002.

[4] M. Bennewitz, W. Burgard, and S. Thrun. Adapting navigation strategies using motions patterns of people. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 2000–2005, Taipei, Taiwan, 2003.

[5] A. Bruce and G. Gordon. Better motion prediction for people-tracking. In *In Proc. Int. Conference on Robotics and Automation (ICRA)*, 2004.

[6] H. Chen and Y. Li. Dynamic view planning by effective particles for three-dimensional tracking. *IEEE Transactions on Systems, Man, and CyberneticsŮPart B: Cybernetics*, 39(1):242–253, 2009.

[7] Z. Chen and T. Birchfield. Person following with a mobile robot using binocular feature-based tracking. In *Int. IEEE Conference on Intelligent Robots and Systems (IROS)*, pages 815–820, 2007.

[8] A. Clodic, S. Fleury, R. Alami, R. Chatila, G. Bailly, L. Brethes, M. Cottret, P. Danes, X. Dollat, F. Elisei, I. Ferrane, M. Herrb, G. Infantes, C. Lemaire, F. Lerasle, J. Manhes, P. Marcoul, P. Menezes, and V. V. Montreuil. Rackham: An interactive robot-guide. In *Proceedings RO-MAN*, pages 502–509, 2006.

[9] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2).

[10] E. Dunn, J. van den Berg, and J.-M. Frahm. Developing visual sensing strategies through next best view planning. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4001–4008, St. Louis, USA, 2009.

[11] A. Fod, A. Howard, and M. Mataric. Laser based people tracking. In *Proc. of the IEEE International Conference on Robotics and Automation(ICRA)*, pages 3024–3029, 2002.

[12] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, pages 34–46, 2006.

[13] H.-M. Gross, H.-J. Boehme, C. Schroeter, S. Mueller, A. Koenig, E. Einhorn, C. Martin, M. Merten, and A. Bley. Toomas: Interactive shopping guide robots in everyday use - final implementation and experiences from long-term field trials. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2005–2012, St. Louis, 2009.

[14] E. Hall. *The hidden dimension*. New York: doubleday, 1966.

[15] E. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science and Cybernetics*, 4:100–107, 1968.

[16] F. Hoeller, D. Schulz, M. Mark Moors, and F. Schneider. Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 1260Ű1265, 2007.

[17] C. Hu, X. Ma, and X. Dai. Reliable person following approach for mobile robot in indoor environment. In *Proceedings of the 8th International Conference on Machine Learning and Cybernetics*, pages 1815–1821, 2009.

[18] T. Kanda, D. F. Glas, M. Shiomi, H. Ishiguro, and N. Hagita. Who will be the customer?: A social robot that anticipates peopleŠs behavior from their trajectories. In *Proc. Of ACM International Conference on Human Robot Interaction (HRI)*, 2009.

[19] J. Kessler. D6.1 report on different navigation strategies to approach elderly people in a polite manner. *EU Deliverable*, 2010.

[20] C. Ku and W. Tsai. Obstacle avoidance in person following for vision-based autonomous land vehicle guidance using vehicle location estimation and quadratic pattern classifier. *IEEE Transactions on Industrial Electronics*, 48(1):205–215, 2001.

[21] V. Kulyukin, C. Gharpure, J. Nicholson, and G. Osborne. Robotassisted wayfinding for the visually impaired in structured indoor environments. *Autonomous Robots*, 21(1):29–41, 2006.

[22] B. Kwolek. Person following and mobile camera localization using particle filters. In *4th Internaltional Workshop on Robot Motion and Control*, pages 265–270, 2004.

[23] R. Luo, N. Chang, S. Lin, and S. Wu. Human tracking and following using sensor fusion approach for mobile assistive companion robot. *Proceedings of the 35th Annual of IEEE Industrial Eletronics IECON*, pages 2235–2240, 2009.

[24] X. Ma, C. Hu, X. Dai, and K. Qian. Sensor integration for person tracking and following with mobile robot. In *Int. IEEE Conference on Intelligent Robots and Systems (IROS)*, pages 3254–3259, 2008.

[25] A. K. Pandey and R. Rachid Alami. A step towards a sociable robot guide which monitors and adapts to the personŠs activities. In *Proceedings of ICAR workshop*, 2009.

[26] M. Piaggio, R. Fornaro, A. Piombo, L. Sanna, and R. Zaccaria. An optical-flow person following behaviour. In *Proceedings of the IEEE ISIC/CIRA/ISAS Joint Conference*, pages 301–306, 1998.

[27] M. Rufli and R. Siegwart. On the application of the d* search algorithm to time-based planning on lattice graphs. In *Proceedings of the 4th European Conference on Mobile Robotics (ECMR)*, pages 105–110, 2009.

[28] S. Satake, T. Kanda, D. Glas, M. Imai, H. Ishiguro, and N. Hagita. How to approach humans?- strategies for social robots to initiate interaction. In *Proc. Of ACM International Conference on Human Robot Interaction (HRI)*, pages 109–116, 2009.

[29] M. Scheutz, J. McRaven, and G. Cserey. Fast, reliable, adaptive, bimodal people tracking for indoor environments. In *Int. IEEE Conference on Intelligent Robots and Systems (IROS)*, pages 1347–1352, 2004.

[30] C. Schroeter, M. Hoechemer, S. Mueller, and H.-M. Gross. Autonomous robot cameraman - observation pose optimization for a mobile service robot in indoor living space. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA 2009)*, Kobe, Japan.

[31] X. Shao, K. Katabira, R. Shibasaki, H. Zhao, and Y. Nakagawa. Tracking a variable number of pedestrians in crowded scenes by using laser range scanners. In *International Conference on Systems, Man and Cybernetics*, page 1545Ű1551, 2008.

[32] E. A. Sisbot. Towards human-aware robot motions. In *PhD Thesis*, Toulouse, 2008.

[33] A. Stentz. Optimal and efficient path planning for partially-known environments. In *In Proc. Int. Conference on Robotics and Automation (ICRA)*, page 3310Ű3317.

[34] M. Svenstrup, S. Tranberg, H. J. Andersen, and T. Bak. Pose estimation and adaptive robot behaviour for human-robot interaction. *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 3571–3576, 2009.

[35] H. Takemura, K. Ito, and H. Mizoguchi. Person following mobile robot under varying illumination based on distance and color information. *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics*, pages 1500–1505, 2007.

[36] C. Tomasi and T. Kanade. *Detection and tracking of point features*. Carnegie Mellon University, 1991.

[37] F. Yuan, M. Hanheide, and G. Sagerer. Spatial context-aware person-following for a domestic robot. In *Proceedings of the CoTeSys Workshop*, 2008.

[38] W. Yun, D. Kim, and J. Lee. Person following with obstacle avoidance based on multi-layered mean shift and force field method. In *International Conference on Systems, Man and Cybernetics*, page 3813Ű3816, 2010.