
Interactive Acquisition of Spatial Representations with Mobile Robots

Fang Yuan

M.Sc. ComputerSc. Fang Yuan
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld
email: meekfang@hotmail.com

Abdruck der genehmigten Dissertation zur Erlangung
des akademischen Grades Doktor-Ingenieur (Dr.-Ing.).
Der Technischen Fakultät der Universität Bielefeld
am 12.04.2011 vorgelegt von Fang Yuan,
am 16.09.2011 verteidigt und genehmigt.

Gutachter:

Dr.-Ing. Marc Hanheide, Universität Bielefeld
Assistant Prof. Patric Jensfelt, Kungliga Tekniska Högskolan
Prof. Dr.-Ing. Franz Kummert, Universität Bielefeld

Prüfungsausschuss:

Prof. Dr. Barbara Hammer, Universität Bielefeld
Dr.-Ing. Marc Hanheide, Universität Bielefeld
Prof. Dr.-Ing. Franz Kummert, Universität Bielefeld
Dr. Christina Unger, Universität Bielefeld

Gedruckt auf alterungsbeständigem Papier nach ISO 9706

Interactive Acquisition of Spatial Representations with Mobile Robots

Der Technischen Fakultät der Universität Bielefeld

zur Erlangung des Grades

Doktor-Ingenieur

vorgelegt von

Fang Yuan

Bielefeld – April 2011

Abstract

Robots that are able to provide services to human users in personal context are known as personal service robots. They are attracting more and more attention, as enriching and easing people's lives are their general missions. Acquisition of spatial representations is crucial for such a robot to move around in a human-shared environment, provide services and interact with the user about the environment. With methods developed in the field of autonomous exploration it is possible for a mobile robot to build a metric representation from sensor data, while exploring an initially unknown environment. However, due to the limited perceptual abilities of the robot and dynamic changes in the environment over time, challenges still pose for robots. Besides, in order to facilitate the communication about the environment, a mapping between the spatial understanding of the user and the robot has to be built in a service context.

To acquire a spatial representation, exploration strategies are necessary for the robot to move around in the environment and gain information from the surroundings. The term *Human-Robot Joint Spatial Exploration* is used for a framework that allows the human user to influence the robot's exploration of the environment and the environment representation built by the robot. By interactively gathering information, individual spatial knowledge from the user's point of view is expected to be integrated into the built environment representation. Therefore, how to make the robot interactively learn from the human user and build a spatial representation incorporating the gathered information is the central problem of this thesis. A *Home-Tour* is assumed as the initial scenario for information gathering. In the interactive guided tour the user presents the environment to the robot, while the robot learns various models of the environment with the help of the user. A link between the spatial representation built by the robot and human understanding of the environment is able to be created.

The major contribution of this thesis is a framework that enables a mobile robot to interactively acquire an environment representation. We first present the interactive *Home-Tour* scenario assumed as the starting point of the integrated solution to the central problem. Benefits and challenges of this solution are discussed as well. Then, a hybrid person-following behavior that facilitates the joint exploration of the environment for the robot and the guide-person is described. The robot is able to choose an appropriate behavior according to strategies designed on the basis of spatial context and relative person's positions with respect to the robot. Afterwards, a metric map of the surroundings can be built by the robot during the joint exploration. Because of the guide-person perceived by the robot in the *Home-Tour*, the violation of the static world assumption used in the mapping process has to be pointed out. Consequently, we provide a technique to reduce the negative impact on the mapping process, resulting from the guide-person. Finally, a topological graph built upon the metric representation is discussed. Human navigation strategies and spatial concepts achieved by the robot in the interactive learning process are incorporated in the graph-based environment representation.

Based on the idea of interactive information gathering, an integrated system is implemented. Experimental results achieved from individual modules validate their effectiveness respectively. By means of a user study, the robustness and practicality of the system consisting of these modules are verified as a whole in real world scenarios.

Acknowledgments

It is a pleasure to thank those who made this thesis possible. First of all, I would like to thank Gerhard Sagerer and Franz Kummert who gave me a chance to show that a foreign student from a university of applied sciences in Germany is competent to complete a doctoral thesis. Their trust is a great encouragement for me throughout these years.

Looking back my work as a whole I owe my deepest gratitude to my supervisor Marc Hanheide who was always able to help me out of a difficult position. With his comprehensive expertise, enthusiasm, and patience he inspired and influenced my work at the most. When I was frustrated by the work, he tried to help me stand up again. I would never forget your help and attention when I fell ill, your encouragement when my paper submission failed at the first time, long time discussion before my presentation in Osaka, working together in Graz. I am heartily thankful to you, Marc, my supervisor, my friend and my brother! I am also grateful that Patric Jensfelt would like to agreed to review the thesis.

Furthermore, I want to thank all my colleagues in the Applied Informatics. The harmonious working atmosphere and your friendly supports make the last years in Bielefeld a really good experience. Especially I would like to thank Thorsten Peter Spexard who gave me the best overview of the robot platform, Jan Moringen who would like to discussed with me for trivial problems that were actually far from his own work and Lars Schillingmann who can give me the best technical support.

Of course, I need to thank my bachelor students Lukas Twardon and Anton Helwart, as well as my diploma student Orhan Engin. It is my pleasure to supervise and benefit from their work. In particular I want to thank again Lukas Twardon who accompanied my research as a student assistant as well. For grammar correction of my thesis I would like to thank Sengupta, Debanjan.

Above all I thank my parents, who have always supported me and helped me to find my way anytime and anywhere. I owe my deepest gratitude to my wife Xixi. You have been always considerate to me, tried your best to support me and exclude the difficulty and anxiety for me. Of course, I thank Meng, our lovable daughter. You have always brought happiness and hope to us.

Contents

1. Introduction	1
1.1. Home-Tour with a Mobile Robot	3
1.2. Background Knowledge and Related Work	7
1.3. Contributions	13
1.4. Organization of this Thesis	15
2. Person-Following	17
2.1. Related Work	18
2.2. Person Tracking and Motion Control	19
2.2.1. Multi-Modal Person Tracking	20
2.2.2. Motion Control	22
2.3. Adaptive Context-Aware Following Behavior	22
2.3.1. Core Strategies	23
2.4. Elementary Person-Following Behaviors	26
2.4.1. General Strategies	26
2.4.2. Direction-Following	29
2.4.3. Path-Following	30
2.4.4. Parallel-Following	32
2.5. Summary	37
3. Map Building in a Home-Tour	41
3.1. Introduction of SLAM with Rao-Blackwellisation	41
3.2. Mapping during the <i>Home-Tour</i>	45
3.2.1. Basic Idea	47
3.2.2. Evaluation on a Simulation Platform	47
3.3. Summary and Discussion	56
4. Path Planning Adopting Human Strategies	59
4.1. Related Work	60
4.2. Path Planning	61
4.2.1. Graph Creation	63
4.2.2. Re-planning	66
4.2.3. Empirical Study	68
4.3. Multisensor-based Motion Generation	73
4.3.1. Motion Generation	74
4.3.2. Data Fusion	79
4.3.3. Experimental Results	83
4.4. Summary and Discussion	85
5. Evaluation	89
5.1. Experimental Design	92
5.1.1. Procedure	92
5.1.2. System Architecture	93

5.2. Assessment of Individual Modules in the Integrated System	99
5.2.1. Person-Following	99
5.2.2. Map Building	110
5.2.3. Path Planning	116
5.3. Observations from a Subject's Perspective	123
5.4. Insights and Discussion	125
6. Conclusion and Future Considerations	129
6.1. Summary	129
6.2. Concluding Discussion and Future Considerations	130
A. A Brief Introduction to Particle Filters	135
Bibliography	139

List of Figures

1.1. Examples of personal service robots.	2
1.2. Two generations of robot BIRON	4
1.3. A regular apartment equipped with standard furniture.	4
1.4. BIRON II in RoboCup competition.	6
1.5. A sketch map.	10
1.6. Elementary tasks lying in robotic exploration.	11
2.1. The test environment.	20
2.2. Results of the person tracking system.	21
2.3. Description of the core strategy.	25
2.4. Geometric representation.	28
2.5. A robot's goal (Px, Py, θ).	29
2.6. Computation of goal-direction.	30
2.7. Comparing direction- with path-following.	33
2.8. Computation of the potential goals.	34
2.9. Decision policy.	35
2.10. Parallel-following performed in <i>Stage</i>	36
3.1. The valid and invalid laser range.	45
3.2. The test environment for the proposed technique.	46
3.3. Comparison of the built grid maps.	46
3.4. Computed information of the built map.	49
3.5. Average information.	50
3.6. The map-based path planning.	51
3.7. Square root of the determinant	52
3.8. Position error of the robot	52
3.9. Angle error computed.	53
3.10. Average curve of the square root.	53
3.11. Average position error.	54
3.12. Average angle error.	55
4.1. Data flow of the proposed path planning.	62
4.2. Human trajectories and robot's paths.	64
4.3. Multiple graph layers.	65
4.4. Finding dynamic obstacles on the grid map.	67
4.5. Test of <i>ND</i>	68
4.6. Occupancy grid map of the environment.	69
4.7. Multiple graph layers.	70
4.8. Path from graph- and map-based approach.	71
4.9. Comparison of the planned and re-planned path.	72
4.10. Comparison of the paths computed.	73
4.11. Overview of sensor fusion.	75
4.12. Motion planning and control loops.	76

4.13.NF1 example.	77
4.14.Example re-planning sequence.	78
4.15.Look-up operations.	79
4.16.SICK LMS.	80
4.17.Head of BIRON I with.	81
4.18.Scenario: table in front.	82
4.19.Comparing the motion generation.	84
5.1. The ground truth map built for the user study.	90
5.2. Introduction of the task.	91
5.3. Overview of the implemented system.	94
5.4. Global localization without exploration.	95
5.5. Global localization with exploration.	96
5.6. Mini-exploration.	97
5.7. Mini-dialog.	98
5.8. Person-following behavior with pure data.	101
5.9. Map built during the second experiment.	102
5.10.Following behavior with data pre-processing.	104
5.11.Person robot distance.	105
5.12.Comparison of the grid map built.	111
5.13.Computed information for the second experiment.	112
5.14.Average map information.	113
5.15.Global path planning using NF1.	114
5.16.Map built during the fourth experiment.	117
5.17.The start and end pose of the robot localizing.	119
5.18.Global localization.	120
5.19.The NF1 path and the subsequent path.	121
5.20.The built graph from the person's trajectory.	122
5.21.Comparison of the map- and person's trajectory-based path.	124
A.1. The distribution function.	136
A.2. The target distribution.	137
A.3. The samples drawn from.	138

1. Introduction

*If I have seen further it is only by standing on the
shoulders of giants.*
[Isaac Newton, 1642–1727]

Acquisition of an environment representation is one of the fundamental problems in mobile robotics. Recently, mobile robots envisioned to coexist with humans and used to perform tasks for human users have attracted more and more attention from researchers. Generally such a robot is known as a personal service robot, whose purpose is to educate, entertain, assist or protect in a domestic or an office-like environment [104]. Examples of personal service robots designed for these goals are shown in Fig. 1.1.

In general, a personal service robot can be thought of as a mobile robotic platform situated in the physical world (in domestic or office context¹) shared with its user. By equipping the robot with hardware, e.g. robot sensors, actuators, microphone, manipulators, etc., and corresponding software components on this platform, the robot is able to gather information from the environment and interactively from the human user. Exploiting the gathered information the robot can work for its user according to concrete situations.

To provide services, the robot needs to move around and work in an environment initially unknown to it. Therefore, acquiring an environment representation containing certain knowledge of the surroundings is crucial for the robot before performing a task assigned by the human user. Leaving the robot alone and making it autonomously explore the environment to build a representation of the surroundings might be a solution. However, there are at least two points amiss:

▷ ***How to support the autonomous acquisition of spatial knowledge for a personal service robot?***

Completing the task of building an environment representation alone, the robot has to face the so-called *integrated exploration* problem [101, 43]. In other words, the robot needs to create the map of the surroundings and simultaneously localize itself in this built map, while making local decisions on where to move next [17, 138]. It is a quite complicated issue, as the three mentioned problems have to be dealt with at the same time. Unfortunately the three tasks are dependent on each other (see section 1.2 for details). Moreover, the efficiency of solving the *integrated exploration* problem in a real world scenario would be influenced directly by the adopted exploration strategy and the environment in which the robot is operating. Besides, to enable an efficient communication about the environment between the robot and the user, the built environment representation needs to correspond to spatial understanding of the human user in a service context. Therefore, from where the robot can learn human

¹Indoor environments are what will be discussed predominantly in this thesis. However, the working areas of personal service robots can be extended to outdoors, such as gardens, garages, normal streets, etc., where the robots have to face more complicated surroundings.

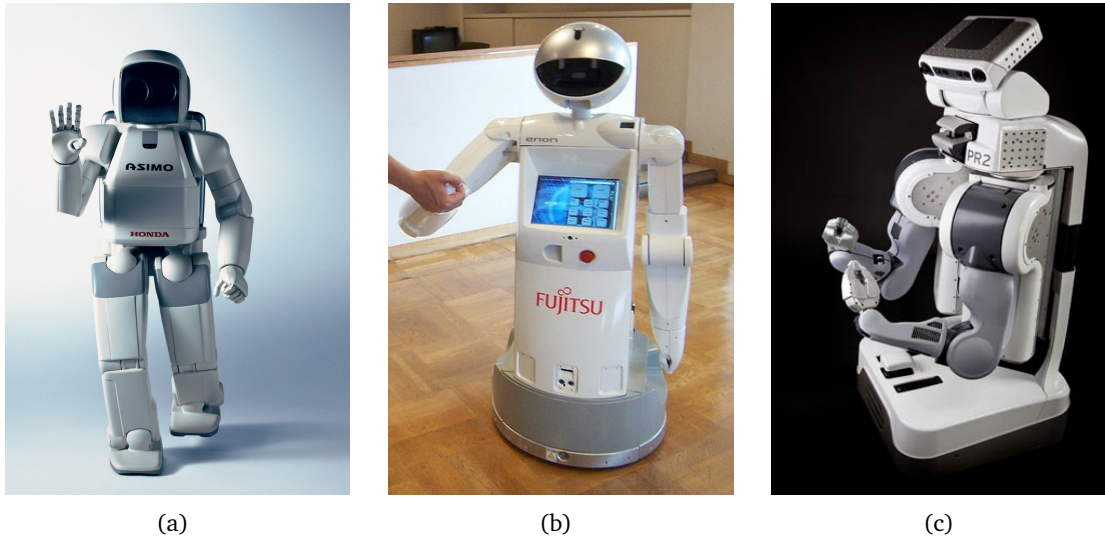


Figure 1.1.: Examples of personal service robots. (a) ASIMO, a humanoid robot created by Honda. The robot's size was chosen to allow it to operate freely in the human living space and to make it people-friendly. (b) Fujitsu Service Robot (enon) that is able to serve in multiple applications, including providing guide information, assistance, transporting items, patrolling, and surveillance. (c) PR2 (Personal Robot 2) provided by Willow Garage is an open and robust robot platform designed for performing useful work for human users.

spatial knowledge and incorporate them into the environment representation built by itself is crucial.

▷ ***Why not invite people into an interactive scenario?***

Recently, Hawes et al. [73] proposed an idea of gathering information from world wide web. Based on this idea the robot tried to identify what kind of rooms the robot was visiting according to the knowledge gathered from web, while autonomously exploring its environment. However, definitions of rooms found in web do not necessarily correspond to spatial understanding of individual human users. In addition, due to limited perceptual abilities of a robot unperceivable obstacles and potential hazards arising from, e.g. unexpected passers-by, are still challenges for the robot moving around in a human-shared environment. Thus, the robot is expected to extend its knowledge by gathering information interactively from its own user. A very effective way to introduce people in a previously unknown indoor environment is in a guided tour scenario. In other words, a person follows a guide, while the guide shows the person around and explains items. It would be an appropriate way for the robot to gain necessary information. In order to perform tasks given by the user, thus to interact with the user about the environment, the built spatial representation needs to be not only useful for the robot but also understandable to the user. With the help of an interactive scenario human spatial knowledge is possible to be integrated in the environment representation, except for the perceived sensor data, e.g. odometry and laser data. Advantages become obvious for the process of information gathering, since on the one hand the robot is able to gain its knowledge of the currently explored environment from the individual user. On the other hand, the gathered information may be adjusted by the user in the interactive scenario. Therefore, the participation of the human user is preferred. The user would perform an interactive scenario as a

guide for his or her personal robot.

Considering these questions an interactive *Home-Tour* scenario which will be described in section 1.1 has been designed. In this scenario the user guides a robot through an arbitrary indoor environment, while presenting general information to the robot, such as the names of rooms, objects in the environment, any comments to the environment, etc.. From the robot's perspective, during the *Home-Tour* it can build a map [146, 147] by means of the perceived metric information of the surroundings, create a conceptual spatial representation [56, 168] or a cognitive map [156] with the help of the user, and even obtain some friendly warnings or reminders like “*You are not allowed to enter this room!*” or “*Please take care of the door sill!*”². This corresponds more or less to what humans would like to do for somebody who is going to work in their house, e.g. a new employed butler. By this means, the robot can learn about the surroundings in the interactive *Home-Tour* scenario influenced by human ontology [5].

Apart from designing a comprehensive spatial model of the surroundings, first of all, the robot is assumed to behave in an indoor environment inhabited by its user. How should the robot *follow* or *accompany* the human user and move around in this environment to perform tasks assigned by its user? What should be taken into account for the acquisition of spatial knowledge in such an interactive scenario? How could make the robot move around safely in highly dynamic and populated spaces, e.g. an unconstrained person's home. These issues are going to be discussed emphatically in this thesis, and will be elaborated on in the following. Let's begin with the interactive *Home-Tour* scenario.

1.1. Home-Tour with a Mobile Robot

The basic idea of an interactive *Home-Tour* scenario can be specified as follows: a typical customer has purchased a robot delivered ready-to-use with all necessary capabilities for assisting in the household. Before the robot can provide any service, it has to familiarize itself with the working environment. As private environments are distinct from each other, little knowledge can be obtained in advance. Therefore, the robot is assumed without any a priori information about the new environment and has to adapt to it. Moreover, partial changes of the environment because of e.g. disarranged or newly bought furniture are inevitable. The robot needs to gather information of the current environment. Besides, considering the discussion of the two questions posed at the beginning of this chapter, the robot should interactively learn about the environment from the human user and incorporate the gathered information into the built spatial representation. In the interactive *Home-Tour* scenario the robot is able to learn basic information about the surroundings, while following³ the user. On the one hand, sensor data perceived by the robot can be obtained from the environment. On the other hand, spatial knowledge, e.g. names of different regions [86], locations [84], objects [153] and their spatial and temporal relations, as well as useful or helpful information from the user's perspective, is expected to be gained by the robot. Once the robot has acquired the knowledge about the surroundings, it can navigate the known part of the environment, autonomously explore the environment in order to fill any knowledge gaps[73], communicate and disambiguate with its human user about the environment[153], and provide services to the user.

²In the current work such warnings are not taken into account though, it would be important for further investigation in the context of robot's social behaviors.

³See details in chapter 2 for the proposed following behavior.

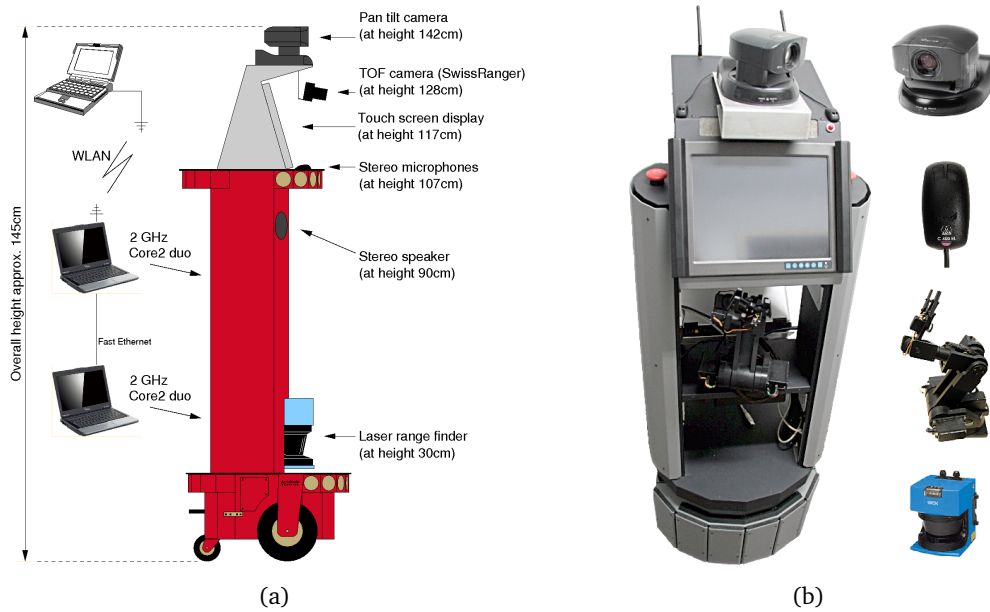


Figure 1.2.: Two generations of robot **BIRON**. a) shows the mobile robot **BIRON I** and the hardware components. Robot **BIRON II** with its hardware components is illustrated in b). From top right: Pan-/Tilt-camera, interfacial microphone, Pioneer 5DOF arm, and laser range finder.



(a)



(b)

Figure 1.3.: A regular apartment with standard furniture and the mobile robot **BIRON**. **BIRON I** placed in the living room is shown in (a), while (b) illustrates the new generation **BIRON II** in the dining room.

The designed *Home-Tour* scenario has been conducted with the mobile robot BIRON (Bielefeld Robot companiON) in a regular domestic environment, e.g. the BIRON apartment equipped with standard furniture [97], or an office environment [164]. Robot BIRON has been developed continuously at the Bielefeld University since eight years with the aim of providing personal services in a domestic environment. The first generation **BIRON I** [66] is illustrated in Fig. 1.2(a). It is based on the Pioneer2 PeopleBot from MobileRobots⁴, Inc. (formerly ActiveMedia). The robot uses one laser range finder mounted at the front at a height of approximately **30cm** for the perception of the surroundings in front of it. Measurements are taken in a horizontal plan covering a **180°** field of view. The color video camera is mounted for visual perception of the scene and for detailed focusing on persons, areas, and objects. For localization of sound direction two interfacial microphones are mounted on the robot's body at the height of about **107cm**. A 3D Time-of-Flight (ToF) camera [159], the SwissRanger SR3000 is fixed under the pan-tilt camera to acquire the depth information from the environment in real time⁵. All computation can be done on-board using two 2GHz dual core processors mounted on the back of the robot and one 1.5GHz Pentium Mobile notebook connected via W-LAN.

The second generation **BIRON II** (see Fig. 1.2(b)) based on the research platform *GuiaBot*TM developed also by MobileRobots. The robot platform customized and equipped with sensors allows analysis of the current situation in a human-robot interaction (HRI) scenario. In order to achieve a system running autonomously in a real world scenario, two piggyback laptops providing the computational power are mounted on the platform. **BIRON II** is a consequent advancement of the BIRON platform, which has been under continuous development since eight years. The robot base is a two-wheeled PatrolBotTM with two passive rear casters for balance. Similar to **BIRON I**, a laser range finder inside the robot base at a height of about **30cm**, a pan-tilt color camera on the top of the robot, as well as two interfacial microphones on the robot's body are mounted. In addition, the robot is equipped with a Pioneer 5 degrees-of-freedom arm for lightweight manipulation and referential gestures. The upper part of the robot's body houses a touch screen as well as the system speaker used for speech output in user interaction. Thus, on the one hand basic functions of the old platform **BIRON I** can be conveniently transplanted to **BIRON II**. On the other hand, new modules may be integrated into the new platform as well.

Fig. 1.3(a) and Fig. 1.3(b) illustrate the real BIRON apartment and the robot platform **BIRON I** and **BIRON II** respectively. Recently the robot has been evaluated several times in the real world [116] with naive subjects, who did not have to be experts in robotics. Gathering the experience and lessons from the experiments in practice, ToBI (Team of Bielefeld) has successfully participated in the RoboCup@Home of the RoboCup German Open 2009 and 2010 in Hannover and Magdeburg respectively, and the world championship RoboCup 2009⁶ and 2010⁷ in Graz, Austria and Singapore respectively. Fig. 1.4(a) and Fig. 1.4(b) shows the arena in Hannover and Graz.

RoboCup@Home focuses on real world applications and HRI (Human-Robot Interaction) with autonomous robots. As a new league inside the RoboCup competitions, RoboCup@Home aims at promoting the development of robotic technology assisting and providing services to humans in everyday life, which is a general goal for personal service robots. Subdomains of technology under RoboCup@Home are e.g. HRI and cooperation,

⁴See the home page www.mobilerobots.com for details.

⁵See section 4.3.2 for details of the SwissRanger camera, which extends the perception area of the 2D laser to 3D. It is important for obstacle avoidance in unconstrained indoor environments.

⁶www.robocup2009.org/

⁷www.robocup2010.org/



Figure 1.4.: **BIRON II** in RoboCup competition. (a) shows the arena of the competition in RoboCup German Open 2009. A dining room-like environment of the arena in the world championship RoboCup 2009 is illustrated in (b).

navigation, spatial representation and understanding, object recognition and manipulation, adaptive behaviors, system integration and standardization, etc.. Since ultimate scenarios should be conducted in the real world, the arenas of the two RoboCup competitions were decorated similar to a real indoor environment, as illustrated in Fig. 1.4(a) and Fig. 1.4(b). Objects made of glass like tea table can easily escape from some distance sensors, such as laser range finder and Time-of-Flight (ToF) camera [55]. Thus, it was difficult for the robot to avoid such unperceivable obstacles. In addition, the arrangement of the arena could be changed at random according to competition items. How the arena would look like was kept secret to all the teams. Hence, the robot had to be capable of learning models of the environment in a real world scenario, e.g. the *Home-Tour*. Otherwise, the robot cannot correctly perform tasks when it only worked on the basis of a static map of the surroundings. In the near future the arena of RoboCup@Home will not be limited in a basic home environment consisting of a living room, a kitchen, etc., but involve other places of daily life like a shop or a garden area. In fact, RoboCup@Home offers a good test opportunity for a personal robot aiming to perform tasks for the human user. The achievements obtained from the four RoboCup competitions have shown the basic navigation capability of the robot platform that will be discussed in the following of this thesis.

In general, the salient differences between a personal service robot and a robot applied in other areas are the object of service and the working environment. As the personal service robot is assumed to behave around humans in the common society, the environment representation built by the robot should correspond to human spatial understanding and be understandable to the human user. Hence, the human user needs to be considered in the learning process of the environment for the robot. With the help of the human user the robot is able to move around in the environment. By this means, where to go in the exploration of the unknown environment is solved. Besides, human spatial knowledge, e.g. where which room is and human navigation strategies implicitly contained in human movement trajectories, can be integrated into the environment representation built by the robot. Thus, how to facilitate interactive information gathering and exploit the gathered information to build a spatial representation becomes the central problem. It is exactly the meaning of the *Home-Tour* scenario, which provides the opportunity for the human user and the robot to interact with each other and cognize their common real world.

1.2. Background Knowledge and Related Work

In the frame of the interactive *Home-Tour* scenario the robot accompanying its user is able to gather information about the new surroundings with the help of the guide-person. In the following the basic definitions for this thesis are given concretely:

Pose A robot's pose (x, y, θ) is defined in a world coordinate system with the pure 2D position (x, y) and the robot orientation θ .

Goal A robot's goal is defined with a 5-tuple $(x, y, \theta, goal_radius, angle_diff)$, where (x, y, θ) shows the robot's pose, and the distance tolerance as well as the angle difference to the set robot orientation are constrained with *goal_radius* and *angle_diff* respectively⁸. In this thesis a goal may be sent to the robot in order to perform the person-following behavior (see section 2.2.2), to reach a *location* defined in the following, or may be selected from the observed pathways of the guide-person (see section 4.2).

Location A *Location* is a 2D point in an indoor environment, at which the guide-person names the region where he or she is according to human spatial concepts. *Locations* indicated by the guide-person and labeled with room (region) names in a *Home-Tour* scenario, e.g. kitchen, living room, hallway, etc., form the nodes of a topological environment representation. Thus, path planning and re-planning that will be discussed in section 4.2.1 can be implemented.

Path A path planned for a robot consists of a set of goals leading the robot from its current pose to the final goal, namely a specific *location* or a goal defined with (x, y, θ) .

Note that various environment models involved in this work are built in two-dimensional space. In the learning process three basic problems or tasks have to be dealt with or performed, i.e. "What does the world look like?", "Where am I?" and "How can I reach the goal?". Answering these problems is one of the most important capabilities and preconditions for a personal service robot to perform tasks assigned by its human user successfully. The three tasks are described in the following.

Mapping – "What does the world look like?"

Autonomous mapping in robotics addresses the problem of representing physical environments interpreted through sensor data. Therefore, the main challenges of a mapping process generally arise from the two central aspects, i.e. the representation of the surroundings and the sensors that enable a robot to gather information from the outside world.

Firstly, each sensor equipped on the robot is subject to measurement errors, not only for sensors (e.g. laser or sonar range finder) perceiving external information of the world, but also for sensors (e.g. dead-reckoning or odometry) providing internal information about robot movement, which accumulate errors over time.

Secondly, all environment entities that are being mapped might build a configuration space with extremely high dimensions. It would be very complex and difficult to tackle, especially when the map needs to be built in real time.

Despite sensor errors, enabling a robot to recognize a place previously entered and an object previously perceived is important for map update. Making a decision on

⁸In order to check whether a goal is reached, *goal_radius* and *angle_diff* are especially effective in practice. When the robot reaches a goal, stop the robot.

whether it is a new feature that should be added into the map or a re-recognized feature already existing in the map is crucial for the correction of the currently built map. It might be ambiguous for humans as well, when the environment is unknown or looks very similar in different places. Such ambiguities will occur when the robot closes a loop [64, 115] or matches the consecutively perceived sensor data [99]. This problem is generally known as the *correspondence problem* or the *data association problem* [114, 67].

Since the main task of mapping is to build a representation of the real world, changes, the scale and the complexity of the environment can all influence the mapping process. Early researches on mapping in large and non-stationary environments can be found in, for example, [61] and [1, 8] respectively.

Besides, range limitations generally existing in robot sensors (e.g. laser range finder) compel the robot to move around in the unknown environment in order to gain more information, other than staying at the start position and perceiving only a part of the surroundings. Once the robot moves, a path planning approach that enables the robot to explore the environment and the accuracy of the pose estimation according to the incomplete map model are of the greatest importance for robotic mapping.

Localization – “Where am I?”

Localization is the problem of estimating the current pose of the robot from sensor perception with respect to a world frame, i.e. the given map. For a personal service robot which should work reliably in an indoor environment, it needs to know where it is in order to provide services to the human user. As internal sensors (e.g. the odometry) accumulate errors over time, the robot has to infer its pose with the help of external sensors mounted on it, such as laser, camera, etc.. How to deal with the errors arising from robot perception and the representation of the surroundings is therefore crucial for a localization task. The solution of the mobile robot localization problem is various considering the main dimensions of this problem [152], as will be summarized as follows:

Local Localization It is also referred to as the position tracking problem [94, 65], in which the initial pose of the robot is known. Pose estimation is therefore confined to the region near the true robot’s pose and can therefore typically be approximated with a unimodal distribution (e.g. a Gaussian). In general, the increment error of robot motion should be relatively small for one step of the movement. Hence, the robot’s pose would be corrected in time when a new perception has been received.

Global Localization In contrast, global localization [20, 89] assumes that no a priori information of the starting robot’s pose is given. As a result the uncertainty of robot’s pose has to be unbounded at the beginning. A more difficult problem can arise when the robot has failed to localize in the environment, but it believes that its current pose estimation is accurate. Moreover, most of the state-of-the-art approaches cannot ensure the correctness of robot localization. Thus, for truly autonomous robots how to make them recover from such failures becomes essential [42, 150, 63].

Passive Localization Sometimes it is difficult for a robot to localize itself when it stands still at the same place and obtains the same knowledge about a part of the surroundings over time. Without the consideration of movement control in order to facilitate robot localization, the robot can only observe the environment

passively. It is actually a pure localization problem and generally known as passive localization [51, 90].

Active Localization Although the majority of approaches did not take into account the problems such as “*where to move*” and “*where to look*”, some researchers have aimed to the active localization [49, 79] problem. In their proposed approaches robot control has been partially or fully integrated into the localization process, in order to make the robot localize itself more efficient and robust. In practice whether the robot can move to the right location is probably crucial for the success of localization especially in situations, e.g. a symmetric corridor illustrated in Fig. 1.5. In this environment it would be impossible for the robot to achieve a unique correct pose estimation, unless it leaves the corridor and drives into one of the opening rooms arranged asymmetrically.

Localization in Static or Dynamic Environments Assuming that the environment does not change over time (static), the necessary estimated quantity is the robot’s pose. A lot of work has focused on localization in static environments [93, 21, 162]. However, most of the environments in the real world are dynamic, e.g. people, movable furniture, doors, daylight, etc.. In general, techniques dealing with localization in dynamic environments need to model dynamic objects perceived by the robot and filter them out when updating the pose estimation. Alternatively non-stationary objects have to be incorporated in the states of the environment and estimated together with robot’s pose. The former group of approaches are relatively easy to be implemented. Fox et al. [50, 52] have evaluated their approach in an environment populated with people. Sensor data can be filtered, so that the damaging effect of sensor readings corrupted by humans or unexpected changes in the environment is reduced. Therefore, the key to solve the localization problem rests in how to detect external unmodeled dynamics. In contrast, adding dynamic entities into the state vector may suffer from the computational complexity [157, 161], since an unknown number of dynamic objects have to be taken into account except for pose estimation.

Path planning – “*How can I reach the goal given the current location?*”

Fundamentally path planning addresses the problem concerning the selection of actions [152]. A planned path can be summarized as a sequence of subgoals leading to the final goal and a list of movement (actions) to be executed sequentially for each subgoal. The status of robot movement is fed back with sensor information (e.g. speed, acceleration, pose, force, etc.) over time, so that the control loop can compare the sensor measurements to the reference values given in the action command that will be executed at next step. Note that in this thesis subgoals and goals for a planned path were given in terms of the definition at the beginning of this section. Navigating to a given goal was the task assigned to the robot.

Once a path is going to be computed, a map of the environment and robot’s pose with respect to this map need to be available. A goal defined in the map is necessary for this plan as well. In general, the uncertainty may exist in many aspects, such as the representation of the surroundings, pose estimation of the robot, robot’s actions, robot perception, etc. [124, 157]. Moreover, the discrepancy between the built map and the real world [68] is inevitable due to dynamic objects in the environment, e.g. doors and people walking by, and errors during the mapping process. Hence, the major questions of path planning are centered on the following areas.

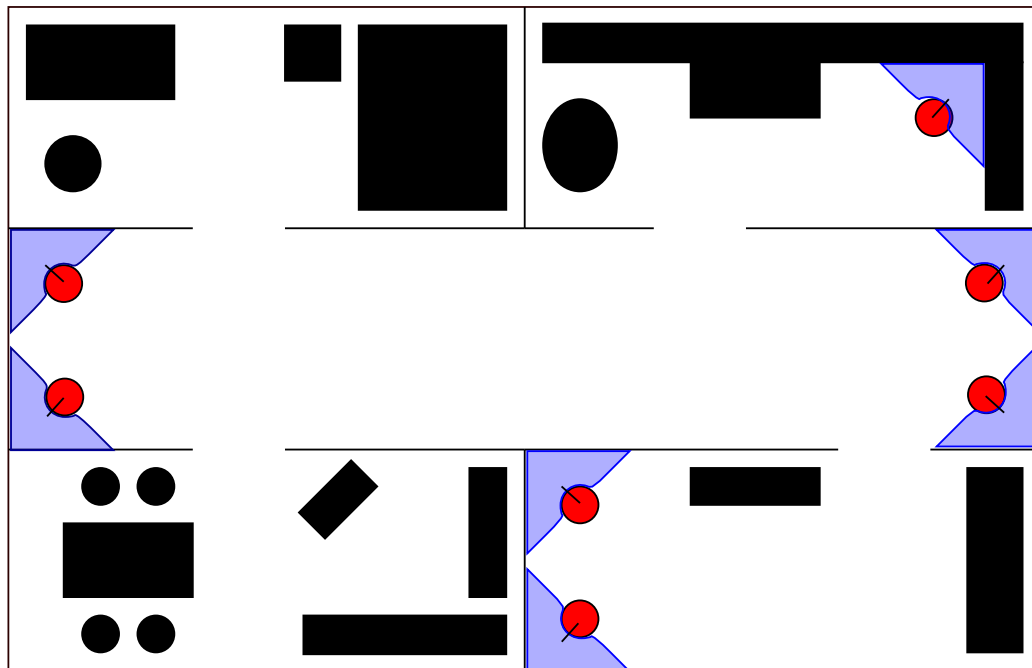


Figure 1.5.: A sketch map containing several positions, at which the robot obtains the same sensor perception. Various situations are illustrated with the robot depicted with a red circle and the corresponding sensor perception depicted with a blue area in front of the robot.

Firstly, *how to model the uncertain elements?* In fact, much work assumed that the map of the environment and the robot's pose were known precisely and robot motion (action) had a deterministic effect on path planning. According to this assumption a path from the current robot's pose to a given goal can be computed with a graph-based search algorithm (e.g. the A^* algorithm [37] and a breadth-first search in a graph [103]), on the basis of a navigation function defined in the so-called *configuration space*⁹ \mathcal{C} (such as the NF1 algorithm that is described in the book of Latombe [92] and will be introduced in section 4.3.1), or by learning reactive rules for planning in terms of the *salient states* [32], etc.. However, once the uncertainty is taken into account, the complexity of the planning task has to be dependent on the non-deterministic factors. When the robot's pose can be sensed at all times, i.e. the robot's state is fully observable, the uncertainty rests in robot motion. Such a paradigm that is able to generate a control policy including non-deterministic aspects in state transition is known as *Markov Decision Processes* (MDPs) [106]. In a more general situation the uncertainty of robot perception has to be considered as well. In other words, the robot's state can be estimated up to a certain degree (partially) due to the noisy sensor data. This case is generally referred to as *Partially Observable Markov Decision Processes* (POMDPs) [125, 109]. A family of techniques generating a control policy for the MDPs or POMDPs problem are on the basis of the iterated computation of a value function¹⁰. *Rewards* and *penalties* for all possible actions leading from the robot's position to the goal are evaluated by the value function under

⁹The \mathcal{C} -space gives all poses of the robot available in its working environment. The pose can be defined with x, y, z for translation and α, β, γ for rotation, the parameters of joints for robot arms in industrial areas, etc..

¹⁰An early description of value iteration goes back to the work of dynamic programming from Bellman [6].

uncertainty in robot motion (MDPs [22]) or in both robot perception and robot motion (POMDPs [72, 170]).

Secondly, *how to react or replan according to dynamic objects in the environment?* Since the global plan generated on the basis of the built map may become invalid because of dynamic obstacles, a local strategy handling with the objects that have not been modeled in the map is needed for the robot to complete the given plan or task. Therefore, most of the current systems for path planning, e.g. the motion generation that will be introduced in section 4.3.1, shall combine a high-level global plan [128, 19] with a local collision avoidance approach continuously reacting to obstacles perceived by robot sensors, such as the Dynamic Window Approach (DWA) [48], Nearness Diagram (ND) [107], Vector Field Histogram (VFH) [14], etc..

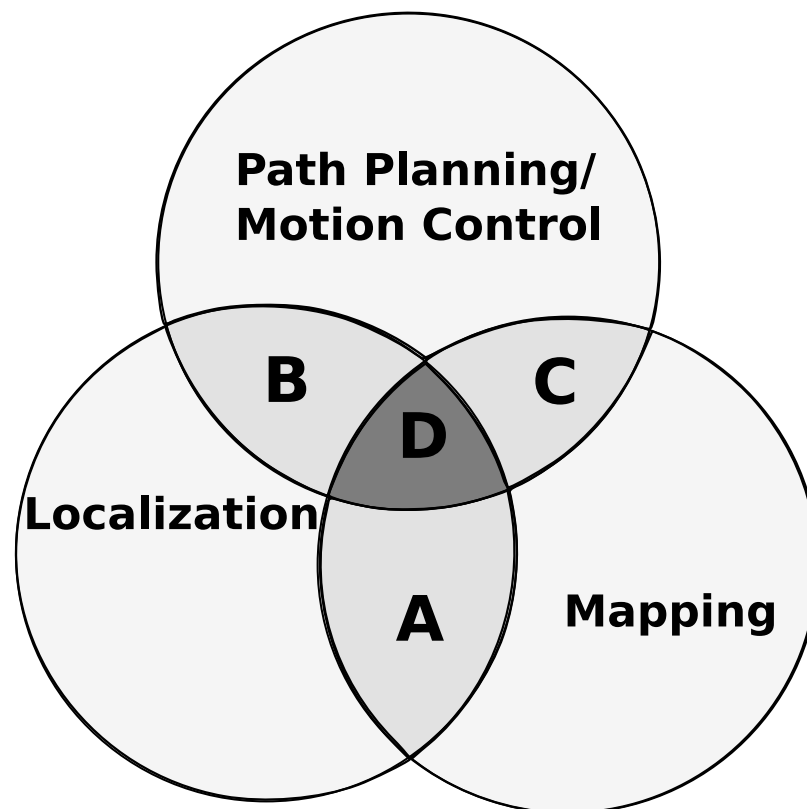


Figure 1.6.: Elementary tasks lying in robotic exploration and the combined problems [101] in the overlapping area marked with A, B, C and D respectively. (A) SLAM, (B) active localization, (C) classic exploration and (D) integrated exploration.

In general, the three processes introduced above are the basic issues in the frame of map-based navigation for mobile robots [45, 105]. Truly autonomous mobile robots need mechanisms to navigate purposefully in an a priori unknown environment [3]. However, as known from the description of the three basic tasks they are actually closely related. Firstly, planning a path towards a goal is tightly coupled with the current robot's pose and the knowledge of the environment at least between this pose and this goal. Next, the environment representation is needed to estimate the robot's pose, and conversely given a set of observations the robot needs to know from which locations these observations have been made, so as to build a map of the environment. Mapping, localization, path planning and their overlapped tasks are illustrated in Fig. 1.6 with different regions. As it is infeasible to provide a robot with the corresponding representations for every possible environment,

it is assumed that no a priori knowledge about a particular environment is available for the robot. Hence, one of the most important assignments for a mobile robot is to gather information of the environment with sensors and create a spatial representation. Since the perception of the surroundings is always related to the current pose of the robot, on the one hand, without localization it is impossible to add new features from the environment into the common world frame and go on updating the map. On the other hand, without the common world frame in which the environment is represented localization alone is also incredible, as the robot needs a reference frame to estimate its pose. Therefore, the problem of *Simultaneous Localization And Mapping* (SLAM) [38] is often known as a chicken and egg problem, which cannot be handled separately. Designing a control policy which can facilitate the pose estimation in a known environment is the goal for *active localization* [18]. If the control policy is created for building the representation of the surroundings with known robot's pose, it is a solution of *exploration for mapping* [135]. The central area which covers the three basic problems is generally referred to as *integrated exploration* and becomes more challenging [137]: *where should the robot go subsequently to obtain information from the environment and localize itself in the built representation effectively and efficiently?*

One of the basic capabilities of a personal service robot operating in an indoor environment is to solve the *integrated exploration* problem. In order to solve this problem, *Human-Robot Joint Spatial Exploration* (HRJSE) is proposed in this thesis to build a framework that allows people to help the robot complete the exploration of the environment initially unknown to it and makes the robot interactively learn from people to build a spatial representation. On the basis of this concept, human-robot interaction is able to be integrated with a general robotic mapping process. As the information incorporated in the spatial representation for the robot is dependent on the individual user in the individual environment, an appropriate interactive scenario is needed for the robot to acquire the corresponding information from the user. Motivated by the two questions posed at the beginning of this chapter, the *Home-Tour* introduced in the last section is assumed as an initial scenario for the robot to interactively gather information from the human user. By this means, the robot is expected to acquire a representation of an indoor environment which is not only useful for the robot to perform the basic tasks, e.g. localization and path planning, but also contains spatial understanding and navigation strategies of the human user. Various solutions to the *integrated exploration* problem based on a *predefined* exploration strategy have been proposed, such as the popular frontier-based exploration strategy used to identify areas of interest tending to lie on the edge of the explored and the unexplored regions [101, 163], which does not take into account the localization issue, the active loop-closing strategy detecting opportunities to actively close a loop [137], the information gain-based exploration strategy seeking a compromise between making the robot re-enter already visited areas to reduce the pose uncertainty and guiding the robot to unknown regions [138], etc.. In contrast to these solutions, a robot is capable of *achieving* an exploration strategy from the user in the context of the framework HRJSE.

Recently building a spatial representation for a robot in an interactive scenario has become a more and more attractive research direction. Many of the environment representations are based on the framework of *(Hybrid) Spatial Semantic Hierarchy* [86, 100] built from a metric map to an intermediate level, in which the environment is segmented into *unknown* regions, and finally up to a conceptual map, where human spatial concepts and understanding are integrated in the map with the help of human-robot interaction. Beeson et al. [5] have proposed a robotic *chauffeur* reasoning about the spatial organization with the route instructions (e.g. "rotate right", "go forward five meters", etc.) obtained from human passengers. However, compared to the interactive human joint *Home-Tour* the

activity of the robot would be impaired, if the robot can only passively obey the explicit descriptions for the route from human passengers. Besides, person's trajectories containing human navigation strategies (see section 4.2) were thoroughly ignored. The interactivity between a human user and a robot has been emphasized in the work of Zender et al. [169] and Topp [153] in an indoor environment. Compared to a sequence of detailed movement dictates the robot can track and follow the potential guide-person under some predefined hypotheses and communicate with the person to gather more comprehensive information about the environment in a guided tour. Similar with the *Home-Tour* the robot was able to build an environment representation together with its human user. However, the environment representation based on a metric map consisting of the line features extracted from the straight structures, e.g. the walls, in an indoor environment can provide a rather sparse description of the surroundings. Such a metric line map is not sufficient to fully support navigation actions, as details for free space in the environment cannot be provided in the built map. In addition, the navigation capability of the robot may be limited, as the path leading the robot to a set goal was computed on the basis of a *static* navigation graph. This graph was built with the nodes selected coarsely from the robot trajectory. Therefore, this thesis focuses on building a topological environment representation based on an occupancy grid map, which does not rely on predefined feature extractors. Furthermore, the grid map is able to provide a more robust and accurate mapping process especially in cluttered indoor environments. Besides, a person-following behavior of the robot in the interactive *Home-Tour* scenario, as well as human navigation strategies implicitly contained in the observed human movement trajectories are taken into account in this work.

More recently the Autonomous City Explorer project [4] has aimed to create a robot capable of navigating unknown urban environments without the use of GPS data or a priori map knowledge. The robot found its way to the set goal by repeatedly asking the pedestrians for movement directions towards this goal. By this means, the robot was able to explore an unknown outdoor environment with the help of human passers-by and enrich its knowledge about the surroundings. Hence, on the basis of the similar idea, i.e. *exploiting* humans to complete tasks assigned to a mobile robot, HRJSE would be appropriate as well for such an exploration in an outdoor environment. Of course, different strategies that should be used to perform individual scenarios, i.e. the *Home-Tour* and the scenario for asking the way, have to be proposed.

1.3. Contributions

This thesis seeks a solution based on *Human-Robot Joint Spatial Exploration* (HRJSE) to the problem concerning autonomous acquisition of spatial representations for a mobile robot operating in a human-shared environment and providing services to the human user. An interactive scenario, i.e. the *Home-Tour* scenario described in section 1.1, is therefore regarded as the cornerstone of information gathering, in order to make the robot not only learn models of the environment, but also exploit the information of the surroundings gathered from a guide-person.

In order to build the proposed framework, the integral parts and the extendibility of the whole system are described as follows:

- ▷ Person-following [131, 28] is a key issue for a robot to interactively acquire spatial representations with the help of the human user. Keeping up with the correct guide-person the robot is able to complete the interactive *Home-Tour* and obtain the infor-

mation from both the environment and the human user. Hence, how should the robot move when accompanying the user becomes crucial to achieve success in acquiring the environment representation interactively [58, 164].

- ▷ Once the robot can follow the guide-person to perform an exploration of the unknown environment, building a metric representation incrementally and estimating the pose of the robot with respect to the environment representation are the next task that the robot has to face. An occupancy grid map-based SLAM (Simultaneous Localization and Mapping) [31, 144] approach that should be appropriate for the interactive guided tour is therefore needed for the robot to acquire a basic (metric) environment representation.
- ▷ In general, a metric representation is the basis and the prerequisite for most of the current work that focused on building a hierarchical environment representation understandable and useful not only for the robot but also for the user in an interactive scenario, such as the guided tour or the *Home-Tour* [169, 153]. Therefore, a partially hierarchical structure for the environment model proposed by Topp [153] is able to be built on the occupancy grid representation adopted in this thesis as well. By this means, different spatial concepts, e.g. regions and objects belonging to the corresponding regions, can be integrated into the metric environment representation. This conceptual environment model would be beneficial to the robot, when the robot provides services to the human user.
- ▷ After building a metric environment representation the *woken up* situation, in which the robot is switched on in an arbitrary position and needs to localize itself in this previously built map, may occur in a practical scenario. Hence, the capability of global localization [151, 15] is necessary for the robot, and this shall make the whole system more robust in practice.
- ▷ To provide services, thus to move around and complete tasks assigned by the human user in the environment, planning a path leading the robot from the current pose to any reachable goal defined in the environment representation is necessary. Therefore, a path planning component based on not only the built map of the environment but also the knowledge gained interactively from the guide-person [166] has been integrated into the system. By this means, path planning is expected to benefit from human navigation strategies. Besides, a 3D capable navigation and obstacle avoidance system [165] is crucial especially when the potential hazards are unperceivable to the robot.

Considering the aspects discussed above three essentials consist the central of the framework HRJSE in this thesis. At first, a person-following behavior is needed to make the robot keep up with the human user and therefore explore the unknown environment. Once the robot can move around, a metric representation of this environment is able to be built incrementally. With the help of the interactive *Home-Tour* additional information can be gathered from the human user as well, apart from the knowledge obtained directly from the environment. By this means, an environment representation incorporating human spatial concepts and understanding can be built by the robot. Thus, the main contributions of this thesis follow below:

- ▷ A framework of HRJSE based on navigation capability of the robot and hybrid environment model for interactive acquisition of a spatial representation.
- ▷ *Where to move in the Home-Tour?* A novel person-following behavior which takes into account spatial context and person's positions relative to the robot.

- ▷ *What to look at?* A robust technique for map building that tackles the dynamic obstacles particularly perceived in the *Home-Tour*, i.e. the guide-person.
- ▷ *What to obtain from the guide-person?* A path planning approach that benefits from human spatial concepts and navigation strategies, as well as environment information.

1.4. Organization of this Thesis

The organization of the thesis follows rather closely the way to solve the central problem of this thesis, i.e. how to make a mobile robot interactively learn from its human user and build a spatial representation incorporating the gathered information?

Chapter 2 – Person-Following

The person-following behavior making the robot keep up with the guide-person in the *Home-Tour* is discussed. Considering the spatial context and the relative position of the human interactive partner with respect to the robot, a novel following behavior that enables the robot to automatically switch between three basic approaches in the interactive *Home-Tour* is designed and described in detail in chapter 2.

Chapter 3 – Map Building in a Home-Tour

Assuming that the robot can follow the user in the initially unknown environment, one of the most important tasks for the robot is to perceive the surroundings and build a basic representation of the environment, i.e. a metric map. An occupancy grid map based environment representation built by a RBPF-SLAM approach is selected and introduced in chapter 3. An extension of the state-of-the-art SLAM approach is proposed as well in this chapter, in order to reduce the negative effect of the guide-person perceived by the robot all along the *Home-Tour* on a mapping process.

Chapter 4 – Path Planning Adopting Human Strategies

Once an environment representation has been achieved by the robot, both the robot's pose and the surroundings can be illustrated in a common global coordinate frame. A goal can be defined as well in this frame for the robot. In the context of the interactive *Home-Tour*, making use of the knowledge gained from an analysis of the observed human trajectory robust path planning is expected to be achieved. The trajectory of the guide-person is stored in a graph, and a topological environment representation can be built upon the metric map. A map- and person's trajectory-based path planning approach is presented in this chapter. The general idea of multisensor-based motion generation is introduced as well in order to deal with the problem of safe navigation in an unconstrained 3D environment.

Chapter 5 – Evaluation

Overview of the experimental design including procedure of the user study and system architecture of the implemented system for interactive acquisition of spatial representations is presented. The user study was conducted by the subjects who used the robot platform for

the first time, in order to validate the robustness of the designed person-following behavior in the interactive *Home-Tour*, the practicality of the mapping process and the advantages of the proposed hybrid path planning approach combining the built environment representation with human navigation strategies. Experimental data collected for the assessment of individual system components, observations from a subject's perspective and evaluation results of the integrated system are discussed in this chapter.

Chapter 6 – Conclusions and Future Considerations

Chapter 6 summarizes the main aspects of the presented work. Conclusions that drawn from the results achieved in this work and future considerations potentially beneficial to the current system are discussed as well.

2. Person-Following

Recalling the basic idea of *Human-Robot Joint Spatial Exploration* (HRJSE), people have been invited in an interactive scenario, namely the *Home-Tour* introduced in section 1.1. In the *Home-Tour* the robot follows its human user in an indoor environment, while the user explains the surroundings to the robot. Thus, the robot can interactively learn and gather information from the user. To perform the *Home-Tour* and interactively acquire a spatial representation, the robot needs to keep up with the guide-person. Person-following actually builds a connection between the movement of the robot and the guide-person. By this means, the robot is able to move around in the environment and complete the interactive *Home-Tour* scenario. A person-following module is regarded as the basis for interactive acquisition of an environment representation, as on the one hand, it provides an opportunity for the robot not only to explore the unknown environment, but also to learn about the environment from the human user. On the other hand, it is possible for the robot to incorporate the interactively gathered information into the spatial representation built by itself. With the help of the *Home-Tour* scenario an additional exploration strategy¹ would not be necessary for the robot. Nevertheless, before building an environment representation and providing services to the user, the robot needs to know how to follow or accompany the guide-person.

Gockley et al. [58] proposed two person-following approaches, i.e. direction-following, where the robot always attempted to drive directly towards the current position of the tracked person, and path-following, where the robot tried to follow the person's path. They demonstrated that the direction-following behavior was more human-like and more closely matched the expectations of humans than path-following. However, people would probably change their following behaviors in the light of concrete situations in practice, rather than using a unique behavior. Imagine two persons walking in a building passing narrow passages, doors and open spaces. Their behaviors and relative positions are probably influenced by concrete situations in the environment. In narrow passages they have to walk in line, while in open spaces they prefer walking side-by-side to ease communication. Therefore, using a unique person-following behavior is generally not appropriate considering various contexts in the environment. The robot needs to change its behavior according to the contexts and keep a close watch on the person's movement, so as to effectively follow the guide-person in the human-robot joint exploration.

In narrow spaces the trajectory of the guide-person would lead the robot to avoid obstacles, when the robot reaches goals on the person's trajectory one after another. However, in open spaces the robot should walk with the person side-by-side in order to facilitate communication between the person and the robot about the environment. Besides, moving towards the guide-person can be regarded as an efficient way to keep up with the person, because the robot is able to react to the movement of the person as soon as possible. Thus, three elementary following behaviors have been designed, i.e. path-following, parallel-following and direction-following. By combining the elementary behaviors, an adaptive context-aware

¹Note that the exploration strategy mentioned here is used for the robot in the *Home-Tour* to build a spatial representation. Due to dynamics of the environment and completeness of the built environment representation, additional exploration strategies are necessary.

and effective following behavior, considering narrow spaces, facilitating human-robot interaction, and exploiting the human as a knowledgeable navigator, is expected to be achieved. However, different partially contradicting goals have to be considered. That is a trade-off between keeping up with the guide-person robustly in highly relative movements while perceiving the unknown environment, being engaged in communication, as well as avoiding obstacles and following the person in appropriate distance.

The remainder of this chapter is organized as follows: In the next section related work on person tracking and following will be discussed. Two prerequisites for a person-following behavior, i.e. person tracking and robot motion control, are described in section 2.2, before core strategies of the adaptive context-aware person-following combining three elementary behaviors is discussed in section 2.3. Afterwards, the elementary following behaviors are presented in section 2.4. Thus, the robot is able to automatically switch between these behaviors, according to core strategies dealing with various situations in a *Home-Tour* scenario. Finally the designed following behaviors are concluded in section 2.5.

2.1. Related Work

An important prerequisite of person-following is person tracking. In general various person trackers can be categorized into the following two groups: single-sensor-based methods, namely uni-modal anchoring, and multi-sensor-based methods, namely multi-modal anchoring. In the first group many researchers have used vision as the main sensor for person detection and tracking. Skin color detection [131], optical flow-based person tracking [119] and stereo vision-based approaches [7] are the most common methods belonging to this group. However, once the robot moves, the variation of lighting conditions in a real environment is prone to make a camera-based algorithms unreliable. In addition to camera-based algorithms many approaches using the laser sensor to look for leg hypotheses have been proposed by e.g. Montemerlo et al. [111]. In their work conditional particle filters were used to estimate the location of people conditioned upon the distribution of robot's poses over time in a previously mapped environment. Schulz et al. [129] used sample-based joint probabilistic data association filters to track people. However, only using laser information is not necessarily sufficient. It is difficult for a robot to confirm the correct target to be followed at the beginning, especially when several candidates, e.g. people or something else with two legs, are standing in front of the robot. Besides, active sensors, such as RFID [88] or other transmitters would be also viable for a person-following task, but people have to wear extra devices and this might be inconvenient in many applications.

Due to the shortages when applying a single sensor, multi-sensor-based approaches have been proposed as well. Feyrer and Zell [44] used a potential field for sensor-based fusion of vision and laser range data. Darrell et al. [30] defined some combination rules to fuse the results of individual algorithms from multiple visual modalities. Nevertheless, the robot is probably confused, when merely the face and leg information is available without an explicit instruction like "*follow me!*" from the real guide-person. Considering the disadvantages mentioned above, the multi-modal anchoring system (see section 2.2) from the work [91] combining camera, laser and microphone data is adopted.

Obtaining the position of the guide-person a robot can move to a goal computed in terms of this position, in order to keep up with this person. Since 10 years there have been many researchers engaging in the area of person-following. Most approaches performed the so-called direction-following [131], where the speed of the robot was generally adjusted by a

feedback of the control loop proportional to the distance between the robot and the person (see e.g. [24] and [28]). Thus, the robot can follow up the person as soon as possible, while maintaining space to avoid collisions. Calisi et al. [23] integrated appearance models and stereo vision for people tracking on the basis of a fixed stereo camera. They assumed that the robot had an a priori map of the environment and used laser-based Monte Carlo Localization (MCL) [2] to estimate the pose of the robot. However, this integrated approach is not appropriate for an unknown environment containing dynamic obstacles. Another drawback of this tracking system is that it is unable to distinguish the people and the objects in the background, when the colors of their clothes and the objects were similar to the target person.

In contrast to the approaches mentioned above, Gockley et al. [58] developed two person-following behaviors, i.e. path-following and direction-following. They tried to find out quantitative and qualitative difference between the two methods of person-following. Kruijff et al. [85] proposed a conceptual representation of the environment with situated dialogue in a guided-tour. However, the key module applied for person tracking was based on the laser sensor. Leg patterns were detected from the surroundings. The robot followed the trajectory of the guide-person like a path-following behavior and preserved a certain distance to avoid disturbing this person. More recently a situation-aware following behavior was proposed by Zender et al. [167]. In principle the robot moved directly towards the guide-person, except for some specific situations, e.g. in a doorway or in a corridor, given in a known environment map. Therefore, the robot can be aware of these situations and use the predefined strategy. Since an known environment representation recording detailed spatial knowledge and organization is necessary for this approach, it is impractical for a *Home-Tour* scenario, in which no a priori information about the environment is available. Considering the discussion about person tracking as well, we have developed the so called parallel-following behavior in addition to the path-, and direction-following behavior, so that the person and the robot can also walk approximately side-by-side in the light of spatial context. Note that some work has been done toward a side-by-side movement of a robot with a person [121]. Combining the velocity obstacle approach [46] with virtual target tracking, robot's speed (heading and velocity) can be computed and varied according to the movement of the accompanied person and the state of the environment. However, under the assumption that current positions and velocities of the robot and obstacles were given, the velocity obstacle approach would not be appropriate for the robot especially in a highly dynamic environment, e.g. a human-shared environment. In contrast, the reactive navigation approaches integrated in our robot systems (see section 4.2.3 and 4.3.1) are more effective and practical than the velocity obstacle approach. Furthermore, a hybrid approach based on the three elementary following behaviors have been designed. By this means, the robot is able to automatically switch its behavior according to the environment and positions of the followed person relative to the robot.

2.2. Person Tracking and Motion Control

Before introducing the adaptive context-aware person-following behavior, two prerequisites have to be discussed. At first, the robot needs to know which person should be kept up with, and then it can try to track the person. In addition, how to reach a goal in a person-following process without collisions must be considered. Dangerous situations for not only people in the vicinity but also the robot should be prevented by controlling the robot's speed.

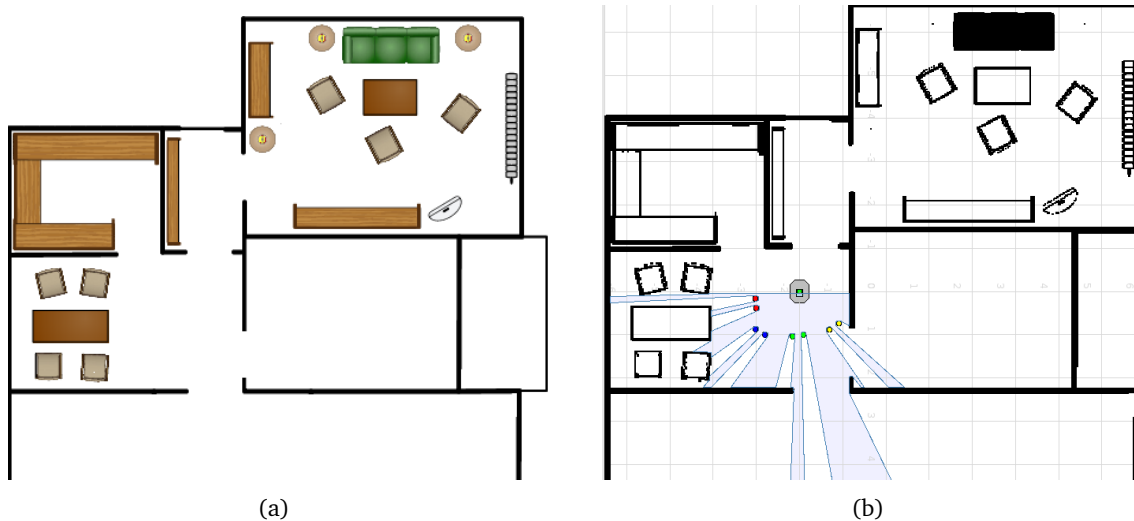


Figure 2.1.: The test environment for the person tracking system on the simulation platform *Stage*. a) shows the plan drawn from the BIRON apartment equipped with standard furniture. This plan, the robot, the four pairs legs of people simulated with small hexagons in front of the robot and the laser plane simulated with the light blue area are illustrated in (b). Notice that the color bitmap a) is read by *Stage* and represented with a black white environment model.

2.2.1. Multi-Modal Person Tracking

To perform a person-following behavior the robot needs to search all potential guide-persons around it and distinguish between different interaction partners. In the laser plane pairs of legs generally form characteristic patterns. These patterns can be detected [54]. In order to avoid including other objects forming the patterns similar to human legs, a pan-tilt camera is used to detect and recognize person's face. In addition, the stereo microphones are used to locate the direction of sound sources [57]. The person who intends to guide the robot around the new environment can directly give speech instructions². The robot will prepare to respond to a person, only when the pair of legs, the face as well as the sound source of this person are anchored at the same time by the corresponding sensors. Note that the person detected by the three signals, namely the legs, the face and the sound source, is assumed as a potential interaction partner, as a speech information is yet needed for the robot to make the ultimate decision on which person should be followed.

Fig. 2.1(a) shows the the plan of the BIRON apartment containing standard furniture, while Fig. 2.1(b) illustrates this plan simulated on *Stage*³. Because of a conversion of the input plan a black white environment model is obtained. Black lines and areas in this figure correspond to obstacles which the robot need to avoid. Four pairs legs of the persons simulated with small hexagons are marked with different colors in front of the robot. The 180° perception area of the laser sensor is illustrated with light blue. *Search*, *Tracking* and *Person Lost* are basic states of the state machine in the person tracking system. The state machine is switched to *Tracking* when person-following is triggered. Outputs of the person tracking system obtained from Fig. 2.1(b) are illustrated in Fig. 2.2. The simulated

²English and German instructions are available for the current system.

³*Stage* is a 2D multiple-robot simulation platform from the *Player* project. With this simulator the robot sensors and actuators, as well as objects can be simply simulated in a two-dimensional bitmapped environment instead of a run on a real agent. For details please see <http://playerstage.sourceforge.net/>.

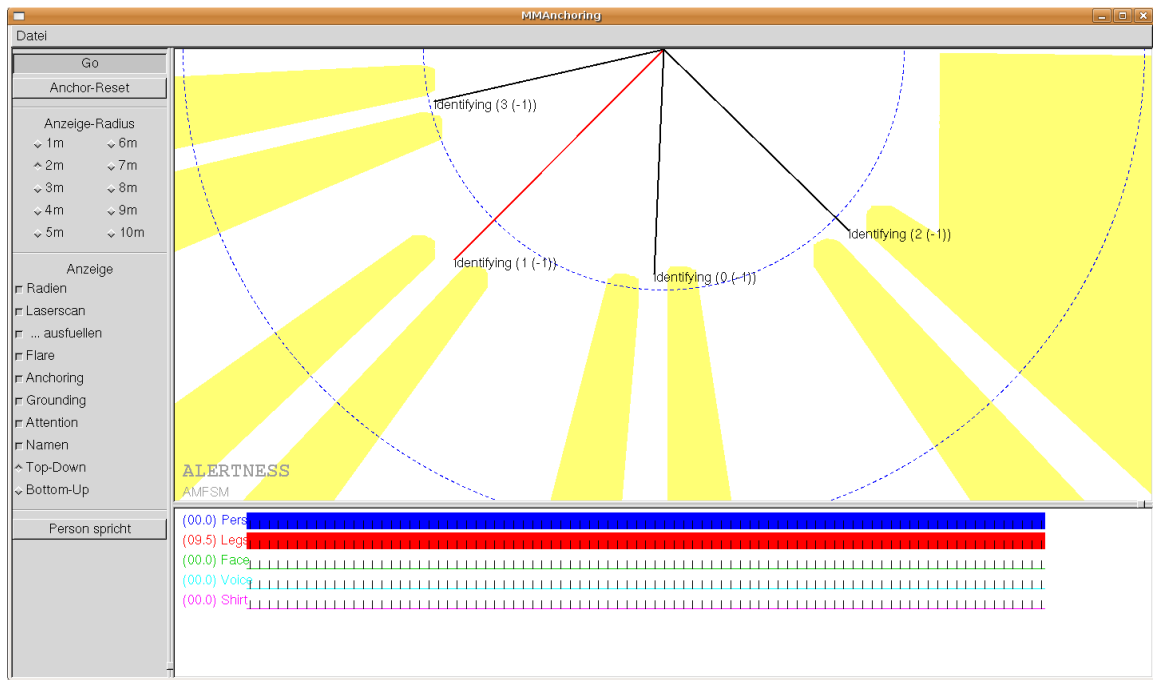


Figure 2.2.: Results of the person tracking system corresponding to the situation illustrated in Fig. 2.1(b).

person currently tracked by the robot is shown with the red line, while the other three candidates indicated with black lines stay in state *Search*. Person's positions outputted by the tracking system are expressed with polar coordinates in the robot frame. Angular coordinates are defined in rad from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. A 0 rad or 0° angle indicates that the person stands right ahead the robot. Notice that each candidate has an identifying number shown nearby. Hence, positions of all candidates recorded in the tracking system can be obtained according to the respective numbers. Once the guide-person is fixed, person-following is triggered. The red line indicates the currently tracked person till the robot loses the target and the state machine of person tracking is switched to *Person Lost*. Only the information of the tracked person and the legs belonging to this person are illustrated with the red and blue bar under the main graphic window shown in Fig. 2.2, because the camera and the microphones are not simulated by *Stage*.

The person tracking system would lose the guide-person when the person moves relatively quickly with respect to the robot in a short time, or the person is outside the perception area of the laser sensor, i.e. a semi-circle with radius $8m$ for the current system. When the tracked person is lost, the state machine of person tracking is switched to *Person Lost*. The guide-person will be prompted by the robot. The robot will restart to follow the guide-person when the person standing in front of the robot is identified by the tracking system with face, voice and legs information. Afterward, person-following is triggered with a speech command⁴ sent by this person.

⁴See section 5.1.2 for details about prompt of the robot and speech commands of the guide-person.

2.2.2. Motion Control

Assuming that the position of the person in the robot frame is known from the output of the person tracking system, the person's position can be calculated in a global coordinate system, in which the current robot's pose is expressed. This position in the global coordinate system is defined as a goal for the robot in direction- and path-following (see section 2.4.2 and 2.4.3 respectively), while a robot's goal in parallel-following (see section 2.4.4) is computed on the basis of this position. As defined in section 1.2 a robot's goal is given by a 5-tuple $(x, y, \theta, goal_radius, angle_diff)$ in the global coordinate system, where (x, y) shows the 2D goal-position. The goal-direction is marked with θ . The distance tolerance as well as the angle difference to the set goal is defined by *goal_radius* and *angle_diff* respectively. The global coordinate system mentioned above is dependent on the coordinate system in which the robot's pose is defined. For the *Home-Tour* scenario pose estimation of the robot is given in the world frame defined by the mapping process (see chapter 3). Alternatively robot's pose can be obtained directly from the intrinsic sensor, e.g. the odometry. Although the accumulated error of odometry increases with the moving distance of the robot, if the goal with respect to the current robot's position is not far and can be updated frequently⁵, the robot is able to keep up with the guide-person [164].

After a robot's goal is computed according to the current person's position, the robot needs to move towards this goal without collisions. As introduced in section 1.2, in general motion planning approaches consist of global planning on the basis of a priori knowledge of the environment and local reactive planning based on sensory information. Under the assumption that the environment is initially unknown to the robot, a global path plan would be impractical at the beginning of a *Home-Tour* scenario. When the robot has completed the exploration of the environment with the help of the guide-person, a global path can be planned for the robot, as an environment representation has been available (see chapter 4 for details). Therefore, a reactive navigation approach that enables the robot to avoid obstacles and reach goals computed in the *Home-Tour* is needed for a person-following behavior.

Another issue to consider is the fact that person's legs tracked in the laser plane would normally be interpreted as obstacles. However, for the following behaviors proposed in this thesis it is required that the robot reaches goals computed according to positions of the person. Hence, reaching goals and avoiding obstacles have to be both considered for the robot performing the *Home-Tour*. It is needed for the reactive navigation approach used in the proposed person-following behaviors not to be impeded by the presence of the person as an obstacle⁶.

2.3. Adaptive Context-Aware Following Behavior

Recalling the introduction of the elementary following behaviors presented at the beginning of this chapter, no one behavior is generally suitable in various situations in the interactive *Home-Tour*. Nevertheless, each elementary behavior has its own characteristic, which is appropriate in a certain situation. In view of human-human following behaviors generally changed according to contexts in the environment, this section aims at designing adaptive context-aware person-following by combining the three elementary following behaviors.

⁵See section 2.4 for strategies of the elementary person-following behaviors.

⁶See section 4.3.1 and section 4.2.3 for the introduction of the obstacle avoidance approach integrated in **BIRON I** and **BIRON II** respectively.

To achieve this goal, core strategies taking into account the advantages of the elementary behaviors in specific situations are discussed in the following.

2.3.1. Core Strategies

Recently combining different following behaviors to enable a robot to accompany a guide-person has been proposed as a long-term goal by Gockley et al. [58]. In addition, an accompanying behavior has been implemented in [121] taking into account situations like narrow doors and narrow hallways. However, positions and velocities estimated for all obstacles perceived by the robot are difficult to be achieved especially in highly dynamic indoor environments.

In general, the key problem of the hybrid approach, i.e. the adaptive context-aware following behavior, proposed in this thesis lies in how to combine the advantages of the three elementary behaviors according to the information obtained from the surroundings, i.e. distance to obstacles measured by the laser range finder mounted on the robot⁷. Considering that person's legs are tracked by the laser scanner as well (see section 2.2.1), environment and person information can be gained both from the laser plane. As a large person's angle with respect to the robot orientation probably results in the state transition *Person Lost* especially when relative movements exist between the person and the robot⁸, a front field (FF) is defined in the laser plane to give an *early warning*. The guide-person should be kept in FF as far as possible, in order to prevent the person tracking system from losing the guide-person. Therefore, the laser plane is divided into three parts, i.e. one front field and the rest two side fields. Core strategies are different as well, depending on whether the guide-person is tracked in the defined front field FF.

Another issue to consider in the core strategies is the appropriateness of the elementary behaviors in certain situations. The robot may keep up with the guide-person more efficiently when direction-following is adopted, because the robot always tries to move in the direction of the current person's position. The angle to the person with respect to the robot orientation can be reduced as well, considering the characteristic of direction-following. Therefore, direction-following is regarded as the most effective behavior to keep the guide-person in FF. Path-following would be more proper, when the guide-person is passing narrow passages, or obstacles are perceived in the vicinity. Hence, it is regarded as the safest behavior as the robot follows the path that the guide-person has just passed by. In contrast to direction- and path-following, the robot should accompany the person to ease communication when free spaces can be detected on at least one side of the person, e.g. the robot moves through wide passages or in open spaces. Free spaces found around the current position of the guide-person is a precondition for the choice of parallel-following⁹.

Besides, situations in an indoor environment, e.g. open spaces, wide, narrow and door-like passages, cluttered regions, are especially considered with characteristics of the elementary behaviors. Fig. 2.3 shows the core strategies in switching behaviors. Unexpected dynamic obstacles, e.g. passers-by, might pass between the person and robot in a real world scenario. Considering the ability of reactive navigation (see section 4.2.3 and 4.3.1), in general the robot is able to avoid obstacles in highly dynamic environments. Hence, obstacle avoidance

⁷See section 1.1 for the introduction of the robot platform.

⁸As mentioned in section 2.2.1, the person tracking system would lose the guide-person when the person is outside the perception area of the laser, i.e. a semi-circle area.

⁹Note that obstacles are checked in the laser plane according to the defined three parts and the area around the guide-person for parallel-following. See the description of the core strategies later on.

is not a special issue taken into account in the core strategies. Potential hazards between the robot and person defined with circle **OCP** in Fig. 2.3 are not especially checked either. In terms of the appropriateness of the elementary behaviors, as well as person's positions and obstacles perceived in the defined three parts of the laser plane, basic rules are elaborated on as follows:

- ▷ The perception area, i.e. the laser plane, is divided into three parts, as shown in Fig. 2.3(a). One front field (**FF**) defined with 2α illustrates a region, in which the guide-person should stay as much as possible. Actually α gives a threshold of the person's angle in the robot coordinate system, so that a large angle between the person and the robot can be reduced as soon as possible by choosing an appropriate behavior. Except for the front field **FF** two side fields **LSF** and **RSF** are defined as well. Especially when the robot is guided to move through a narrow passage, obstacles lying near the robot are probably detected in the two side fields.
- ▷ The area occupied by the guide-person (**OCP**) is represented with a circle centered on the current person's position P . r_P gives the radius of circle P according to the distance between the two legs of the person detected by the person tracking system.
- ▷ The check-regions on the left and right side of **OCP** with respect to the robot orientation are marked with **CR_P_L** and **CR_P_R** respectively. Free spaces are checked in the two regions for the choice of parallel-following. The two yellow boundaries on the left and right side of **OCP** are tangent to circle P . The areas detected in **CR_P_L** and **CR_P_R** are given by angle γ in the laser plane. The calculation of γ is illustrated in Fig. 2.3(b) in detail. The two boundaries **P_{LL}** and **P_{LR}** on the left side of circle P are tangent to circle PL . Notice that **P_{LR}** is tangent to circle P as well. Similarly the two tangents of circle PR on the right side of circle P are depicted with **P_{RL}** and **P_{RR}**. Angle γ can be calculated simply with the following equations:

$$\theta = \arctan \frac{r}{\sqrt{|\overrightarrow{RP}|^2 - |r_P|^2}} \quad \text{with } |\overrightarrow{RP}| > |r_P| \quad (2.1)$$

$$\gamma = 2\theta. \quad (2.2)$$

In order to reserve free space for the robot when it reaches the goal **P_L** or **P_R**, radius r is chosen according to the real size of the robot. The check-regions in **CR_P_L** and **CR_P_R** illustrated in Fig. 2.3(a) are therefore defined with sector areas **SEC_P_L** and **SEC_P_R** respectively. The radiuses for the two sectors¹⁰ are both $|\overrightarrow{RP_L}| + r$. As a condition for using parallel-following, no obstacles in **SEC_P_L** and **SEC_P_R** are checked by the laser sensor.

- ▷ When the person moves in front field **FF**, facilitating human-robot interaction is considered combined with specific situations in the environment. The robot conservatively chooses path-following in case obstacles in **LSF** and **RSF** are both near to it, e.g. when the robot is guided in a narrow passage, obstacles are likely to be detected both in **LSF** and **RSF**. Otherwise, free space is found at least on one side of the robot. Path-following, direction-following and parallel-following may all be regarded as candidates. If the guide-person is going through a door-like passage, obstacles are in proximity to the person. These obstacles are probably detected in both **CR_P_L** and **CR_P_R**, namely the sector areas **SEC_P_L** and **SEC_P_R** depicted with yellow lines in

¹⁰According to the geometrical relation of the robot **R**, the person's position **P** and the centers of the two circles **P_L** as well as **P_R**, it is simply to obtain that $|\overrightarrow{RP_L}| = |\overrightarrow{RP_R}|$.

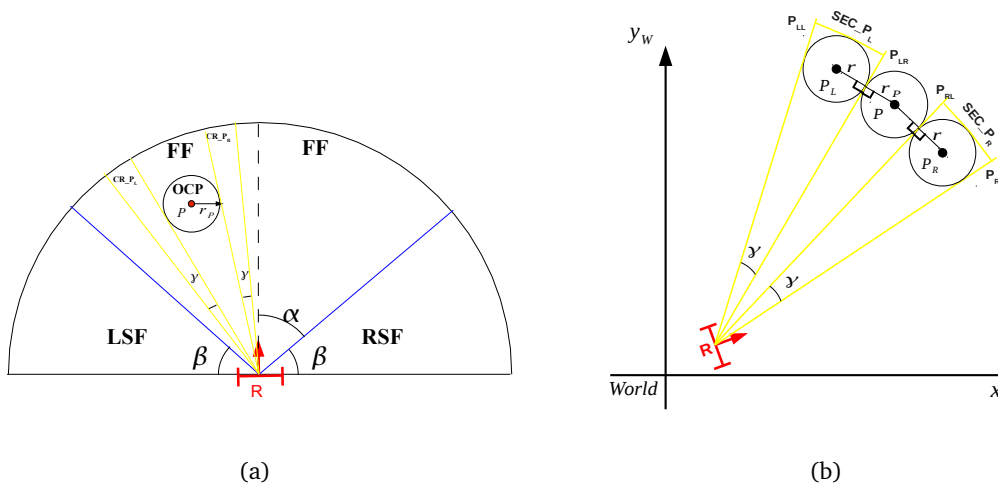


Figure 2.3.: Description of the core strategy with which the robot can switch following behaviors autonomously. (a) illustrates the perception area of the robot R , the current person area OCP , the predefined check-regions LSF and RSF on the left and right side of R , and FF in front of the robot, as well as the check-regions CR_{P_L} and CR_{P_R} defined by γ on the two sides of OCP . The calculation of γ is showed in (b) at length. The check-regions on the two sides of the person are illustrated with the two yellow sector areas SEC_{P_L} and SEC_{P_R} .

Fig. 2.3(b). Path-following is therefore chosen for the robot. By this means, the robot is able to follow behind the guide-person, and adjust from an improper behavior, e.g. parallel-following, as soon as possible¹¹. If free space can be detected in CR_{P_L} and/or CR_{P_R} (keeping in mind that free space is detected at least in one side field of the robot), e.g. the robot moves through wide passages or in open spaces, parallel- and direction-following are candidates. Note that the computation and decision of a robot's goal are carried out in parallel-following. When no goals can be computed by parallel-following, core strategies temporally choose direction-following¹² on the one hand. On the other hand, check-information gathered from parallel-following is concurrently inquired, so that parallel-following can be chosen once a goal is computed.

- ▷ When the person is out of the front field, the robot needs to bring its target back to FF again. Direction- and path-following can be selected, as the relative angle between the robot and the person possibly increases when parallel-following is adopted. Direction-following will be carried out, as long as no hazards lying in the whole laser range plane, i.e. LSF , RSF and FF , can be detected in the vicinity. Otherwise, cluttered obstacles are implied near the robot, and path-following will be performed for the robot to follow the path chosen by the guide-person.

The guideline for design of the adaptive context-aware following behavior is made by combining safe navigation with specific situations in an indoor environment and appropriate-

¹¹In principle interactive navigation enables the robot to avoid obstacles detected in the laser range finder without switching behaviors. However, the overall goal is to produce an effective, situation-adequate following behavior, exploiting human spatial knowledge, considering narrow spaces and facilitating interaction.

¹²As a goal in parallel-following is computed on the left or right side of the person with respect to the person's movement direction (see section 2.4.4), it is possible that parallel-following is chosen according to the current spatial context, but the robot has to wait for an appropriate goal (see algorithm 2).

ness of the elementary behaviors. Adopting the core strategies the robot was able to change its behaviors according to the guideline in the user study conducted by inexperienced subjects in real world scenarios¹³.

2.4. Elementary Person-Following Behaviors

The two important prerequisites for a person-following behavior have been discussed in the section 2.2. In this section elementary person-following behaviors constituting the adaptive context-aware behavior are elaborated on. Behavior changes for a mobile robot in a person-following scenario have been discussed in [58] and [121]. Imagine a human-human following scenario, their behavior might be changed sometimes to adapt spatial restrictions, such as narrow passages and open spaces. In order to facilitate and support communication between the guide-person and the robot in the interactive *Home-Tour*, changes of the robot's behavior according to spatial context need to be taken into account in a real environment. Considering human-human following schemes and specific situations in an indoor environment mentioned in section 2.3.1, three basic behaviors, i.e. direction-, path- and parallel-following, are assumed as the elements capable of constituting a more complex person-following behavior.

2.4.1. General Strategies

Before the individual person-following behaviors are discussed in detail, common strategies adopted in all behaviors are firstly extracted. In this way, the characteristic (distinction) of one basic behavior would become more prominent compared to the others.

Departure and Arrival Except for the pure person-following phases that will be described in the next sections, *departure* and *arrival* are designed for the individual behaviors. When should a person-following behavior start (*departure*) and how should the robot react when the guide-person halfway stops (*arrival*) are two episodes belonging to the whole person-following process. The same rules for departure and arrival are used in the three basic person-following behaviors. The robot starts to follow a person (*departure*), if the person standing in front is identified by the person tracking system, and person-following is triggered. In order to prevent the guide-person from dangerous situations, the robot just turns to the person without translational movements as long as the distance to this person is greater than a threshold $T_{ShortDis}$. According to the study of interpersonal distances in human-human interaction (HHI) by Hall [69], *personal distance* implying that extremities can be used to get in contact with each other is limited in the interval [46cm, 122cm]. Moreover, the level of familiarity of one person with another is marked by the exact distance. Although the results of interpersonal distances are based on HHI, it has been generally concluded that human participants preferred *personal distance* when interacting with a robot [77]. Therefore, threshold $T_{ShortDis}$ is set to 1.2m for the three basic following behaviors in the user study (see section 5.2.1.).

Arrival deals with what the robot should do if the distance between the robot and the person is less than threshold $T_{ShortDis}$ defined for *departure*. During a *Home-Tour Arrival* can occur, e.g. when a speech command for stopping the robot is sent

¹³See details in section 5.2.1.

from the guide-person and accepted by the robot, or the guide-person stops to show the environment while keeping a short distance (within $T_{ShortDis}$) to the robot. For *Arrival* the robot speed is set to zero, so that the robot can brake in time and does not cause danger to the guide-person. Once the translational speed of the robot has been reduced to zero, a rotational speed is set in order to make the robot turn to the person until the angle of the person with respect to the robot is less than $angle_diff$ defined in section 2.2.2. The sequence of actions, i.e. the robot brakes and turns to the person, is called *short distance behavior*. Only rotation is allowed for the robot in *short distance behavior* unless the person moves far from the robot, i.e. the distance between the person and the robot exceeds threshold $T_{ShortDis}$.

Interpolated Robot's Pose As positions of the person with respect to the robot frame and global robot's poses are output with different frequency, at first, the robot's pose at each time stamp is interpolated to the time point when a person's position is obtained, applying equation 2.3 and 2.4.

$$\overrightarrow{RPose_{t_{Person}}} = \overrightarrow{RPose_{t_1}} + \left(\overrightarrow{RPose_{t_2}} - \overrightarrow{RPose_{t_1}} \right) \cdot \lambda \quad (2.3)$$

$$\lambda = \frac{t_{Person} - t_1}{t_2 - t_1} \quad \text{with } t_1 < t_{Person} \leq t_2 \quad (2.4)$$

Where $\overrightarrow{RPose_{t_{Person}}}$, $\overrightarrow{RPose_{t_1}}$ and $\overrightarrow{RPose_{t_2}}$ is the pose of the robot at time stamp t_{Person} , t_1 and t_2 respectively. Then the person's position is computed in the world frame according to the interpolated robot's pose at the time stamp t_{Person} , as if the two data streams were synchronized.

Go back to Robot In the interactive *Home-Tour* the robot follows the guide-person generally along the movement direction of the person. However, the person is allowed to go back to the robot if it is necessary, e.g. the person would like to come back to see what happened, when the robot moves too slowly. For safety the robot would stop and turn to the person (similar with *short distance behavior*) and wait for the next goal in terms of the movement of the person. The criterion judging whether a person moves back to the robot, is explained in Fig. 2.4, assuming that the guide-person has moved from P_1 to P_2 in the world frame¹⁴. All positions of the person P satisfying the following inequation are regarded as *go back to robot* starting from person's position P_1 at a previous time stamp.

$$\left| \overrightarrow{RP_1} - \overrightarrow{RP} \right| > T \quad \text{with } T > 0 \quad \text{and} \quad \left| \overrightarrow{RP_1} \right| > \left| \overrightarrow{RP} \right|, \quad (2.5)$$

where R is the position of the robot in the world frame and T is the minimal distance between P_1 and a p inside the gray circle area illustrated in Fig. 2.4. Notice that person's position P_2 marked in this figure is an element of the set defined by equation 2.5.

Stop the Robot due to Person Lost Sometimes it is inevitable that the multi-modal person tracking system loses the target, particularly when the relative speed between the person and the robot is high, or the person is outside the range of robot sensors (the laser plane in our case). The state in the state machine of the person tracking will transit to *Person Lost*. Theoretically when the person is lost by the tracking system, the last person's position stored as a goal for the robot should be sent out. It is especially important for path-following, in which the robot reaches goals extracted from

¹⁴The position of the person is updated about $10Hz$. Because the person is unlikely to move far in $0.1s$, P_2 is recorded as a *real* movement of the person only when $\left| \overrightarrow{P_1P_2} \right|$ is greater than a predefined threshold.

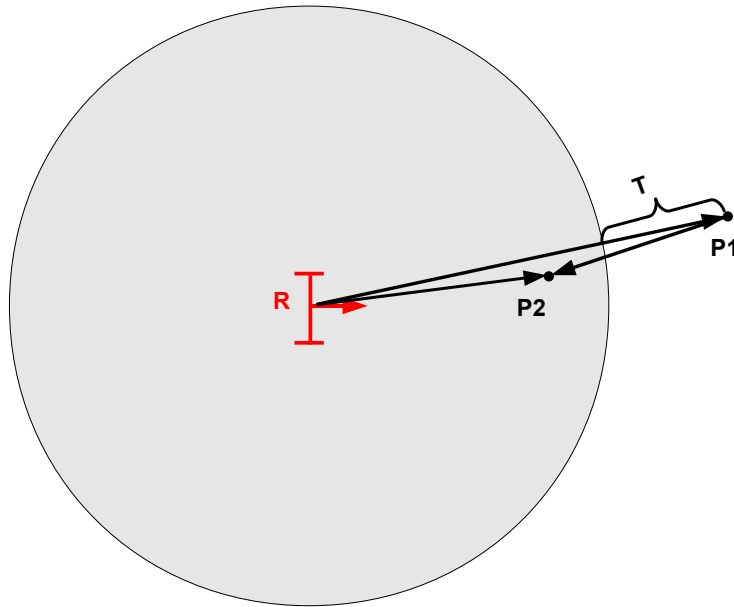


Figure 2.4.: Geometric representation of equation 2.5 judging *go back to robot*. The robot is depicted with the red R . Assuming that the person has moved from P_1 to P_2 . The gray area inside the circle with radius $|\overrightarrow{RP_1}| - T$ illustrates possible positions of the person satisfying this criterion.

the path of the person one after another. By this means, the robot is able to keep up with the lost person as far as possible. After the robot reaches the last person's position, it stops to wait for the presence of the guide-person. A process of person re-recognition [75] by means of face identification combining the pattern information of people's clothes has been proposed¹⁵. The robot would try to find out the person whom it has just followed from one or several candidates, and then interact with the person according to the result of the person re-recognition module.

However, the last goal computed before *Person Lost* is confirmed by the person tracking system probably informs a spurious position not on the real path of the person¹⁶. That is because the person tracking system outputs the same person's position for some time, before the ultimate *Person Lost* is confirmed. It is important for the tracking system to find corresponding target with respect to the last tracked position in a short time span¹⁷, which is crucial as well for the robot to perform the interactive *Home-Tour*. Note that *Person Lost* cannot be determined only by the same person's positions output by the person tracking system, because the same situation occurs when no relative movement exists between the person and the robot. Therefore, an online detection is difficult to be performed¹⁸. In order to keep the tracking ability and prevent spurious positions of the person, the robot simply stops to wait for the subsequent instructions given by the person, when *Person Lost* occurs.

¹⁵The module of person re-recognition has been conceptually evaluated with the pan-til camera equipped on **BIRON I** (see Fig. 1.2(a)) and **BIRON II** (see Fig. 1.2(b)). In the immediate future it will be completely integrated in the robot system.

¹⁶An example is illustrated in Fig. 5.8 in section 5.2.1.

¹⁷Please see [91] for details.

¹⁸*Person Lost* was recorded in the user study presented in chapter 5. See section 5.2.1 for an offline detection.

2.4.2. Direction-Following

The simplest method of following a person is to make the robot drive in the direction of the guide-person. In order to design an efficient behavior, a goal for the robot is defined at the current person's position. Moreover, robot's goals are updated as soon as the guide-person moves. Thus, the robot is able to react to the person's movement as fast as possible.

Fig. 2.5 illustrates some implementation details to achieve the direction-following behavior proposed in this work. A goal for the robot is defined with the position (Px, Py) , at which the person currently is. By means of the goal-orientation θ computed from the current robot's position to the current person's position, the robot always tries to move in the direction of the person. In principle the person's position should be computed in the world frame as a new robot's goal, as soon as the person moves. However, the goals that are really sent to the robot do not need to be updated so frequently as the frequency (circa 10Hz) of the person tracking system, because the person cannot move far in such a short time (circa 0.1s). Hence, a threshold of time difference $T_{TimeDiff}$ is defined to reduce the unnecessary high frequency of the goals sent to the robot. When the robot moves in proximity of the person, the *short distance behavior* described in the last section will be performed to make the robot brake and turn to the person. After the robot faces to the person, i.e. the person stands still and the angle tolerance *angle_diff* is reached, no goals need to be sent out anymore until the person moves again. The robot starts to follow the person (*departure*), as introduced in the last section.

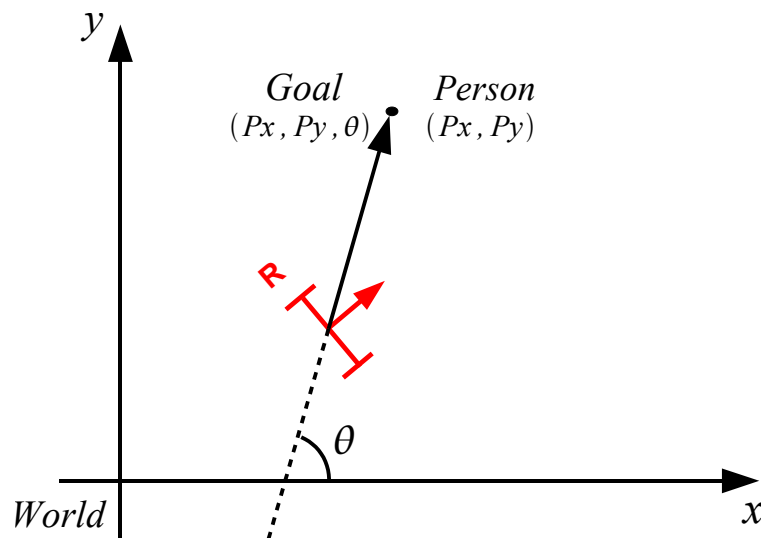


Figure 2.5.: A robot's goal (Px, Py, θ) computed in the world frame for direction-following. The current person's position (Px, Py) obtained from the person tracking system is transformed into the world coordinate system. θ shows the direction from the robot to the person in the world frame.

2.4.3. Path-Following

In comparison to direction-following, path-following enables the robot to move along the path taken by the guide-person. Considering human navigation strategies no obstacles would lie on the person's path in a following scenario¹⁹. Therefore, the trajectory of the person can be regarded as a safe path for the robot²⁰. Note that the trajectory (the path) of the person consists of a set of positions that the person has passed by. The robot should reach all positions selected from the person's trajectory, i.e. all robot's goals, one after another when performing path-following. Selecting goals from person's path effectively and making the robot follow the path taken by the person as closely as possible need to be both taken into account. In principle all person's positions output from the tracking system constitute the complete person's path. However, not all person's positions are necessarily computed as robot's goals due to *goal_radius* defined in section 1.2 for the robot. All robot's goals are stored in a queue *RobGoal*. Basic considerations are outlined as follows:

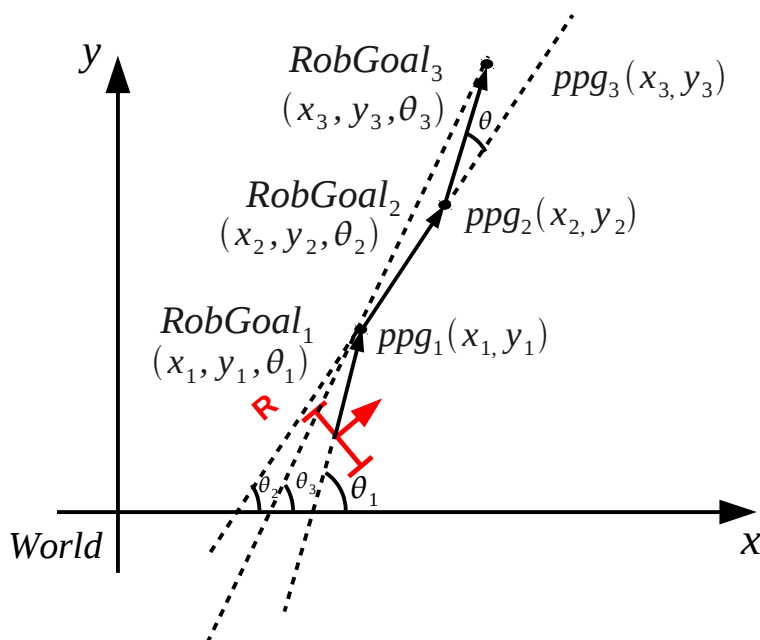


Figure 2.6.: Computation of goal-direction in path-following. The corresponding person's positions and robot goals are marked with the same subscripts. The orientation of the first goal θ_1 shows the direction from robot to ppg_1 . The direction θ_2 from ppg_1 to ppg_2 is calculated for the goal *RobGoal*₂. Assuming that the person approximately moves straight, i.e. θ is smaller than $\theta_{MinDirChange}$, ppg_2 is removed from *RobGoal* and the direction from ppg_1 to ppg_3 is calculated for the new goal *RobGoal*₃

- ▷ Recalling *goal_radius*, the robot reaches the goal-position when it is inside *goal_radius*. In other words, all goals lying inside the circle defined by

¹⁹In an extreme case unexpected dynamic obstacles might move between the guide-person and the robot. However, reactive navigation would make the robot avoid such obstacles.

²⁰Note that obstacles like thresholds are no hazards for people, but they might be walls for most robots and cannot be detected. Such situations should be taken into account in future work.

goal_radius are equivalent to the robot. Therefore, adding a new goal is restricted by a threshold of minimal distance $T_{MinDist}$ between the two adjacent positions on the person's trajectory²¹ computed in a global coordinate system (ppg in algorithm 1). Thus, useful information representing the person's path can be extracted in the context of the designed path-following.

- ▷ In order to select robot's goals from the person's path more effectively, whether the guide-person goes straight is checked. When the person does not change the movement direction much, i.e. the trajectory of the person can be regarded approximately as a straight line, two end positions of this line are sufficient for the robot to follow the person's path. Hence, a threshold of minimal movement angle $\theta_{MinDirChange}$ is defined. As illustrated in Fig. 2.6 when θ is greater than $\theta_{MinDirChange}$, *ppg3* will be computed. Otherwise *ppg2* will be removed from *RobGoal* and *ppg3* will be added as a new goal (see algorithm 1).
- ▷ As introduced in section 2.4.1 the robot stops and turns to the guide-person when *go back to robot* is detected. Therefore, all goals are deleted from *RobGoal*, if it is detected that the person moves back to the robot. The robot observes the next step of the person's movement, and does not have to reach the goals on the person's path back to it.
- ▷ Except for goal-positions, goal-directions are important for the robot to move along the person's path. Direction from the last to the current goal-position is computed as the orientation of the current goal. By this means, all previous movement directions of the guide-person till the current goal can be achieved. For the first goal extracted from the path of the person, direction from the robot to this goal is calculated. As illustrated in Fig. 2.6 θ_1 shows the orientation of the first goal, while θ_2 for goal *RobGoal₂* shows the direction from *ppg1* to *ppg2*.

The complete implementation of the process of goal selection is summarized in algorithm 1. As the robot has to spend time on reaching the goals extracted from the path of the guide-person one after the other, path-following might be not so efficient as direction-following. Besides, an updated person's position does not influence the moving direction of the robot until the old goals are reached. For instance, the robot may not react quickly to the person's turns, as the robot has to turn at the location where the person turned rather than at the same time as the person did. Therefore, the possibility of *Person Lost* should be higher than when direction-following is adopted. Nevertheless, following the path of the person would be beneficial to the robot. As shown in Fig. 1.2(a) the laser sensor is equipped at a height about **30cm** for the perception of a half-plane²². All obstacles out of the laser plane become potential hazards to the robot. Fig. 2.7 shows two experimental runs performed on **BIRON I** in a lab-environment. All the obstacles which the laser can perceive are colored with black areas. The outline of a table laid in the lab is represented with a rectangle, but only the four small legs of the table can be perceived by the robot. On the one hand, the robot was able to successfully avoid the table using path-following as shown in Fig. 2.7(b). On the other hand, adopting direction-following the robot collided with the table at the end of the experiment. As illustrated in Fig. 2.7(a) when the person was at the position P_5 , a collision with the table occurred at the robot's position R_5 . Besides, objects absorbing light (e.g. black cases) or transparent obstacles (e.g. glass windows) are hazardous to a

²¹Note that minimal distance $T_{MinDist}$ was used to collect real person's positions during path-following in the user study. See the evaluation of person-following in section 5.2.1.

²²Applying an additional sensor, e.g. the SwissRanger camera described in section 4.3.2, the robot can perceive the 3D environment partially. However, transparent objects are still unperceivable to SwissRanger.

Algorithm 1 Goal Selection in Path-Following

```

Input: Person position in global (ppg)           // world frame shown in Fig. 2.6
Input: Radial coordinate of person in robot frame ( $r$ )
Require: Queue RobGoal                        // store goal-position (ppg) and goal-direction
Require: Threshold  $T_{ShortDis}$  given          // for departure and arrival
Require: Threshold  $T_{MinDist}$  given         // minimal distance between two adjacent goals
Require: Variable  $dist$                     // distance between current ppg and last ppg in RobGoal
while true do
  if new ppg is computed then
    if not go back to robot then
      if  $r > T_{ShortDis}$  then
        if RobGoal is not empty then
          compute  $dist$                           // remember threshold  $T_{MinDist}$ 
        end if
        if RobGoal is empty then
          insert ppg and goal-direction into RobGoal           //  $\theta_1$  in Fig. 2.6
        else if  $dist < T_{MinDist}$  then
          return                                             // select goals according to  $T_{MinDist}$ 
        else if RobGoal.size == 1 then
          insert ppg and goal-direction into RobGoal           //  $\theta_2$  in Fig. 2.6
        else
          if person goes straight then
            delete last ppg, insert new ppg and goal-direction into RobGoal
          else
            insert ppg and goal-direction into RobGoal         // see Fig. 2.6
          end if
        end if
      else
        short distance behavior
      end if
    else
      clear queue RobGoal
    end if
  else
    return                                             // wait for new ppg
  end if
end while

```

robot due to the limited perception abilities. Carefully moving along the trajectory of the guide-person is therefore necessary for the robot to avoid such unperceivable obstacles.

2.4.4. Parallel-Following

The last elementary following behavior is the so-called parallel-following. The basic idea stems from the observation of a human-human following scenario in open spaces or wide passages. Gockley et al. [58] have aimed in the long run for robotic assistants that could accompany a person side-by-side. A model for side-by-side accompaniment has been implemented by Prassler et al. [121] for a robotic wheelchair in the context of transportation services, e.g. transportation of disabled or elderly people, or transportation of patients in a

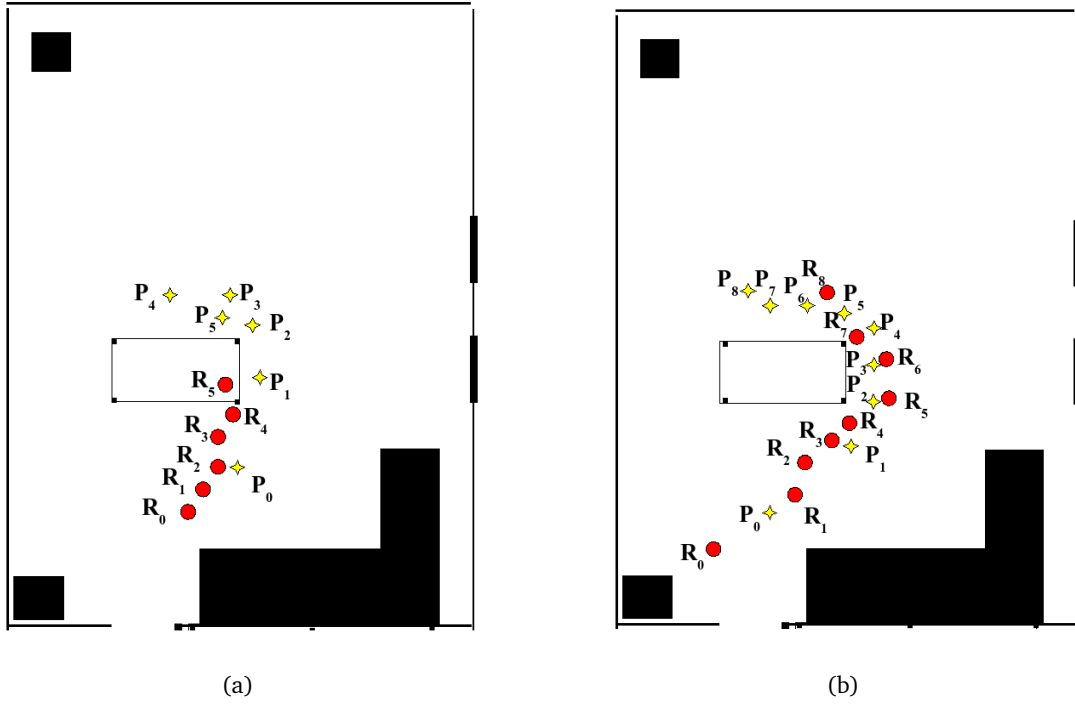


Figure 2.7.: Comparing direction-following (a) with path-following (b). The two experiments were performed by **BIRON I** in the same environment. The start points of the robot and the person are depicted with R_0 and P_0 respectively and the corresponding positions of the robot and the person during the following are labeled with the same number.

hospital. Following these ideas, parallel-following is incorporated in the adaptive context-aware behavior discussed in section 2.3. By this means, the robot is able to walk with the guide-person side-by-side in order to facilitate interaction, while taking available free space into account. Therefore, how to compute a robot's goal set around the current position of the guide-person becomes the key problem of parallel-following.

In order to approximate a side-by-side person-following behavior, a path parallel to the person's trajectory needs to be computed. In other words, person's movement direction that should be kept approximately by the robot and goal-positions deviated from original person's path are two points considered in the computation. Besides, whether a goal can be computed on the left or right side of the person is still uncertain without further information obtained from the surroundings, i.e. free space detected by the laser scanner. Hence, two potential robot's paths lying on both sides of person's trajectory are taken into account.

Fig. 2.8 shows the method proposed to achieve such a robot's goal in parallel-following. Two adjacent person's positions P_1 and P_2 are recorded to calculate the movement direction of the guide-person. As illustrated in Fig. 2.8(b), a local coordinate system is defined with the origin P_2 , i.e. the current person's position, and axis x_L showing the movement direction from the last person's position P_1 to the origin P_2 . According to axis x_L deviation from the current person's position P_2 on the original person's path can be expressed with radius r and angle α in local coordinate system $\{x_L, y_L\}$ at first, and then transformed into the world coordinate system. Two potential robot's goals $G_L(r, \alpha)$ and $SG_L(r, -\alpha)$ are set symmetrically to axis x_L (see α and $-\alpha$ in this figure). In contrast to two adja-

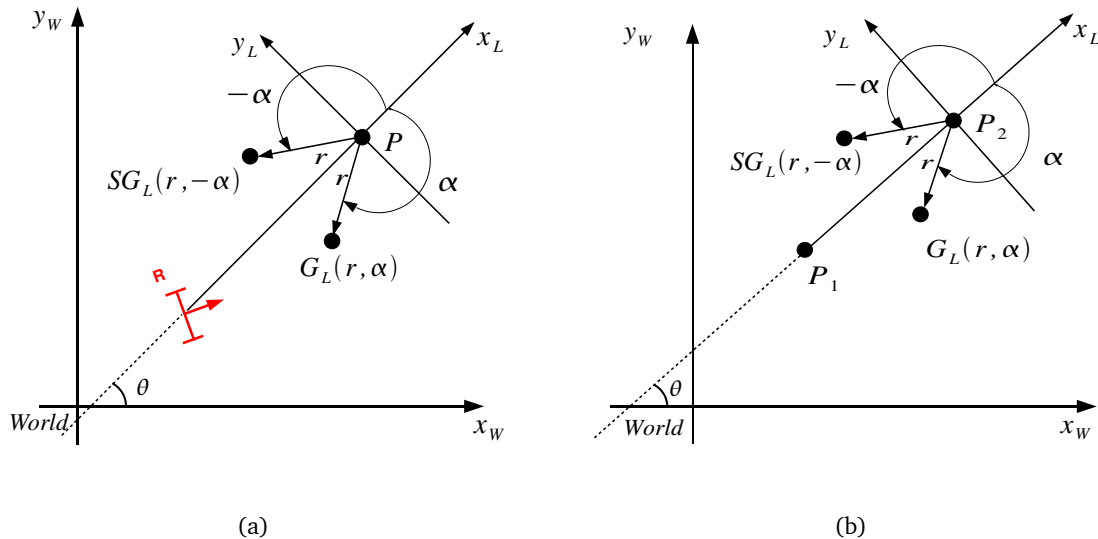


Figure 2.8.: Computation of the potential goals defined with $G_L(r, \alpha)$ and $SG_L(r, -\alpha)$ for the robot in parallel-following. With respect to the direction of axis x_L , namely the goal direction for the robot, G_L and SG_L lies on the right and left side of the guide-person respectively. (a) depicts the initial state, in which only the most current position of the person is stored for the robot. (b) illustrates the goals computed during the person-following process.

cent person's positions observed by the robot during the parallel-following behavior, before this behavior is started by $T_{ShortDis}$ defined in section 2.4.1, the most current person's position is stored. Fig. 2.8(a) shows the two possible goals G_L and SG_L defined in the local coordinate system, where the origin is the current person's position P , and x_L shows the direction from the robot to P . Note that the positive direction of x_L illustrated in Fig. 2.8(a) is not necessarily consistent with the movement direction of the person at the next step. However, goals computed according to Fig. 2.8(b) will correct robot heading, as soon as the guide-person moves away from the robot and parallel-following is triggered. In terms of the decision policy that will be introduced later in this section, whether a goal is available for the robot can be determined. The intersection angle θ from the world axis x_W to the local axis x_L gives the goal direction of the robot. Thus, the robot can move right or left behind the person with respect to person's movement direction.

After two candidates of a robot's goal are computed according to the current person's position, which candidate should be chosen needs to be determined. As the information of the surroundings, i.e. distance to obstacles, is represented in the laser plane, checking free space for the two potential goals is regarded necessary for goal decision. Candidate lying in free space is chosen as the ultimate goal for the robot. When free space can be detected for both candidates, according to the angle of the person in robot frame the unique goal for the robot is able to be chosen (see Fig. 2.9(b)). If the person walks on the left side with respect to the robot orientation, the potential goal on the right side behind the person is chosen as the final goal for the robot. Thus, the relative position between the person and the robot can be kept. In other words, the robot does not have to move on the left side of the person, i.e. keep the person on the right side, when the person is already on the left side of it.

Fig. 2.9(a) illustrates in detail how to compute the regions checked for the two candidates in

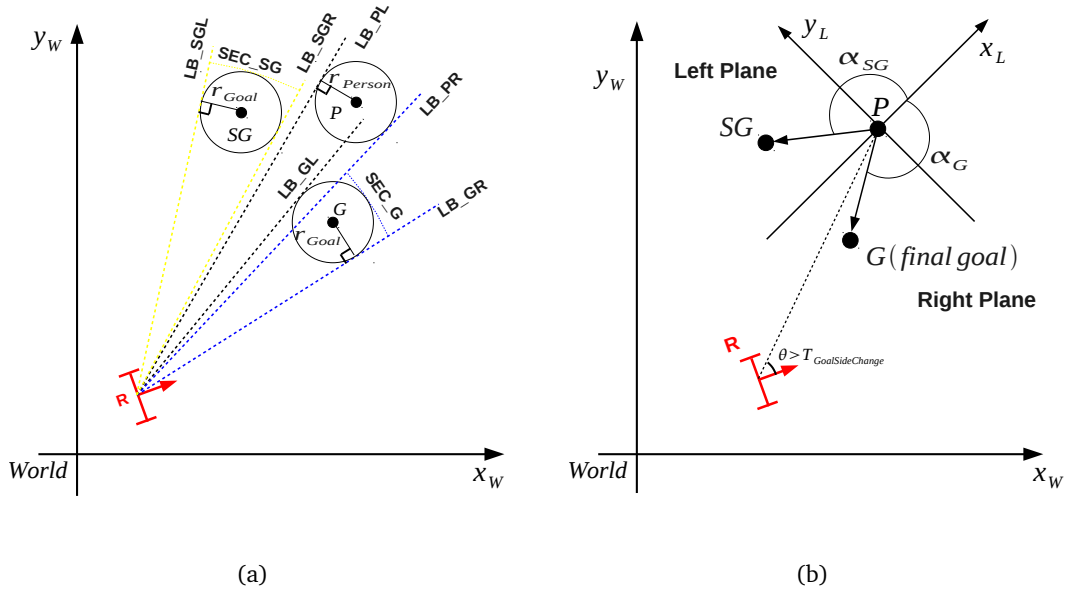


Figure 2.9.: Decision policy of robot's goals in parallel-following. The robot, the guide-person and the two potential goals are marked with R , P , G and SG respectively. (a) illustrates the general criteria for goal decision. The occupied areas of the guide-person and the two potential goals are represented with the circles centered on P , G and SG respectively. The radius of circle P is marked with r_{Person} according to the distance between the two legs of the person, while r_{Goal} gives the radius of circles G and SG in terms of the real size of the robot. The two sector regions SEC_G and SEC_{SG} depicted with blue and yellow dashed lines show the check-areas excluding the area occupied by the person in the laser plane. When obstacles are detected inside SEC_G or/and SEC_{SG} , the corresponding goal(s) is(are) not available for the robot. A special case, i.e. no obstacles are detected in both SEC_G and SEC_{SG} , is shown in (b). With respect to the guide-person P and the moving direction, goal G lying on the right side is chosen as the final goal, as the person stands on the left side relative to the current robot's orientation.

the laser plane. Areas occupied by the two potential goals G and SG , as well as the current person's position is depicted with circle centered on G , SG and P respectively. Radius r_{Person} of circle P is obtained²³ according to the distance between the two legs of the person detected by the person tracking system. In order to reserve free space for the robot when G or SG is reached, radius r_{Goal} is chosen on the basis of the real size of the robot. Thus, two tangents of the respective circles in the robot frame can be computed. Three regions for P , G and SG are obtained. Notice that these regions may overlap each other. Regions checked for the two potential goals G and SG exclude the circle area occupied by P , since the person should not be regarded as a real obstacle during a person-following process. Moreover, the *short distance behavior* defined in section 2.4.1 will be performed to stop the robot, when the distance between the person and the robot is shorter than $T_{ShortDis}$. The computed sector region checked for G and SG is shown in Fig. 2.9(a) with SEC_G (the region depicted with blue dashed lines) and SEC_{SG} (the region depicted with

²³Checking the area occupied by the person is applied as well in the mapping process discussed in section 3.2.

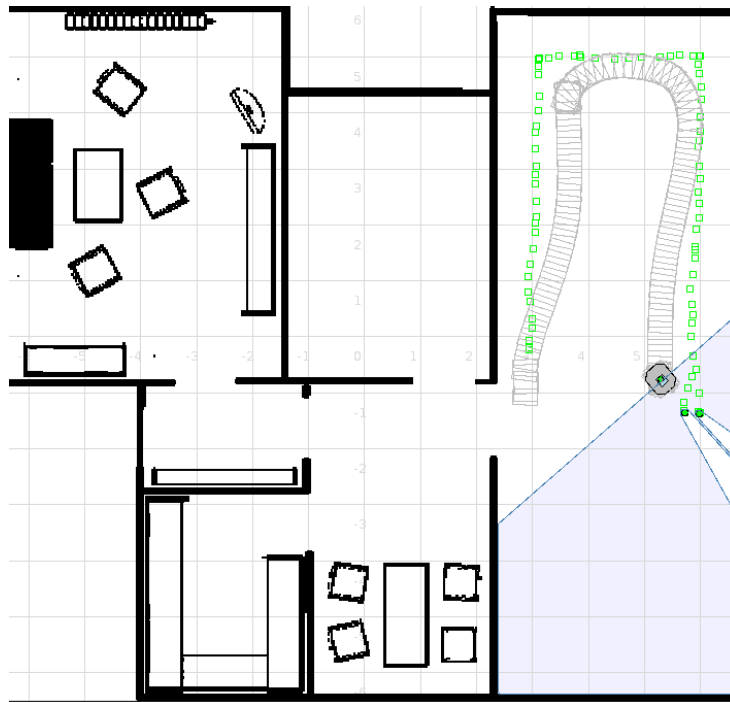


Figure 2.10.: Parallel-following performed in *Stage*. The robot's trajectory depicted with gray approximately forms a parallel curve with respect to the green path illustrating the movement of the guide-person. The robot chooses the goals on the side, in which there are no obstacles.

yellow dashed lines) respectively. The radius of the sector region SEC_G is $\|\overrightarrow{RG}\| + r_{Goal}$, recalling the circle area reserved for the robot. G will be chosen as the ultimate goal for the robot, when no obstacles are detected by laser scanner inside SEC_G . Analogically for SG , free space is checked in the sector region with radius $\|\overrightarrow{RS\hat{G}}\| + r_{Goal}$.

In the special case, i.e. G and SG are both available for the robot, the unique goal can be determined according to the angle of the guide-person with respect to the robot. As illustrated in Fig. 2.9(b) θ lying in the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$ shows the angle of the person in the robot coordinate system. A positive threshold $T_{GoalSideChange}$ is predefined considering the consistency of robot's movement²⁴. Axis x_L divides the whole plane into the left and right plane with respect to its positive direction. According to the person P and the moving direction, i.e. the positive direction of axis x_L , G is on the right side as α_G is in the interval $(0, -\pi)$. In contrast, SG is on the left side as α_{SG} lies in the interval $(0, \pi)$. G is picked up as the final goal, since the person is on the left side (θ is greater than $T_{GoalSideChange}$) with respect to the orientation of the robot, and G lies on the right side. Analogically SG will be selected, if θ is smaller than $-T_{GoalSideChange}$. When the person stands in front of the robot in the middle area, i.e. θ lies in the interval $[-T_{GoalSideChange}, T_{GoalSideChange}]$, the goal on the side consistent with the previous one will be chosen.

Except for the criteria for goal decision discussed above, how to enable the robot to keep up

²⁴It should be avoided when the robot often changes the trajectory on the right and left side behind the guide-person.

with the person effectively needs to be taken into account. As a robot's goal is set around the current person's position, the robot does not drive in the direction of the guide-person. It is more challenging for the person tracking system not to lose the target. Robot's goals are therefore updated as soon as a new movement direction of the guide-person, e.g. the direction from P_1 to P_2 illustrated in Fig. 2.8(b), is observed. By this means, the robot is able to react to person's movement as soon as possible. Considering the computation efficiency the movement direction of the person is recorded only when the person changes the position exceeding a threshold²⁵ $T_{PersonChange}$. Once the potential goals are computed, the real goal chosen for the robot is determined according to the decision policy illustrated in Fig. 2.9.

The implementation of goal decision illustrated in Fig. 2.9 is presented in algorithm 2. This behavior is called parallel-following, since the goals defined with r , α and θ (see Fig. 2.8) may approximately form a curve parallel to the trajectory of the guide-person. Theoretically the curve consisting of the robot's goals is parallel to the path of the guide-person only when the perceived person's trajectory is a straight line. In addition, as the robot's goals are defined on both sides behind the guide-person, and a goal on one side is chosen according to the decision policy (see Fig. 2.9), the entire path of the robot may intersect with the trajectory of the guide-person. Fig. 2.10 illustrates the parallel-following behavior simulated in *Stage*. The gray curve depicting the movement trajectory of the robot lies on the right side²⁶ of the green person's path. Notice that all the goals on the right side of the two potential goals are selected for the robot, since obstacles, namely the walls in the simulated environment, are on the left side according to the movement direction of the guide-person.

Adopting the parallel-following behavior the robot is able to enter the area around the person flexibly. Frequently updating the goals for the robot, the robot can keep the moving direction consistent with the movement of the guide-person. By this means, the robot is able to move approximately parallel to the person. An issue to be considered is that parallel-following is restricted by the free space available nearby the person. It cannot be used alone in the *Home-Tour*. Therefore, the adaptive context-aware following approach presented in section 2.3 is necessary for the robot.

2.5. Summary

Person-following enables the robot to keep up with a guide-person and complete a human-robot joint exploration. Moreover, as will be discussed in section 4.2 *locations*, i.e. room names indicated by the guide-person, can be obtained by the robot during an interactive *Home-Tour* scenario. These *locations* form the nodes of a topological representation of the environment.

In this chapter two prerequisites for the person-following behavior, i.e. person tracking and robot motion control were introduced at first. The guide-person is identified and tracked in the interactive *Home-Tour* with face and voice information, except for person's legs detected in the laser plane. Reactive navigation and obstacle avoidance in dynamic cluttered indoor environments is needed as well for the robot to follow the person, which cannot be impeded by the presence of the guide-person.

²⁵Note that this threshold was used to collect real person's positions during parallel-following in the user study. See the evaluation of person-following in section 5.2.1.

²⁶With respect to the movement direction of the simulated guide-person.

Afterwards, the adaptive context-aware following approach integrating three elementary behaviors was presented. By dividing the perception area of the robot, i.e. the laser plane, into different parts, various situations in an indoor environment have been taken into account in the core strategies combining characteristics of the elementary behaviors with the gained person information. Using the core strategies an effective person-following behavior exploiting the human-user as a knowledgeable navigator, considering narrow space and facilitating interaction, has been implemented. In section 5.2.1 the evaluation results of the proposed person-following approach achieved in a user study will be discussed.

At last, three elementary person-following approaches constituting the adaptive context-aware behavior are discussed. General strategies available for all the elementary following behaviors are introduced. Robot's reaction to the situations, in which the robot is in proximity to the guide-person, the person is checked to go back to the robot, or the person is lost by the person tracking system, are discussed. Path-following has been regarded as a smart solution for obstacle avoidance, especially when the robot needs to navigate in narrow spaces or unperceivable obstacles lie on the robot's path. However, the robot cannot promptly react to changes of the person's movement, as the goals extracted from the person's path have to be reached one after another. Therefore, direction-following is designed for the robot to keep up with the guide-person as soon as possible. The drawback of this behavior has been shown in Fig. 2.7 in comparison with path-following. Parallel-following is proposed to find goals for the robot nearby the guide-person and track the movement direction of the person. By this means, the robot can move approximately side-by-side with the person in wide passages or open spaces to facilitate interaction, while taking available free space into account.

Algorithm 2 Goal Decision in Parallel-Following

```

Input: G computed in world frame           // one potential goal, see Fig. 2.9(a)
Input: SG computed in world frame        // the other potential goal, see Fig. 2.9(a)
Input: Person position P computed in world frame
Input: Robot's position in world frame (R)
Input: Radial coordinate of person in robot frame (rp)
Input: Angular coordinate of person in robot frame (ap)
Input: Laser scan
Require: Left and right laser border for P: LB_PL, LB_PR           // see Fig. 2.9(a)
Require: Left and right laser border for G: LB_GL, LB_GR           // see Fig. 2.9(a)
Require: Left and right laser border for SG: LB_SGL, LB_SGR       // see Fig. 2.9(a)
Require: Sector region checked for G: SEC_G
Require: Sector region checked for SG: SEC_SG
Require: Positive threshold  $T_{GoalSideChange}$  // when SEC_G and SEC_SG are both free
Require: Radius  $r_{Person}$  // circle area occupied by person
Require: Radius  $r_{Goal}$  // circle area occupied by robot's goal
Require: Distance between R and G:  $Dist_{RG}$ 
Require: Distance between R and SG:  $Dist_{RSG}$ 
while true do
  if new P, G and SG are computed then
    compute LB_PL and LB_PR with  $r_{Person}$  and  $rp$  // see Fig. 2.9(a)
    compute  $Dist_{RG}$  and  $Dist_{RSG}$ 
    compute LB_GL, LB_GR, LB_SGL and LB_SGR with  $r_{Goal}$ ,  $Dist_{RG}$  and  $Dist_{RSG}$ 
    compute radius  $Dist_{RG} + r_{Goal}$  and  $Dist_{RSG} + r_{Goal}$  for SEC_G and SEC_SG
    determine SEC_G and SEC_SG excluding circle area of person with radius  $r_{Person}$ 
    check free space in SEC_G and SEC_SG with laser scan
    if SEC_G and SEC_SG both free then
      if  $ap \geq T_{GoalSideChange}$  then
        select the goal on the right side // correspond to G in Fig. 2.9(b)
      else if  $ap \leq -T_{GoalSideChange}$  then
        select the goal on the left side // SG in Fig. 2.9(b)
      else
        select the side consistent with last one
      end if
    else if SEC_G free but SEC_SG not then
      select G as goal
    else if SEC_SG free but SEC_G not then
      select SG as goal
    else
      send no goal is available to core strategies, return // remember section 2.3.1
    end if
  else
    return // wait for available P, G and SG
  end if
end while

```

3. Map Building in a Home-Tour

As mentioned in previous chapters, one of the important tasks for a robot during the interactive *Home-Tour* is to build a representation of the new environment for itself. Since the problem “*where to move in the Home-Tour?*” has been solved by the person-following behavior proposed in the last chapter, the robot can explore the unknown environment and perceive the surroundings. This chapter is devoted to building a metric map of the environment in the guided tour. Considering that the details of the environment are crucial for the robot performing a navigation task, an occupancy grid map built with a SLAM (*Simultaneous Localization And Mapping*) approach on the basis of a Rao-Blackwellised Particle Filter [35] (RBPF) is selected as the metric representation of the surroundings. Moreover, without predefined feature extractors needed to be appropriate for some known structures of the environment, an occupancy grid-based spatial representation should be more robust for cluttered indoor environments. The key idea and the kernel technique of the RBPF-SLAM is introduced in section 3.1. A problem that can specifically occur in a mapping process during a *Home-Tour* is the dynamic obstacle resulting from the guide-person. Due to the guide-person who may be perceived by the robot all along a *Home-Tour* and regarded as dynamic obstacles, the Markov assumption [143], sometimes known as static world assumption, is violated. This violation negatively affects not only the mapping process, but also the estimation of the robot’s poses. Therefore, a practical solution dealing with the special dynamic objects, as well as the evaluation results achieved from a simulated environment are presented in section 3.2.

3.1. Introduction of SLAM with Rao-Blackwellisation

Simultaneous Localization And Mapping (SLAM) is a general problem for a mobile robot placed at an unknown location in an unknown environment. The ability of building a consistent map of the environment incrementally and estimating the location simultaneously within the map makes a mobile robot truly autonomous [38]. SLAM has been formulated and solved dominantly as a probabilistic problem

$$p(\mathbf{x}_{0:t}, \mathbf{m} \mid \mathbf{u}_{1:t}, \mathbf{z}_{1:t}), \quad (3.1)$$

i.e. if it is possible to concurrently estimate the robot’s trajectory $\mathbf{x}_{0:t}$ and the environment map \mathbf{m} given the movement $\mathbf{u}_{1:t}$ and perception $\mathbf{z}_{1:t}$ up to the current time step. The quantities and basic models are defined in the following:

- ▷ \mathbf{x}_t describes the robot state at the time step t . Especially for the RBPF-SLAM (Rao-Blackwellised Particle Filter SLAM) application the robot’s pose (x, y, θ) constructs the robot state. $\mathbf{x}_{t_1:t_2} = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_1+1}, \dots, \mathbf{x}_{t_2}\}$ denotes the set of robot states from time step t_1 to t_2 with $0 \leq t_1 \leq t_2$.
- ▷ The control actions which are executed from the time step $t_1 - 1$ to $t_2 - 1$ and make a robot change its state from \mathbf{x}_{t_1-1} to \mathbf{x}_{t_2} are denoted with $\mathbf{u}_{t_1:t_2}$, where $\mathbf{u}_{t_1:t_2} = \{\mathbf{u}_{t_1}, \mathbf{u}_{t_1+1}, \dots, \mathbf{u}_{t_2}\}$ and $1 \leq t_1 < t_2$. Particularly when $1 \leq t_1 = t_2$, the

control action u_{t_1} or u_{t_2} is applied at time step $t_1 - 1$ to drive the robot from state x_{t_1-1} to state x_{t_1} .

- ▷ The motion model, which specifies the effect of the control action u_t on the state x_{t-1} and describes the probability distribution of the state transition from x_{t-1} to x_t on the basis of the control u_t , is generally given as follows:

$$p(x_t | x_{t-1}, u_t). \quad (3.2)$$

- ▷ The robot perception from state x_{t_1} to x_{t_2} is given with $z_{t_1:t_2}$, where $t_1 \leq t_2$ and $z_{t_1:t_2} = \{z_{t_1}, z_{t_1+1}, \dots, z_{t_2}\}$.
- ▷ In general, for a metric map m the environment can be represented either by a set of landmarks detected in the environment, i.e. $m = \{m_1, m_2, \dots, m_N\}$, where N is the total number of objects in the environment, or by the occupancy grid, in which the environment is discretized into a high resolution grid, namely small cells, and the probability of being occupied by an obstacle is assigned to each cell.
- ▷ The observation model describes the likelihood of an observation z_t from a robot state x_t in the built map m , as shown in the following equation:

$$p(z_t | x_t, m). \quad (3.3)$$

The key idea of RBPf-SLAM is to divide the complete joint posterior presented in equation 3.1 into a pose estimation and a map update process with the following factorization:

$$p(x_{0:t}, m | u_{1:t}, z_{1:t}) \stackrel{\text{product rule}}{=} p(m | x_{0:t}, u_{1:t}, z_{1:t}) \cdot p(x_{0:t} | u_{1:t}, z_{1:t}) \quad (3.4)$$

$$\stackrel{\text{independent}}{=} p(m | x_{0:t}, z_{1:t}) \cdot p(x_{0:t} | u_{1:t}, z_{1:t}). \quad (3.5)$$

Assuming that m is independent from the control actions $u_{1:t}$ given the sequence of robot states $x_{0:t}$ and the corresponding robot perception $z_{1:t}$, the first part of equation 3.5 can be obtained. This problem is generally known as *mapping with known poses* [146]. The posterior about the map $p(m | x_{0:t}, z_{1:t})$ can be computed analytically, given the knowledge of $x_{0:t}$ and $z_{1:t}$. In the built map the estimation of robot's poses is feasible, when the sequence of control actions and robot perceptions are known, as presented in the second part of equation 3.5. This factorization, which offers a solution of SLAM with efficient computation, is referred to as Rao-Blackwellisation [35]. The name is related to the Rao-Blackwell formula [26].

In fact, equation 3.5 provides a basic probabilistic framework. Hence, a solution to the SLAM problem involves finding an appropriate representation for the both generative models, namely the state transition model and the observation model shown in equation 3.2 and 3.3 respectively. The localization and the mapping problem described in equation 3.5 needs to be tackled effectively. One class of approaches to solve the SLAM problem are based on Kalman filters [81] (KFs), which represent the joint posterior shown in equation 3.1 with Gaussians. Linear Gaussian systems [123] are assumed, i.e. the motion and the observation model needs to be linear with additive Gaussian noise, and the initial uncertainty of robot state has to be Gaussian [146]. Considering nonlinearities, e.g. robot motion and perception, existing in almost all practical systems, linearization techniques based on Taylor series expansion are applied on original KFs. Such extended techniques are generally known as Extended Kalman Filters [80] (EKFs), which have been developed in indoors [27], outdoors [60], subsea [160] applications, etc. to build an environment

representation and make the robot localize itself in the built map [38]. However, linear Gaussian assumptions, which may be not appropriate for real models are essentially retained by those approaches. In contrast, sample (particle)-based approaches, which are generally referred to as particle filtering [25] or sequential Monte Carlo sampling [36], directly represent the non-linear process model and non-Gaussian distribution of the robot state. Each sample or particle with its weight is an instance of the estimated robot state in the real world¹. Therefore, particle filtering based on Rao-Blackwellisation is assumed as a robust approach to deal with non-linear and non-Gaussian systems.

For RBPF (Rao-Blackwellised Particle Filter) the estimation of the posterior shown in equation 3.1 is approximated by the distribution of the samples populated in the real world. Notice that the probability distribution of the map illustrated in equation 3.5 is conditioned on the robot's trajectory $x_{0:t}$ rather than a single pose x_t . This actually indicates the independence of the map states as well, because the observations are conditionally independent given the known history of pose states [38]. Consequently, each particle represents a potential robot's trajectory instead of a single robot's pose, and an individual environment map is associated with the corresponding particle. Assuming that the set of particles at time step $t - 1$ is $X_{0:t-1} = \{x_{0:t-1}^{(1)}, x_{0:t-1}^{(2)}, \dots, x_{0:t-1}^{(N)}\}$, where N is the number of particles, and each particle $x_{0:t-1}^{(i)}$ has its own weight $w_{t-1}^{(i)}$ and its own map $p(m | x_{0:t-1}^{(i)}, z_{1:t-1})$, the recursive mapping and localization process based on RBPF for the next time step t can be exemplarily described as follows [59]:

- ▷ The set of particles $X_{0:t}$ at the time step t is generated by drawing samples based on the generation $X_{0:t-1}$ from a proposal distribution $\pi(x_{0:t} | u_{1:t}, z_{1:t})$. Usually a probabilistic motion model, such as the Gaussian odometry model proposed in [41] for our case, or the improved proposal distribution described in [59], is used as the proposal distribution.
- ▷ Particles are weighted according to importance sampling principle, as introduced in [34].

$$w_t^{(i)} = \frac{p(x_{0:t}^{(i)} | u_{1:t}, z_{1:t})}{\pi(x_{0:t}^{(i)} | u_{1:t}, z_{1:t})} \quad (3.6)$$

The computed weights arise from the difference between the target distribution $p(x_{0:t} | u_{1:t}, z_{1:t})$ which is generally not be able to directly sample from and the proposal distribution which is selected close to the target as far as possible and easy to sample.

- ▷ Particles are drawn with replacement proportional to their weight, i.e. the sample $x_{0:t}^{(i)}$ is selected with probability $w_t^{(i)}$. This process is generally referred to as resampling [33], which realizes sampling from the approximation of the target distribution.
- ▷ For each particle, the associated map $p(m | x_{0:t}^{(i)}, z_{1:t})$ is updated on the basis of the robot trajectory and the history of observations.

For the resampling step, although the particles with low weight can be removed, particle degeneracy or depletion is inevitable. As shown in [36], most of the particles probably have a weight close to zero. The so-called Effective Sample Size (ESS) introduced in Kong et al. [83] is generally known as a suitable measurement of degeneracy, which is defined as:

¹See the brief introduction to particle filters in appendix A.

$$ESS = \frac{1}{\sum_{i=1}^N (w_t^{(i)})^2}. \quad (3.7)$$

ESS takes values in the interval $[1, N]$. A resampling process is invoked only when ESS is below a threshold N_T . In an extreme case, all samples are replaced by a single particle carrying the largest weight. Thus, ESS tends to 1. This implies that the number of particles in use is one. Typically N_T is set to $N/2$ [34].

Apart from the trajectory of the robot, namely the robot state represented by a set of weighted samples, the map representation is the other point that still needs to be discussed. In general, two types of environment representations on the level of metric maps² have been proposed. Feature- or landmark-based representations are attractive because of their compactness. In a feature-based representation the Cartesian coordinates of the set of features extracted from the environment, such as points [149], line segments [134], geometric models [95], key-points extracted by SIFT (Scale-Invariant Feature Transform) [130], etc., are stored in the map. EKF-based map update fusing new observed data into the map for each particle is generally applied, after the data association problem (see section 1.2), i.e. to determine whether the currently obtained features (landmarks) are fit into the map built so far, has been solved. Solutions to the probabilistic SLAM problem that are based Rao-Blackwellisation and apply extended Kalman filters combined with data association approaches for map estimation constitute the kernel of the so-called FastSLAM introduced by Montemerlo et al. [110, 112]. As the metric map consists of features (landmarks) extracted from the environment, FastSLAM relies on predefined feature extractors under the assumption that some structures in the environment are known in advance. Furthermore, except for the strong assumption of linear Gaussian systems the features (landmarks) are assumed to be uniquely identifiable by EKF approaches. Once these assumptions are violated, EKFs may diverge [53].

In contrast, a free space representation of the environment, which illustrates the portion of the environment that is accessible for the robot, does not rely on predefined landmarks. It uses raw sensor data to represent free space and arbitrary objects in the environment. The occupancy grid map developed by Elfes and Moravec [113, 40] is the most popular approach to this idea. In this case, the environment is discretized into a grid with a high-resolution. Each grid cell is assigned a value implying the probability of the occupancy covered by an obstacle. As the environment can be represented finely with the occupancy grids [144], the built map is capable of supporting navigation actions more sufficiently in comparison to a sparse description of the environment [169], e.g. feature- or landmark-based maps. Besides, the information contained in an occupancy grid map can be computed with entropy-based measurements that are generally used for the uncertainty of a belief, as will be discussed in the next section. For some existing exploration strategies, such as frontier-based strategies [163], the active loop-closing approach [136], etc., entropy-based measurements computed from an occupancy map have been widely used. In summary, considering the capability of dealing with non-linear and non-Gaussian systems, the feasibility of direct application of particle filters with the help of Rao-Blackwellisation, the robustness against predefined feature extractors, the detailed description of the objects and free space in the environment, as well as the available measurements of uncertainty in SLAM, grid mapping with Rao-Blackwellised Particle Filters (RBPF) is chosen in this thesis.

²Except for metric maps representing objects with coordinates, topological framework focuses on place identities and their spatial relations [87]. On the basis of metric and topological environment representations, more complicated structures, such as multi-hierarchical semantic maps [56], cognitive maps [156], conceptual spatial representations [169], etc., have been proposed since recent years.

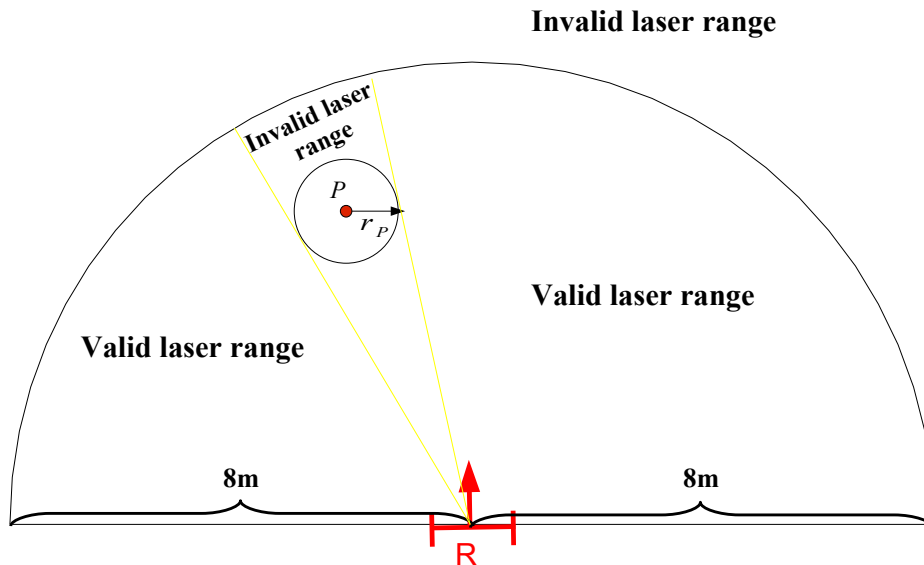


Figure 3.1.: The valid and invalid laser range applied for the mapping process during the *Home-Tour*. The robot R and its orientation are shown with red. The valid laser range inside the semi-circle is restricted to $8m$ away from the robot. Furthermore the occupied area of the guide-person that is defined with the center P and the radius r_P is regarded as invalid.

3.2. Mapping during the *Home-Tour*

Recalling the idea of *Human-Robot Joint Spatial Exploration* (HRJSE), knowledge of the surroundings can be gathered by a mobile robot in an interactive scenario. Furthermore, an environment representation incorporating human spatial understanding is able to be built. Therefore, human inhabitants sharing an indoor environment with the robot are regarded as a prerequisite to realize a human-robot joint exploration. The interactive *Home-Tour* actually provides an opportunity for the robot to explore the unknown environment with the help of a guide-person. On the one hand, once the robot is able to move around in the environment, building a metric representation of the surroundings is possible. On the other hand, additional information can be obtained from the guide-person, as will be discussed in section 4.2. However, the guide-person tracked by the robot violates the Markov assumption, namely the static world assumption generally used in a solution of the SLAM problem [27, 64, 145]. Of course, human inhabitants should not be forced to leave the environment in order to meet this assumption. In addition, making the robot interactively learn from the human user and build a spatial representation is the central problem discussed in this thesis. Thus, the problem of interactive acquisition of an environment representation lies in how to reduce the negative impact of the perceived guide-person³ on a mapping process.

³In our case the legs of the guide-person are detected and tracked by the person tracking system introduced in section 2.2.1 on a laser plane. The maximum perceiving distance of the laser sensor is assigned $8m$ for the current system.

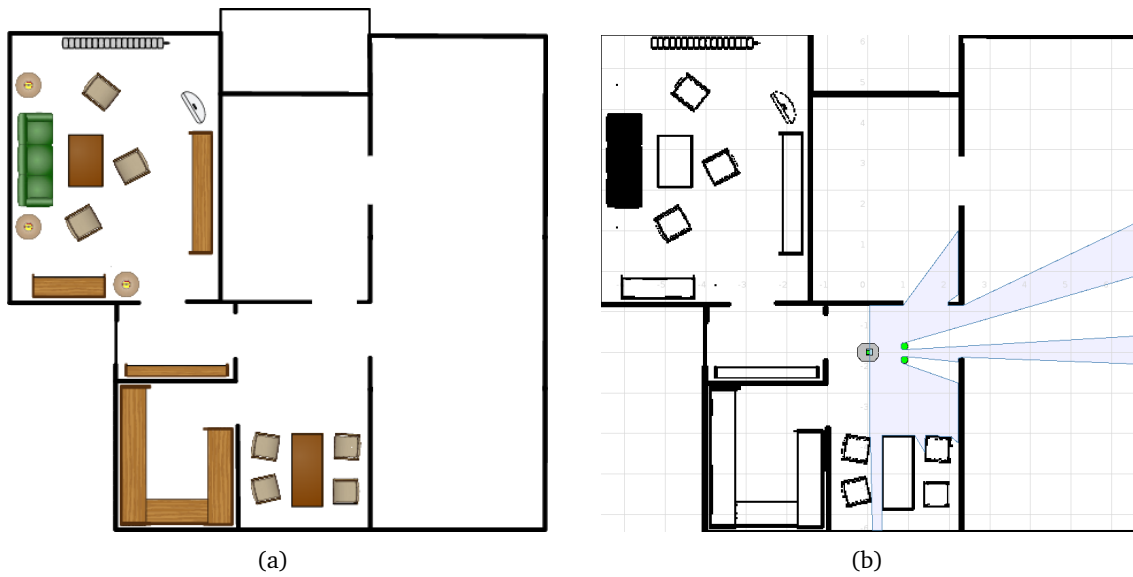


Figure 3.2.: The test environment for the proposed technique to tackle the problem arising from the perceived person's legs in the mapping process. a) shows the plan drawn partially from the BIRON apartment equipped with standard furniture. b) shows the plan displayed on *Stage*. Notice that the color bitmap a) is read by *Stage* and represented with a black white environment model. The start pose of the robot, the two small green hexagons simulating the pair of person's legs in front of the robot, as well as the light blue area simulating the laser scan of the robot are shown in this figure.

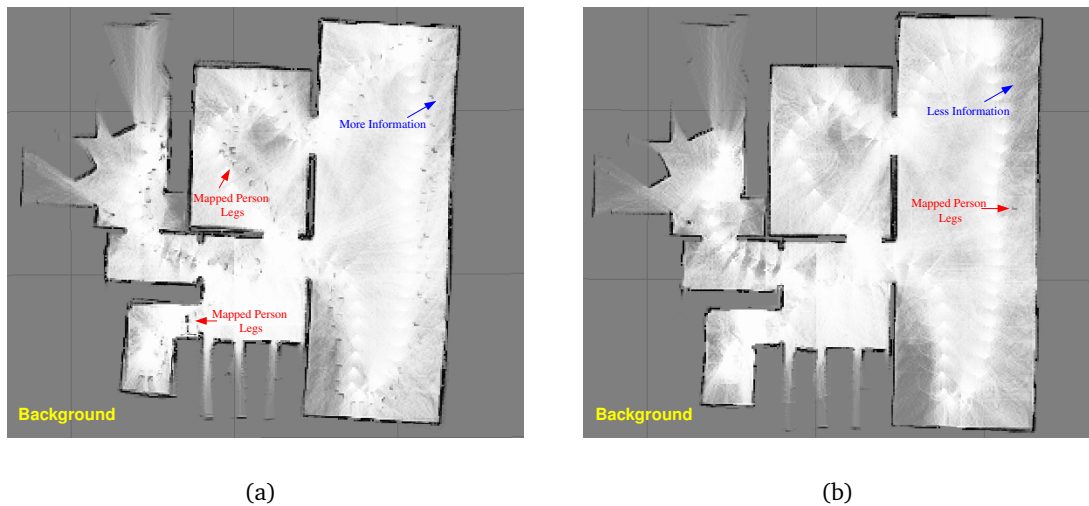


Figure 3.3.: Comparison of the grid maps built with and without the proposed technique. a) shows the mapping result including the person's legs in the laser scans. However, when regarding the person's legs perceived in the laser plane as invalid, the built map is illustrated in b). Some examples of the recorded person's legs are shown with red arrows in a) and b) respectively. The probability of the unperceivable grid cells, such as the background and the invalid laser range, is set to **0.5**. Due to the invalid laser range different map information can be obtained in the two cases, as the areas shown with a blue arrow in a) and b) illustrate.

3.2.1. Basic Idea

In order to deal with the specific situation occurring in a guided tour, i.e. the guide-person perceived by the robot, the area occupied by the tracked person needs to be found and removed from the perception area of the robot (the laser plane in our case) during the mapping process. The person area is defined with a circle centered on the person position P with radius r_P ⁴, as illustrated in Fig. 3.1. The person tracking system introduced in section 2.2.1 provides polar coordinates of person positions in the robot coordinate system. The two borders shown in yellow and tangent to the circle area can be computed to find the corresponding laser points. All laser points inside the two borders are regarded as invalid. Thus, the robot may follow the guide-person, build the environment representation and localize itself in this representation, when the invalid laser points are not used in the mapping process. If this technique is adopted, the violation of the Markov assumption arising from the guide-person should be reduced. Furthermore, false laser readings resulting from person's legs⁵ are expected to be excluded from the built map as far as possible. A better pose estimation of the robot should be achieved.

3.2.2. Evaluation on a Simulation Platform

To verify our hypothesis that by reducing the negative impact of the guide-person a better map can be achieved, the map built in the *Home-Tour* as well as the estimated robot's poses were evaluated on a 2D simulation platform *Stage*. A population of mobile robots, sensors and objects in a two-dimensional bitmapped environment can be simulated by *Stage*⁶. Fig. 3.2(a) shows the plan of the partial BIRON apartment containing standard furniture. In Fig. 3.2(b) the plan is shown on *Stage*. Due to a conversion of the input image a black white environment model is shown in Fig. 3.2(b). The black areas in this figure are *walls*, and the robot can move everywhere else. The simulated pair of person's legs that can be tracked in the laser plane, namely the light blue area, are illustrated as well in Fig. 3.2(b) with the two small green hexagons in front of the robot. Performing the adaptive context-aware following behavior discussed in section 2.3 the robot was able to move around in the simulated environment. Hence, the surroundings can be perceived by the robot with the laser sensor and the environment representation can be built as well on the simulation platform *Stage*. Note that only when the speech command "*Follow me!*" from the mini-dialog module⁷ was sent out, the target detected by the person tracking system can be confirmed. Since the occupied areas of the two green hexagons (a simulation of the guide-person) were detected by the person tracking system and regarded as invalid in the laser plane during the simulated *Home-Tour*, the corresponding laser points were not taken into account in the mapping process. The built map was recorded step by step. Besides, the two necessary inputs of the RBPF-SLAM approach, i.e. the pairs of the increment odometry data and laser scans *including* the information of the person's legs were stored. By this means, the same mapping process except for the use of different laser data can be reproduced. Therefore, the environment representations built with and without the proposed technique can be compared with each other.

⁴The size of the guide-person is given with r_P . Theoretically it can be obtained according to the distance between the two person's legs detected by the person tracking system introduced in section 2.2.1.

⁵Of course person's legs are actually perceived by the laser sensor. However these information should not be added into the environment representation.

⁶see more information in section 2.2.1

⁷See the introduction of the mini-dialog module in section 5.1.2.

The built environment representations with laser scans including and excluding the person's legs are illustrated in Fig. 3.3(a) and Fig. 3.3(b) respectively. As mentioned at the end of section 3.1, in a grid map each grid cell is assigned a value implying the probability of the occupancy covered by an obstacle. Black implies occupied with high certainty, while white corresponds to free with high certainty. No *Person Lost* was detected in the experimental run⁸. Many gray points resulting from the person's legs, namely the two green hexagons illustrated in Fig. 3.2(b), can be found in Fig. 3.3(a). Some examples are shown with red arrows as well. Notice that in Fig. 3.3(b) the person's leg depicted on the map and shown with a red arrow arises from the fact, that during this *Home-Tour* one of the legs of the guide-person was occasionally out of the defined person area at that place and thus recorded by the mapping process.

Map information

In order to measure the uncertainty of the built map, entropy-based measurements were selected. The entropy of one cell m_{xy} in an occupancy grid map m can be computed with the following binary entropy function, since each grid cell is a discrete random variable with a Bernoulli distribution, where the two possible outcomes of the random variable are occupied or not.

$$H(m_{xy}) = -p(m_{xy}) \ln p(m_{xy}) - (1 - p(m_{xy})) \ln(1 - p(m_{xy})) \quad (3.8)$$

Where x and y in m_{xy} gives the index of this grid cell in x - and y -direction. The maximum entropy value for m_{xy} is attained when $p(m_{xy}) = 0.5$. This is the maximum uncertainty obtained from the unobserved grid cells or the cells lying in the area of invalid laser scans. However, the minimum uncertainty, namely the maximum certainty can be obtained, when a cell is certainly occupied or free. Assuming statistical independence between the grid cells, the entropy of the whole map m can be computed as follows:

$$H(m) = \sum_{\forall x,y} H(m_{xy}). \quad (3.9)$$

Although the entropy of a grid map m has been widely used as a measurement providing information in maps [138], the drawbacks that have been discussed in [10] lie mainly in the contribution to the computed entropy from the unobserved grid cells. On the one hand, as shown in equation 3.9, the entropy of the whole map is dependent on the given extent of the map, which contains all the unobserved cells with their maximum value of uncertainty. Hence, it is difficult to sift the pure contribution obtained from the actually observed cells. On the other hand, the resolution of the grid map can also influence the uncertainty measurement due to the fact that the entropy of the map containing unobserved areas must increase, when the resolution of the map increases. Considering these drawbacks the information (I) of a map m can be computed with the following entropy-base measurements.

$$I(m_{xy}) = 1 - H(m_{xy}) \quad (3.10)$$

$$I(m) = \sum_{\forall x,y} I(m_{xy}) \quad (3.11)$$

Where $H(m_{xy})$ has been defined in equation 3.8. Note that the information of the whole map $I(m)$ becomes irrelevant to the given size of the map, as the cell m_{xy} with the occupancy probability 0.5 has the minimum value, i.e. the null information. Hence, such

⁸One experimental run was performed on *Stage*, while the laser data including and excluding the person's legs were reused in the mapping process to build respective maps.

cells do not have contribution to $I(m)$. Furthermore, the maximum information of a cell is obtained with the occupancy probability 0 for free or 1 for occupied.

Before we show the map information computed from the both built maps illustrated in Fig. 3.3, expected results should be discussed firstly. Since the grid cells lying in the area of the tracked guide-person are regarded as invalid, their occupancy probability is set to 0.5. Null information is contributed by these cells. These invalid cells could be observed again only when the person area moves. A comparison of the map information is illustrated in the areas shown with the blue arrows in Fig. 3.3(a) and 3.3(b). When the perceived person's legs are added into the built map, the whole map information would increase. Hence, the information computed from map 3.3(a) should be more than that computed from map 3.3(b). Our assumption is verified in Fig. 3.4. The map information $I(m)$ was computed every 20 iterations in the map process and stored in a log file. At the beginning of the experiment, e.g. from step 0 to step 50 in the log file, there is little difference between the two curves, as the person-following behavior was not started by the speech command "Follow me!". Therefore, the original laser scans were used. After "Follow me!" had been sent out by the mini-dialog module (see section 5.1.2), the proposed technique was triggered. In Fig. 3.4 the red curve shows the map information computed by using the proposed technique, while the blue one was obtained when the original laser scans were used. As

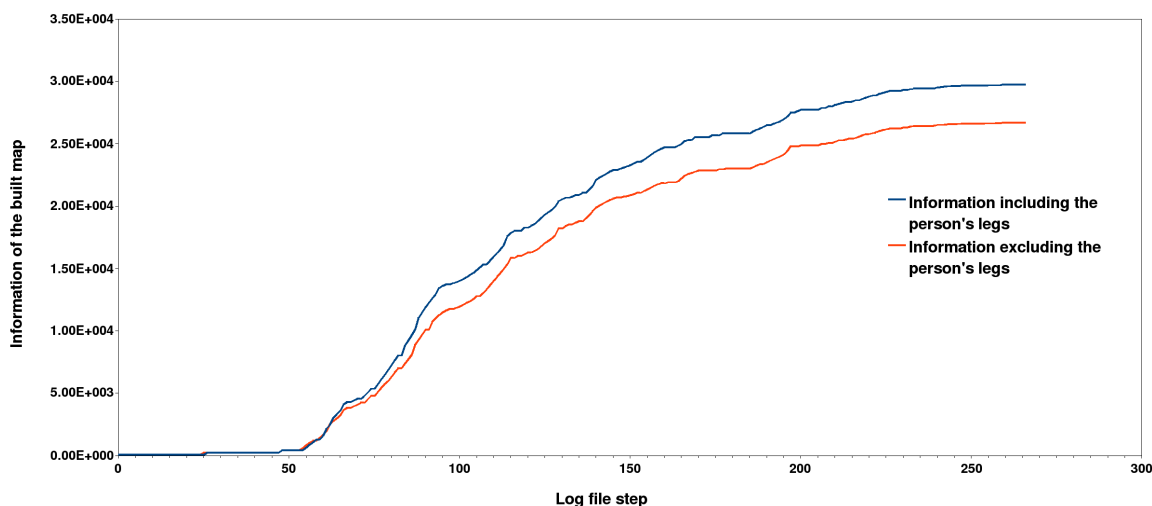


Figure 3.4.: Computed information of the built map according to equation 3.11. The results obtained by applying and not applying the proposed technique are illustrated in the red and the blue curve respectively. The information of the map is stored every 20 iterations in the mapping process.

illustrated in this figure the blue curve lies above the red one on the whole. Recalling the gray points illustrated in Fig. 3.3(a) the additional information arising from the grid cells occupied by the guide-person has been added into the environment representation. However, this additional map information does not reflect the real environment, because they are only the temporary imprints left by the guide-person. Therefore, this map information is actually *false* and should be removed from the environment representation. With the help of the proposed technique the false information can be reduced, as the discrepancy between the two curves in Fig. 3.4 illustrates. In other words, at the price of relatively less information about the surroundings, false information arising from the tracked guide-person can be excluded from the built environment representation as much as possible.

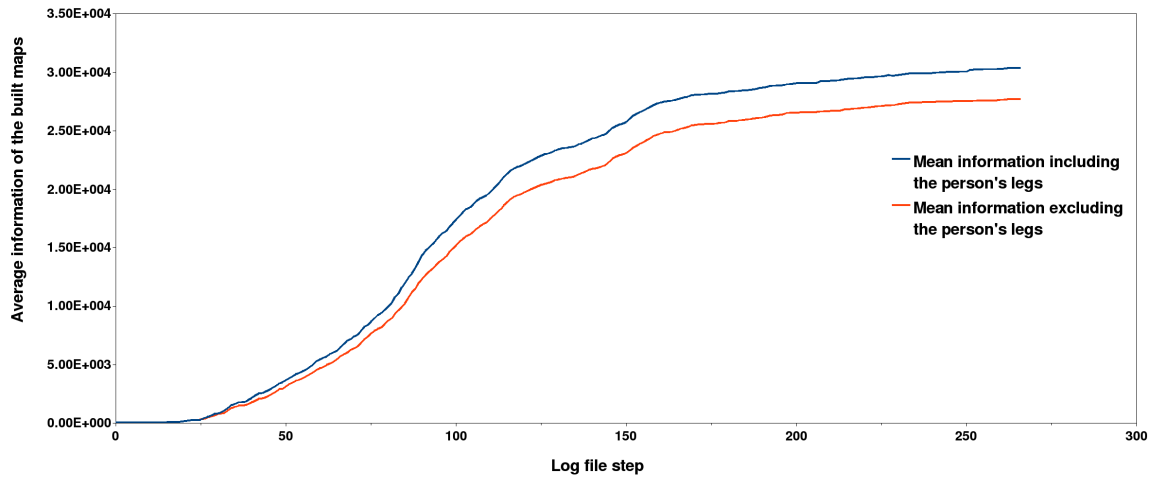


Figure 3.5.: Average information of the built maps computed from the six runs on *Stage*.

In order to verify the conclusion drawn from a unique experiment run five additional experiments were performed in the same simulated environment, except that different guide-paths were used. 266 steps of the mapping process were recorded in the log files for all the six runs. The same frequency, i.e. all 20 iterations in the mapping process, was used as before. For each step, the average map information was computed from the six runs. Besides, the exploration of the whole environment shown in Fig. 3.2(b) was limited to complete inside 266 steps. In other words, the robot has been guided throughout the environment before the 266-th step. Note that for the evaluation of pose uncertainty, results achieved from the six experimental runs will be discussed later on. The average information of the built maps is shown in Fig. 3.5. Comparing to the result illustrated in Fig. 3.4, consistent conclusion can be drawn. Applying the proposed technique, the information computed from the built maps is though less than what can be obtained by using the original laser scans, false information arising from the guide-person can be rejected as far as possible.

Performing a path planning task

Another issue to consider is the negative impact of the gray points shown in Fig. 3.3(a) on a path planning task. Particularly when the *spurious* obstacles mapped in the environment representation lie between the current robot position and a given goal, a map-based planning approach probably fails to find a path. An example is given in Fig. 3.6. The environment maps illustrated in Fig. 3.3(a) and Fig. 3.3(b) were used for the map-based path planning approach NF1 [92]. Because of the *spurious* obstacles on the way from the current robot's position to the given goal, NF1 cannot find a path in the map built without the proposed technique, as illustrated in Fig. 3.6(a). In contrast a path depicted in green has been planned for the robot, as Fig. 3.6(b) shows. Notice that comparing to Fig. 3.6(a) the lighter orange circle shows the estimated robot poses obtained directly from the mapping process, while the darker orange circle overlapping on the lighter one was computed on the basis of the estimated robot's poses. An interpolation was combined with the incremental odometry data in order to obtain a higher frequency of the poses of the robot, considering that the accuracy of the incremental odometry data were acceptable for a short time step.

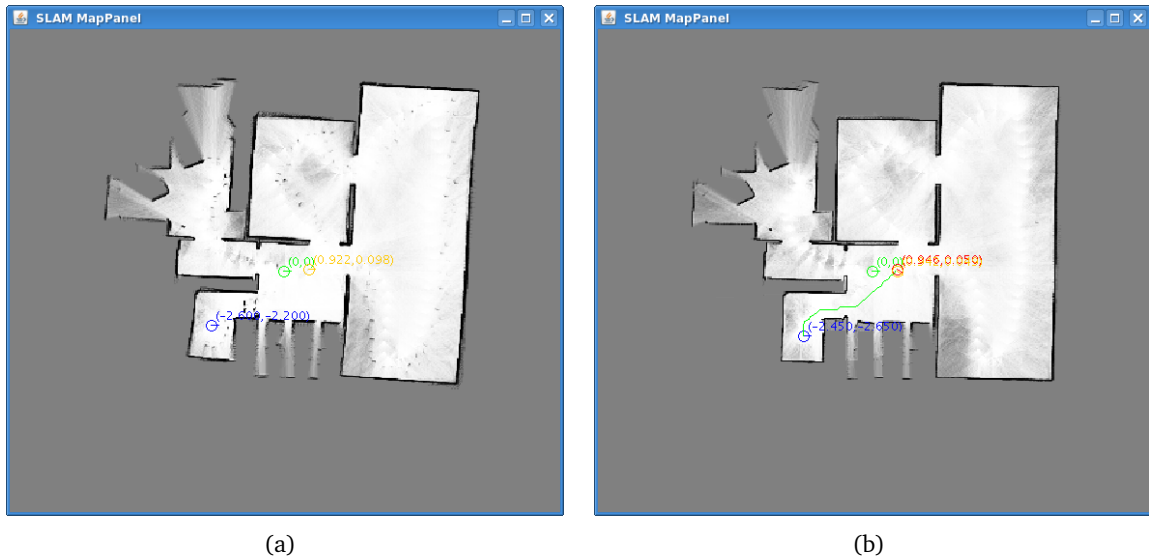


Figure 3.6.: The map-based path planning approach NF1 applied for the two maps shown in Fig. 3.3(a) and Fig. 3.3(b) respectively. The start, current and goal position of the robot is depicted with the green, orange and blue circle respectively. All these positions are given in the world coordinate system defined in the corresponding maps. The robot's goal is set on the similar location for the both maps. Because of the false obstacles lying on the way to the set goal, no path can be found in a). However, a path depicted with green is found in b).

Pose uncertainty

Apart from building a metric map in the *Home-Tour*, estimating the poses of the robot, i.e. the robot's path in the incrementally built map is the other task of a SLAM approach. The estimated robot's poses and the relevant information are therefore stored in a log file as well in the same experimental run on *Stage* for the two cases, i.e. RBPF-SLAM with and without the proposed technique. Considering the consistency with the computed map information the same frequency, namely all 20 iterations in the mapping process, was adopted. Since a mean pose vector and the corresponding covariance matrix⁹ are the two of the most widely used quantities for a probabilistic representation of robot's poses [152], these two quantities are selected to estimate the pose uncertainty of the robot. Fig. 3.7 shows the square root of the determinant of the pose covariance matrix for the two cases. The red curve illustrates the determinants of the pose covariance matrices when the disturbance from the guide-person has been reduced by our proposed technique, while the blue one was obtained when the original laser scans were used in the experiment. Notice that the red curve lies below the blue curve on the whole. In general, on the one hand, the pose uncertainty of the robot should increase when the robot is exploring an unknown area of the environment, as the ascending parts of the both curves are shown in Fig. 3.7. On the other hand, as the descending parts of the both curves illustrate, by revisiting a known area of the environment or by perceiving some parts of the known environment the robot would be more certain about its locations. Besides, the invalid areas occupied by the guide-person may affect the estimation of covariance matrices, because the real perception area of the

⁹In our case the mean pose of the robot is represented with a three dimensional vector $\mu(x, y, \theta)$ containing the robot's position (x, y) and the orientation θ , while the corresponding covariance Σ is a 3×3 matrix that is symmetric and positive-semidefinite.

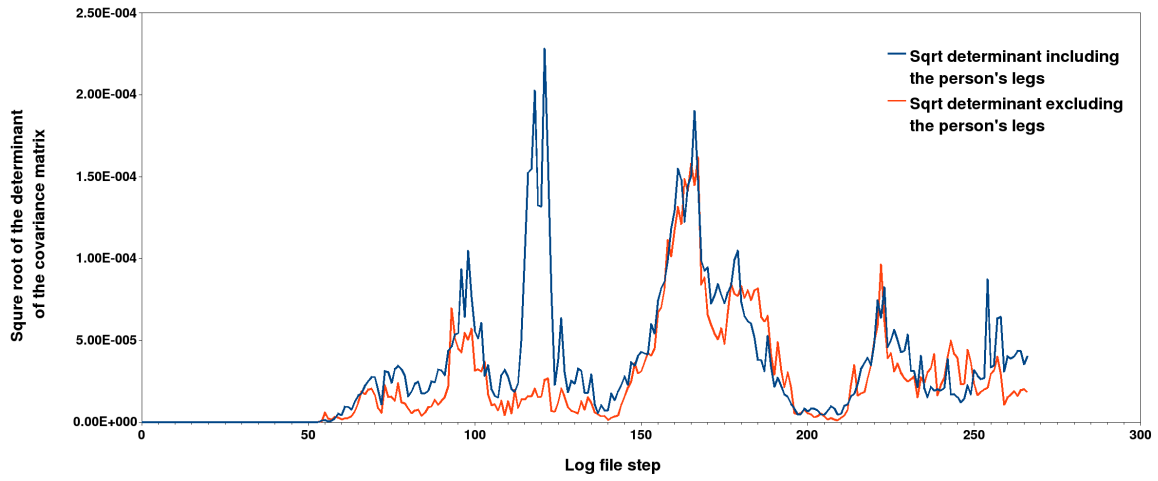


Figure 3.7.: Square root of the determinant of the pose covariance matrix in the two cases. The result curve obtained with our proposed technique lies below the blue curve at large.

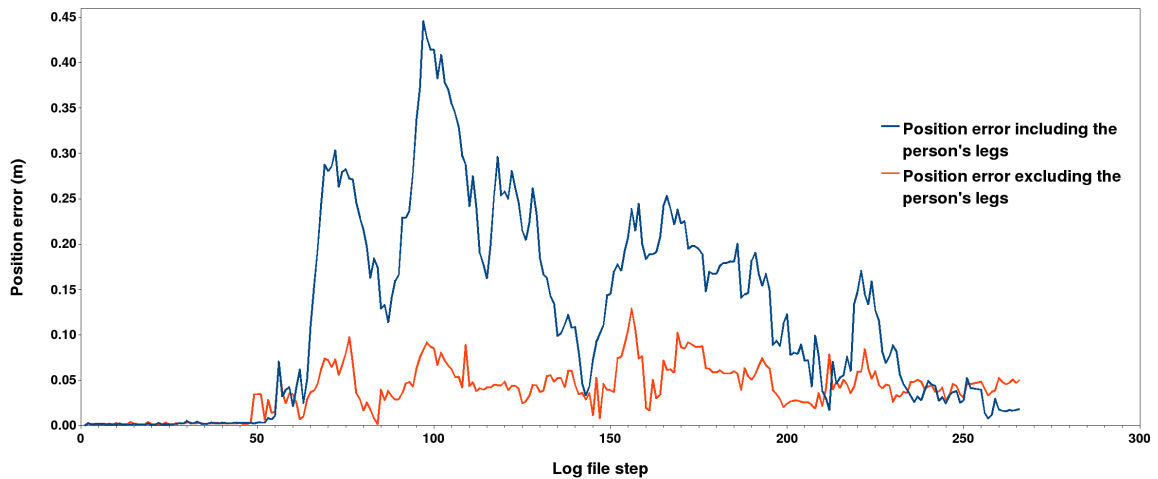


Figure 3.8.: Position error of the robot during the simulated *Home-Tour* in the two cases. The red curve presents a better result than the blue one achieved without our proposed technique.

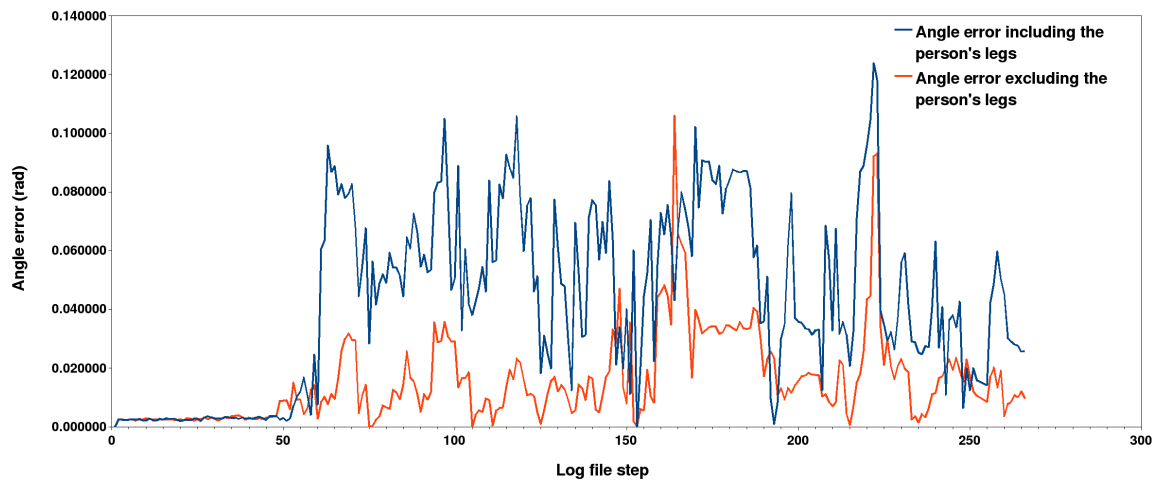


Figure 3.9.: Angle error computed during the experimental run on the simulation platform *Stage* for the two cases. The red curve illustrating the result obtained with the proposed technique lies below the blue curve at large.

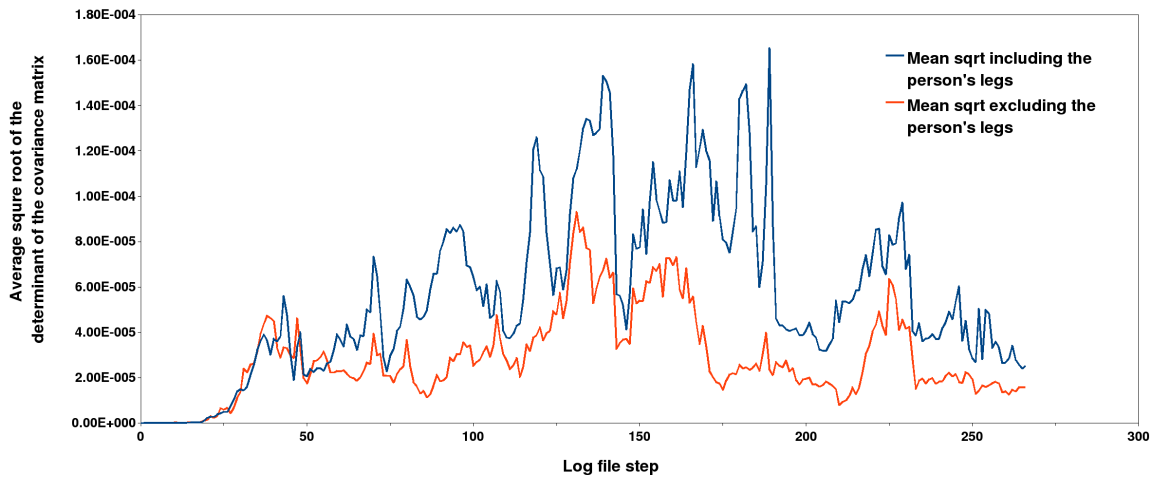
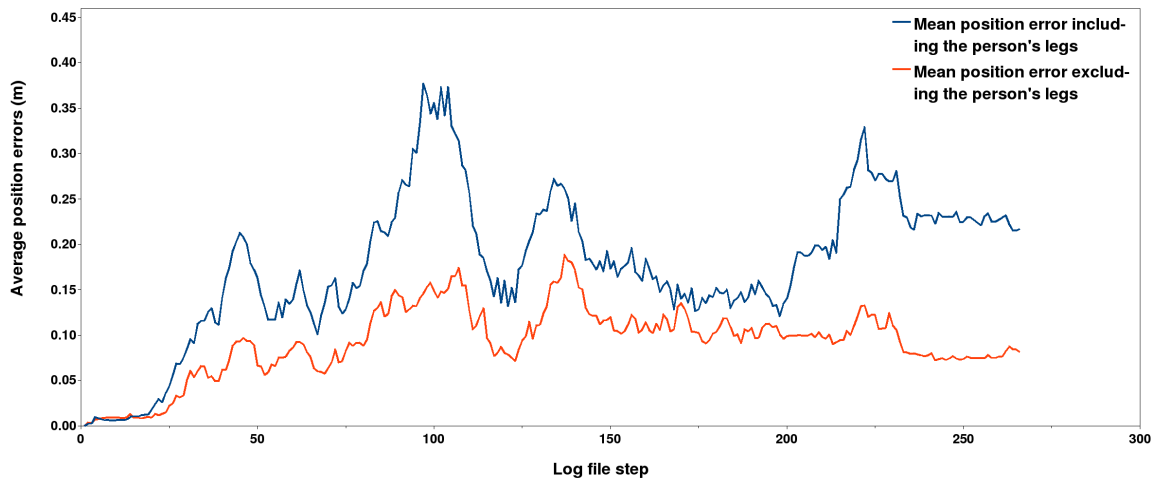


Figure 3.10.: Average curve of the square root determinant of covariance matrices achieved from the six runs on *Stage*.

Table 3.1.: Statistic results computed from Fig. 3.7, Fig. 3.8 and Fig. 3.9.

	Mean	Std-deviation	Reduced percent of mean	Reduced percent of variance
Sqrt-det inc. legs	3.50×10^{-5}	4.01×10^{-5}	28.6%	22.4%
Sqrt-det exc. legs	2.50×10^{-5}	3.11×10^{-5}		
Position-error inc. legs [m]	0.122	0.108	67.2%	75.9%
Position-error exc. legs [m]	0.040	0.026		
Angle-error inc. legs [rad]	0.043	0.030	62.8%	50.0%
Angle-error exc. legs [rad]	0.016	0.015		

robot is actually reduced. Nevertheless, the mean and the standard deviation computed from the red curve are both less than the results achieved from the blue one. The reduced percentage of the mean and the deviation is 28.6% and 22.4% respectively, as shown in table 3.1.

Figure 3.11.: Average position error obtained from the six runs on *Stage*.

As mentioned above, the covariance pose estimation of the robot is computed on the basis of the perceived sensor data and the built map. However, actual pose error of the robot cannot be obtained directly from the covariance matrices, since the covariance matrices do not provide any reference value of robot's pose. Therefore, mean position and orientation of the robot estimated in the mapping process, as well as odometry data obtained directly from *Stage* were recorded during the simulated *Home-Tour* on *Stage*. In addition, position error of the robot, i.e. the distance between the estimated and the odometry positions, and angle error of the robot, namely the absolute difference¹⁰ between the estimated and

¹⁰Note that only the acute angle intersected between the estimated and the odometry angle of the robot is considered and the angle error is therefore limited in the interval $[0, \pi]$.

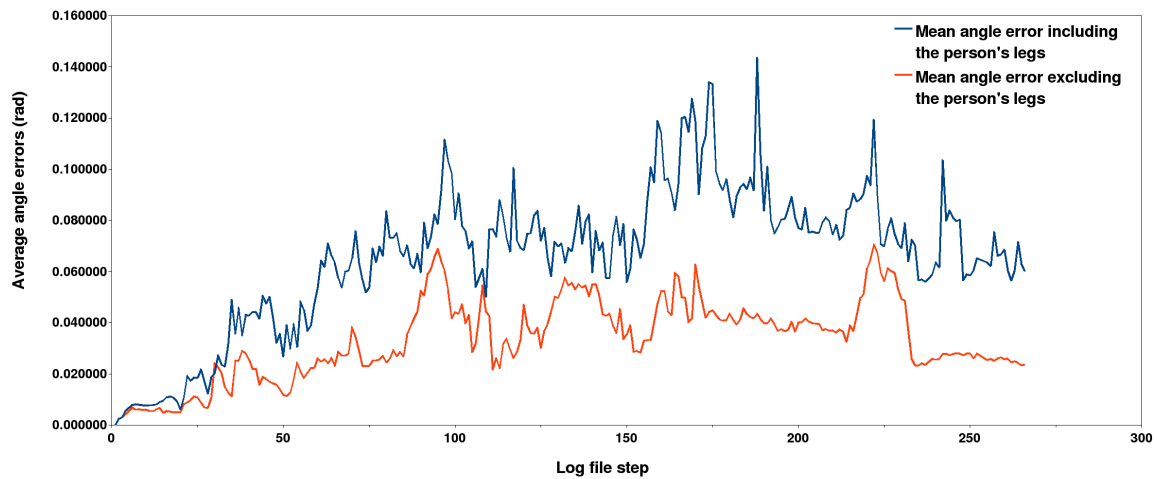


Figure 3.12.: Average angle error computed from the six runs on *Stage*.

the odometry orientations of the robot, were stored. Considering the two parameters of the position model defined in *Stage*¹¹, namely *odom* and *gps*, the latter was used in the experiment in order to get *perfect* knowledge about where the robot was in the simulation. Note that using position model *odom* noise can be added in the odometry recordings, and the odometry data will become increasingly inaccurate as the simulation goes on. Although this situation is closer to the reality, *perfect* odometry providing accurate poses of the robot is chosen because the odometry data can be regarded as the ground truth, which are necessary for the computation of pose error. By this means, conclusions of the proposed technique can be drawn from the experiment performed on the ideal simulation platform. Besides, the proposed technique has been integrated in the current system (see section 5.1.2) and evaluated in a user study conducted by inexperienced subjects. Evaluation results will be discussed in section 5.2.2. Fig. 3.8 and Fig. 3.9 shows position and angle errors of the robot respectively. Robot localization benefits from the proposed technique, as the red curves in the two figures illustrate. The negative effect arising from the perceived person's legs are illustrated by the blue curves of the two figures. Statistic results shown in table 3.1 verify again the superiority of our approach. The reduced percentage of mean and deviation for position errors are **67.2%** and **75.9%** respectively, while for angle errors a **62.8%** reduction for the mean and a **50%** reduction for the deviation is achieved.

As mentioned before in the discussion of map information, additional five experimental runs were performed in the same simulated environment with different guide-paths. Average values of sqrt (square root) determinant of the covariance matrices, position errors, as well as angle errors computed step by step from the six experiments performed on *Stage* are shown in Fig. 3.10, Fig. 3.11 and Fig. 3.12 respectively. Comparing to the results illustrated in Fig. 3.7, Fig. 3.8 and Fig. 3.9 respectively, errors in pose estimation of the robot during the mapping process can be reduced when the proposed technique is used.

Furthermore, statistic results calculated from Fig. 3.10, Fig. 3.11 and Fig. 3.12 are presented in table 3.2. In contrast to the results of table 3.1 obtained by one experiment, consistent results can be achieved from the six runs, i.e. due to the disturbance of the perceived person's legs in the *Home-Tour*, pose errors of the robot will increase.

¹¹See more information about the position model in the web page of Player/Stage Manual <http://www-users.cs.york.ac.uk/jowen/player/playerstage-manual.html>.

Table 3.2.: Statistic results computed from the Fig. 3.10, Fig. 3.11 and Fig. 3.12.

	Mean	Std-deviation	Reduced percent of mean	Reduced percent of variance
Average sqrt-det inc. legs	5.70×10^{-5}	3.72×10^{-5}	49.5%	48.9%
Average sqrt-det exc. legs	2.88×10^{-5}	1.90×10^{-5}		
Average position-error inc. legs [m]	0.178	0.079	48.3%	51.9%
Average position-error exc. legs [m]	0.092	0.038		
Average angle-error inc. legs [rad]	0.066	0.027	50.0%	44.4%
Average angle-error exc. legs [rad]	0.033	0.015		

3.3. Summary and Discussion

Simultaneous Localization And Mapping (SLAM) is a basic problem in robotics, which has been discussed for over 20 years. In this chapter, I proposed an extension to a state-of-the-art SLAM approach [9] that improves its performance in interactive mapping. The basic idea of Rao-Blackwellisation-based SLAM approaches was introduced firstly. By applying Rao-Blackwellisation it is possible to reduce the sample space. A two-step process, i.e. an incremental mapping process with known poses and the pose estimation in the built map, is therefore feasible.

Next, the negative effect arising from the perceived guide-person was discussed. Both the mapping process and the pose estimation of the robot may be influenced, because the person's legs are tracked by the robot and regarded as dynamic obstacles all along the *Home-Tour*. Considering the person detection and tracking system introduced in section 2.2.1, I presented a robust technique, in which the area occupied by the guide-person can be detected in the laser scans. The corresponding laser points were regarded as invalid for the SLAM approach. As the regions occupied by the guide-person were removed from the perception area of the robot as far as possible, they cannot contribute any information to the built map. The total information of the environment gathered with the proposed technique is less than what can be obtained using the original laser scans. Nevertheless, dynamic obstacles arising from the person's legs may be excluded from the built environment representation as far as possible. On the one hand, pose estimation of the robot computed in the incrementally built map is disturbed by the dynamic obstacles, i.e. the guide-person. On the other hand, map-based path planning approaches probably fail to find a path for the robot due to the false obstacles added into the environment representation.

At last, I discuss the evaluation results of the proposed technique achieved from one experiment and from six experiments on the simulation platform *Stage*. Consistent conclusions have been drawn. Applying the proposed technique less information computed from the environment actually results in a more accurate environment representation, since most of the false obstacles arising from the simulated person's legs can be rejected by the mapping process. Hence, negative effect of the false obstacles on a map-based path planning approach can be reduced, assuming that the environment is static. Besides, the localization

process performed in a relatively static map may provide more accurate pose estimation of the robot, because the disturbance of the guide-person is moderated. Recalling that the areas occupied by the guide-person are regarded as invalid in the laser plane, the corresponding laser points are used neither for the mapping process nor for the pose estimation of the robot. A finer definition of the area actually occupied by the guide-person (in our case, the two or occluded person's legs in the laser scans) would reduce the number of the invalid laser points. Thus, the extended perception area might benefit a SLAM approach. However, the results of the experiments validate the effectiveness of our technique, apart from verifying the negative effect resulting from the perceived guide-person on the RBPF-SLAM approach.

Note that the mapping process discussed in this chapter was performed in a simulated small-scale indoor environment, i.e. the BIRON apartment partially simulated by *Stage*. The pose uncertainty of the robot may be decreased, when the robot re-enters known terrains during the *Home-Tour*, as those local valleys of the two curves in Fig. 3.7 illustrate. In contrast to SLAM in small-scale environments, many aspects, such as improving the computational efficiency [62], closing multiple nested loops [11], some problems of standard particle filters in large-scale mapping discussed in [102], etc., have to be considered for the solution to the SLAM problem in a large-scale environment [16]. However, in theory the results shown in this chapter verified the feasibility of our approach in a small-scale home environment, because the robot may naturally revisit some known areas or at least perceive the part of them under the guidance of the person. Besides, an occupancy grid representation in a small-scale environment and the relatively small number of particles required on the robot's path actually moderate the computation and memory problems in comparison with large-scale SLAM. In chapter 5 the robustness of the proposed technique will be validated on our robot platform in a real environment.

4. Path Planning Adopting Human Strategies

One of the tasks performed in a *Home-Tour*, i.e. building a metric environment representation, has been discussed in the last chapter in detail. So far, the map of the environment initially unknown to the robot has been built in a human-robot joint exploration. Using this environment map how to reach a goal defined in the map has to be solved, so that the robot can move around in the environment and provide services to its user. The capability of safe navigation in the environment is necessary for a service robot placed in a person's home. However, natural human-inhabited environments still pose significant challenges for robots despite the impressive progress that has been achieved in the field of path planning and obstacle avoidance. These challenges mostly arise from the fact that (i) the perceptual abilities of a robot are limited, thus sometimes impeding its ability to perceive relevant obstacles (e.g. transparent objects), and (ii) the environment is highly dynamic due to e.g. movable objects and humans. Recalling the idea of *Human-Robot Joint Spatial Exploration* and the interactive *Home-Tour* scenario, an integrated solution building upon the analysis and use of implicit human knowledge in path planning to these challenges will be elaborated on in the following in this chapter. The knowledge gained interactively from the guide-person is therefore incorporated in the environment representation, except for the information gathered from a standard robotic mapping process. By this means, a topological representation can be built.

As introduced in section 1.2, in general path planning answers the question “*How can I navigate to a goal position from my current location?*”. Most of the established approaches assumed that a priori knowledge of the environment was available (usually a 2D map representation denoting obstacles and free spaces¹). They tried to compute a plan towards a goal for the robot [105]. These approaches are purely map-based, without the consideration of any prior experience in navigating the environment. Generally, planning algorithms can be classified in two types. The first is to compute a path from the robot's start position to the goal deterministically [29], using a navigation function, such as NF1 introduced in [92]. In contrast, the other type of approaches tackle the problem probabilistically, which try to plan a sequence of actions, i.e. a path, associated with all possible positions of the environment to the goal [128]. Probabilistic nature of robot motion and perception [72, 170] are considered explicitly in the planning domain. Such a plan is generally referred to as a universal plan [105].

As discrepancies resulting from dynamic obstacles, e.g. movable doors and people, almost always exist between the a priori map and the real environment, local strategies dealing with dynamic environments are necessary for the robot to execute the computed path. Several classical approaches have been proposed to continuously react to obstacles perceived by robot sensors, e.g. DWA (Dynamic Window Approach) [48] achieving optimal actuator commands by searching in a velocity space, VFH (Vector Field Histogram) [14] employing

¹Note that very recently, using the Microsoft Kinect (see <http://www.xbox.com/en-US/kinect>) based on the design by PrimeSense (see <http://www.primesense.com/>), 3D scene information is achieved from a continuously projected infrared structured light.

a two-stage data-reduction process to represent the world model with a polar histogram and select the most suitable steering direction among this histogram with a low polar obstacle density, *ND* (*Nearness Diagram*) [107] describing an environment (robot and goal location, obstacles, as well as free space) with sectors centered on the robot's location and designing actions of the robot according to predefined situations. More recently, reactive navigation solved with the help of a parametrized space, namely a trajectory parameter space has been proposed by Minguez et al. [108] and extended by Blanco et al. [12]. On the basis of trajectory parameter spaces, decoupling the problems of kinematic restrictions and obstacle avoidance is feasible. Furthermore, the robot is able to efficiently carry out the long-term (global) plan, while reacting to unexpected obstacles quickly.

However, a major drawback of the path planning approaches mentioned above should not be left unconsidered. Due to the limitation of robot perception, potential hazards may not be all contained in the built representation of the environment. Besides, the computational complexity has to be taken into account especially for the calculation of a universal plan in real time. Therefore, we propose a path planning approach that not only relies on the static configuration of the environment but instead utilizes the knowledge gained from an analysis of human pathways. A combination of human-aware and map-based path planning, dynamic re-planning abilities, and dynamic obstacle avoidance will be discussed in this chapter. People sharing the environment with the robot should be taken into account under the assumption that path planning might benefit from human strategies. Following the path selected by the guide-person in the *Home-Tour* no unconquerable obstacles should be left for the robot, when the environment is assumed to be static. Thus, robust path planning is achieved. In addition, avoiding potential dynamic hazards, e.g. movable objects and people near the doorway, is expected to be considered by the person guiding the robot in the *Home-Tour*, and thus benefits the robot as well. Consequently, people are invited in the designed *Home-Tour*, in which the robot can not only explore the environment and build a map effectively (see chapter 2 and chapter 3 respectively), but also observe the behavior of the guide-person during the exploration of the environment to positively affect its own navigation abilities. Note that in an unconstrained environment the two-dimensional perceptual space of a fixed mounted laser is not sufficient to ensure safe navigation. 3D capturing techniques are therefore necessary, especially when dynamic obstacles out of the 2D laser plane obstruct the path planned for the robot.

The remainder of this chapter is organized as follows. After discussing the most recent work on path planning in the context of an interactive scenario in section 4.1, details of path planning and the re-planning scheme are explained in section 4.2. Besides, results of the proposed path planning and re-planning approach achieved in an empirical study are presented as well at the end of section 4.2. Considering dynamic obstacles that may become potential hazards for the robot, a multisensor-based motion generation is discussed in section 4.3, before the summary and discussion of this chapter in section 4.4.

4.1. Related Work

In 2007, Gockley et al. [58] were one of the the first to propose that knowledge about paths of a human guide could be exploited by robots needed to make regular trips between specific locations in hospital environments. Path planning for a mobile robot in a domestic environment has been widely studied, e.g. by Topp [153] and Zender et al. [169]. In both work a robot followed a person during a guided tour. The geometric features of the environment representation are lines extracted from straight structures like walls, without

details of small objects. Therefore, the built map is not necessarily sufficient for navigation actions. The so-called *Navigation Nodes* leading the robot to a goal were selected from the robot trajectory and kept sparse. The robot's path computed from these coarse navigation nodes was based on a *static* graph, in which the shortest route was found. Their representation does not contain any information about the actual pathways that people walked along. It creates new nodes only if no node exists within a distance of approximately $1m$. Moreover, the choice of sparse nodes for path planning might neglect details of people's movement, i.e. strategies of the guide-person. These strategies could be helpful for obstacle avoidance, e.g. going through a door, or avoiding objects that might not be perceived by robot sensors. The robot may get into trouble as well, when it is far from those *Navigation Nodes* and without connection to them, or the connections between the neighboring nodes are cut off by dynamic obstacles. Therefore, appropriate approximation and representation of person's trajectories, an occupancy map-based path planning approach and the capability of re-planning are key for robust navigation in an unconstrained indoor environment.

Recently Sisbot et al. [133] have proposed a human aware motion planner (HAMP). In their work the robot's path was calculated according to the recognized status of people, obstacles distributing in the near of people, as well as a known environment representation. In order to achieve a socially acceptable robot behavior in a human-shared environment, the path planned for the robot took people into account. However, accurate human postures, such as standing and sitting, the field of view and the gaze direction of a person required by their approach are not necessarily recognized stably and in time in practice. Besides, due to the limitation of robot perception the computed path may be infeasible. In contrast, exploiting the knowledge gained interactively from the guide-person unperceivable obstacles may be avoided.

On the basis of the insight obtained from the observation that how people can safely navigate through crowds, the freezing robot problem² can be solved by engaging in joint collision avoidance³. By this means, the robot was able to proceed through crowd to reach the given goal. However, observations of all dynamic agents are needed for this approach, which are difficult to achieve precisely by robot sensors from crowds in a real world scenario.

4.2. Path Planning

Considering the disadvantages of pure map-based approaches, static trajectory-based systems, the human aware motion planner, as well as the approach for unfreezing the robot discussed above, a graph-based planning and re-planning approach using the information gathered interactively in the guided tour has been designed. The graph is built on the basis of pathways selected by the guide-person during the *Home-Tour*. Since the search space of the graph is constrained to the person's trajectory, no path can be computed especially when the robot is far from the pathways of the guide-person. Therefore, a map-based path planner has been integrated into the proposed approach as a supplement. Besides, dynamic objects obstructing the path planned for the robot can be detected in the built occupancy grid map discussed in chapter 3. Thus, the robot is expected to benefit from human navigation strategies considering the following points:

²When every path planned for the robot is expected to make a collision, the robot stands still or performs unnecessary evasive action to avoid crowd. See [154] for details.

³It is called social forces model in [74].

- ▷ On the basis of the person's trajectory a robust path can be computed, as no obstacles that cannot be overcome by the robot should be left on the way.
- ▷ The robot follows a conventional path that would be expected by people as well.
- ▷ The trajectory of the robot would be predictable, since the path computed for the robot has been selected by the guide-person in the *Home-Tour*.

Besides, the re-planning process will be triggered when the current plan is obstructed by dynamic obstacles.

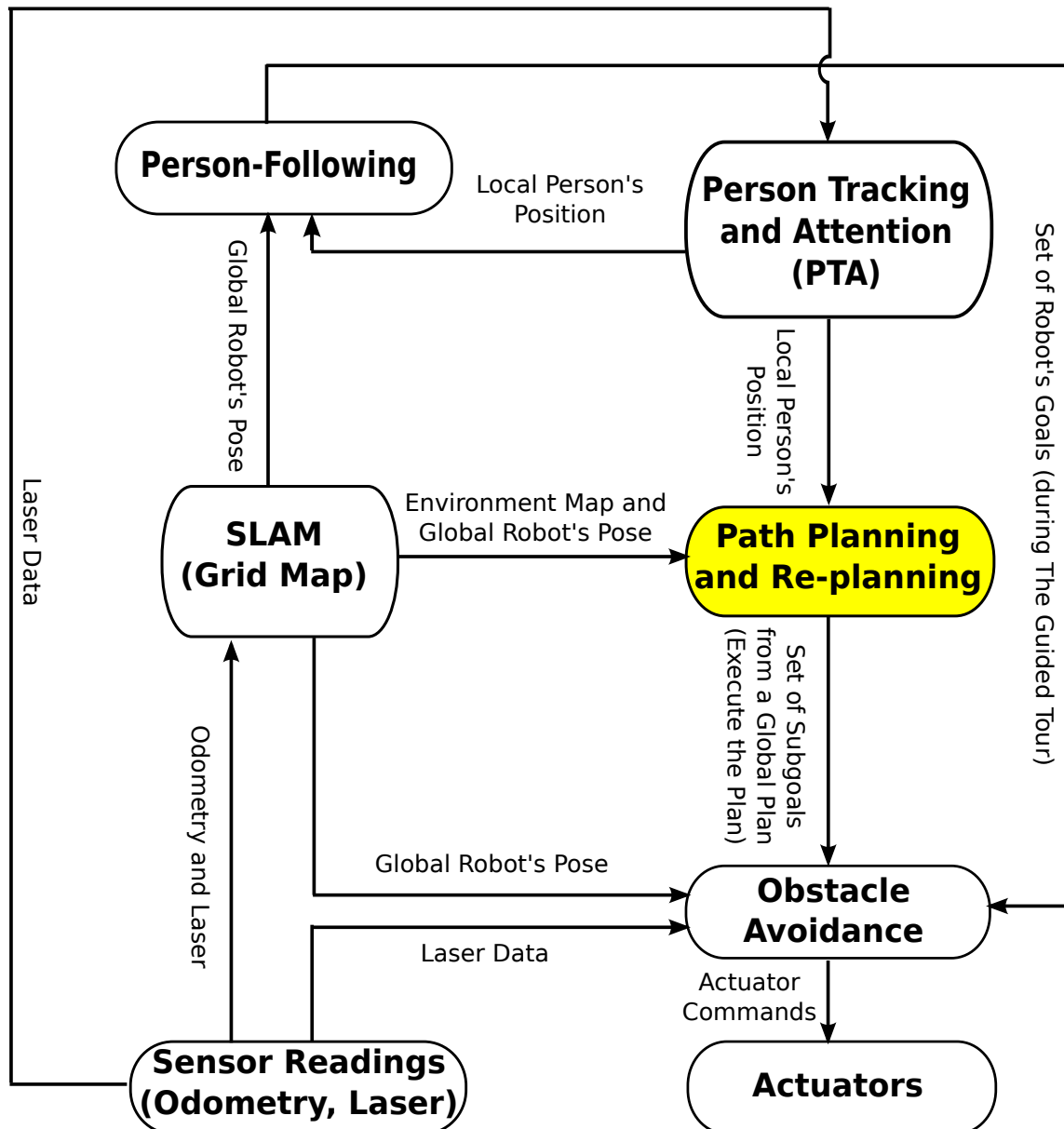


Figure 4.1.: Data flow of the proposed path planning and re-planning approach. Directions of the data flow are shown with arrows. Outputs of the components are commented nearby the corresponding arrows. The person's trajectory during an interactive scenario and the environment map are both observed by the path planning module, in order to compute or re-compute a path for the robot.

The data flow centered on the component *Path Planning and Re-planning* marked with yellow is presented in Fig. 4.1. Recalling the person-following behavior discussed in chapter 2, the guide-person is tracked by the component *Person Tracking and Attention* during the *Home-Tour*. Person positions are expressed with polar coordinates in the robot's local frame. Combining the local person's positions with global robot's poses estimated in the mapping process (see section 3.2), the person's trajectory is represented in the built map as well. In terms of the built environment representation and the person's trajectory a sequence of actions, i.e. the set of robot's subgoals, are computed by *Path Planning and Re-planning*. Notice that the set of robot's goals computed by *Person Tracking and Attention* are used to perform the *Home-Tour*, while the subgoals sent from *Path Planning and Re-planning*⁴ constitute the global path computed for the robot after the guided tour when a *location* (see definition in section 1.2) is sent to the robot. In order to execute the path computed or re-planned for the robot, a component of obstacle avoidance is necessary. Checking the current robot's pose, perceived laser data and the sequence of subgoals, actuator commands, i.e. translation and rotation speed, are sent out.

For the proposed path planning approach it is assumed that the person's trajectory represents basic behavior information of the guide-person. Thus, following the pathway of the guide-person, the robot actually adopts human navigation strategies. As will be elaborated on in the following section, the person's positions selected from the trajectory are recorded to build a graph. Recalling the definition of *location* given in section 1.2, human spatial concepts, e.g. rooms (regions) in the environment, are able to be incorporated as well in the built graph representation. Paths connecting the regions are computed from the trajectory of the guide-person. By means of the *locations* and paths connecting these *locations*, a topological graph can be built upon the occupancy grid environment representation during the interactive *Home-Tour*. Thus, *Path Planning and Re-planning* can be carried out according to the topological representation, while taking into account the information obtained from the metric representation, i.e. free spaces and obstacles represented in the occupancy map.

4.2.1. Graph Creation

Movement trajectory of the guide-person is observed and recorded by the robot during the guided tour. *Person Tracking and Attention* shown in Fig. 4.1 provides person's positions with respect to the robot frame. Combining the global robot's poses estimated in the built occupancy grid map, the person's trajectory can be computed in this map, i.e. the world frame. The naive path planning approach is to make the robot follow the person's positions stored in the guided tour in chronological order. By this means, the robot follows the person's trajectory exactly. Then a path from A to B can be found, only if the guide-person has gone directly from A to B before. Thus, trajectory changes are not possible, as shown in Fig. 4.2(a). Besides, human paths containing unnecessary detours (see Fig. 4.2(b)) cannot be detected. Hence, utilizing the original human trajectory needs to be compromised with the basic requirement for path planning, i.e. making the robot to reach a given goal. In other words, following the human trajectory as far as possible, sticking to the trajectory of the guide-person if possible and changing trajectories to reach a goal must be all taken into account. Derivation of the proposed solution begins with building a graph based on the recorded person's trajectory.

⁴Note that the subgoals can be computed by either the built graph (see section 4.2.1) or a map-based path planner (see section 4.2.2).

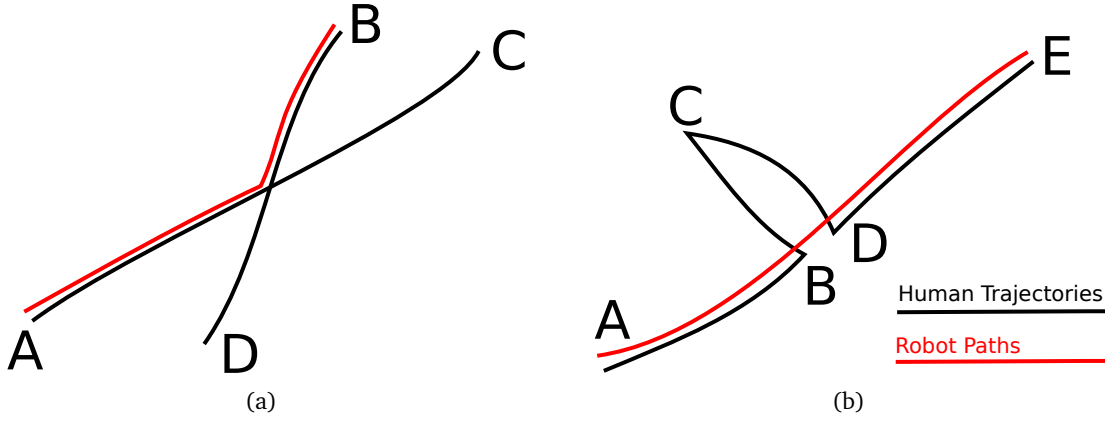


Figure 4.2.: Human trajectories (black) and robot's paths (red): In a) the original human trajectories are from A to C and from B to D. If the robot intends to navigate from A to B, a trajectory switch is necessary. In b) the person went from A, via B, C and D to E. If the robot navigates from A to E, the robot does not need to take the detour over C (unless there is an obstacle between B and D, which can be avoided using the path via C).

At first, the set of nodes in the graph are investigated. All positions recorded in chronological order for one guide-person in the *Home-Tour* are inserted as the set of nodes N into a graph $G_{PerTraject}$, where $N = \{N_1, N_2, \dots, N_i, N_{i+1}, \dots, N_n\}$, i is the subscript and n is the number of the nodes⁵. Note that N_1 and N_n records respectively the start and end person's position in the *Home-Tour*. The creation of edges in the graph will be discussed later on in this section. Assuming that m locations $L_1, \dots, L_j, \dots, L_m$ are specifically shown to the robot one after another, the corresponding person's positions firstly recorded at the m locations are denoted as $N_{SL_1}, \dots, N_{SL_j}, \dots, N_{SL_m}$, where $SL_1, \dots, SL_j, \dots, SL_m$ represents the respective subscripts of the nodes in graph $G_{PerTraject}$. Considering the start and end position of the guide-person, i.e. N_1 and N_n , $m + 2$ key nodes $N_1, N_{SL_1}, \dots, N_{SL_m}, N_n$ are selected from graph $G_{PerTraject}$. Thus, n nodes in graph $G_{PerTraject}$ can be divided into $m + 1$ subsets defined as follows:

$$N = \{N_{sub_1}, \dots, N_{sub_j}, \dots, N_{sub_{m+1}}\}, \quad (4.1)$$

where

$$N_{sub_1} = \{N_1, \dots, N_{SL_1-1}\} \quad (4.2)$$

$$N_{sub_j} = \{N_{SL_{j-1}}, \dots, N_{SL_j-1}\} \quad \text{with } 1 < j < m + 1 \quad (4.3)$$

$$N_{sub_{m+1}} = \{N_{SL_m}, \dots, N_n\}. \quad (4.4)$$

In other words, the whole graph $G_{PerTraject}$ that comprises the n nodes is divided into $m + 1$ graph layers $G_{Layer_1}, \dots, G_{Layer_{m+1}}$, according to the corresponding subsets. Each graph layer records the person's positions of the corresponding sub-trajectory. Each node N_i of the graph $G_{PerTraject}$ can be assigned to a corresponding layer $G_{Layer(N_i)} = G_{Layer_k}$, where $k \in \{1, \dots, m + 1\}$, when $N_i \in G_{Layer_k}$ ⁶. Actually graph $G_{PerTraject}$ can be regarded as *graph union* or *graph join* of all graph layers [70], since no edges have

⁵Note that whether the person's positions are different from each other is not checked. In other words, different nodes may store same person's positions.

⁶Note that a node with the same coordinate might be passed by the guide-person not only once. For such nodes they might belong to different graph layers.

been connected between nodes yet. Furthermore, the whole person's trajectory consisting of set N can be correspondingly segmented into $m + 1$ sub-trajectories. Recalling equation 4.2, the first sub-trajectory represents the subset of person's positions from the first node recorded after the *Home-Tour* starts, i.e. N_1 , to the last node recorded before the first *location* L_1 is shown to the robot, i.e. N_{SL_1-1} . The last sub-trajectory, which represents the subset of person's positions from the first node recorded after the last *location* L_m is indicated by the guide-person to the last node recorded before the *Home-Tour* is completed, is given in equation 4.4 The rest sub-trajectories are given in equation 4.3, where one sub-trajectory begins at the first node recorded when *location* L_{j-1} is shown to the robot, i.e. node $N_{SL_{j-1}}$, and ends at the last node recorded before the subsequent *location* L_j is indicated to the robot, i.e. node N_{SL_j-1} . Considering the fine resolution of the occupancy grid map ($0.05m$ in our case) a relative rough virtual grid with a predefined resolution is laid over the built map, in order to store person's position effectively. Furthermore, the number of graph nodes is limited to one per virtual grid cell.

The connection of graph nodes is created simply under the restriction of distance. In other words, newly added nodes get connected by an edge to all neighbor nodes whose grid cell distance is smaller than a certain threshold⁷. By this means, on the one hand nodes lying on different graph layers might be connected, when they are near to each other. On the other hand, no edge might be connected between nodes on the same layer, when they are far from each other. The complete graph $G_{PerTraject}$ can be therefore created with an operation resembling *graph join* of multiple layers. However, the edges of graph $G_{PerTraject}$ are built in terms of the predefined restriction of distance. Edge weight $\omega(N_i, N_j)$ is initially set to the euclidean distance between the connected points N_i, N_j . The A^* search algorithm [71] is applied on the graph $G_{PerTraject}$ to find the least-cost path if it exists. According to the initial weights of the edges the shortest distance path will be found.

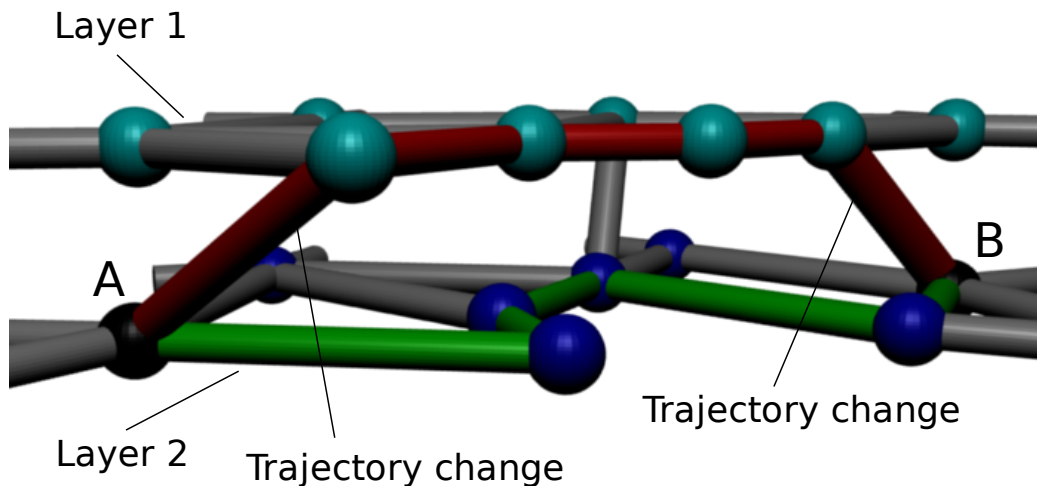


Figure 4.3.: Two layers created from the corresponding human sub-trajectories. Light blue nodes belong to layer 1, while dark blue nodes belong to layer 2 including the two black nodes A and B. Two possible paths from A to B are depicted with red and green edges. The graph search algorithm would prefer the green path, since the red one contains two expensive layer (sub-trajectory) changes.

However, the shortest path is probably not consistent with the path originally chosen by the

⁷Note that obstacles between two connected nodes are not checked by graph creation, but instead effectively in the re-planning process. See next section for details.

guide-person, as some details of the person's trajectory might be ignored by A^* . In order to make the robot follow the trajectory of the guide-person as far as possible, timestamp $ts(N_i)$ memorizing when node N_i is added into the graph is taken into account. Considering the fact that a long time span between two nodes implies that they are not neighbors on the original human path, the difference of the timestamps between two selected nodes is computed. If the timestamp difference is greater than a defined threshold Θ_t , the initial edge weight will be multiplied by a penalty factor α_t .

In addition, frequent changes between different human sub-trajectories should be avoided. The robot ought to follow one sub-trajectory as long as possible, unless trajectory switches are necessary for the robot to reach a goal. As described above, instead of building one graph recording all person's positions, multiple graph layers are created (see Fig. 4.3). Edges connecting nodes from different layers are allowed. Nevertheless, they will suffer from a multiplied penalty factor α_l on the basis of the initial edge weights. This, again, makes graph layer changes rare, although not impossible.

In summary, the path computed by A^* is a compromise between the shortest distance way leading the robot to a goal and an imitation of human behavior watched by the robot during the *Home-Tour*. Thus, the edge weight $\omega(N_i, N_j)$ between two nodes N_i, N_j of the graph $G_{PerTraject}$ is defined as follows:

$$\omega(N_i, N_j) = p_t(N_i, N_j) \cdot p_l(N_i, N_j) \cdot \left\| \overrightarrow{N_i N_j} \right\|, \quad (4.5)$$

where

$$p_t(N_i, N_j) = \begin{cases} \alpha_t & \text{if } |ts(N_i) - ts(N_j)| > \Theta_t, \\ 1 & \text{otherwise} \end{cases} \quad (4.6)$$

and

$$p_l(N_i, N_j) = \begin{cases} \alpha_l & \text{if } G_{Layer}(N_i) \neq G_{Layer}(N_j). \\ 1 & \text{otherwise} \end{cases} \quad (4.7)$$

Note that $\alpha_t > 1$ and $\alpha_l > 1$. In section 4.2.3 we will show that the proposed path planning and re-planning approach is a trade-off allowing the robot both to benefit from qualities of human navigation and to autonomously make decisions on the switch of graph layers.

4.2.2. Re-planning

A path planned for the robot consists of a set of subgoals leading to a goal. A precise plan computed by means of a static environment representation may be inappropriate to be executed due to dynamic obstacles, such as people or movable furniture. Using the laser sensor, component *Obstacle Avoidance* shown in Fig. 4.1 is able to make the robot circumvent obstacles on the way to subgoals. Whenever a subgoal near to a dynamic obstacle cannot be reached within a certain time period, the robot tries to go to the next subgoal that could lie in free space. Thus, *Obstacle Avoidance* is expected to overcome temporal obstacles, such as passers-by, or objects obstructing the path not completely, and drive the robot to the goal using the current plan.

If the effort of reactive navigation (component *Obstacle Avoidance*) is not successful⁸, the current plan is marked invalid. Obstacles between the robot and the current subgoal are

⁸When the robot cannot reach the current and the next subgoal within a certain time period, it is assumed that *Obstacle Avoidance* fails.

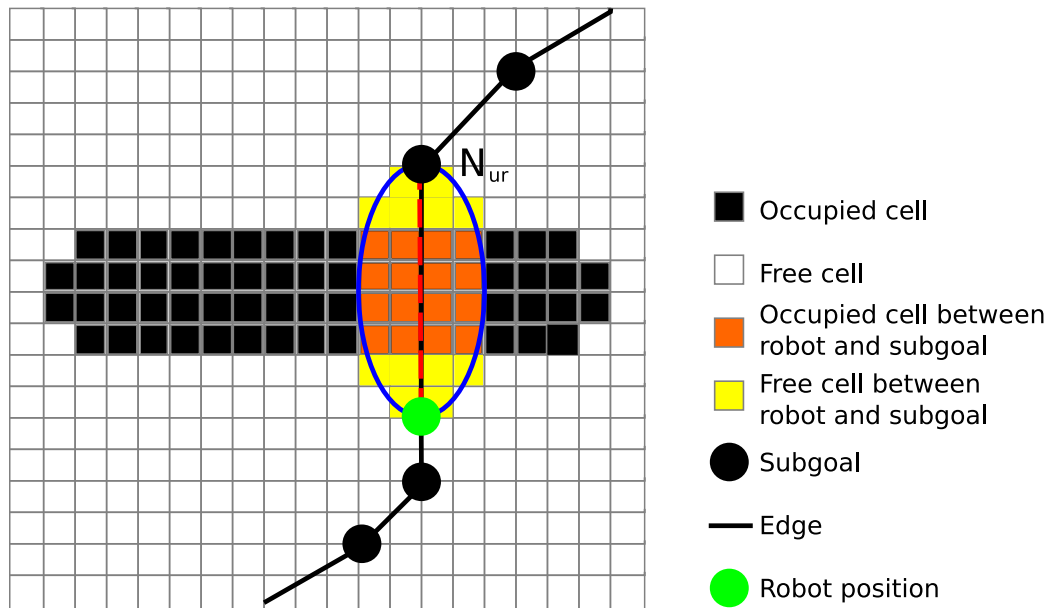


Figure 4.4.: Finding dynamic obstacles on the grid map. Free grid cells are depicted in white, occupied cells in black. An obstacle is detected, since the occupied cells (orange) outweigh the free cells (yellow) in the ellipse area between the robot and the subgoal.

checked in the built occupancy grid map. For this purpose, grid cells contained in an ellipse covering the area between the robot and the subgoal are inspected, as Fig. 4.4 illustrates. When the percentage of occupied cells in this area exceeds a threshold, it is assumed that obstacles have obstructed the current plan. Re-planning is therefore triggered. Since the value of a grid cell indicates the occupied probability of this cell, in order to avoid recognizing false occupation of temporal obstacles, only the cells with high value are regarded as occupied. Considering the failure of reactive navigation and the detected obstacle between the robot and the subgoal, this subgoal is marked as an unreachable node N_{ur} . Given the radius of the robot, all neighbors of the N_{ur} inside this radius are marked as unreachable as well. A^* -search is then repeated, ignoring those unreachable nodes. If no obstacle can be detected in the path found by A^* but *Obstacle Avoidance* still fails, map-based path planning is used for path planning as the last resort⁹.

Furthermore, map-based path planner can be seen as a general fall back strategy whenever A^* cannot find a path from the graph $G_{PerTraject}$ discussed above. This is the case if either no connection between the start and the target node can be found in the graph, or the robot's position is far away from any node of the graph. The second case mainly occurs when the robot is manually placed at a location, where it has not been before, or it is exploring autonomously. While the robot is following the NF1 path, A^* constantly tries to find a path switching back to the behavior computed from the graph $G_{PerTraject}$ as soon as possible.

Considering the graph-based path planning approach and re-planning strategies discussed above, the robot is expected to benefit from both map-based path planning and navigation strategies of the guide-person. Thus, the robot is able to effectively solve the planning problem “How can I navigate to a goal position from my current location?” posed at the

⁹Due to errors in robot sensors and the mapping process, as well as the real response of the robot to actuator commands, such a situation might occur in practice.

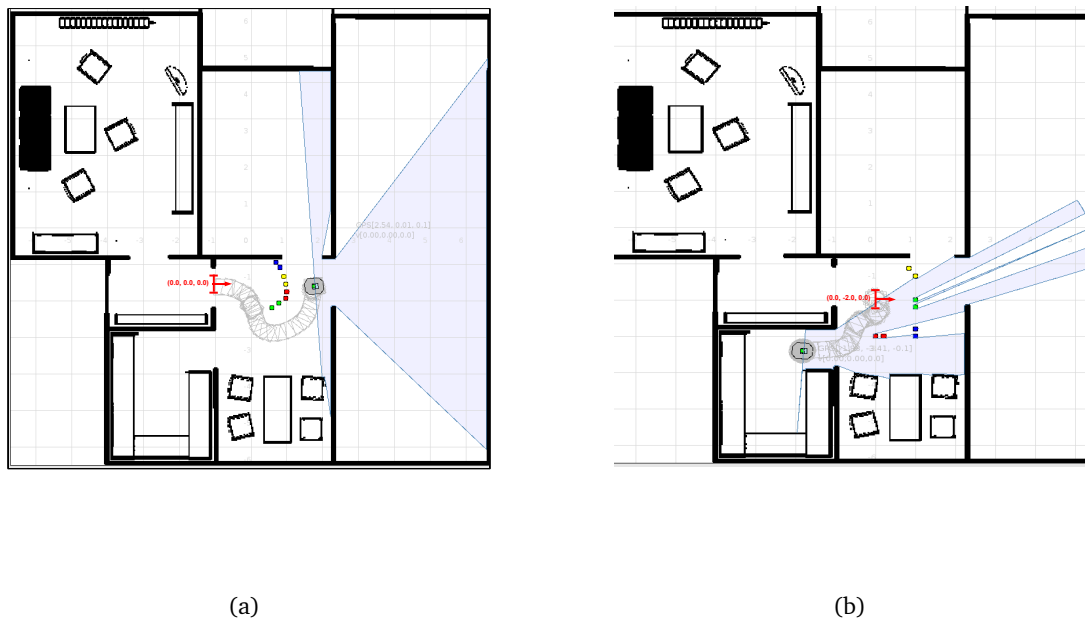


Figure 4.5.: Test of *ND* on the simulation platform *Stage*. The pairs of the simulated legs shown in Fig. 2.1(b) form now the U-shape and the cluttered obstacles in (a) and (b) respectively. The start position of the robot is depicted with red. The red arrow shows the start orientation of the robot. Moving along the gray trajectory the robot can successfully reach the goal within the predefined goal tolerance.

beginning of this chapter.

4.2.3. Empirical Study

The proposed approach has been studied on the robot platform **BIRON II** in a real environment resembling an apartment with real subjects. NF1 [92] was used to compute a global path based on the built occupancy grid map¹⁰. In order to make the robot avoid obstacles, *Nearness Diagram (ND)* navigation [107] was integrated in **BIRON II**. *ND* aims to make the robot get out of the troublesome situations, such as the obstacles forming U-shape (see Fig. 4.5(a)) or a very dense and cluttered scenario illustrated Fig. 4.5(b). To compute collision-free motions for a robot in a complex environment, *ND* divides the perception area into sectors with respect to the robot center and describes the robot, the goal location, the obstacles as well as the free space with a set of situations. Afterwards, actions, i.e. the translational and rotational velocities, corresponding to the assigned situation are specified with a *decision tree* according to the given criteria. By this means, difficulties lying in reactive navigation can be well solved, as the available situations in *ND* have overlapped most of the problematic cases in cluttered indoor environments.

As an environment representation and *locations* recorded in the built map are necessary for

¹⁰See a short introduction for NF1 in section 4.3.1.

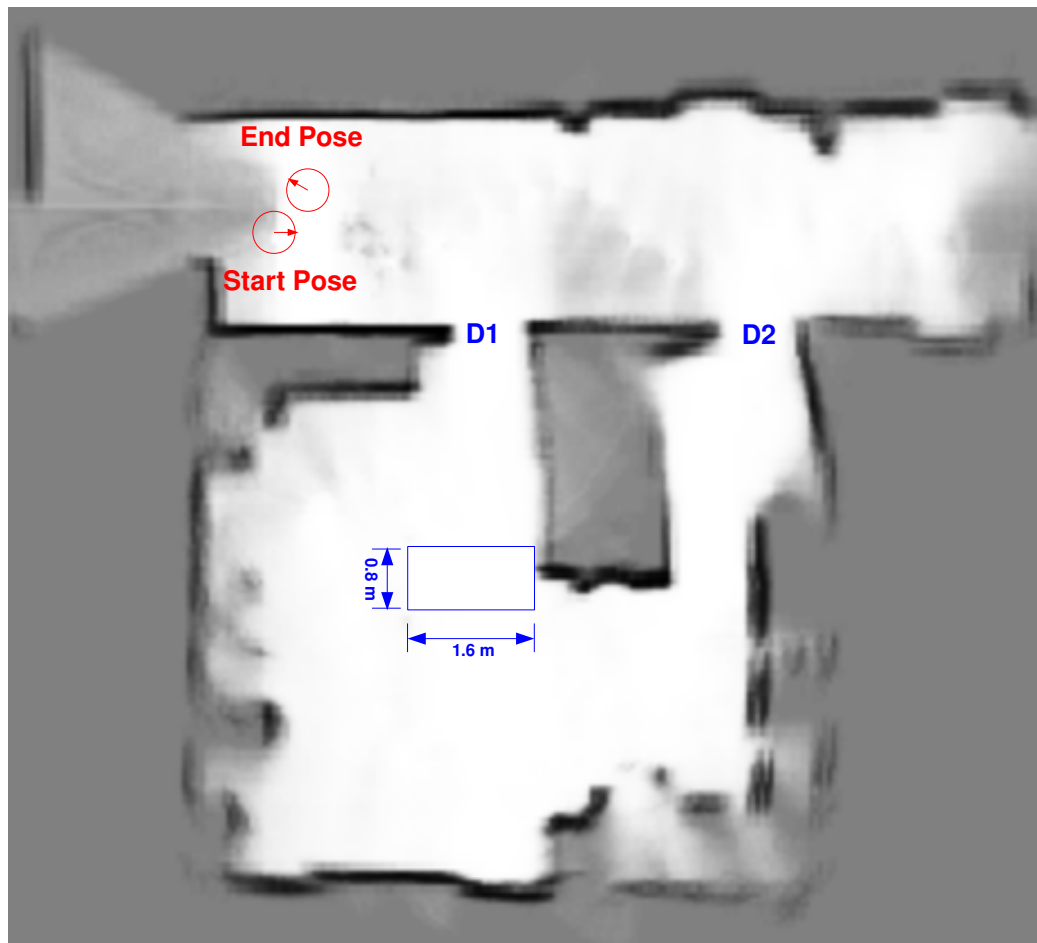


Figure 4.6.: Occupancy grid map of the environment. Two doors D1 and D2 open during the *Home-Tour*, a table whose four legs can be perceived by the robot and the start as well as end pose of the robot are depicted on the map.

our path planning approach, the test scenario was divided into two parts. At first, a person guided the robot in the initially unknown environment and showed different *locations* to the robot. Afterwards, the robot was required to navigate to the learnt *locations*. Besides, in order to validate the robustness of the re-planning process, a dynamic object was purposely laid on the way to the goal to obstruct the path planned for the robot. In the following the experiment procedure will be discussed in detail.

Fig. 4.6 illustrates the grid map of the environment built after the guided tour, and the start as well as end pose of the robot. The gray-scale indicates the probability whether the grid cells are occupied by obstacles perceived from laser sensor. As explained in section 3.1, white corresponds to free with high certainty, while black implies occupied with high certainty. Since the guide-person has to be regarded as an obstacle by *ND* during the *Home-Tour*, the strategy described in section 3.2.1 was adopted in the test scenario. By this means, the occupied area of the person was defined with the help of the component *Person Tracking and Attention* illustrated in Fig. 4.1 and removed from the laser plane. A table about $1.60\text{m} \times 0.80\text{m} \times 0.72\text{m}$ is depicted with a blue rectangle. As the laser range finder can perceive a plane in front of the robot at a certain height, only the four legs of the table

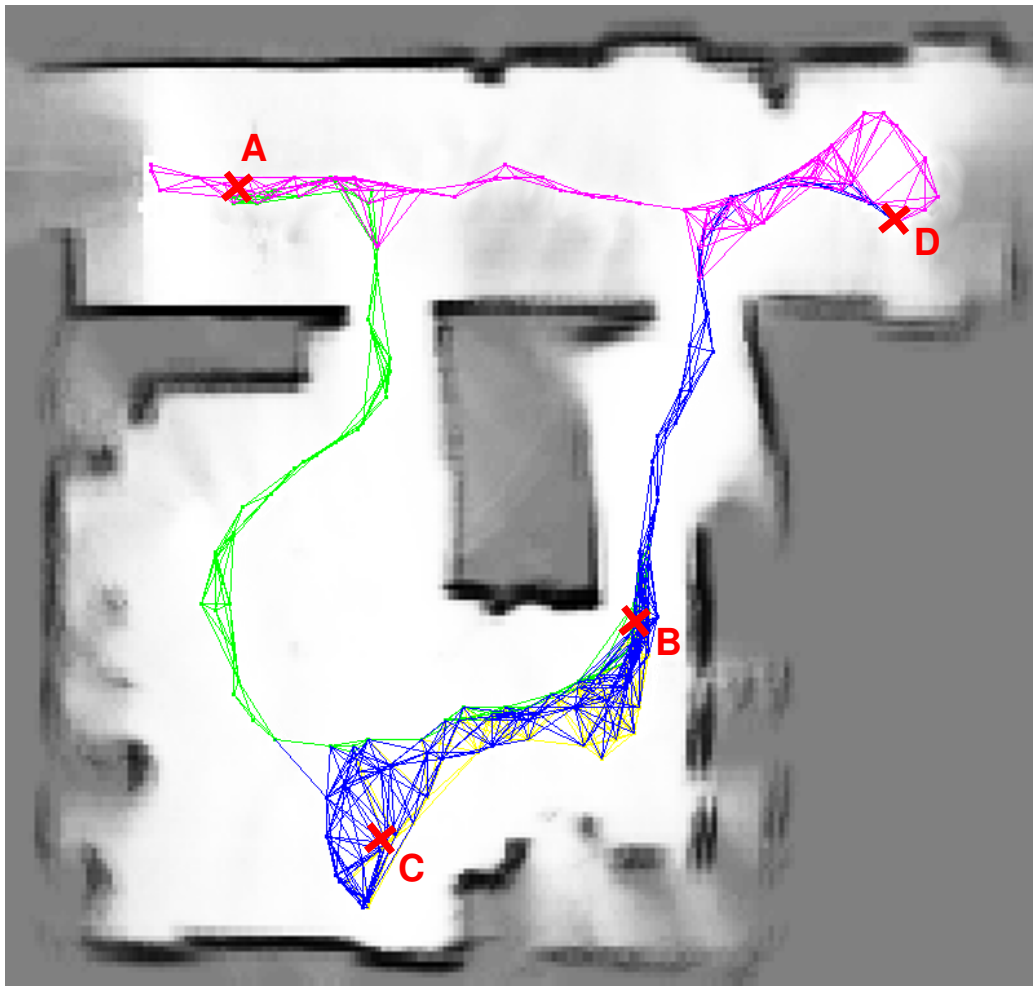


Figure 4.7.: Multiple graph layers created during the interactive *Home-Tour* scenario. Different colors illustrate different sub-trajectories of the guide-person from one *location* to another, as described in section 4.2.1.

can be perceived by the robot. The two doors D1 and D2 were opened, while the robot was following the guide-person. The effective elementary following behavior discussed in section 2.4.2, i.e. the direction-following process, was used in the empirical study¹¹.

The multiple graph layers illustrated with different colors were created (see Fig. 4.7), when different *locations* were shown to the robot. As discussed in section 4.2.1, each layer stores the person's positions from one *location* to another, if the *locations* exist. For example, the green sub-trajectory provides the movement information of the guide-person from *location* A to B, while the violet graph layer illustrates the set of person's positions from D to the end position of the guided tour because no subsequent *location* was shown to the robot. When the sub-trajectories of the person interweave each other, edges connecting nodes from different layers are allowed according to the distance restriction for edge creation mentioned in section 4.2.1. As illustrated in Fig. 4.7, nodes on the blue, yellow and green graph layer between *location* C and B are connected by edges with high cost. Thus, switches

¹¹In this section only the proposed path planning approach is evaluated. A complete evaluation of the system containing person-following behavior, map building process and path planning is presented in chapter 5.

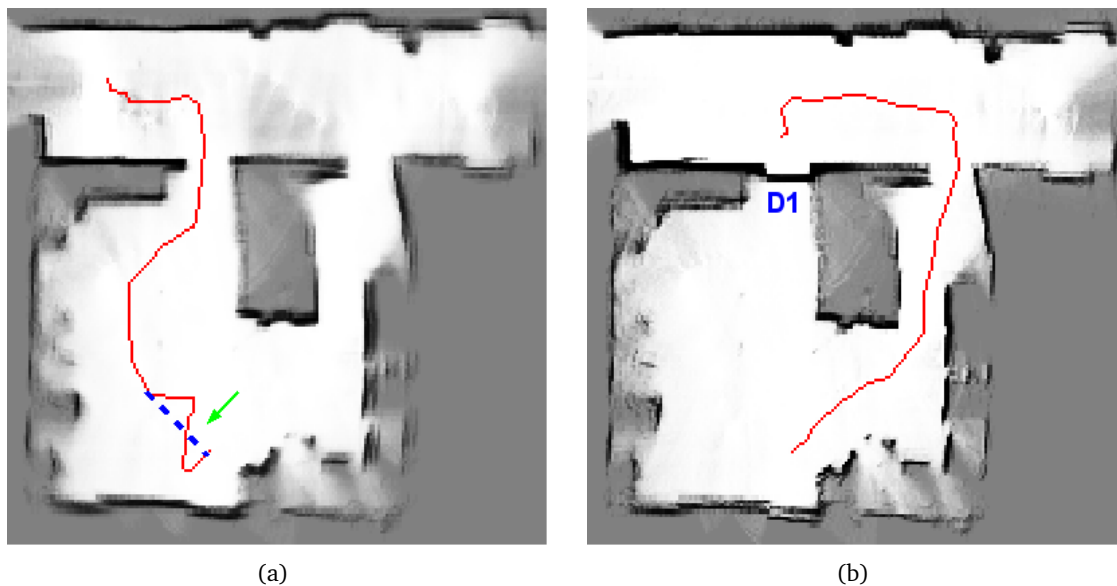


Figure 4.9.: Comparison of the planned and re-planned path. a) shows the planned path on the base of the built map during the *Home-Tour*. Tuning edge weights defined in equation 4.5 the blue dashed shortcut would be an alternative neglecting potential obstacles in the region shown by the green arrow. Once the new obstacle D1 had been added in the grid map and detected by the robot, re-planning was started. The resulting path is illustrated in b).

the robot's path was closed purposely, in order to disturb the current plan. As discussed in section 4.2.2, when the local approach *ND* was not successful, and the robot was aware of the new obstacle D1 obstructing the path in front, re-planning was started. The re-planned path computed from the graph and the closed door D1 added into the grid map are shown in Fig. 4.9(b). After the robot had successfully reached *location C* using the re-planned path, it was commanded to go back to A. Notice that the two built grid maps illustrated in Fig. 4.9(a) and Fig. 4.9(b) distinguish from each other in some details, e.g. the aisle between the two doors and the area inside the room. Compared to Fig. 4.9(a), the uncertain (gray) regions become relatively certain (white) in Fig. 4.9(b), as the robot entered or perceived these areas again. In addition, a shortcut marked with the blue dashed in Fig. 4.9(a) can be achieved by tuning edge weights defined in equation 4.5. However, potential hazards that can be avoided with the red path may exist in the area shown by the green arrow.

One of the advantages of the proposed path planning approach has been illustrated in Fig. 4.8(b), when obstacles are unperceivable to the robot. Besides, it is meaningful to compare the plan computed from NF1 and our approach in a fair situation. In other words, the obstacles on the way to the set goal can be perceived by the robot entirely. Therefore, the robot's position and orientation were marked in the experiment, when the robot went back to *location A*. The same goal B was sent to the robot following the path computed by our approach and NF1. Since the post-closed door D1 was mapped in the environment representation, the map-based plan obtained this information as well. The result is illustrated in Fig. 4.10. As NF1 tries to find a shortest path, the green trajectory leads the robot near to obstacles, e.g. the wall in the aisle and the door D2 marked in Fig. 4.6. In contrast, more free spaces were considered by the guide-person for safely planning, as the red trajectory shows, especially in the area of the aisle. In addition, the guide-person tried to keep dis-



Figure 4.10.: Comparison of the path computed from NF1 and our approach for the robot from A to B given in Fig. 4.7. The green path from NF1 is close to obstacles, while the red path reveals navigation strategies of the guide-person.

tance in the doorway of D2, where potential hazards, e.g. movable people or unexpected objects, might arise.

Notice that only the four thin legs of the table illustrated in Fig. 4.6 can be perceived by the robot. Adopting navigation strategies gained from the guide-person such an unperceivable obstacle should not impede the robot in performing navigation tasks. However, in a dynamic environment such a movable table may be placed on the path planned for the robot, which becomes a potential hazard. Therefore, two dimensional perceptual space of a fixed mounted laser is not sufficient to ensure safe navigation in an unconstrained environment. A motion generation taking 3D perception into account is meaningful and necessary. In the following section a solution will be discussed in detail.

4.3. Multisensor-based Motion Generation

As mentioned at the end of the last section, unconstrained human-inhabited environments still pose significant challenges for robots. Apart from effective interaction skills, personal service robots are expected to navigate safely and effectively in a human-shared environ-

ment. Several established motion planning and obstacle avoidance techniques are formulated in planar domains. They frequently rely exclusively on two-dimensional sensory input [148, 60]. However, an indoor environment, e.g. a home or an office, contains many objects which cannot be appropriately sensed and represented in this way. Thus, these approaches may lead to difficulties in collision avoidance [78]. For example, tables are large obstacles for robots (except miniature appliances like floor sweeping devices), but a laser scanner mounted parallel and close to the ground can only perceive the four legs (see Fig. 4.6). In this case, the robot is possibly to be misled through the midst of dinner. One solution to this issue is the use of 3D sensors, e.g. stereo vision, sweeping laser scanners, or time-of-flight (ToF) cameras. Consequently, there are ongoing research projects investigating full 3D mapping, localization, motion planning, and obstacle avoidance [139, 140, 149]. However, for an overwhelming majority of robots, indoor environments are still essentially two-dimensional. Existing 2D techniques are quite sufficient, when the three-dimensional world information is appropriately prepared according to their requirement. The feasibility of using a SwissRanger for navigation tasks has been presented in [159, 39], where the authors have focused on the ToF technology. A comparison of robot navigation systems either using a standard 2D laser scan or a SwissRanger camera has been discussed in [159]. Furthermore, the suitability of 3D ToF technology for robot navigation has been verified in [39]. The SwissRanger's limited field of view can be overcome by using several SwissRanger cameras. However, the number of Swissrangers parallel used is currently limited to three, as each camera has to work at its own modulation frequency. Otherwise, distance measurements obtained from different cameras may be impact on each other. Nowadays using the Microsoft Kinect mentioned at the beginning of this chapter 3D point cloud based SLAM [76] and obstacle avoidance [158] is becoming more common. By this means, the problem of 3D information capturing is able to be addressed more appropriate than using SwissRanger cameras.

In contrast to the work mentioned above, a solution fusing the 2D laser data and the complete 3D ToF data into a common representation is proposed. Fig. 4.11 gives an overview of sensor fusion. Each data stream has different characteristics (frame-rate, noise, and precision). By means of the sensor fusion, advantages of the two sensors, i.e. a wide field of view and accurate measurements of laser scans and 3D information of the environment achieved by a SwissRanger camera, are expected to benefit the robot. Considering the laser and SwissRanger data fused on the laser plane, a 2D motion generator is thoroughly competent to perform a navigation task. Besides, an approach of ToF camera calibration, i.e. determination of the position and orientation of the camera in the laser frame, has been proposed, so that the corresponding 3D points can be represented in the laser and camera coordinate system.

4.3.1. Motion Generation

In general, path planning and obstacle avoidance deal with the problem “*how to move towards a goal?*” and the problem “*how to move safely?*” respectively [132]. The motion generation discussed in this section is based on the approach controlling ten autonomous tour-guide robots during a five-months exhibition in 2002 [118, 117]¹² The map-based approach NF1 provides a global plan for the robot on a two-dimensional grid [92], while a flexible path representation based on the elastic bands formulation [122] is combined with the Dynamic Window Approach (DWA) [48]. By this means, the global path planned by NF1 can be smoothed by the elastic bands. Moreover, the robot is able to react unexpected

¹²See <http://libsunflower.sourceforge.net/> for details.

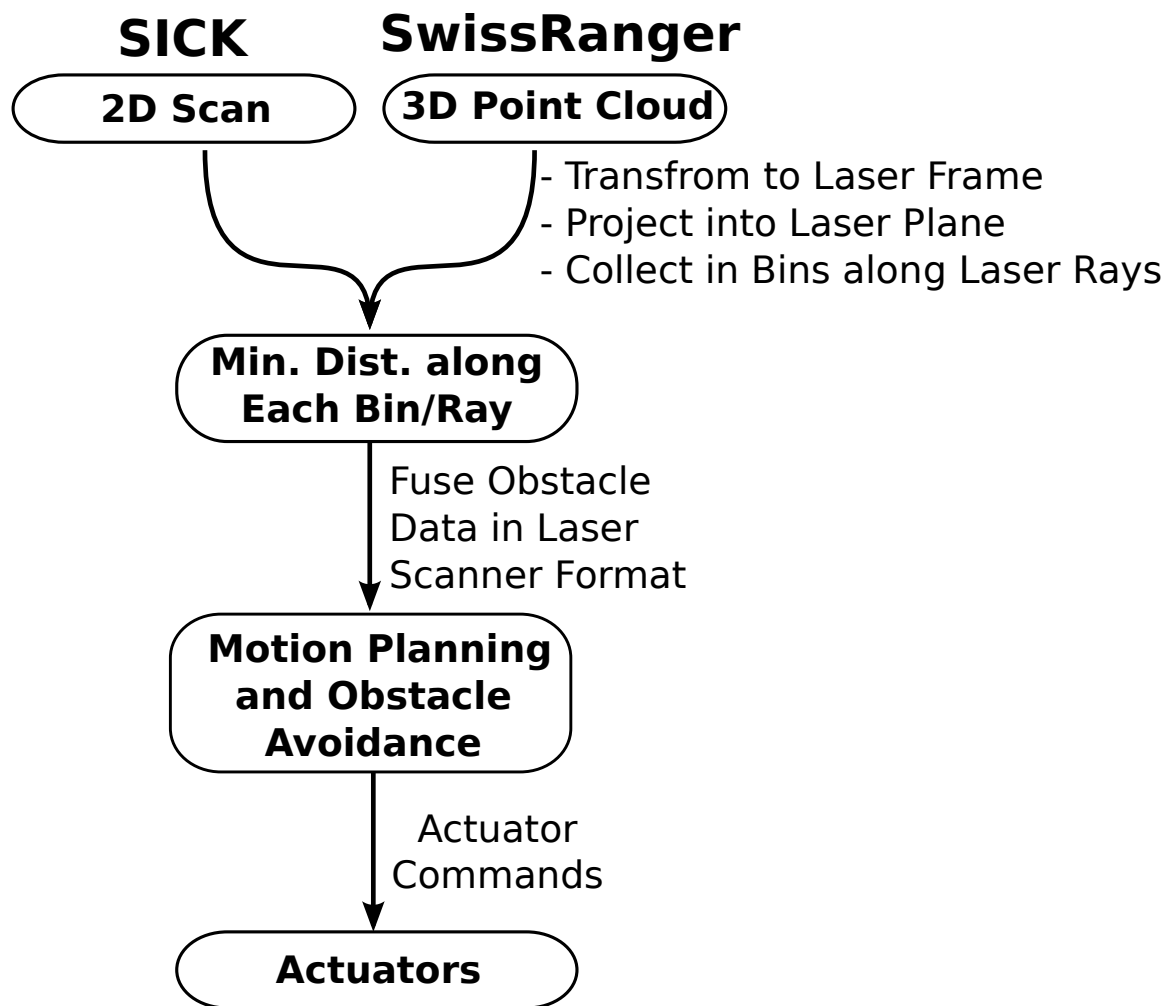


Figure 4.11.: Overview of sensor fusion. An available navigation component is reused by means of the obstacle information fusing 2D and 3D data.

obstacles according to the actuator commands computed from DWA. Fig. 4.12 summarizes how planning and controlling of the available navigation component are interwoven to achieve smooth goal-directed behavior, while avoiding collisions. In the following the main components of the motion generation will be briefly introduced¹³. Furthermore, the basic idea of the integration with 3D data will be discussed.

Global Path Planning The global plan created by NF1 is a list of global grid cells that lead from the robot's position to the goal. As can be seen in Fig. 4.13, such plans are constrained to lie along integer multiples of 45° . These corners are smoothed out using the elastic strip, as explained below. In the motion generation system, the NF1 grid is repositioned at each re-planning step, so that it forms a corridor from robot to goal. By this means, grid effects can be minimized. Moreover, the computation time of the wavefront propagation can be bounded. The elastic band, which translates and adapts the plan for execution, is initialized by placing circles (termed *bubbles*) in the centers of the grid cells. This band starts at the robot's position and proceeds towards the goal, while skipping cells that are already covered by the preceding bubble.

¹³See the work of Philippsen [117] for a detailed description of the motion generation system.

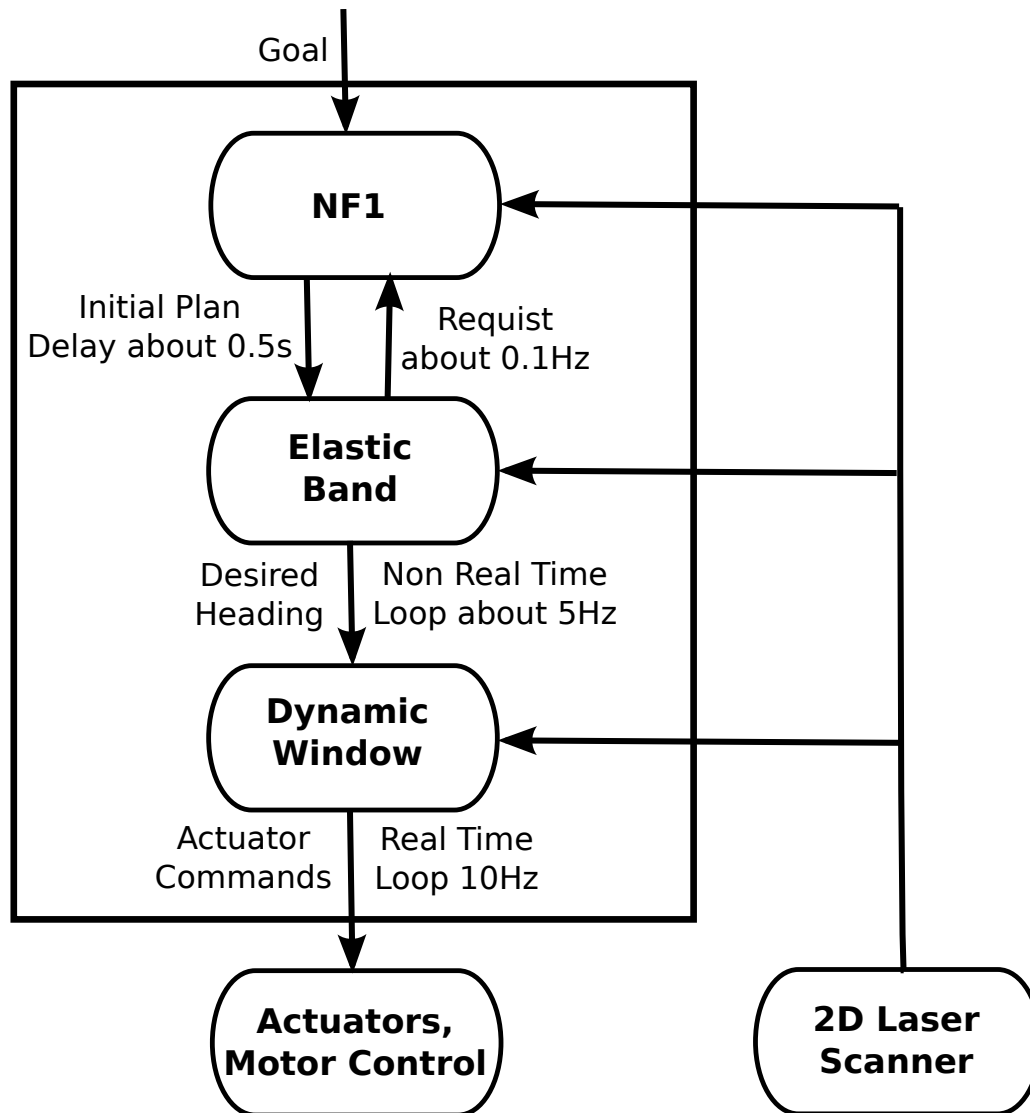


Figure 4.12.: Motion planning and control loops of various temporal and spatial scopes are integrated into the motion generation system. A global goal position is sent to the motion planner. The output is in the form of actuator commands sent to the motor control level.

Plan Adaptation Plan adaptation reduces the amount of re-planning by changing the path according to changes in the environment and the motion of the robot. In order to maintain a feasible homotopic path, the adapted plan might deviate from the plan. The initial plan inherits some of the grid effects from the NF1 path, notably the non-smooth directional changes which are integer multiples of 45° . However, the first iteration of plan adaptation already significantly smooths the elastic band. During each iteration, bubbles are pushed away from obstacles (in order to increase free space around the path) and attracted to their neighbors (to remove slack and eliminate sharp corners). The first bubble always follows the robot's position, and the last bubble is fixed on the goal. New bubbles are inserted along the band whenever the overlap between two successive bubbles is too small to let the robot pass, and spurious bubbles are pruned from the band whenever their predecessor and successor already overlap sufficiently. Fig. 4.14 illustrates the interaction between NF1 and

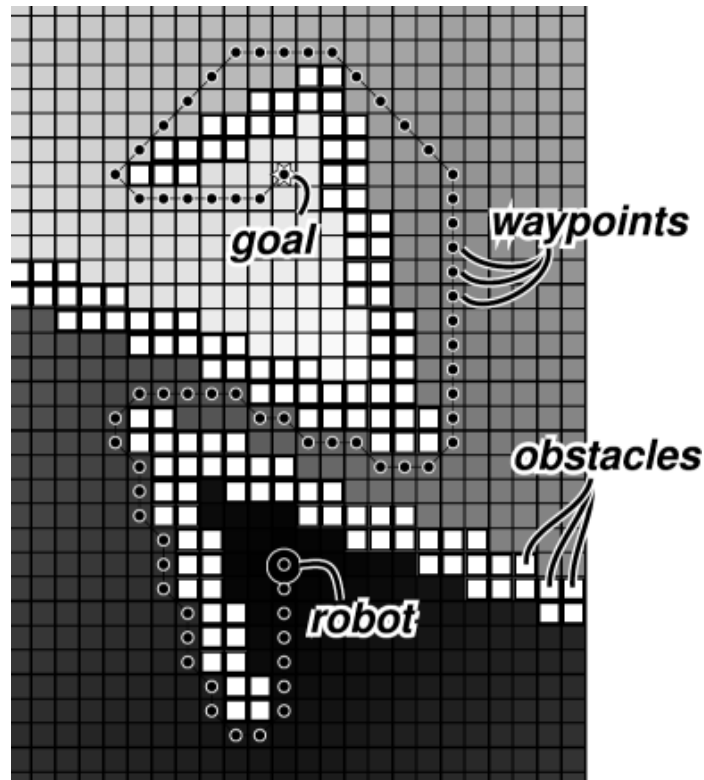


Figure 4.13.: NF1 example.

elastic band.

Obstacle Avoidance The plan adaptations performed by the elastic band are able to anticipate general direction changes in order to circumvent most collision. However, it cannot guarantee obstacle avoidance because it runs at a slower (non-realtime) rate than the motor controllers and obstacle detectors. Moreover, it does not take into account robot kinematics and dynamics, which leads to tracking errors that could force the robot to leave the safety zone of the bubbles. Additionally, even when no global path is available (during re-planning or temporary blockage) the robot must guarantee safe motion. The process responsible for obstacle avoidance takes into account vehicle kinematics and dynamics by determining which motion commands are available at each instant (bound by acceleration and velocity constraints), and predicting how the robot would move for each candidate control within the reachable velocity region. Motions that would result in collisions are discarded, while an optimization process is used to choose among the remaining safe velocity commands.

In order to allow sensor fusion and implement an upper bound on computation time, the local space around the robot is discretized into an obstacle grid. Furthermore, for each grid cell which motions would lead to a collision with an object inside the cell, similar to [126], are pre-computed. As the velocity space is discretized as well, the total effort for the DWA computations has a constant known upper bound, which allows us to guarantee real-time execution. Besides, the pre-computed collisions eliminate the need for subsequent complex operations. The required data structures are summarized in Fig. 4.15.

Integration with 3D Data The motion generation system depicted in Fig. 4.12 uses planar geometry for representing and reasoning about obstacles and movements. The en-

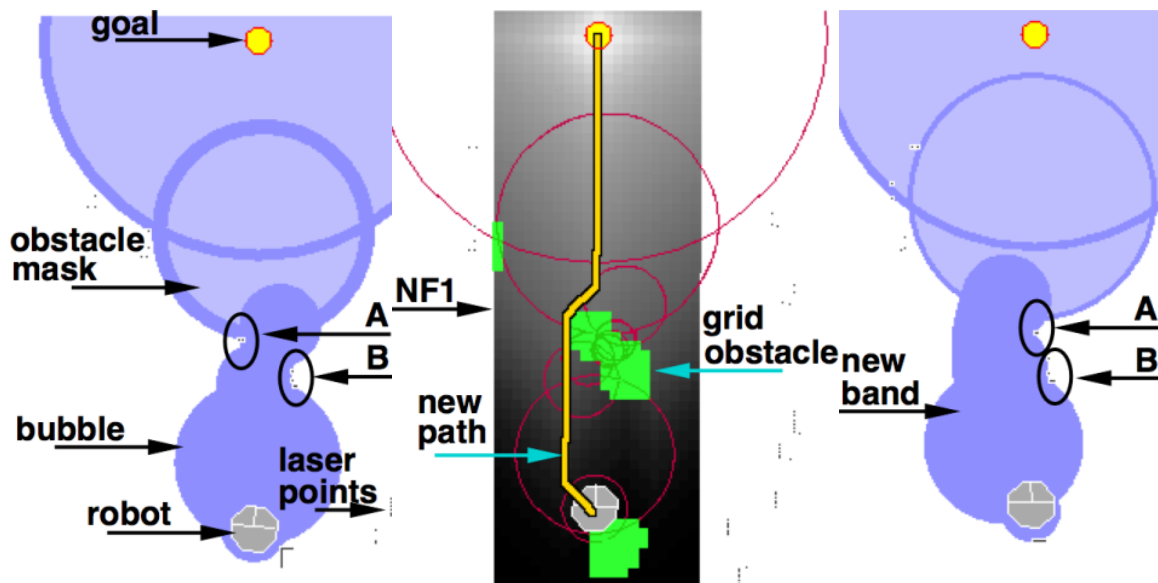


Figure 4.14.: Example re-planning sequence involving NF1 and elastic band: two moving obstacles (labeled A and B) squeeze the bubbles until the robot would not fit through their overlapping region anymore. This event triggers an invocation of the NF1 planner, which produces a new path beset with grid effects. This new path is transformed into a new elastic band, which now leads around the region that became blocked.

environment is considered to be projected onto the ground plane on which the robot moves. The main sensory input comes from a laser scanner with beams that are parallel to the ground. This results in fast robust computations based on reliable high-resolution laser readings, but neglects the true three-dimensional nature of the environment.

There are several possible ways to integrate 3D perception into the presented motion generation system. Given that the shape of BIRON (see Fig. 1.2(a) and 1.2(b)) can be approximated by an upright cylinder without gravely mis-representing its navigational constraints, there is little to be gained from extending the motion generation components to three dimensions. Instead of incurring the added computational costs and the effort of changing a smoothly running system, projecting all available sensory data into the plane of motion is chosen.

Assuming that the navigational component remains two-dimensional, we are now faced with the question of how to fuse the projected 3D data into the planar components. There are basically two possibilities. Either insert point-cloud data into the grids of NF1 and DWA's lookup table, and adapt the elastic band to use general point-clouds. This requires changing the data structures that are fed into those components and possibly extending the rasterization algorithms. Or transform the fused data into a virtual laser scanner which allows us to reuse all existing data structures and concentrate all change to the entry point of sensory data.

For the sake of minimally invasive changes to a working system we have chosen the latter approach, bearing in mind that robots which, unlike BIRON, deviate significantly from a cylindrical shape would require more elaborate techniques. The resulting data fusion and reduction can be summed up as follows:

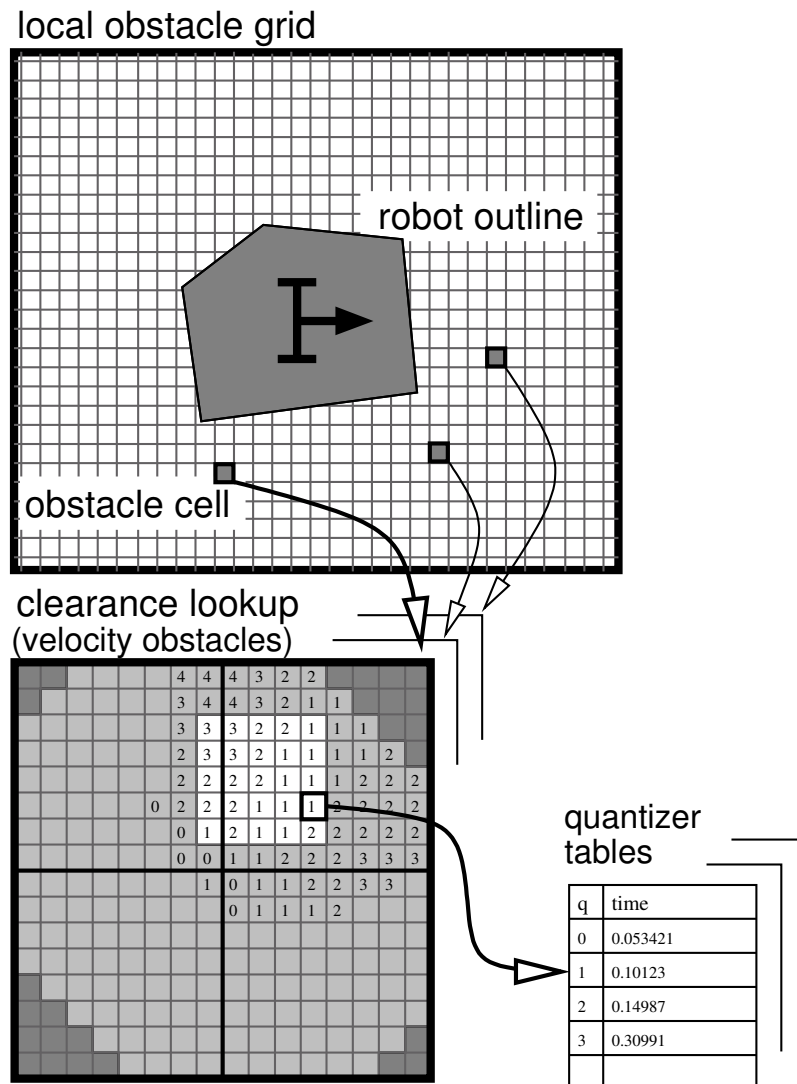


Figure 4.15.: Look-up operations in the dynamic window. Obstacles are stored in a two-dimensional grid, where each cell stores associated velocity obstacle information in a collision table. To reduce the amount of memory taken up by this four-dimensional lookup-structure, the collision table is compressed using a variant of the Lloyd-Max algorithm.

- ▷ capture and preprocess 3D point clouds
- ▷ project 3D data points into laser plane
- ▷ cluster projected points into bins that correspond to the laser beams
- ▷ in each bin, find the point which is closest to the robot
- ▷ use the distance to the closest point as (virtual or real) laser reading

4.3.2. Data Fusion

As already discussed, a computationally expensive 3D motion control is not needed for a service robot working in a two-dimensional environment. Hence, instead of attempting to implement full 3D motion control, a fusion of 3D ToF data and laser data in the domain

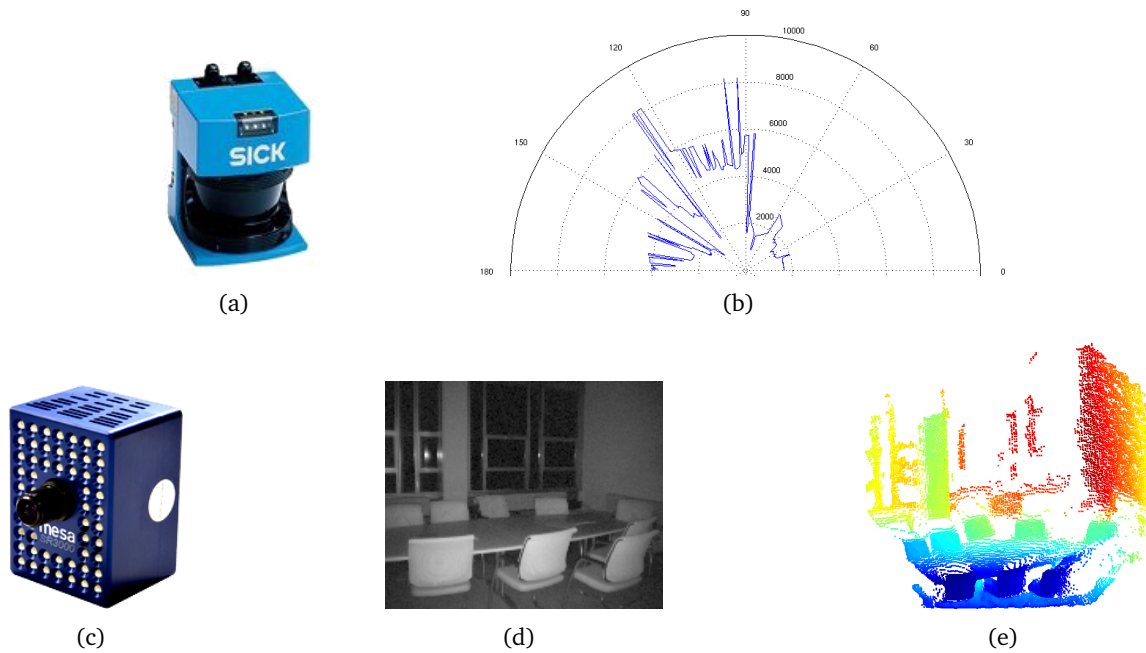


Figure 4.16.: (a) SICK LMS, (b) example laser scan, (c) SwissRanger SR3000, (d) example amplitude image, and (e) example 3D point cloud.

of laser scans has been proposed to create the so-called *virtual laser*. Fig. 4.16(a) and Fig. 4.16(b) show the 2D SICK laser range finder equipped on the robot platform. The laser sensor provides a 2D data stream at a frequency of about 10Hz. The laser scans the scene within a 180° arch in intervals of 0.5° measuring distances with an accuracy of approximate $10mm$. Objects in the environment can be perceived by the laser scanner at a height about $30cm$. In order to extend the perception area limited on the 2D laser plane, a 3D Time-of-Flight (ToF) camera, in our case the SwissRanger SR3000 (Fig. 4.16(c)) developed by Swiss Center for Electronics and Microtechnology (CSEM) [159], is mounted on top of the robot below the pan-tilt camera (see Fig. 4.17(a)). This camera independent of texture and lighting conditions provides 3D data up to 30 fps. 176×144 CMOS active pixel sensors measure distances between the optical center of the camera and the real 3D world points via the time-of-flight of a near-infrared signal. Additionally, each sensor delivers an amplitude value indicating the amount of light reflected by a world point. The amplitude value can be used to rate the quality of the provided distance measurement. Fig. 4.16(d) presents an example of an amplitude image and Fig. 4.16(e) the corresponding 3D point clouds.

In order to fuse the ToF data with the 2D laser data, firstly, the 3D camera data are transformed into the coordinate system of the laser. Afterwards, the transformed camera data are projected on the xz -plane, i.e. the laser plane. Re-sampling this projection in bins of 0.5° in a 180° arch and choosing the minimum value per bin produce a kind of ToF laser stream. This data stream encodes information of objects like tables that is not presented in the regular laser stream. Finally, the original laser stream and the ToF laser stream are merged into a *virtual laser stream*. Camera calibration aligning the camera coordinate system to the laser coordinate system and the method generating the *virtual laser* will be discussed in the following.

Calibration In order to realize the fusion of 3D ToF data and laser data, firstly, it is needed to determine the position and orientation of the ToF camera in the laser coordinate

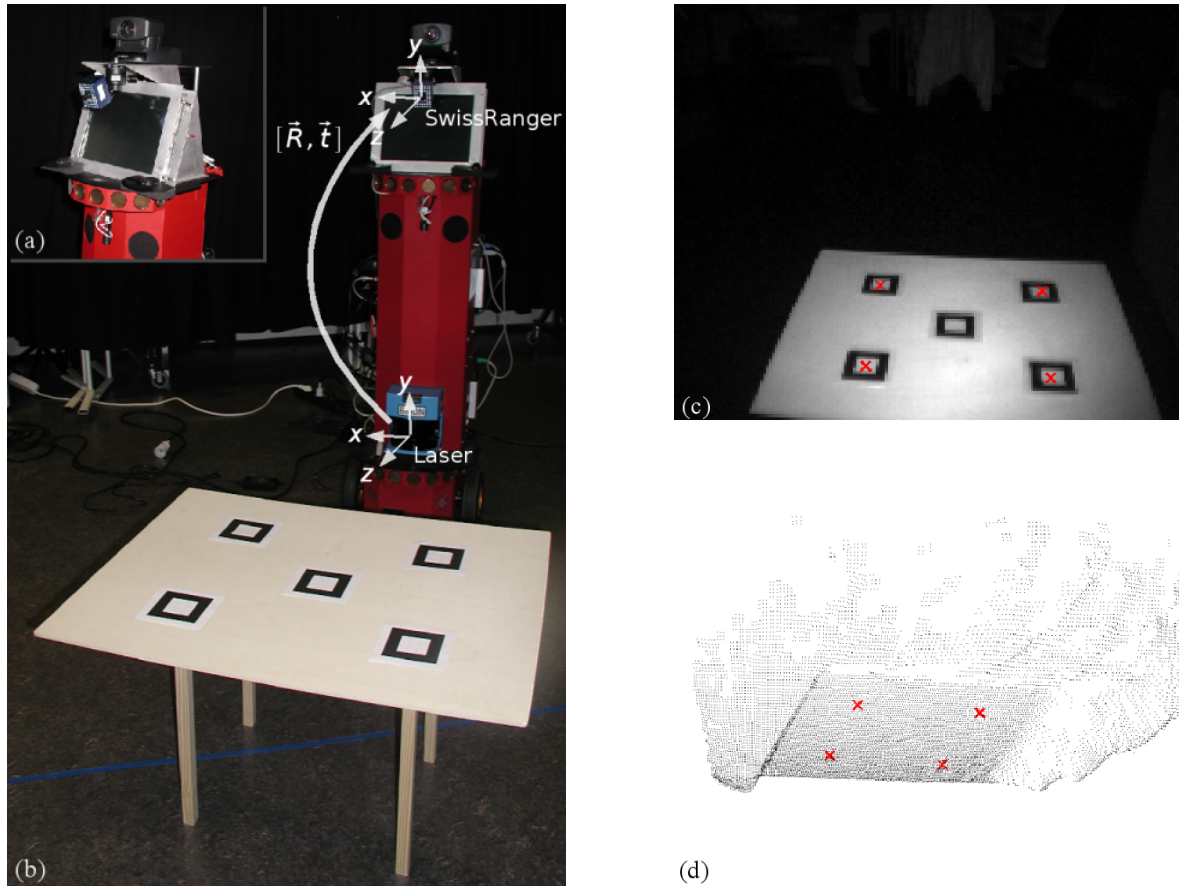


Figure 4.17.: (a) head of **BIRON I** with the ToF camera mounted below the pan-tilt camera, (b) the calibration setup, (c) the ToF view of the calibration pattern as amplitude image, and (d) as 3D point cloud. The red crosses mark the hand chosen points in the pattern. The central marker on the pattern is skipped.

system set as the reference coordinate system. As illustrated in Fig. 4.17(b), the transformation of the coordinate system from the laser plane to the ToF camera is given by a rotation and a translation $[\vec{R}, \vec{t}]$. To determine the parameters of this transformation an extrinsic calibration is required. As the two sensors are inherently different, none of the established calibration methods is directly applicable. For robustness explicit calibration is used. Hence, a special calibration pattern with dedicated markers that can be both detected in the laser scan and the amplitude image of the ToF sensor was built. The pattern looks like a table with legs mounted right below some visual markers attached on top of it. The laser can detect the four legs and return the respective xz -coordinates of the visual markers. As the height of the calibration pattern and the laser plane are known, the 3D coordinates $\vec{p}_{1,L}, \vec{p}_{2,L}, \vec{p}_{3,L}, \vec{p}_{4,L} \in \mathbb{R}^3$ of the markers can be computed in the reference coordinate system¹⁴, i.e. the laser plane.

The second step of calibration is to determine the coordinates $\vec{p}_{1,S}, \vec{p}_{2,S}, \vec{p}_{3,S}, \vec{p}_{4,S} \in \mathbb{R}^3$ of the visual markers in the SwissRanger coordinate system. The markers are visible in the amplitude image, which can either be detected automatically or selected by hand (red cross markers in Fig. 4.17(c)). In the ToF data each 2D pixel corresponds to a 3D point in the camera frame, as illustrated in Fig. 4.17(d). Thus, the 3D

¹⁴It is assumed that the laser plane is parallel to the floor.

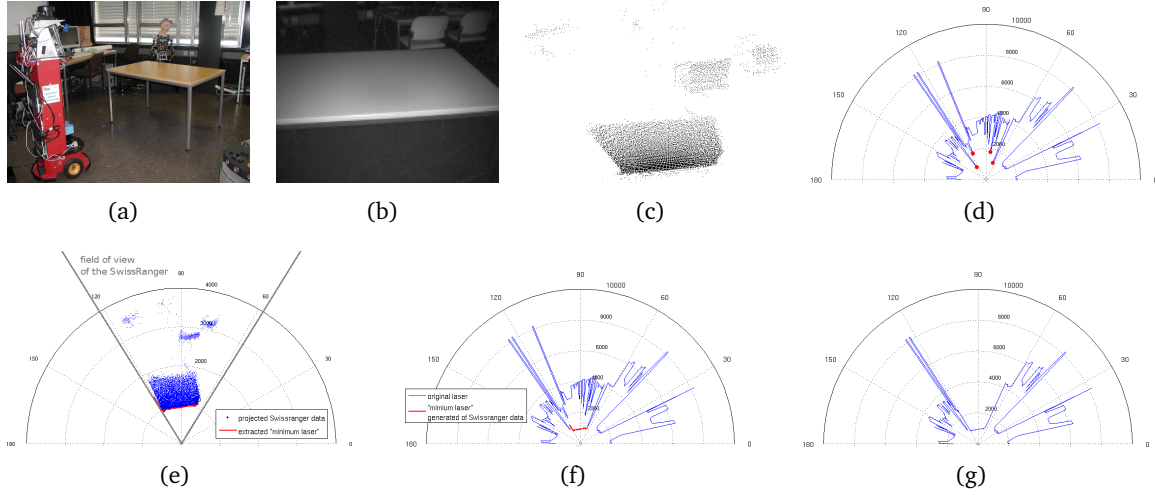


Figure 4.18.: (a) Scenario: table in front of BIRON; (b), (c) view of the SwissRanger camera on the table ((b) amplitude image and (c) 3D point cloud); (d) a laser scan of the scenario where only the four legs marked with red points are visible; (e) the SwissRanger 3D points of (c) projected in the laser plane and the minimum value per bin marked in red; (f) the original laser and the minimum distances extracted from SwissRanger data; (g) computed virtual laser.

coordinates of the markers can be extracted.

In the ideal case it holds that

$$\forall i \in \{1, 2, 3, 4\} : \vec{p}_{i,L} = \vec{R} \cdot \vec{p}_{i,S} + \vec{t}, \quad (4.8)$$

but on noisy data the following mean square objective function has to be minimized [127]:

$$f_{\min}(\vec{R}, \vec{t}) = \frac{1}{4} \sum_{i=1}^4 \|\vec{p}_{i,L} - \vec{R} \cdot \vec{p}_{i,S} - \vec{t}\|^2. \quad (4.9)$$

The optimization problem for $[\vec{R}, \vec{t}]$ can be solved using Singular Value Decomposition (SVD) [98, 155] with \vec{m}_L and \vec{m}_S being the centroids of the point sets $\{\vec{p}_{i,L}\}$ and $\{\vec{p}_{i,S}\}$, respectively:

$$\vec{K} = \sum_{i=1}^4 (\vec{p}_{i,L} - \vec{m}_L)(\vec{p}_{i,S} - \vec{m}_S)^T \xrightarrow{\text{SVD}} \vec{V} \vec{D} \vec{U}^T \quad (4.10)$$

$$\vec{R} = \vec{V} \vec{U}^T \quad (4.11)$$

$$\vec{t} = \vec{m}_L - \vec{R} \vec{m}_S. \quad (4.12)$$

Three non-colinear 3D points are sufficient to uniquely compute the transformation. In other words, three rotation angles and three translation parameters can be determined, while ensuring \vec{R} to be a rotation but not a reflection. Four markers of the pattern are used to compute the transformation. Experimental results that will be discussed in section 4.3.3 verify the robustness of the proposed extrinsic calibration.

Integration as a virtual laser Using conventional laser-based navigation in a situation shown in Fig. 4.18(a), a collision to the table is inevitable when a goal behind the table is sent to the robot. This navigation error is due to the erroneous information

obtained from the laser. As Fig. 4.18(d) illustrates, free space in front of the robot is perceived by the laser scanner except for the table's legs marked with red points.

To overcome this problem the ToF 3D points (see Fig. 4.18(b) and 4.18(c)) are exploited to achieve more information about the table. First, several pre-processing steps need to be performed on the 3D ToF point clouds. By means of the pre-processing, the 3D points are smoothed, and the bad measurements are rejected if their amplitude values do not meet the criteria given in [142]. Afterwards, the points are transformed into the laser coordinate system according to equation 4.11 and 4.12, i.e. $\vec{p}'_{k,S} = \vec{R} \cdot \vec{p}_{k,S} + \vec{t}$, where $\vec{p}_{k,S}$ is the k -th point in the camera coordinate system, while $\vec{p}'_{k,S}$ is the point transformed in the laser plane. As the transformed point cloud contains measurements of the floor and objects higher than the robot top, these points can be discarded according to their y -coordinates.

The remaining points $\{\vec{p}'_{l,S}(x, y, z)\}$ are projected into the laser plane, transformed to polar coordinates $\{\vec{p}'_{l,S}(r, \theta)\}$, where $r = \sqrt{x^2 + z^2}$ and $\theta = \cos^{-1}(x/r)$. By means of θ , each point can be assigned to one of the 361 bins defined as follows: $[0, \frac{1}{720}\pi)$, $[\frac{1}{720}\pi, \frac{3}{720}\pi)$, \dots , $[\frac{717}{720}\pi, \frac{719}{720}\pi)$, $[\frac{719}{720}\pi, \pi]$. Considering that more than one point can be accumulated in one bin, for each bin the point with the smallest radius r is regarded as the minimum distance $MinDisSR_i$ extracted from SwissRanger data for bin i . Finally, these minimum distances and the original laser points are fused to a *virtual laser*, where the smaller distance of $MinDisSR_i$ and $DisLaser_i$ from the original laser is chosen as the *virtual laser* point for bin i . Fig. 4.18(e) to Fig. 4.18(g) illustrate the steps described above. Thus, the *virtual laser* is able to output data stream with the frequency of the original laser.

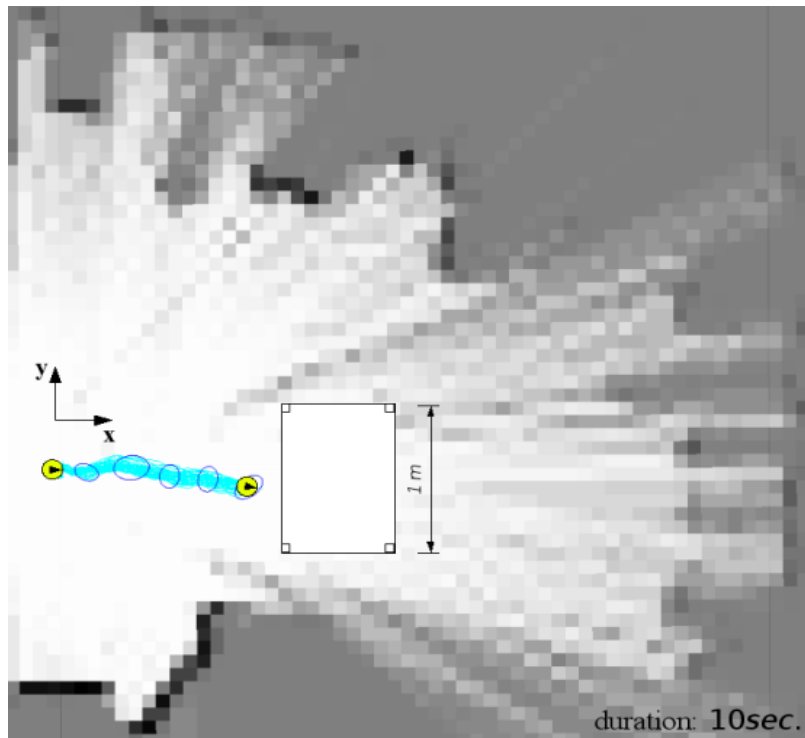
4.3.3. Experimental Results

To evaluate the effectiveness of the proposed approach, the motion generator (see section 4.3.1) integrated in **BIRON I** was used to perform the same navigation task in the same environment. The original laser and *virtual laser* was adopted respectively in five experimental runs. Hence, ten trails were performed. In the ten experimental runs the robot was started in the same pose $(0.0m, 0.0m, 0.0rad)$. A table about $1.00m \times 0.70m \times 0.73m$ was placed in front of the robot as a main obstacle on the way to the set goal. The shortest distance from the start position of the robot to the table's edge is about $1.63m$. The goal $(2.5m, 0.0m, 0.0rad)$ behind the table was set for the ten runs with the same *goal_radius*¹⁵. The RBPF-SLAM approach introduced in section 3.1 was started as well during the ten trail runs to record the map of the environment and the robot's path estimated in the built map.

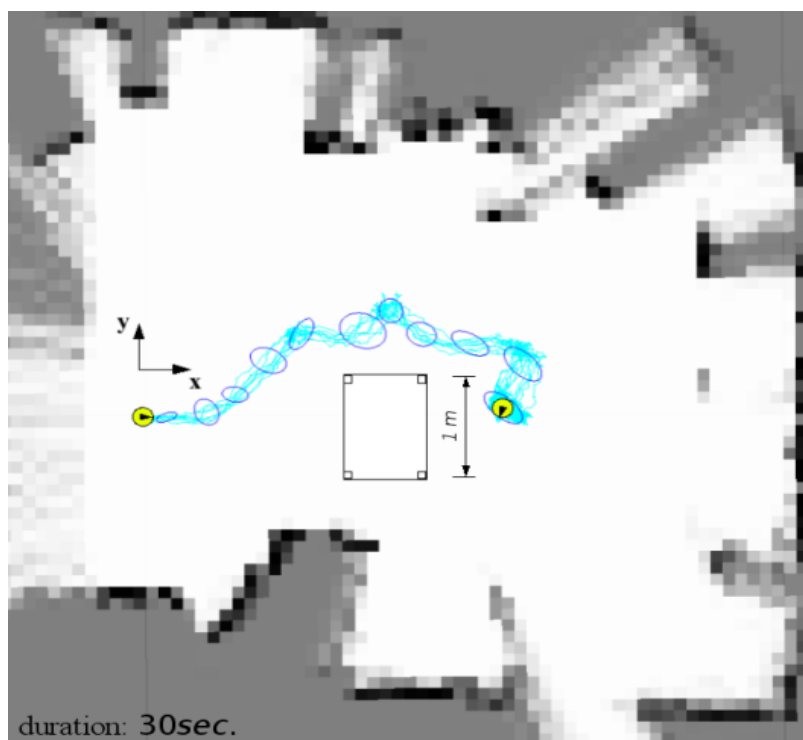
Fig. 4.19 shows the occupancy grid map built in the two of the ten trail runs, in which the original laser and *virtual laser* was used respectively. The robot is represented with a yellow circle¹⁶. The center of the circle is the robot's position, and a black arrow points out the robot orientation. The start and end pose of the robot is depicted on the built map. Each green curve illustrates a robot's path estimated by a particle according to the map. The uncertainty of the estimated robot's poses are displayed with blue error ellipses. The semimajor axis and the semiminor axis of the error ellipses are derived from the covariance

¹⁵*goal_radius* is a distance tolerance to determine whether a goal is reached. It was set $0.3m$ for the experiment. See the definition in section 2.2.2

¹⁶Note that the circle is only for visualization. The radius of the circle is not relative to the real size of the robot.



(a)



(b)

Figure 4.19.: Comparing the motion generation with real and virtual laser in the same environment. (a) the robot makes a collision with the table, as the real laser provides only the four thin legs of the table and a relative large free space in the middle of the both front legs. (b) shows the trajectory of the robot with a successful obstacle avoidance, when the virtual laser fused the pure laser and 3D information from SwissRanger was used.

matrices of the robot's poses. These ellipses are centered on the best estimated path¹⁷. The table placed in front of the robot is outlined with a rectangle. The four legs of the table, which are the only parts of the table perceivable in the laser plane, are marked with four small squares. Besides, the x -axis parallel to the start orientation of the robot and the y -axis of the world coordinate system are drawn on the map to point out the two directions of the world frame. The actual origin of the world coordinate system is the start position of the robot.

When the robot used the original laser scans, it moved straightforward and tried to go through the table due to the large free space perceived between the two front legs, just as Fig. 4.19(a) illustrates. Although the robot took the goal $(2.5m, 0.0m, 0.0rad)$, it was manually stopped in time in this trail run, so that the robot would not really crash into the table. In contrast, using the *virtual laser* the robot saw the table. Fig. 4.19(b) shows the robot moving around the table and reaching the goal successfully.

Furthermore, it can be observed that the robot's path is not smooth but zig-zag. This is because the robot tried several times to correct its direction towards the goal when moving around the table (see Fig. 4.19(b)). The reason hiding behind this phenomenon is due to the limited field of view of the SwissRanger. Since the goal behind the table was sent to the robot, the motion generator drove the robot in the direction towards this goal as far as possible, when available free space was detected. According to the current setup of the SwissRanger (see Fig. 4.17(a)), the 3D data projected on the laser plane approximately covers an angle range between 70° and 120° , as Fig. 4.18(e) shows. Therefore, outside this angle range the original laser data are used. In other words, the information of the table obtained from the SwissRanger is probably ignored by the *virtual laser*. For the *virtual laser* used in the experiment, movement commands driving the robot towards the goal need to be computed by the motion generator. The zig-zag robot's path leads to various durations in the experiment though, the robot can reach the goal set for all trails. Considering the problem of false path adaption, the SwissRanger should be controllable. Thus, the view direction of the camera can be changed towards the robot's goal. Obstacle information in this field of view is provided instead of sending a real rotation command to the robot. Using the data fused by the laser sensor and the movable SwissRanger, unnecessary zig-zag plans should be avoided. Incorporating SwissRanger information in the mapping process should be a more ideal solution. By this means, the perceived obstacle can be still *remembered* by the robot, although it is out of the field of view of SwissRanger. However, errors in SwissRanger need to be taken into account, so that false obstacles can be reduced as far as possible.

4.4. Summary and Discussion

Navigating to a given goal is one of the most important capabilities especially for a service robot working in a human-shared environment. Natural human-inhabited environments still pose significant challenges for robots despite the impressive progress that has been achieved in the field of path planning and obstacle avoidance. In view of the fact that these challenges mostly arise from the limited perceptual abilities of a robot, as well as the environment that is highly dynamic being populated by human beings, analysis and exploitation of human spatial knowledge should be taken into account. In the context of *Human-Robot Joint Spatial Exploration*, it is possible for the robot to interactively learn from

¹⁷The best estimated path is associated with the most likely particle.

the guide-person, and incorporate the gathered information into the built spatial representation. Thus, an integrated solution dealing with these challenges has been achieved.

In this chapter, at first I presented a path planning approach on the basis of a graph with multiple layers, representing the movement trajectory of the guide-person in the interactive *Home-Tour*. Considering navigation capabilities implicitly contained in human pathways, following human-like paths the robot is able to avoid unperceivable obstacles and potential hazards. The trajectory of the guide-person actually leads to a collision free path for the robot under the assumption of static environment. A robust path planning approach is therefore achieved, by means of which the robot follows a conventional path expected by people. In addition, the robot's path would be predictable, as the robot follows the path chosen by the guide-person. For the graph-based approach, efficiency, effectiveness and person's trajectory have to be considered synthetically in order to compute a robot's path. Therefore, edge weights combining multiple factors need to be defined. Distance to the goal, timestamp along person's trajectory and switch of graph layers have been taken into account. Since the ultimate goal for path planning is to find a path from any location to any other location in the free space of the environment, the map-based approach NF1 has been integrated into the system as a supplement. Thus, the limitation of the graph-based approach, i.e. it is almost impossible to get a trajectory of the guide-person covering any free space of the environment, can be overcome.

Afterwards, I discussed the problem of dynamic objects in path planning. Especially when unexpected dynamic obstacles obstruct the path, a re-planning process will be triggered, trying to compute another plan. This integrated approach enables the robot to move safely, while still effectively navigating in the environment. The results achieved in an empirical study are discussed as well. Different graph layers are created in terms of *locations* shown by the guide-person. The robot can reach the visited *locations* by means of the path computed on the basis of the person's trajectory. Unperceivable obstacle, i.e. the table shown in Fig. 4.8(b), was avoided by the robot. Furthermore, free spaces and potential hazards were considered by the guide-person, as Fig. 4.10 illustrates. Thus, exploiting pathways of the guide-person containing human navigation strategies is beneficial to the robot. Besides, the re-planning capability of the robot verifies the robustness of the proposed approach performed in a real world scenario.

At last, a data fusion technique extending a purely two-dimensional motion generator to a 3D capable navigation system is presented. As locations of unperceivable obstacles can be changed, it is dangerous for a robot performing a navigation task. The extension is realized by means of the *virtual laser* fusing original laser scans and 3D camera data into a common two-dimensional frame. Thus, the available motion generator presented in section 4.3.1 is minimally invasive. Moreover, obstacles out of the original laser plane can be avoided by the robot. The main advantages of the approach lie in the robustness and computational modesty. Fusing the two sensors working asynchronously, the *virtual laser* is able to produce a data stream at a high rate, which enables reliable robot motion in dynamic cluttered environments.

As the proposed path planning approach is based on the analysis of the observed human trajectories, the implication containing in the person's trajectories should be taken into account. Trajectories interwoven with or closed to each other are generally caused by repeated movements of the person through the same areas. Merging these trajectories can refine the graph and make the path planning and re-planning process more efficient. Besides, according to the information obtained from the environment, an assessment of the observed person's trajectories might be meaningful to determine the intention of the human

behavior. E.g. unnecessary detours of the guide-person occurring in an interactive *Home-Tour* scenario could be detected by the assessment. Furthermore, this hypothesis is expected to be clarified in the interactive scenario in a manner of inquiry. Considering that the robot shall perform navigation tasks by means of the built map, it needs to localize itself on the known environment representation. Hence, a component of global localization (*GLOC*) has been integrated in the robot system as well, which will be introduced in section 5.1.2.

For the proposed data fusion approach, at first, without laser scanner using 3D ToF camera alone might be an alternative. However, the limited field of view of the camera would not be suitable in a cluttered home environment in spite of mounting a movable camera. When multiple cameras are equipped, signals from the individual cameras would probably interfere each other. The accuracy of measurements is therefore affected. In addition, keeping the person tracking system (see section 2.2.1) integrated in the robot system in mind, laser scanner is necessary for the robot to complete the *Home-Tour*.

Next, instead of the SwissRanger camera, an additional 2D laser sensor can be equipped on top of the robot to overlook the surroundings. In contrast to the horizontal field of view (47.5°) and the vertical field of view (39.6°) of a SwissRanger SR3000 [39], the perception area of laser scans is still limited to a two-dimensional plane.

The field of application of the presented novel calibration of the 2D laser scanner and the 3D ToF camera reaches far beyond the presented one. For instance, it allows to compute 3D positions for all laser ranges, which is useful for precise object positioning to compensate the noise in the depth perception of the ToF camera. An enhancement to overcome the limitations of a cylindrical robot model would be a more elaborate environment representation used in the motion generation subsystem. Navigational constraints due to the robot's shape and characteristics of the mounted sensors need to be both taken into account. However, the proposed approach enables the robot to successfully pass doorways and avoid crashing into different types of furniture in the *Home-Tour* scenario.

In order to make the *virtual laser* more robust against noise, sensor readings in each bin should be weighted by their distance to the neighbor point. Minimum distance readings with neighbors far away should not be considered. Moreover, only meaningful structures extracted in the 3D ToF data [141] should be projected in the laser plane.

Besides, the camera data projected on the laser plane might be utilized for a pure mapping process, while the robot's pose can be estimated by the RBPF-SLAM approach using the original laser. By this means, objects outside the laser plane can be represented on the camera map. However, noise of the 3D camera and the relatively narrow horizontal field of view (47.5°) compared to the laser (180°) might negatively influence the online mapping process.

Note that the path planning and obstacle avoidance component integrated in **BIRON I** and introduced in section 4.3.1 is different from the variant¹⁸ discussed in section 4.2 and integrated in **BIRON II**. Due to the relatively long time delay of the control loop in robot actuators, the implemented obstacle avoidance strategy combining DWA and elastic band is substituted by *Nearness Diagram (ND)* on the new robot platform **BIRON II**. Hence, the motion generator presented in section 4.3.1 is not evaluated in the current system shown in section 5.1.2. Nevertheless, the *virtual laser* fusing the 2D laser data and the 3D ToF data is transplantable to **BIRON II**, as two-dimensional laser scans are needed for *ND* as well. Furthermore, real world experiments on platform **BIRON I** have shown the effectiveness of the proposed *virtual laser*.

¹⁸See component *Path Planning and Re-planning*, as well as *Obstacle Avoidance (ND)* shown in Fig. 4.1.

5. Evaluation

According to the basic idea of *Human-Robot Joint Spatial Exploration* (HRJSE) introduced in section 1.2 the interactive *Home-Tour* is assumed as an initial scenario of information gathering. The *Home-Tour* scenario provides an opportunity for a service robot to explore an unknown environment and interactively learn from a human user. Thus, metric information of the surroundings obtained from perceived sensor data and knowledge gained from human user can be integrated in the spatial representation. The framework of interactive acquisition of an environment representation can be built on the basis of HRJSE and the available *Home-Tour* scenario. The central modules that construct this framework, i.e. the person-following behavior, the mapping process and the path planning approach, have been discussed in chapter 2, chapter 3 and chapter 4 respectively. The proposed approaches have been tested in a simulated environment or on a mobile robot platform. The test results discussed in the corresponding chapters have verified the functionality of the modules. However, whether the integrated framework as a whole is appropriate and convincing for solving the main problem, i.e. acquisition of environment representation in an interactive scenario, has not been discussed in detail yet. The proposed and implemented approaches must undergo serious evaluation, in order to validate their feasibility in practice. Therefore, the system consisting of the three modules needs to be run as a whole and evaluated on the robot platform in a real world scenario.

The main objectives of the evaluation corresponding to the framework of interactive acquisition of an environment representation can be condensed as follows:

- ▷ How performs the person-following behavior presented in chapter 2 according to spatial context in real world scenarios?
- ▷ In section 3.2.1 a strategy has been proposed to detect and remove the false information arising from the perceived person's legs in a mapping process. Evaluation results of the mapping process performed in a simulated environment have been discussed in section 3.2.2. To characterize the performance under real constraints, how are the environment representations built with and without the strategy in comparison to the results obtained from the simulated environment?
- ▷ How performs the proposed path planning approach according to the map built in a real world *Home-Tour* scenario.

The three points are associated with the individual modules working together in the whole system. Besides, behaviors of the subjects guiding the robot in real world scenarios will be shortly discussed. Of course, the goal of evaluation is not only to confirm the appropriateness of the proposed methodologies, but also to identify their drawbacks and limits.

As the focus of this chapter is the assessments of the three modules working together for interactive acquisition of an environment representation, firstly, experimental design for the user study will be introduced. Next, I will present and analyze the evaluation results of the individual function modules from a system perspective. Afterwards, I turn to a brief discussion of the observation from a subject's perspective. Summary and insights arising from the evaluation will be presented at the end of this chapter.

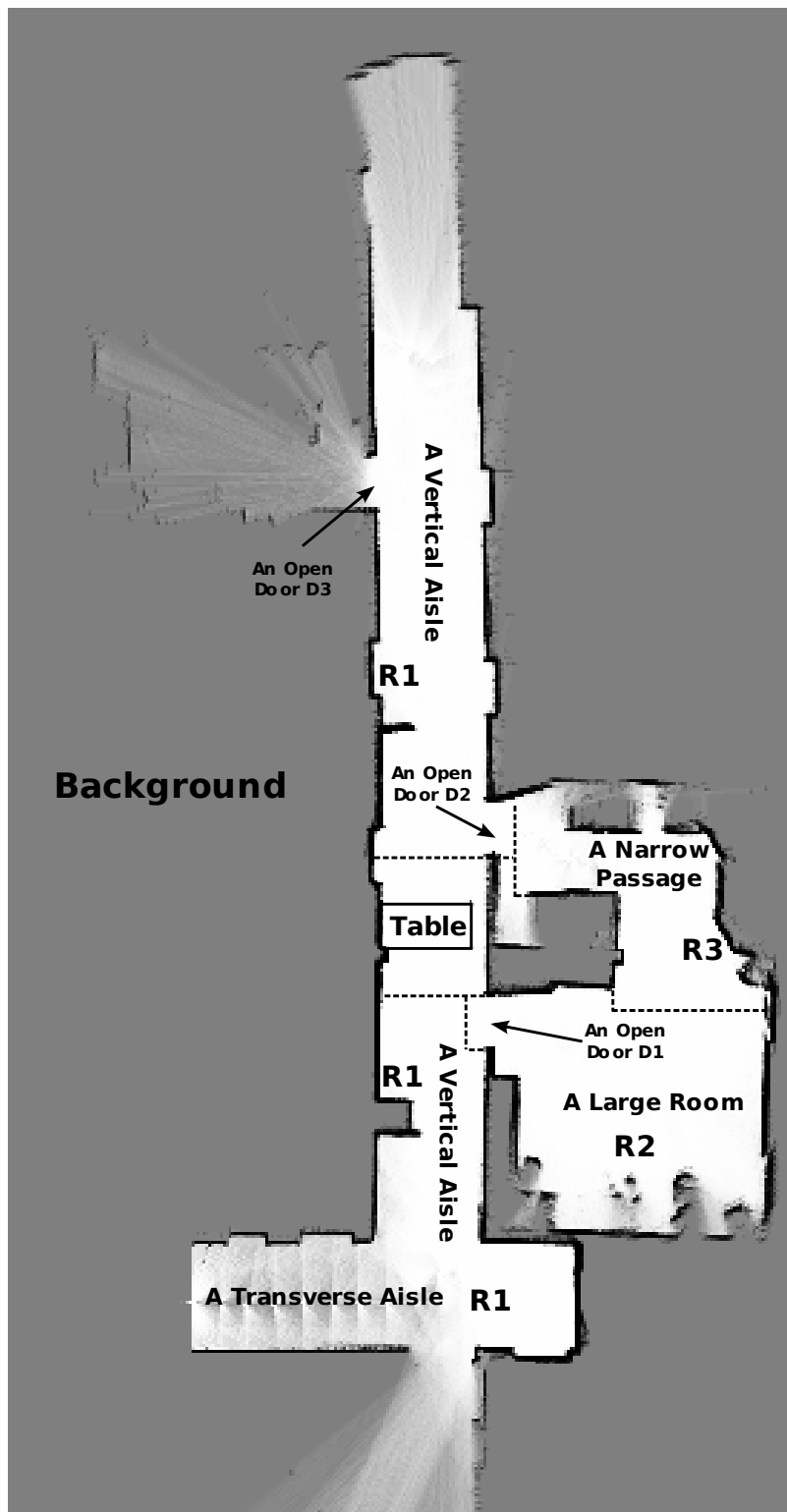


Figure 5.1.: The ground truth map built for the user study. This map was built by driving the robot platform **BIRON II** with a joystick. The gray background indicates the areas that cannot be perceived by the robot. Each grid cell of these areas is set to the maximum uncertainty 0.5. Free space with high certainty corresponds to the white areas, while the black areas imply occupied with high certainty. A transverse and a vertical aisles, three open doors, a large room and a narrow passage are shown on the environment map. Three regions are marked with R1, R2 and R3 on this map as well.

Introduction of your task

The mobile robot whose behavior will be evaluated called BIRON. Your task is as follows: please take BIRON a tour around the environment depicted below and show the 4 rooms to it.

Besides, BIRON can understand the following commands:

- **"Follow me"** will make BIRON come after you.
- **"This is the hallway/bedroom/living room/kitchen"** will make BIRON remember the name of the room where it currently is.
- With **"Stop"** you can say to BIRON anytime, when it should stop.

Enjoy yourself!

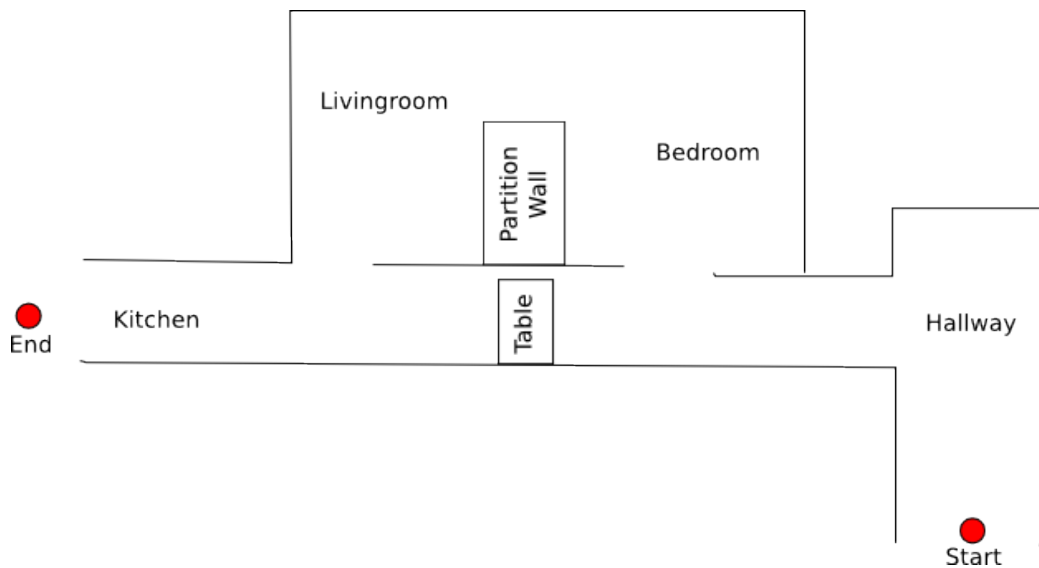


Figure 5.2.: Introduction of the task assigned to the subjects. A brief description about what a subject should do and a sketch map of the environment are given. The start and the end position for each trial run is depicted with two red circles respectively. Four predefined *locations* (hallway, bedroom, living room and kitchen) are marked on this sketch as well. A table is laid between *location* kitchen and *location* hallway.

5.1. Experimental Design

In order to evaluate the acquisition of spatial representation in an interactive scenario, the system consisting of the three individual modules needs to be run as a whole in the *Home-Tour*. The procedure and the general information of the user study is introduced at first. Next, an overview of the implemented system will be given in section 5.1.2. All relevant software components will be introduced in detail.

5.1.1. Procedure

The scenario of the user study was an interactive *Home-Tour* through a portion of a floor in an office building. The integrated system (see section 5.1.2) has been evaluated on the new mobile robot platform **BIRON II** introduced in section 1.1 in this environment with real subjects. On the basis of the RBPF-SLAM approach introduced in section 3.1 a grid map of the environment can be built, as illustrated in Fig. 5.1. Three open doors, a transverse and a vertical aisles, a large room and a narrow passage are marked on the map. As will be discussed in section 5.2.1, according to the existing spatial context and the robot's trajectory recorded in an experimental run the environment is divided into three regions R1, R2 and R3. The areas that the robot cannot perceive are regarded as the background of the environment. Grid cells in these areas obtain the maximum uncertainty 0.5. Instead of building the map in a guided tour, the robot was controlled with a joystick and moved slowly through the environment. By this means the environment can be perceived in detail. This map is regarded as a ground truth map. In comparison with the occupancy grid map built in the same environment during a *Home-Tour* and shown in Fig. 5.9, the black areas implying obstacles and the white areas corresponding to free spaces are more clearly recorded in the ground truth map. The purpose of creating the ground truth map will be explained in the next section.

Different experimental runs were performed in this environment. The user study was conducted by eight subjects who were found randomly on the campus of the university Bielefeld. All participants in the user study used the robot platform for the first time and seven of them had no background of mobile robots. Before an experiment was started, the corresponding subject had received a brief introduction of the task shown in Fig. 5.2. The main task assigned to the subjects was to guide the robot through the given environment. Three types of commands were available to control the robot and send *locations* predefined in the environment to the robot. A mini-dialog system (see section 5.1.2) was necessary for both the robot and the guide-person to perform the interactive *Home-Tour*. A sketch map of the environment, in which the start and the end position for each trial run, as well as the four predefined *locations* are marked, is illustrated in Fig. 5.2 as well. Notice that a table is laid between *location* kitchen and *location* hallway on purpose, in order to fully evaluate the proposed path planning approach. As all subjects were not familiar with the mobile robot and the sketched environment, we answered their questions, when they had any doubt about the task and the new surroundings. Time consumption discussed later on was therefore influenced. Of course, any detail of the system, the parameters of the robot movement, e.g. maximum speed, as well as the goals of the study presented at the beginning of this chapter were not revealed to the subjects until after their experiments were finished. How the guided tour would be performed, such as the route of the tour, was entirely free to the subjects.

After the subjects had made sure of their task, and the mobile robot platform had been

made ready at its start position, i.e. all system components had been successfully started, the robot needed to localize itself in the ground truth map illustrated in Fig. 5.1. Thus, the trajectory of the guide-person and the robot were recorded in a common frame. The person-following behavior performed in the eight experiments can be compared with each other and evaluated (see section 5.2.1). The component of global localization (*GLOC*) will be introduced in the next section. In addition, the subjects were allowed to take the brief introduction illustrated in Fig. 5.2 during the experiment. Subsequently, no prompts from our side were given to the subjects during the rest of the experiment, unless dangerous situations for the guide-person would occur or the robot would collide with any objects in the environment. However, no such situations occurred in all eight experimental runs. Of course, if the person was not sure which command should be chosen, a kindly reminder was provided.

Each subject conducted the *Home-Tour* once in the user study. The eight subjects were all able to guide the robot from the start position to the end position and solve the task described in the brief introduction successfully, although bystanders may pass in the environment now and then. The main distinction of the performance was the behavior of the guide-person. The subjects walked in a forward-, a backward- or a mixing-manner in the user study. It took some time to answer the questions posed by the subjects though, time consumption including instruction and system preparation was quite consistent and ranged between 15 minutes and 20 minutes (mean 16.38m and standard deviation 1.49m). After the trial runs five of the subjects who walked in a backward-manner in the experiment were asked about the reason (see section 5.3).

5.1.2. System Architecture

The user study was conducted by eight guide-persons on the robot platform **BIRON II**, as shown in Fig. 1.2(b). In the following the implemented system and the components additionally integrated for the user study will be discussed.

The relevant system components and connections between them are illustrated in Fig. 5.3. The modules constructing the core of the implemented system are marked with yellow. The adaptive context-aware person-following behavior, the mapping process and the path planning approach have been discussed in detail in chapter 2, chapter 3 and chapter 4 respectively. The *Person Tracking and Attention System* (PTA) sub-system introduced in section 2.2.1 provides person positions with respect to the robot coordinate system. These person positions output by PTA are necessary for the three yellow modules and the separate *GLOC*.

For the evaluation of the integrated system, *GLOC* played an important role in the assessment of person-following and path planning. As mentioned in the last section, in order to compare the results of the person-following behavior performed in the eight trial runs, a common frame, in which the trajectory of the robot and the person are recorded, is necessary. Since all the experiments were carried out in the same environment, a ground truth map (see Fig. 5.1) was built by controlling the robot to move step by step in the environment. This ground truth map is regarded as the common frame for different experiments. Thus, the robot can localize itself in this ground truth map using the separate *GLOC*. Moreover, all robot's poses and person's positions from different trial runs were defined and may be compared with each other in this frame. The separate *GLOC* was used *only* for the eight trial runs in the user study. Besides, as discussed in section 4.4 reusing the map built in a *Home-Tour* is practical for the robot when the environment is assumed static or the change

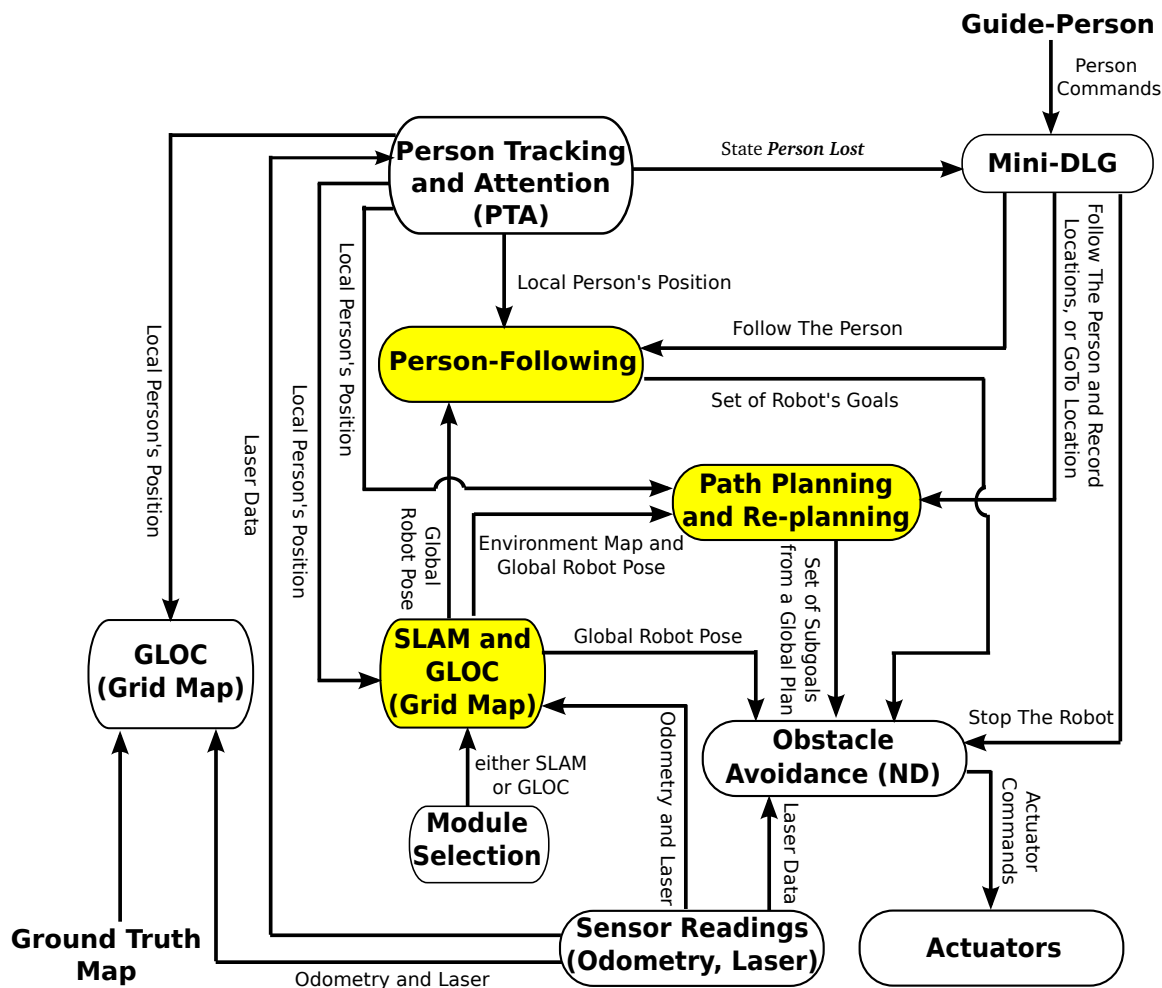


Figure 5.3.: Overview of the implemented system. Directions of the data flow are shown with arrows. The outputs of the components are commented nearby the corresponding arrows. The three modules constituting the core of the system are marked with yellow. Module *Mini-DLG* is responsible to the communication between the robot and the guide-person, which sends signals to trigger the corresponding modules. A separate module *GLOC* used for the evaluation of the person-following behavior was started *only* for the eight trial runs. *SLAM* integrated in module *SLAM and GLOC* was chosen to build the environment map in the user study, while *GLOC* of this module was used *only* for the evaluation of the proposed path planning approach on *Stage*.

of the environment is trivial. Therefore, *GLOC* has been integrated in the mapping process as well, i.e. module *SLAM and GLOC* illustrated in Fig. 5.3. This *GLOC* will be performed when an available map of the environment built in a trial run of the user study can be found by *Module Selection*. Otherwise, the RBPF-SLAM approach introduced in section 3.1 will be executed. Note that *Module Selection* just checks whether a map is available for *SLAM and GLOC*. Whether this map is corresponding to the real environment will not be checked. Especially for the user study the robot was guided by the subjects and chose *SLAM* in module *SLAM and GLOC* to build the respective map of the environment for each trial run. Since a map of the environment was available, the simulation platform *Stage* can be started. Moreover, *GLOC* in module *SLAM and GLOC* was chosen and the robot can local-

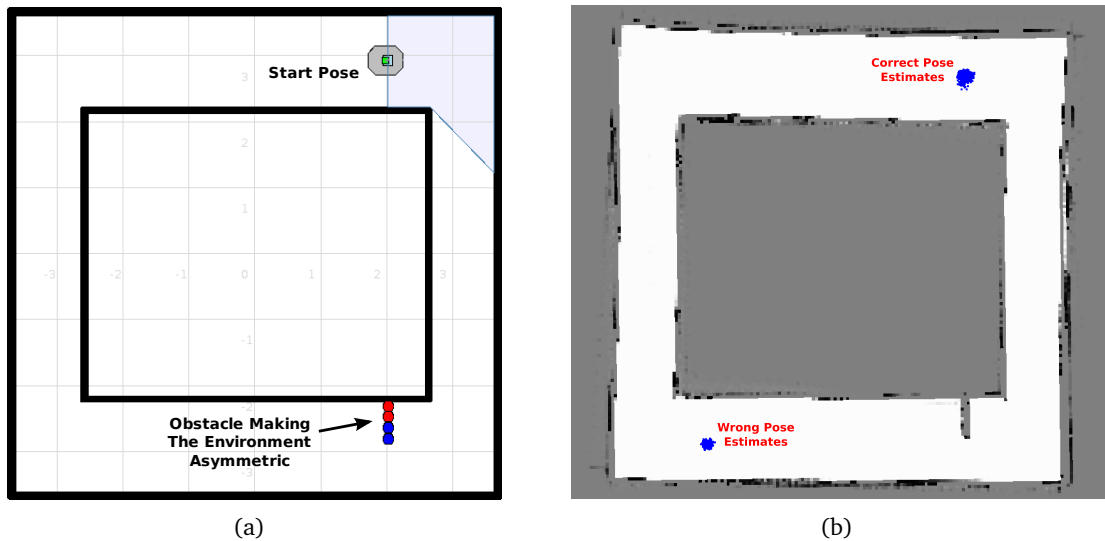


Figure 5.4.: Global localization tested on the simulation platform *Stage* without the designed mini-exploration. a) shows the start pose of the robot in a symmetric environment when omitting the small obstacle. This obstacle added on purpose is pointed with a black arrow. The perceived area of the laser sensor is illustrated with light blue in front of the robot. The result of localization is given in b), when the robot stands still at the start position. Two groups of blue particles center on the two hypotheses of robot's pose in terms of the perceived sensor data. Compared to the start pose of the robot illustrated in a), correct pose estimates are marked as well.

ize itself in the environment map. Thus, path planning can be assessed with *Stage* as will be discussed in section 5.2.3. In comparison to the separate *GLOC* used for the eight trial runs, *GLOC* integrated in module *SLAM and GLOC* was used *only* for the evaluation of the proposed path planning approach in the simulation platform *Stage*.

In general, *GLOC* is responsible for a global localization task when the map of the environment is given. As mentioned in section 4.4 the ability of global localization is necessary for the robot when the environment map is built in the *Home-Tour*. The robot needs to reuse the built map and perform navigation tasks without the help of the human user. Therefore, *GLOC* has been integrated to make the current robot system more robust. The global localization approach integrated in *GLOC* is based on particle filters. Integrating the ideas of adaptive sample size (KLD-sampling [47]), auxiliary particle filters [120] and rejection sampling [96] Blanco et al. [13] have proposed this approach and implemented in the MRPT library [9]. Recalling the problem illustrated in Fig. 1.5 multiple hypotheses of the robot's pose are probably obtained, if the robot stands still. In other words, the information perceived by the robot does not sufficient to determine its pose uniquely. Therefore, the robot needs to move around in the environment and tries to disambiguate the multiple hypotheses of the pose estimate. Fig. 5.4 and Fig. 5.5 illustrate an example in a simulated environment. Notice that it is difficult for a robot to get a correct pose estimate in the symmetric parts of the simulated environment, unless the added obstacle is perceived. Actually the obstacle added into the environment on purpose becomes a landmark for robot localization. If the robot did not move and stood still at the start position, in terms of the perceived sensor data two hypotheses of the robot's pose were obtained and illustrated with the two groups of particles (see Fig. 5.4(b)). In contrast, when the robot moved around

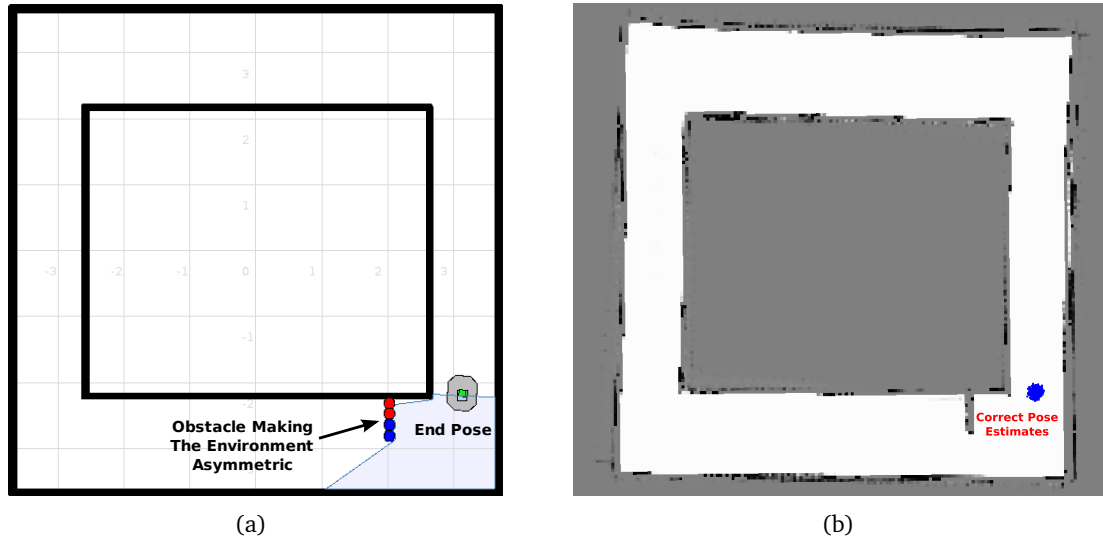


Figure 5.5.: Global localization tested on the simulation platform with the designed mini-exploration. a) shows the end pose of the robot, when it localizes itself in the environment. The light blue region in front of the robot shows the perception area of laser sensor. b) illustrates the result of localization, when the mini-exploration is performed. The robot is quite certain about its position, after the obstacle has been perceived. As the majority of particles are centered tightly around the correct position, the robot is stopped.

and perceived the landmark that cannot be *seen* in the other group of the estimated poses, the particles converged to the unique and correct hypothesis (see Fig. 5.5(b)). Hence, the idea of a mini-exploration has been integrated in *GLOC*, which can help the robot to localize itself correctly in the environment.

Note that the assignment of the mini-exploration is to send movement commands to robot actuators. Fig. 5.6 illustrates the computation of a robot's goal. Firstly, the perception area of the laser sensor is divided equally into n sectors¹ with the same field angle $180^\circ/n$. The minimum distance to an obstacle is computed for each sector, and the n minima are sorted for all the sectors. The maximum distance r_i is chosen from the n minima for sector i , in which the robot's goal G is computed. Thus, G can be defined in the polar coordinate system centered on the origin of the robot,. The pole is the origin of the robot, and the polar axis is the orientation of the robot. The radius and the polar angle of goal G is l and α defined respectively with the following equations:

$$l = k \times r_i \quad \text{with} \quad 0 < k < 1 \quad (5.1)$$

$$\alpha = \frac{180^\circ}{n} \times \left(i - \frac{1}{2}\right) \quad \text{with} \quad n \geq 1, \quad (5.2)$$

where the factor k enables the robot's goal to keep away from the obstacles in sector i and the angle α is chosen in the middle of sector i . In addition, goal G defined in the polar coordinate system needs to be transformed in a global frame and sent to the component *Obstacle Avoidance* for local navigation. After the robot has been certain about its pose, the task of the mini-exploration is successfully completed and the robot should stop. Therefore, a criterion for stopping the mini-exploration is necessary, which is given with the following

¹In our case n is assigned to 6 for the user study.

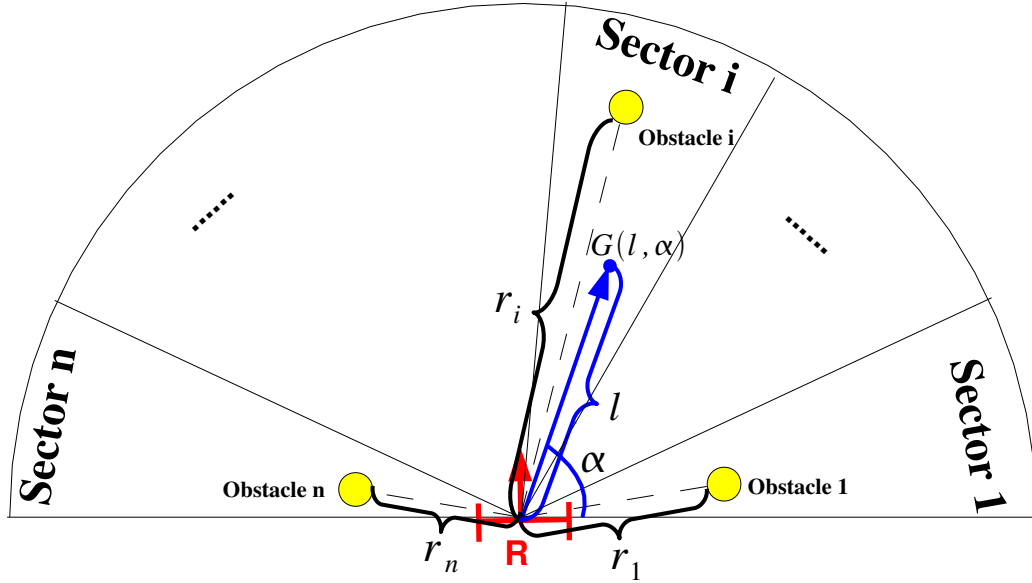


Figure 5.6.: Goal decision for the mini-exploration strategy. The laser plane is equally divided into n sectors. The robot is shown with R , and its orientation is marked with the red arrow. For the n sectors the minimum distance to obstacles is recorded with r_1, r_2, \dots, r_n respectively. Robot's goal G is selected in the sector in which the distance to obstacles r_i is maximum. G is set in the middle of sector i , which can be computed in the robot's polar coordinate system according to radius l and polar angle α .

inequation:

$$\sqrt{\det(\Sigma_{RobotPose})} < T_{LocUncertainty}, \quad (5.3)$$

where $\Sigma_{RobotPose}$ is a 3×3 covariance matrix for the estimation of robot's poses, and $T_{LocUncertainty}$ is a predefined threshold. Once the robot is assumed to localize itself in the environment², i.e. the inequation 5.3 is fulfilled, it will stop and wait for movement commands sent by other components, such as *Person-Following*.

Furthermore, a mini-dialog module has been designed as a state machine, in order to facilitate the guided tour performed in the user study. Five states of the mini-dialog system, namely *Initial State*, *Follow*, *Stop*, *Go To* and *Person Lost*, are defined, as Fig. 5.7 illustrates. State transitions are triggered by speech instructions shown nearby the corresponding arrows, while the corresponding responses are written inside the state circles.

Note that in the user study the mini-dialog system started at state *Initial State*, at which the robot did not get any speech instruction from the guide-person. When the movement command "Follow me!" is sent by the person and accepted by the robot, *Initial State* will be translated to state *Follow*. Compared to the system architecture shown in Fig. 5.3 *Person-Following* will be triggered, after the acceptance of "Follow me!" is sent from *Mini-DLG*. Notice that *Path Planning and Re-planning* will be triggered as well, as the path planning approach discussed in section 4.2 is based on the observed trajectory of the guide-person.

²Note that the problem of recovering from catastrophic localization failures still challenges the proposed mini-exploration. See e.g. the robust algorithm Mixture-MCL discussed in the work of Thrun et al. [151].

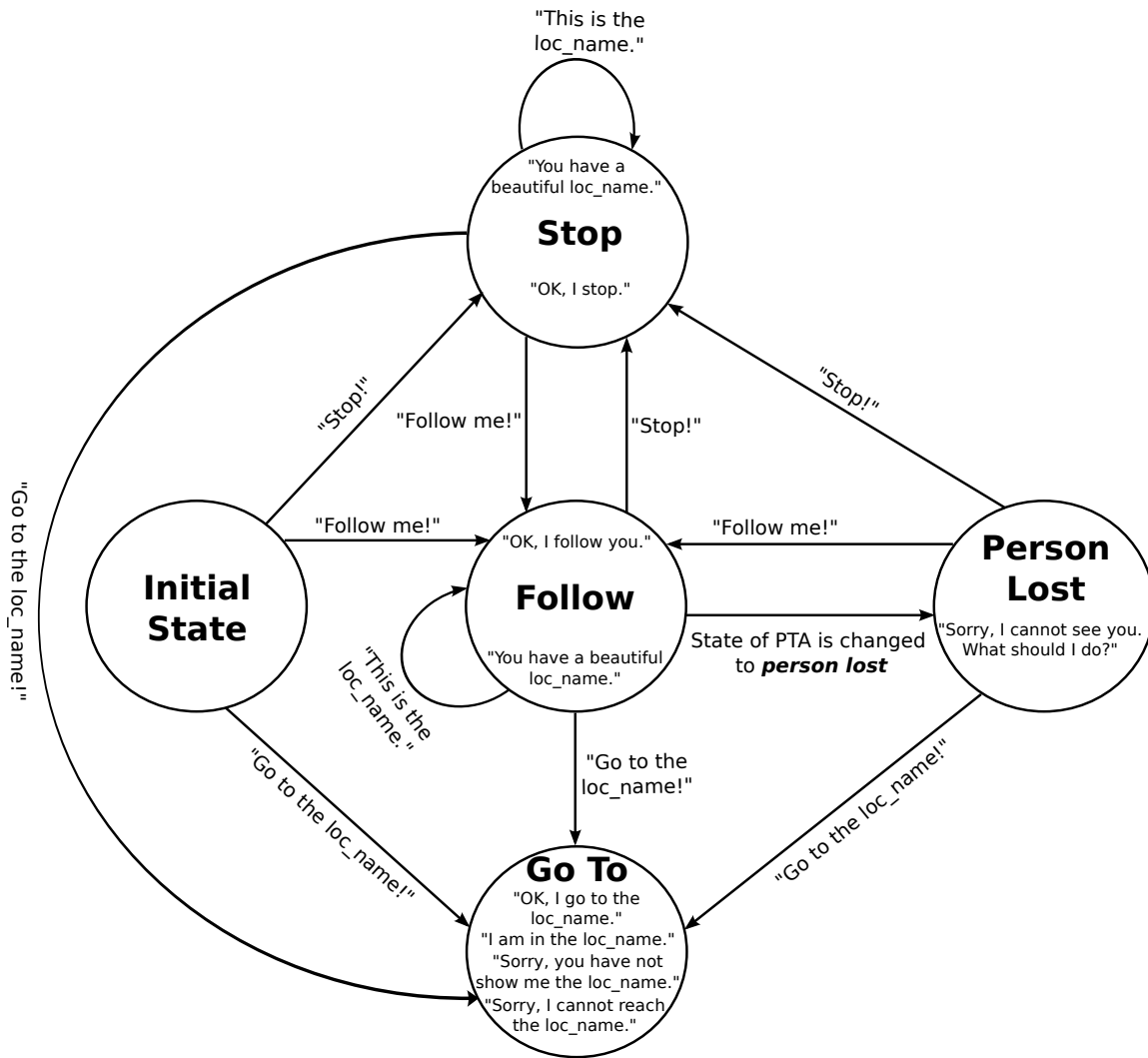


Figure 5.7.: The mini-dialog module designed for the user study. The dialog system starts at the *Initial State*, from which state *Stop*, *Follow* and *Go To* can be reached in terms of the corresponding speech instructions commented nearby the arrows. The corresponding responses are written inside the state circles. At state *Follow* and *Stop* the guide-person may show *locations* labeled by specific *loc_name* to the robot. State *Person Lost* will be reached, when the signal *person lost* from PTA is true.

The guide-person may show the *locations* specified with *locame* to the robot at state *Follow* or *Stop*. The indicated *locations*, such as the *livingroom* shown in Fig. 5.2, are stored in module *Path Planning and Re-planning*. The speed of the robot will be set to zero by module *Obstacle Avoidance* when the command “*Stop!*” is obtained. The state of *Mini-DLG* will be changed to *Person Lost*, when the guide-person is lost by the *Person Tracking and Attention System* (PTA) at the state *Follow*. A prompt “*Sorry, I cannot see you. What should I do?*” will be sent out. By this means, the guide-person can know the current situation of the robot and select a proper movement command to continue the trial run.

State *Go To* is designed to perform a path planning task. A path will be planned for the robot from the current robot’s pose to a *location* which has been shown by the guide-

person. Note that in the user study the robot did not really carry out such a task, as the planned path can be computed and illustrated in the simulation platform *Stage* in terms of the online collected data (see section 5.2.3). Nevertheless, whether the set location is reached successfully should be considered. Various situations have been taken into account, e.g. the goal was reached successfully, the goal location has not been shown, or the goal cannot be reached in a predefined time limitation. Thus, different responses are provided as shown in Fig. 5.7 inside state *Go To*.

The SwissRanger camera was not integrated in the robot platform for the user study, as illustrated in Fig. 1.2(b). However, the feasibility of the motion generation based on *virtual laser* has been validated in section 4.3.3. As illustrated in Fig. 5.3 the *ND* approach has been integrated in the current system for local navigation tasks. Due to the consistent input data, i.e. the laser data, the proposed *virtual laser* is 100% compatible and transparent for the current *Obstacle Avoidance (ND)* module.

5.2. Assessment of Individual Modules in the Integrated System

As introduced at the beginning of this chapter, the system consisting of the three core modules has run as a whole in the user study. In Fig. 5.3 an overview of the system components and the connections between them has been illustrated. In terms of the experiment procedure presented in section 5.1.1 eight subjects have conducted the robot platform **BIRON II** in the user study. Though some problems occurred during the experiments, the subjects can solve them by themselves. E.g. if the robot was not able to understand an instruction given by a guide-person, she or he tried to repeat the command again till the robot replied. Or when the subjects were lost by the *Person Tracking and Attention System (PTA)* and got the prompt “*Sorry, I cannot see you. What should I do?*” from the robot (see Fig. 5.7), seven persons of the eight subjects chose the command “*Follow me!*” to continue the guided tour. Only one subject was not sure to use this command until a confirmation was received from our side. Moreover, all subjects were able to accomplish the task illustrated in Fig. 5.2 without causing any dangerous situations either for the surroundings or to themselves. No technical or system problems occurred during the various trial runs, e.g. no signals from robot sensors and hardware control, no response from the system, system restart, etc..

Apart from the information of the whole system obtained from the user study, in the following I will focus on the evaluation of individual modules constituting the core of the implemented system. Consistent results of the mapping process and the path planning approach discussed in section 3.2 and 4.2.3 respectively were achieved from the user study.

5.2.1. Person-Following

The adaptive context-aware person-following behavior discussed in section 2.3 was adopted in the user study. Recalling the general information given in section 5.1.1 the robot can successfully follow the subjects in the user study. Qualitatively, the proposed person-following behavior is competent to accomplish a *Home-Tour* scenario in a real environment. Hence, in the following a quantitative assessment of the person-following behavior will be mainly discussed.

Raw data collected from different trial runs are introduced at first. Next, the raw data

need to be pre-processed, so that the data really used for the person-following behavior are sifted out and the spurious trajectory of the guide-person can be removed. Afterwards, quantitative measurements extracted from the pre-processed data are used to assess the person-following behavior. A discussion of the evaluation results will be given at last.

Raw data

Adopting the adaptive context-aware following behavior, the robot can switch its behavior according to the core strategies given in section 2.3.1. Spatial context, i.e. the perceived surroundings and the relative position of the robot and the guide-person, provides the basis for the change of the robot's behavior. Therefore, built environment map, trajectory of the person and poses of the robot are necessary for the evaluation of person-following. As the origin of the environment map is the start pose of the robot, both the poses of the robot and the positions of the person can be represented in the map built in the individual trial runs. However, a comparison of results obtained from different built maps, i.e. different world coordinate systems, is not possible. A common frame is therefore necessary, so that trajectory of the person and the robot from different trial runs can be represented in this frame. Since all the eight trial runs were carried out in the same environment sketched in Fig. 5.2, a ground truth map of this environment (see Fig. 5.1) was built by controlling the robot with a joy stick. With the help of the separate *GLOC* illustrated in Fig. 5.3 the robot can localize itself in this map. Moreover, the trajectory of the guide-person and the robot can be recorded in this map as well. Thus, a common frame for analyzing and assessing the person-following behavior in the user study was available. Pairs of robot's poses and person's positions were stored one after the other in chronological order as a part of raw data. Hence, information of the person-following behavior can be recorded as detailed as possible. In other words, the poses of the robot and the positions of the person which were both sent out at a frequency circa 10Hz were completely recorded.

On the basis of the common frame, i.e. the ground truth map of the environment, both trajectories of the person and the robot can be represented in this map. However, the information of behavior changes of the robot has not yet been taken into account. Therefore, which following behavior the robot chose and where this behavior was used needed to be recorded as well. The selected behaviors were marked for all pairs of the poses of the robot and the positions of the person collected in each trial run. Besides, polar coordinates of the guide-person output by *Person Tracking and Attention* (PTA) express positions and orientations of the person with respect to the robot. It is important to record these information³, because behavior changes are dependent on relative positions of the robot and the person. As mentioned in section 2.4.1, state *Person Lost* in PTA was recorded as well for the pre-processing that will be discussed in the next section.

In summary, a time-stamped sequence of tuples comprising person's positions directly measured by PTA, state *Person Lost* in PTA, robot's poses and person's positions transformed into the global coordinate frame (the ground truth map), as well as the selected person-following behaviors is recorded as raw data. The raw data were regarded as a reference of the assessment. Fig. 5.8 shows the raw data collected from the second experimental run and represented on the ground truth map (see Fig. 5.1). Trajectories of the robot and the person are both depicted in the ground truth map with the thick and the thin curve respectively. In order to annotate the robot's behavior, different colors are applied for the trajectory of the

³Note that person's positions with respect to the robot coordinate system can be also computed in terms of the person's and robot's positions in the ground truth map.

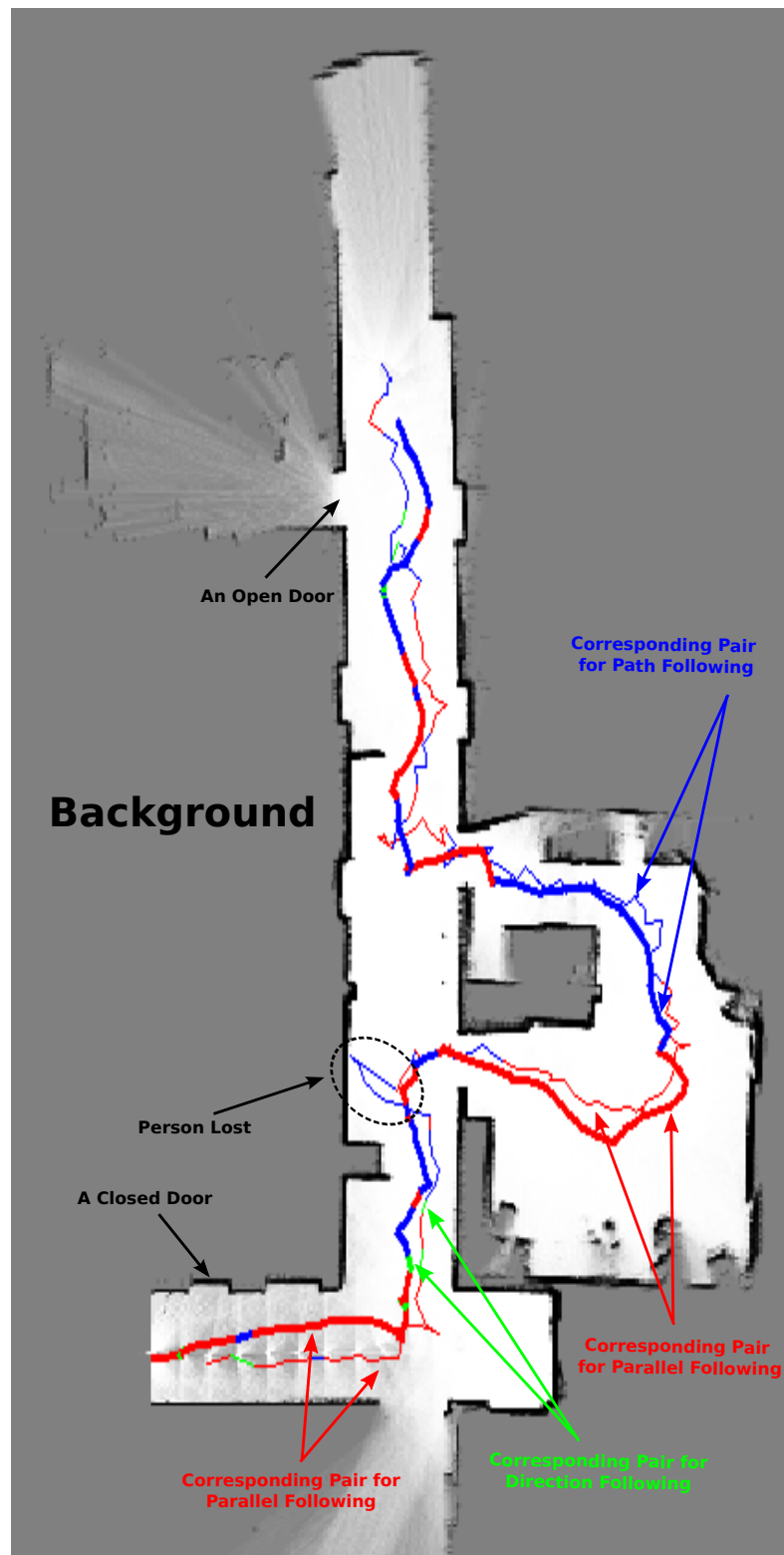


Figure 5.8.: Person-following behavior represented on the ground truth map. The raw data of the second experiment are used. Robot's and person's trajectory is depicted with thick and thin curve respectively. The robot's trajectory is illustrated with red, green and blue. The three colors represent the parallel-, the direction- and the path-following behavior respectively. The corresponding parts of the person's trajectory are marked with the same color. Examples for the corresponding pairs on the two trajectories are shown. The location at which *Person Lost* occurred in this experiment is illustrated with the black dashed ellipse.

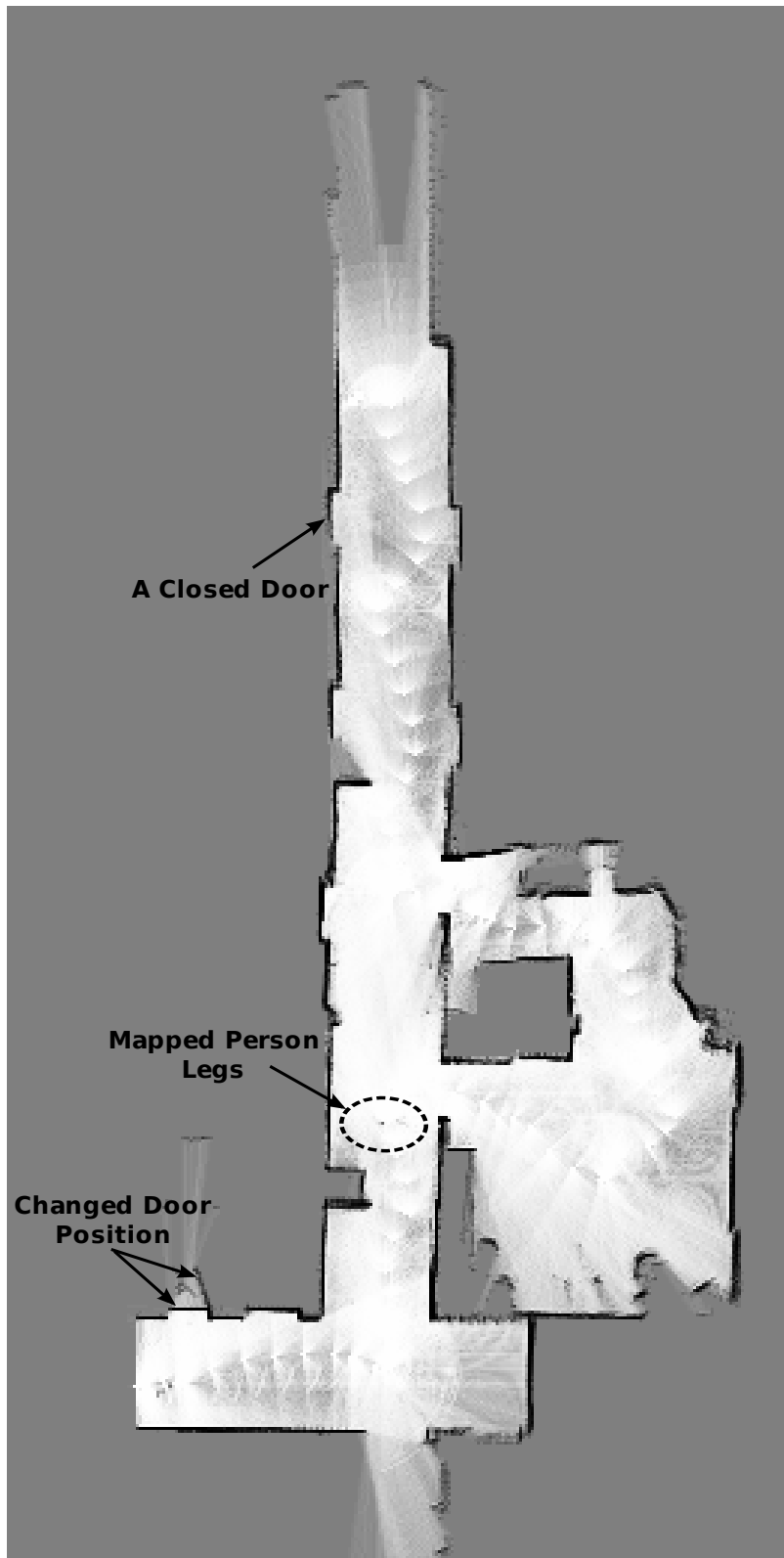


Figure 5.9.: Map built during the second experiment. In contrast to Fig. 5.8 the corresponding mapped pair of person's legs are illustrated. The changed door positions are marked as well.

robot. Red, green and blue parts of the trajectory indicate parallel-, direction- and path-following behavior respectively. The three colors are marked for the corresponding parts of the trajectory of the person as well. Examples for the corresponding pairs of trajectory parts are illustrated with different colored pairs of arrows in Fig. 5.8. The map built in this experimental run is shown in Fig. 5.9. In comparison with Fig. 5.8 uncertainty of the occupancy grids becomes higher in the human guided *Home-Tour*, keeping in mind that white grids imply free space, while black ones correspond to obstacles. Besides, small changes in the environment, e.g. the open or closed door, are illustrated as well in Fig. 5.8 and 5.9.

Pre-processing of the raw data

Notice that the location at which the guide-person was lost is marked with the black dashed ellipse⁴ in Fig. 5.8. It can be easily detected from the raw data, as *Person Tracking and Attention* (PTA) sent the same person angle computed from the last track step till the person was really lost⁵. The thin blue part of the curve inside the ellipse leads the guide-person against the wall, because the robot went on moving while the wrong orientation of the person was sent by PTA. This part of the curve does not reflect the real movement of the person and is regarded as a spurious trajectory part of the person. Moreover, these spurious person positions may influence the selection of the robot's behavior⁶. Therefore, a pre-processing should be done for the raw data, so that the trajectory of the person before the state in PTA is changed to *Person Lost* can be detected and deleted.

Besides, the zigzag person's trajectory, i.e. the thin curve, is notable in the ground truth map. It is mainly due to the *delay* of person's positions sent by PTA⁷. Thus, noised movement trajectory of the guide-person is presented. Smoothing the person's trajectory obtained from the raw data is necessary. According to the two thresholds defined for path- and parallel-following in section 2.4.3 and section 2.4.4 respectively, the person's positions that were really used for the person-following behavior should be taken into account⁸. Therefore, except for obliterating the tracks resulting from *Person Lost*, the distance between two neighboring person positions was checked. The result of the pre-processed raw data is shown in Fig. 5.10. The part of *Person Lost* has been removed. Note that the person's trajectory was smoothed by the distance threshold given in path- and parallel-following behavior respectively, instead of the use of a smoothing filter. By this means, on the one hand person's positions really used for behavior selection were completely preserved. On the other hand, the thin trajectory of the person becomes relatively smooth compared to the corresponding trajectory in Fig. 5.8. In terms of the pre-processed data and the map built in each trial run, all behavior changes of the robot can be explained by the core strategies discussed in section 2.3.1. In the following I will discuss the results of the person-following behavior achieved from the pre-processed data.

⁴Note that the person's legs can be added into the map built with the technique discussed in section 3.2, when *Person Lost* occurs. See the results of the mapping process in section 5.2.2.

⁵Please see [91] for details.

⁶Recalling the general strategies presented in section 2.4.1, *Person Lost* is difficult to be detected in an online person-following process. In order to reduce the negative impact of the false person's positions on behavior selection, an offline detection is discussed.

⁷The positions of the person are interpolated on the basis of the history of the person's legs detected by the laser sensor, the person's face detected by the camera and the person's voice located by the stereo microphones. See [91] for details.

⁸Note that for direction-following no restrictions of person's positions were applied. Therefore, all person's positions were recorded for direction-following.

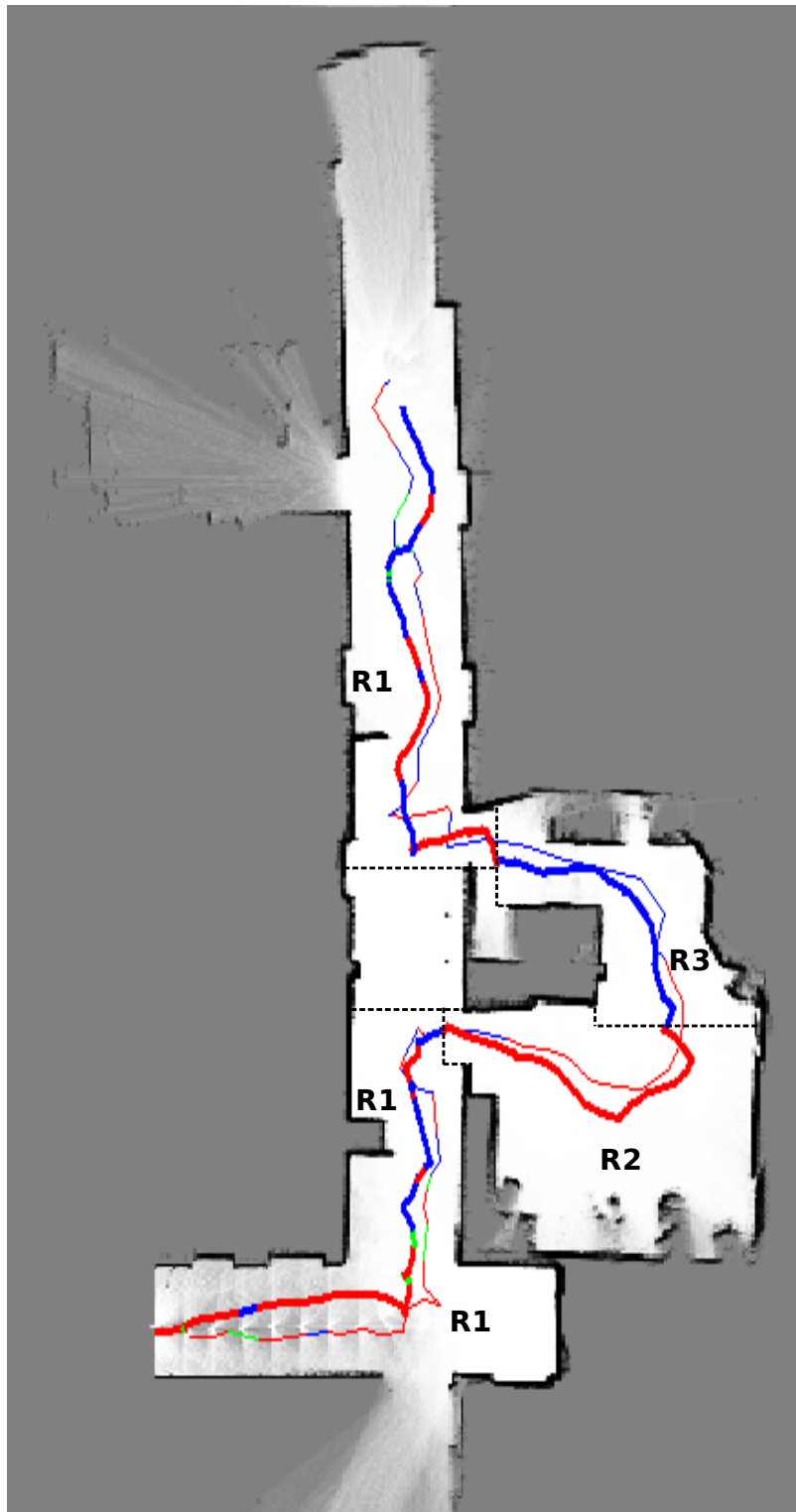


Figure 5.10.: The person-following behavior recorded in the second experiment. The trajectories of the guide-person and the robot are represented on the ground truth map. Applying the pre-processing of the person's trajectory the thin curve becomes relatively smooth with respect to Fig. 5.8. The definitions of the thick and the thin curve, as well as the three colors are the same with Fig. 5.8. The region R1, R2 and R3 corresponds to the transverse and vertical aisles, the large room and the narrow passage respectively illustrated in Fig. 5.1.

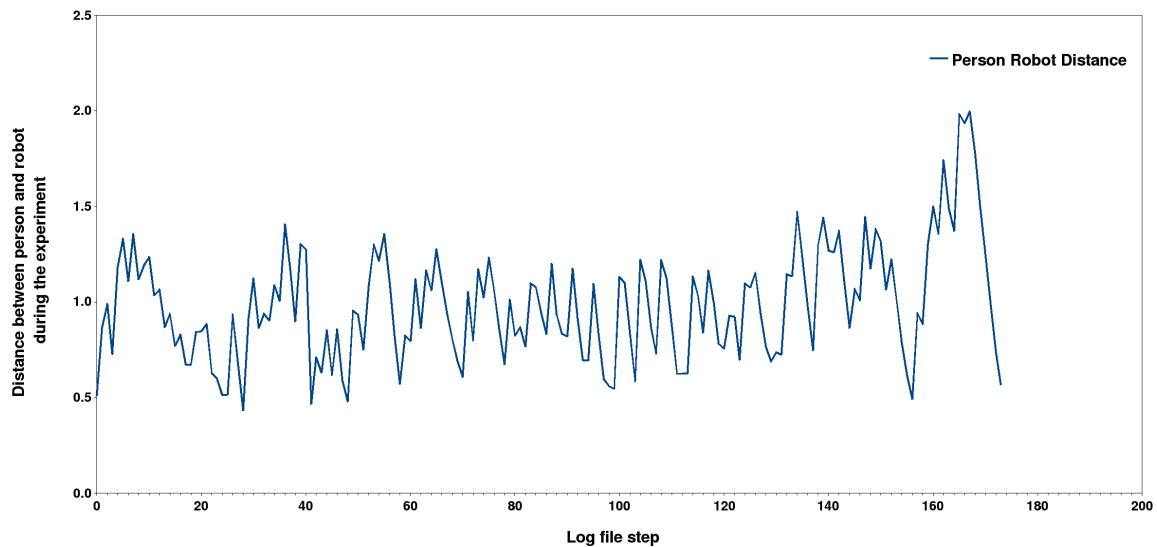


Figure 5.11.: Distance between the person and the robot in the second experiment.

Study I: Relative positions of the person and the robot

As mentioned in section 2.3.1 keeping up with a guide-person has been regarded as the most important issue for a person-following behavior. In other words, on the one hand the guide-person should not move far away from the robot, while the robot is not allowed to be in close proximity to the guide-person in order to prevent dangerous situations. On the other hand, the angle of the person with respect to the robot should not be very large. Hence, the relative positions of the person and the robot, i.e. distance and relative angle between the person and the robot, are selected as quantitative measurements to describe their positional relation. By this means, the ability of the proposed following behavior in keeping spatial relation of the guide-person and the robot can be assessed.

As the most current pair of robot's pose and person's position were stored one after another in chronological order, the Euclidean distance between the positions of the person and the robot can be computed. Person angles defined in robot coordinate system were obtained directly from PTA. The second experiment is chosen as an example and shown in Fig. 5.10. Fig. 5.11 illustrates the distance between the guide-person and the robot step for step. As discussed in section 2.4.1 threshold $T_{ShortDis}$ was defined for *departure* and *arrival* of all basic person-following approaches. In terms of Hall's study about interpersonal distances [69] a distance lying in the interval $[46cm, 122cm]$ is referred to as *personal distance*. Extremities can be used to get in physical contact with each other. Moreover, the concrete distance marks the level of familiarity of one person with another. Hall's study about interpersonal distances was based on human-human interaction (HHI) though, it was generally concluded by Hüttenrauch et al. [77] that human participants preferred Hall's *personal distance* ($46cm$ to $122cm$) when interacting with a robot. Hence, in the user study $T_{ShortDis}$ was set to $1.2m$, which implied a *personal distance*, for the robot platform **BIRON II** aiming at providing personal services to a human user.

The third row of table 5.1 shows the mean $0.984m$ and the standard deviation $0.296m$ of the recorded distances between the person and the robot for the second experiment. The computed mean distance lies in the interval $[46cm, 122cm]$, and the difference be-

Table 5.1.: Statistic results of distance between person and robot, as well as relative angle of the person with respect to the robot computed from individual experiments.

	Mean Distance [m]	Std-deviation of Distance [m]	Mean Angle [degree]	Std-deviation of Angle [degree]
E1	1.149	0.422	2.499	16.785
E2	0.984	0.296	-1.296	23.974
E3	1.197	0.382	-5.545	32.680
E4	0.990	0.220	-1.926	35.093
E5	1.099	0.448	1.394	24.831
E6	1.175	0.408	0.597	26.435
E7	1.189	0.446	1.631	31.043
E8	1.435	0.469	0.609	31.785

tween threshold $T_{ShortDis}$ and the computed mean distance is $0.216m$. Therefore, on the one hand the mean distance $0.984m$ computed from the second experiment supports the argument about the distance between human participants and the robot drawn by Hüttenrauch et al.. On the other hand, the distance between the guide-person and the robot can be effectively controlled by the proposed person-following behavior performed in a real world interactive scenario. As shown in the third row of table 5.1, although the person angles with respect to the robot coordinate system have the deviation 23.974° , the mean angle is -1.296° in the second experiment. As explained in section 2.2.1 a zero person angle indicated that the person stood right in front of the robot. Adopting the core strategies given in section 2.3.1 the robot always tried to keep the guide-person in its front field **FF**. Besides, the *short distance behavior* introduced in section 2.4.1 made the robot turn to the person once the distance to the person was inside threshold $T_{ShortDis}$. Thus, the effectiveness of the proposed core strategies and the *short distance behavior* is verified by the relative small mean angle.

In the following the distance between the person and the robot, as well as person angles with respect to the robot coordinate system will be computed for each experiment. Table 5.1 shows the computed mean distance and mean angle, as well as the standard deviation of the distances and the angles for each experiment. Seven mean distances are in the range of Hall's *personal distance*, i.e. from $46cm$ to $122cm$, except the mean distance $143.5cm$ obtained from the last experiment. The mean person angles computed for each experiment are relative small and comparable. Furthermore, the computed mean values and the standard deviations for all the eight experiments are given in table 5.2. The mean

Table 5.2.: Statistic results of distance between person and robot, as well as relative angle of the person with respect to the robot computed from all eight experiments.

Mean Distance [m]	Std-deviation of Distance [m]	Mean Angle [degree]	Std-deviation of Angle [degree]
1.146	0.411	-0.393	28.446

distance $1.146m$ and the mean person angle -0.393° are generally consistent to the results obtained from the individual experiment. Therefore, the conclusions drawn from the second experiment are verified by all trial runs in the user study.

Study II: Context and choice of a following behavior

Apart from the basic requirement for a person-following behavior, i.e. keeping up with the guide-person, choosing a following behavior in terms of the spatial context is the characteristic of the following behavior proposed in this work. As mentioned at the end of the section introducing the pre-processed data, all behavior changes of the robot can be explained by the core strategies given in section 2.3.1. Considering general situations in a domestic environment, such as open space, narrow and wide passages, etc., a relation between spatial context and the choice of a following behavior is expected to be built. According to the core strategies of the proposed following behavior, parallel-following is rather appropriate for open space and wide passages, while path-following is chosen preferably in narrow passages.

Based on these rather intuitive ideas the environment should be divided into regions in terms of spatial context, so that the selected one or multiple following behaviors can be analyzed in a relatively small and typical area. As illustrated in Fig. 5.1 the aisles, a large room with rather open space and a relative narrow passage exist in the environment for the user study. According to these available situations and the recorded robot's path, the environment is divided into three regions as shown in Fig. 5.1 and Fig. 5.10. Comparing the two figures region R1 is corresponding to the transverse and vertical aisles. Notice that the area in which a table is laid is excluded from R1, because the robot did not move through this area in the user study. The two black dashed lines defining this area are selected in terms of one side of door D1 and D2 respectively. Besides, region R2 and R3 is chosen according to the large room and the narrow passage respectively. The exact boundaries between region R1 and R2, region R2 and R3, as well as region R3 and R1 are selected in terms of the start position of the robot for a following behavior newly selected by the robot. E.g. the start position of the robot on the part trajectory of path-following is chosen, so that the region R2 and R3 is divided by the transverse black dashed line going through this position, while the vertical black dashed line goes through the robot's position at which parallel-following begins. These region boundaries marked with black dashed lines are regarded as a reference on the ground truth map for the pre-processed data achieved from all trial runs. Since the robot's positions defining these boundaries have been recorded in the second experiment, the robot's trajectories stored in all experiments can be precisely divided according to the predefined regions R1, R2 and R3.

Dividing the environment in terms of the available spatial context and the robot's trajectory, the percentage of the following behaviors selected by the robot can be computed for each region with the following equation:

$$F_{ij}\% = \frac{P_{ij}}{Q_j} \times 100\% \quad \text{with } i \in \{1, 2, 3\} \quad \text{and } j \in \{1, 2, 3\}, \quad (5.4)$$

where $F_{ij}\%$ stands for the percentage of the following behavior i in region j . P_{ij} represents the number of points in the robot's trajectory stored for behavior i in region j , while Q_j is the total number of points in the robot's trajectory recorded in region j .

Table 5.3 shows the analysis of the second experiment. The percentage of the following behaviors selected in the three regions are computed according to equation 5.4. Notice that in region R2 and R3 parallel-following and path-following is uniquely chosen by the robot respectively. Actually this result is not a big surprise, as on the one hand in terms of the data collected from this experiment the start robot's position for a new following behavior defines the corresponding new region. On the other hand, according to the core strategies discussed in section 2.3.1 parallel-following and path-following is the behavior that should

Table 5.3.: Analysis of following behaviors selected in the defined three regions for the second experiment.

	Parallel-Following	Direction-Following	Path-Following
Region1	60.55%	4.59%	34.86%
Region2	100.00%	0.00%	0.00%
Region3	0.00%	0.00%	100.00%

be chosen in the large room (R2) and through the narrow passage (R3) respectively. In contrast, three basic following behaviors are selected in region R1, i.e. the aisles in the environment. Direction-following is rarely chosen by the robot, which implies that the guide-person is generally kept in the front field (FF) of the robot. However, parallel- and path-following are both appropriate for R1 according to the spatial context. Restricted by the width of the aisles and the relative person's position with respect to the robot, one of the two following behaviors can be chosen. Note that the width of the aisles ranges from $1.1m$ to $2.1m$. Hence, parallel-following cannot be performed in the relative narrow parts of the aisles. In addition, when the guide-person moves in the middle of the aisles no goals can be computed as well on the left and right side of the person for some parts of the aisles because of the detected walls. Looking back the results shown in table 5.3 and the recorded person's and robot's trajectory in Fig. 5.10 the percentage of parallel-following is higher than path-following, as the guide-person did not always move in the middle of the aisles, especially in the transverse aisle.

Although the relation between the spatial context and the selected following behaviors can be well explained for the second experiment, whether consistent results can be achieved for all the eight experiments? To answer this question the selected following behaviors in the defined three regions are recorded for the remaining seven experiments. An average measurement can be computed for the eight experiments using the following equation:

$$F_{Mij}\% = \frac{P_{Tij}}{Q_{Tj}} \times 100\% \quad \text{with } i \in \{1, 2, 3\} \text{ and } j \in \{1, 2, 3\}, \quad (5.5)$$

where $F_{Mij}\%$ stands for the mean percentage of following behavior i chosen in region j . P_{Tij} represents the total number of points in the robot's trajectory stored for behavior i in region j , while Q_{Tj} is the total number of points in the robot's trajectory recorded in region j .

The analysis of all eight experiments is shown in table 5.4. The percentage of the following behaviors selected in the three defined regions is computed according to equation 5.5. Compared to the results of the second experiment shown in table 5.3 the same following behavior, i.e. parallel- and path-following, is predominantly chosen in region R2 and R3 respectively. Notice that the order of the selected three following behaviors, i.e. parallel-

Table 5.4.: Statistics of following behaviors selected in the defined three regions for all eight experiments.

	Parallel-Following	Direction-Following	Path-Following
Region1	46.65%	12.17%	41.18%
Region2	89.15%	3.88%	6.97%
Region3	9.76%	6.62%	83.62%

path- and direction-following in region R1 is consistent with the second experiment. However, the choice of parallel- and path-following is comparable, as the both following behaviors might be performable in the aisles. In summary, using the adaptive context-aware following behavior discussed in section 2.3 parallel- and path-following would be predominantly chosen in region R2 (the large room) with open space and R3 (the narrow passage) respectively for a guided tour in the same environment. However, the selection of following behaviors in region R1 (the aisles) is dependent on the width of the aisles at different locations and the relative position of the person in the robot coordinate system.

Discussion

Using the proposed adaptive context-aware following behavior the robot was able to follow the guide-person in each trial run successfully. Thus, the task assigned to the subjects can be all accomplished. Besides, behavior changes can be explained by the data collected from the eight experiments. In general, the feasibility of the proposed person-following behavior has been verified.

The two studies focused on quantitative assessments of the following behavior. Keeping up with the guide-person is regarded as the most important issue for a person-following behavior in the interactive *Home-Tour* scenario. This is the center of study I. Although the *Home-Tour* was conducted by the inexperienced subjects, the mean distance between the person and the robot can be well kept according to threshold $T_{ShortDis}$ defined in the proposed following behavior. As shown in table 5.2 due to the threshold $T_{ShortDis}$ set to $1.2m$, the mean distance is generally kept about $1.2m$ in the user study. Apart from the mean distance between the person and the robot that can be controlled by the proposed following behavior, the relative small mean angle of the person in the robot coordinate system has been achieved for all experiments. In other words, the robot tried to turn to the interaction partner, i.e. the guide-person, in all trial runs. In view of the fact that the inexperienced subjects used the robot platform for the first time, both the distance between the person and the robot and the relative angle of the person with respect to the robot are varied in the user study. As illustrated in table 5.2 the deviation of the distance and the relative angle is $0.411m$ and 28.446° respectively. Nevertheless, the relative position of the person and the robot can be controlled as a whole by the proposed following behavior in real world scenarios. Thus, the capability of keeping up with the guide-person has been verified in study I.

Study II focused on the relation between the spatial context and the selected basic following behaviors. As discussed in study I the small relative angle of the guide-person can be achieved by the proposed following behavior. Direction-following is therefore rarely chosen in the user study, which is independent on the spatial context. In rather open and narrow spaces, e.g. the large room (region R2) and the narrow passage (region R3) illustrated in Fig. 5.1, parallel- and path-following is predominantly to be chosen by the robot respectively. In contrast, multiple following behaviors may be appropriate for a region when the spatial context is varied, e.g. the aisles (region R1) illustrated in Fig. 5.1. The robot needs to change its behavior according to the currently observed spatial context. Hence, a predominant following behavior cannot be achieved in such a region.

Besides, it is notable that the two parts of path-following in the robot's trajectory (the thick blue parts) are actually not through door D1 and D2, as Fig. 5.10 illustrates. Parallel-following had been chosen before the robot moved through the doors, as free space nearby the guide-person had been detected. By this means, the robot is able to switch to a behavior

appropriate for the spatial context currently perceived by the robot as soon as possible.

5.2.2. Map Building

Creating an environment representation is one of the tasks performed in the *Home-Tour*. On the basis of the built map the robot is able to navigate in the environment and provide services to its human user(s). However, the guide-person perceived by the robot violates the static world assumed in the RBPF-SLAM approach introduced in section 3.1. Therefore, the person's legs detected by the robot are removed from the laser plane. By this means, negative effect arising from the guide-person is expected to be reduced as far as possible. The results achieved in a simulated environment have proved the benefit of the proposed technique for the built map and pose estimation of the robot. In this section the proposed technique will be assessed in a real environment. Besides, the results of the user study will be compared with what have been achieved from the simulated environment.

As mentioned in section 3.2, the map built by using the proposed technique can be stored for each experimental run. Laser scans used in the mapping process are distinguished from the original ones. The environment representations built in the two cases are therefore different. In order to show the difference of the map built in the two cases, the map information $I(m)$ defined with equation 3.11 is selected as one measurement. In principle the maps built on the basis of the laser scans containing and removing the detected person's legs can be both recorded, when an additional mapping process using the original laser data runs parallel. Nevertheless, extra burden has to be added to the system, which may negatively influence the performance of the online *Home-Tour* scenario. A more practical and economical way is to store the necessary sensor data, i.e. the odometry and laser data, in a *rawlog file*. Note that the most current pair of odometry and laser are stored for each loop of the mapping process as the input of the RBPF-SLAM approach introduced in section 3.1. By this means, the complete mapping process can be equally replayed offline, using the data stored in the *rawlog file*. Moreover, the results of the mapping process achieved from the sensor data collected in different experimental runs can be compared fairly.

Apart from the map built in the *Home-Tour* pose estimation of the robot is the other task of the RBPF-SLAM approach. In contrast to the mapping process performed on the simulation platform *Stage*, no ground truth of robot's poses is available in the real environment. Therefore, absolute pose error of the robot cannot be achieved in the user study. Nevertheless, the results illustrated in Fig. 3.11 and Fig. 3.12 have validated the advantage of the proposed technique for pose estimation in the simulated *Home-Tour*. The other measurement selected in the user study is the square root of the determinant of the pose covariance matrix (see section 3.2). Localization errors can be analyzed on the basis of the pose covariance matrix estimated on the built map.

According to the two measurements selected for the user study, i.e. the map information $I(m)$ and the square root of the determinant of the pose covariance matrix, three studies are designed as follows. First, the information of the built map is computed with and without the proposed technique for each experiment. The difference of the built maps can be illustrated and analyzed with the map information $I(m)$ computed in the user study. Afterwards, a map-based path planning approach is performed on the map built with and without the proposed technique. The capability of the built maps in performing a path planning task can be compared with each other. At last, pose covariance matrices of the robot are analyzed in order to assess the localization error estimated on the basis of the

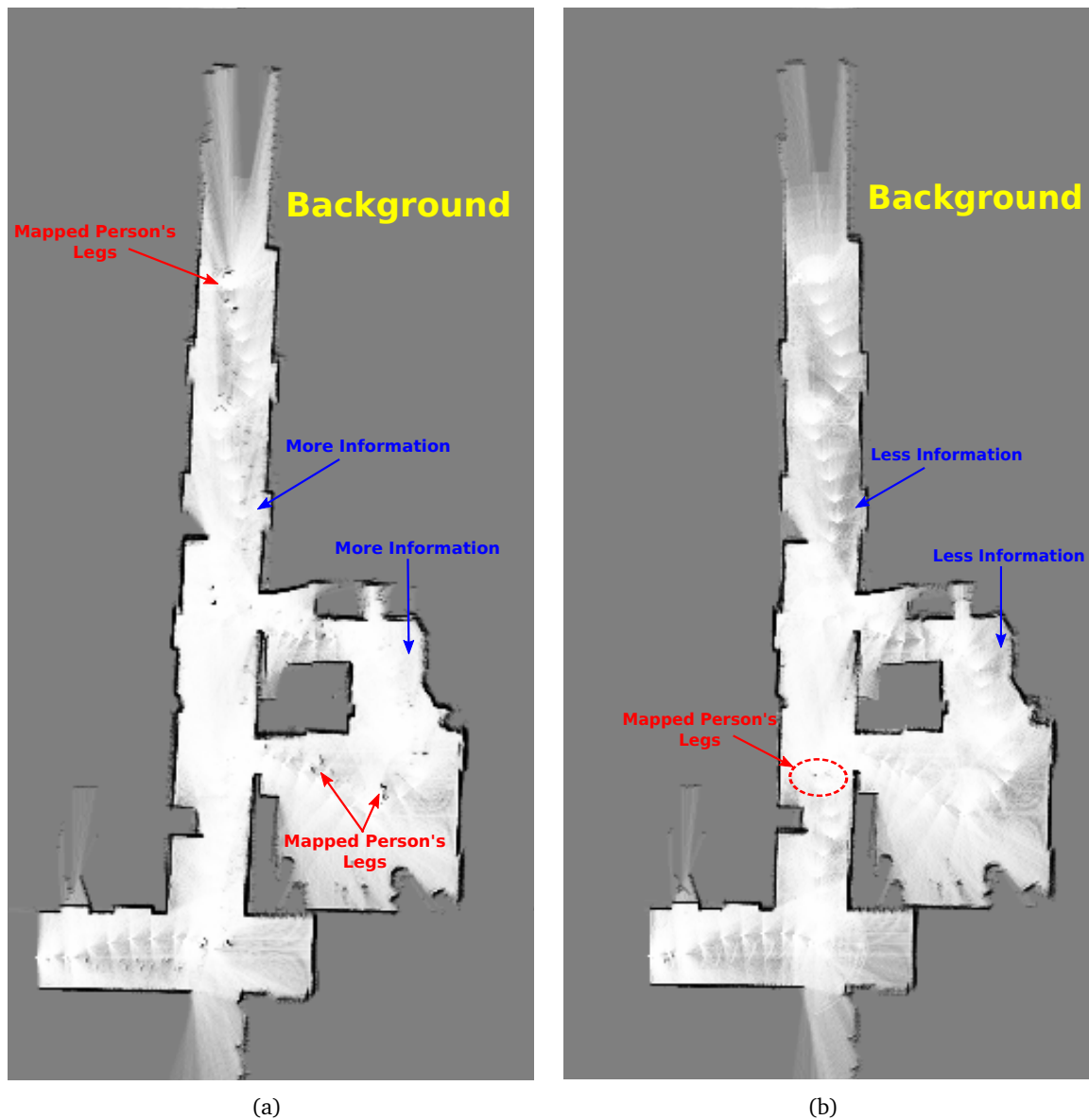


Figure 5.12.: Comparison of the grid maps built during the second experiment with and without the approach proposed in section 3.2. a) shows the mapping result using the original laser scans, while b) illustrates the built map excluding the area occupied by the person from the laser scans. The probability of unperceivable grid cells, such as the background, is set to 0.5. Examples of mapped person's legs are shown with red arrows in a) and b). Besides, examples of more and less information gained in the two cases are illustrated with the blue arrows in a) and b).

built map. Analyzing the selected two measurements and comparing the results of the built map in the user study, more general conclusions can be drawn from the following three studies.

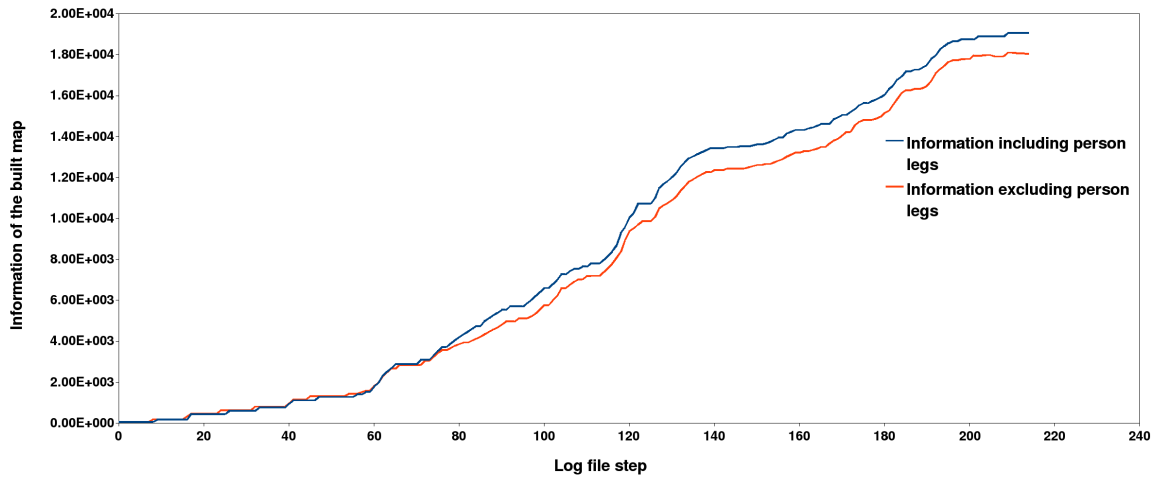


Figure 5.13.: Computed information of the map built in the second experiment according to equation 3.11. The blue and red curve illustrates the result achieved without and with the technique proposed in section 3.2 respectively. The map information is stored every 20 loops of the mapping process.

Study I: Map information

As discussed in section 3.2 the uncertainty of obstacles and free space in the built grid map can be computed by map information $I(m)$. Fig. 5.12 shows the maps built during the second experiment without and with the technique proposed in section 3.2. When the detected person's legs are left in the laser plane, i.e. the original laser is used in the mapping process, they have to be mapped. As marked with the red arrows in Fig. 5.12(a) many gray points are depicted on the path of the guide-person, which indicate the mapped person's legs. In contrast, the gray points arising from the perceived person's legs cannot be easily found in Fig. 5.12(b), since the area occupied by the person has been detected and removed from the original laser plane. One pair of the mapped person's legs are denoted by the red dashed ellipse for the second experiment. Recalling the location of *Person Lost* illustrated with the black dashed ellipse in Fig. 5.8, the mapped person's legs are due to the *Person Lost* occurred once in this experiment. After the guide-person had been re-recognized by the robot, the detected person's legs were removed again from the laser plane. Thus, no mapped person's legs can be clearly made out on the rest path of the guide-person.

The computed map information $I(m)$ in terms of equation 3.11 for the second experiment is shown in Fig. 5.13. At the beginning of the both curves little difference can be made out, as the component *Person-Following* was not started. The original laser was used by the mapping process. When the speech command "*Follow me!*" was accepted by the mini-dialog module and the pair of the person's legs were detected in front of the robot, the proposed technique was triggered. In Fig. 5.13 the blue curve shows the map information computed by using the original laser scans, while the red one was achieved when the proposed technique was used. Note that the occupancy probability of the background as well as the detected person area is set to 0.5. As discussed in section 3.2 zero map information can be gained from the grid cells with occupancy probability 0.5. Therefore, the robot got relatively less information about the environment using the proposed technique, because false information resulted from the guide-person was removed from the built map as far as possible. Examples of the more and less information for the two cases are shown by the

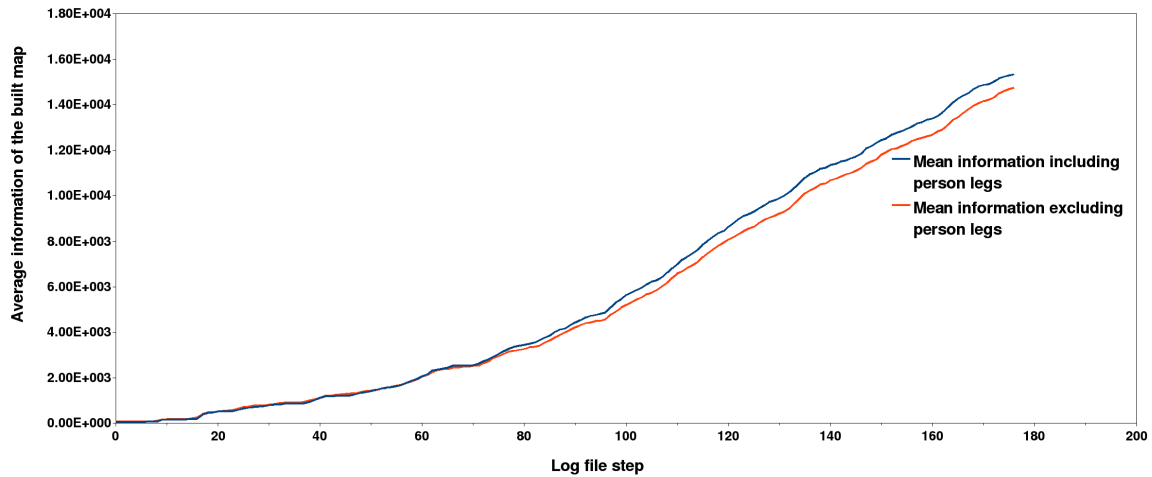


Figure 5.14.: Average map information computed from eight experiments.

blue arrows in Fig. 5.12(a) and Fig. 5.12(b) respectively. Compared to the maps built in the simulated environment (see Fig. 3.3 in section 3.2) consistent conclusions about the map information can be drawn from the user study performed in the real environment. In other words, using the proposed technique less information gained from the environment is better than adding false information arising from the guide-person into the built map.

In order to verify the results achieved from one experiment, i.e. the second experiment, map information is computed for the rest seven experiments. Fig. 5.14 illustrates the average map information computed from the eight experiments. As the log file steps of the individual trial runs are not necessarily the same, the steps from the beginning of one experiment to the minimum steps recorded from the eight runs⁹ are selected for the computation. The results achieved from the second experiment are consistent with the computed average map information. Removing false information arising from the guide-person, average map information is shown with the red curve. However, when adding these false information into the built map, the average map information illustrated with the blue curve lies over the red one as a whole. Notice that little difference can be found at the beginning of the two curves, as the original laser scans were used in the mapping process. Once the guide-person was identified, the proposed technique was triggered. Thus, the blue curve gradually exceeds the red one.

Study II: Performing a path planning task on the built map

In order to provide services, the robot needs to move around and navigate in the environment. Certain knowledge about the surroundings, e.g. an environment map, is therefore necessary for the robot. Using the proposed technique the environment map can be built incrementally in the interactive *Home-Tour*. Now whether the built map is capable of performing a path planning task is the center of this study.

The map-based approach NF1 is used on the two maps shown in Fig. 5.12(a) and

⁹In our case the eighth experiment recorded the minimum log file steps, namely 177 steps. Therefore the log file steps for the computation of the average map information are limited to 177 for all the eight experiments.

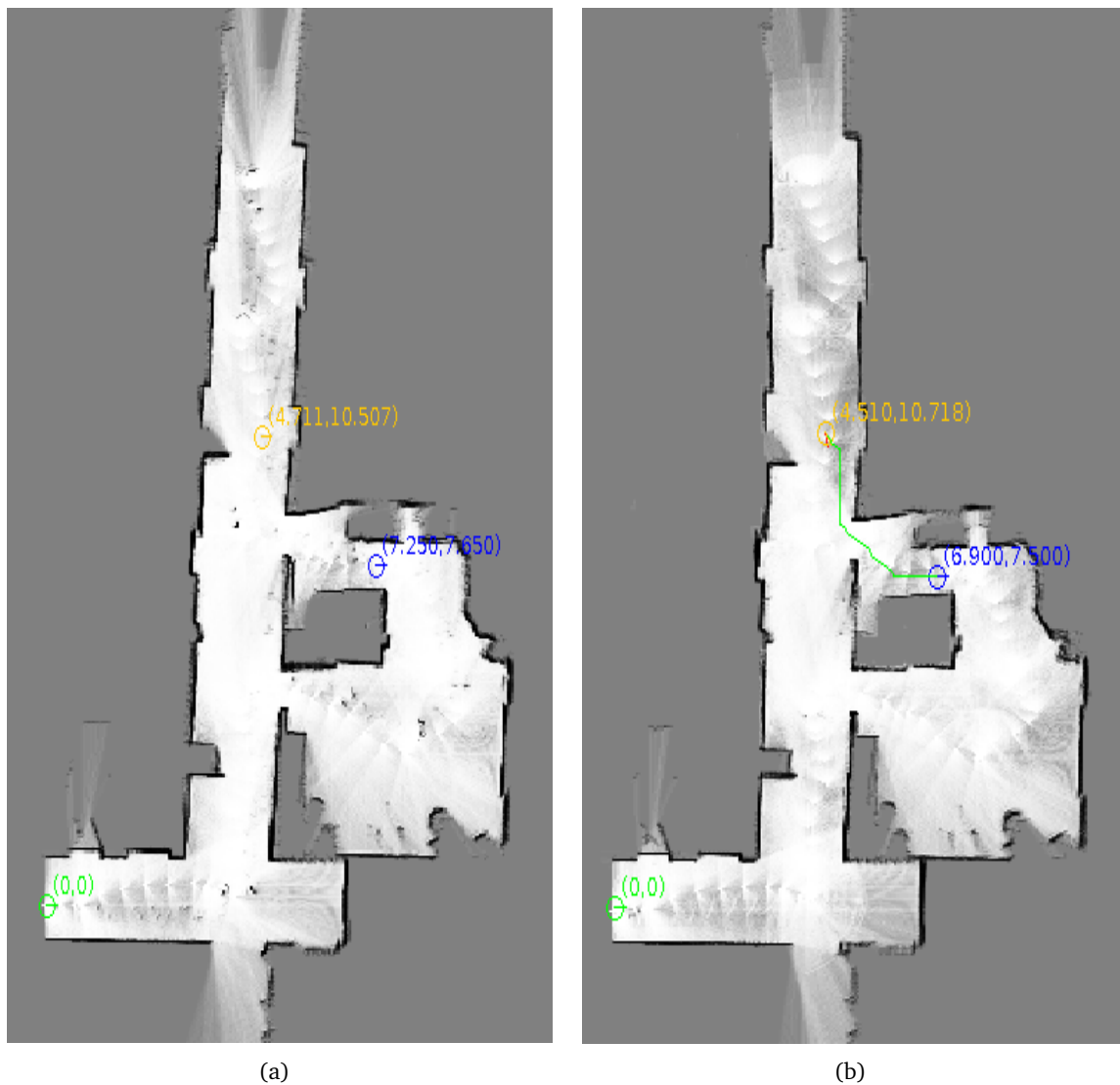


Figure 5.15.: Results of global path planning using NF1 on the map built during the second experiment and shown in Fig. 5.12(a) and Fig. 5.12(b) respectively. The start, current and goal position of the robot is marked with green, orange and blue circle respectively. All these positions are computed with *GLOC* introduced in section 5.1.2 on the corresponding maps. The robot's goals are set in close proximity to each other on the two maps. In b) the path illustrated with green curve leads the robot from the current position to the set goal, while no path can be found in a) due to the false obstacles resulting from the mapped person's legs.

Fig. 5.12(b) respectively. The results of performing a path planning task are illustrated in Fig. 5.15. On the one hand, the robot get into trouble on the map containing many false obstacles arising from the guide-person. No path can be found from the current robot's pose depicted with the orange circle to the set goal denoted with the blue circle (see Fig. 5.15(a)). On the other hand, as illustrated in Fig. 5.15(b) a green path has been planned on the map, in which false obstacles are removed by the proposed technique as far as possible. Notice that the start pose of the robot in the trial run is depicted with a green circle. The current robot's poses and the robot's goals are set in close proximity on the two

Table 5.5.: Statistic results of sqrt determinant of covariance matrices recorded in eight experiments.

	Mean	Std-deviation	Reduced percent of mean	Reduced percent of variance
Sqrt-det inc. legs	4.67×10^{-4}	1.60×10^{-3}	24.1%	46.1%
Sqrt-det exc. legs	3.54×10^{-4}	8.63×10^{-4}		

maps, in order to fairly compare the NF1 approach in both cases.

Thus, the assumption proposed in study I, i.e. less map information is better than adding false information into the map, is proved. Besides, the capability of the map built by removing the false information in performing a path planning task is verified.

Study III: Pose uncertainty

Due to the lack of a ground truth reference of robot's poses, absolute localization error is difficult to be obtained in the real environment. However, the error of the robot's pose can be estimated on the basis of the built map. Pose uncertainty can be analyzed by the covariance matrix of the robot's pose. Hence, the square root of the determinant of the pose covariance matrix is computed and recorded step by step for all the eight experiments. By this means, the mean value and the standard deviation of the square root of the determinant can be computed for the user study.

As shown in table 5.5 the reduced percentage of the mean and the deviation is **24.1%** and **46.1%** respectively, when using the technique proposed in section 3.2. Therefore, the perceived dynamic obstacles arising from the guide-person result in relatively high pose uncertainty of the robot. The same result achieved in a simulated environment has been discussed in section 3.2.

Discussion

From the reported findings in the three studies it can be concluded that the results achieved in the user study are consistent with what have been obtained in a simulated environment. Because of the negative influence arising from the guide-person in the *Home-Tour*, false information has to be added into the built environment representation. Using the proposed technique the false information can be reduced as far as possible. The less information gained from the environment is more beneficial to the robot, in comparison with the more information containing the false obstacles arising from the guide-person.

This assumption can be proved when a path planning task is performed on the built map. Due to the false obstacles mapped on the environment representation, no path can be found, although no obstacles lie between the current robot's position and the goal. Besides, compared to the robot's pose estimated with the laser scans removing the detected legs of the guide-person, false obstacles result in relatively high pose uncertainty of the robot. The absolute pose error of the robot is increased as well, as the result achieved in a simulated environment in section 3.2 has indicated.

5.2.3. Path Planning

For a personal robot providing services to its user(s) one of the most important capabilities is to reach a given goal from the current robot's pose. This is exactly the task of path planning. On the basis of the built environment representation during the interactive *Home-Tour*, a planning approach adopting human navigation strategies has been proposed in section 4.2. The results achieved in an empirical study have been discussed in section 4.2.3. compared with the map-based approach NF1 the path planning and re-planning capabilities of the proposed approach have been validated on the map built in a guided tour.

In contrast to the empirical study in which the robot was guided by a person who was familiar with the robot system, the proposed path planning approach was performed in the user study conducted by the subjects who used the robot platform for the first time. Furthermore, instead of the direction-following behavior (see section 2.4.2) chosen in the empirical study, the adaptive context-aware following behavior discussed in section 2.3 was used in all eight trial runs to build the environment representations. Besides, the map built in the empirical study was not reused. Therefore, as a complementary study the benefit of the combination of the map- and person's trajectory-based approach is the focus of the evaluation under the assumption that the environment is static and known to the robot. On the one hand, exploiting the knowledge gained from an analysis of the person's trajectory, human navigation strategies would be beneficial to the robot. There should be no unconquerable obstructions along the path of the robot for it to reach its goal, since this path was computed from the person's trajectory. Especially when the obstacles on the way are unperceivable, the pathways of the guide-person are valuable to the robot. On the other hand, as the person's trajectory cannot cover all free space of the environment, map-based approach may be necessary suppose the robot is far from the pathways of the guide-person. The robot may get into trouble, when the path can be computed only on the basis of the person's trajectory.

With the help of the component *SLAM and GLOC* illustrated in Fig. 5.3 the map of the surroundings can be reused. The robot is able to localize itself in the environment map. Starting the map built in the user study on the simulation platform *Stage*, a path planning task can be performed on the built map. Thus, the proposed path planning approach can be compared with the map-based approach NF1 on *Stage*.

In order to assess the capability of the proposed map- and person's trajectory-based approach in performing a path planning task, first, experimental data collected in the user study are introduced. On the basis of the experimental data the proposed path planning approach can be assessed on the simulation platform *Stage*. Afterwards, the map built in one experiment of the user study is reused. The robot needs to localize itself on this map. The necessity of the combination of the map- and person's trajectory-based approach is shown before a discussion of the study.

Experimental data

The proposed path planning approach is based on the trajectory of the guide-person. Hence, positions on the person's trajectory in the user study were recorded. Note that the recorded person's positions were transformed into the respective global coordinate frames defined by the map built in each experimental run. *Locations*, i.e. kitchen, hallway, etc., illustrated in Fig. 5.2, indicated by the guide-person were recorded on the person's trajectory as well. As discussed in section 4.2.1 a graph representation with multiple layers can be created

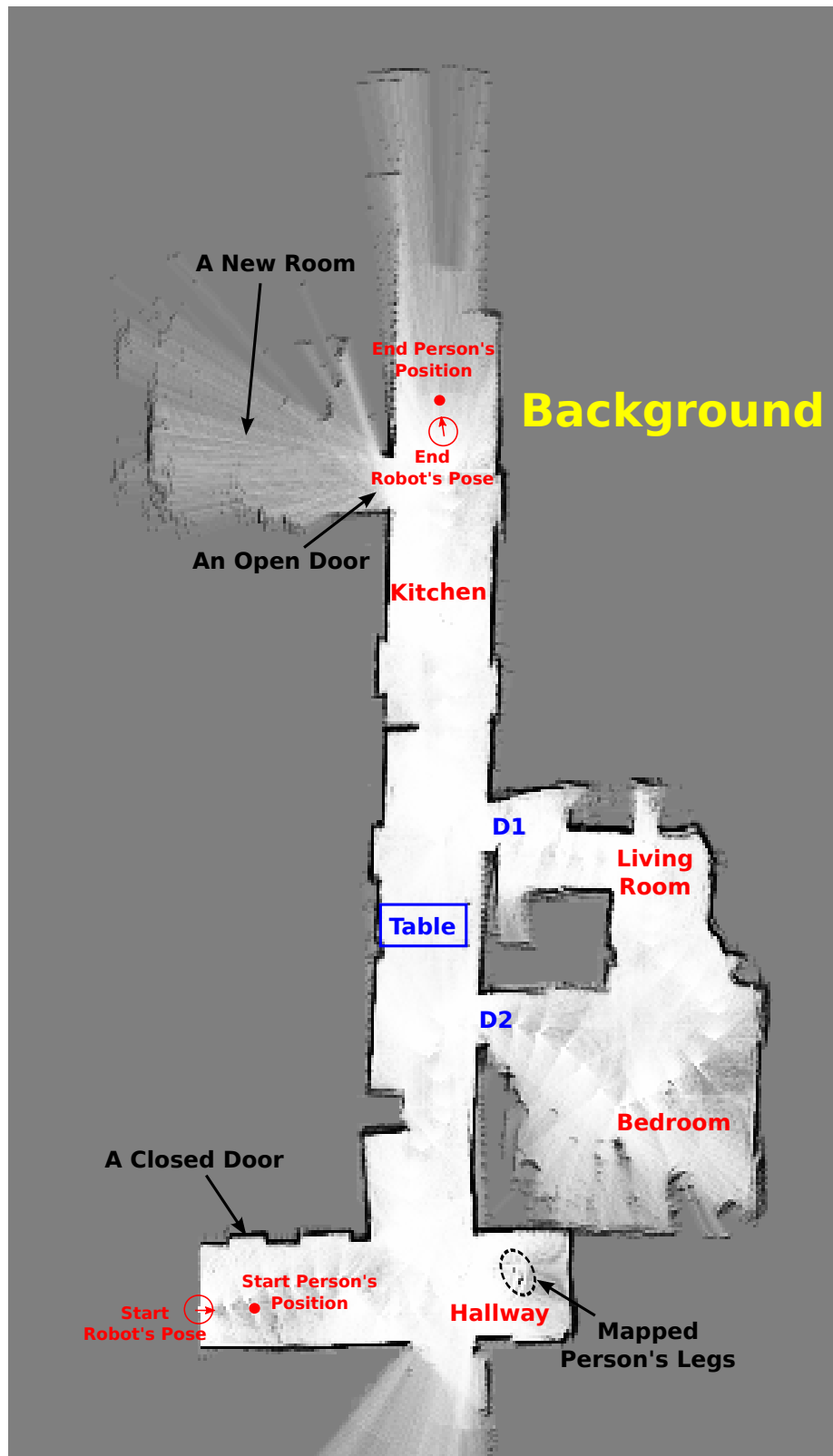


Figure 5.16.: Map built during the fourth experiment. An open and a closed door, a new room compared to Fig. 5.9, as well as the mapped person's legs are marked with black arrows. The start and end position of the person, the start and end robot's pose, and the four locations shown to the robot during this experiment are marked. One table and two open doors D1 and D2 are illustrated as well.

on the basis of the person's trajectory and the indicated *locations*. Graph nodes, namely the positions on the person's trajectory, corresponding graph layer assigned to each node, timestamp of each node, edges of the graph nodes, as well as the weights of the edges computed with equation 4.5, 4.6 and 4.7 need to be stored. In summary, a time-stamped sequence of person's positions (graph nodes) expressed in the respective built maps, *Locations* shown to the robot, corresponding graph layer assigned to the nodes, edges as well as edge weights are recorded as experimental data for all trial runs. Since the map built in each experimental run has been stored for the assessment of the mapping process (see the last section), the proposed map- and person's trajectory-based path planning approach can be performed on the simulation platform *Stage* with the collected experimental data.

Global localization

Starting the map built in one experiment of the user study on *Stage*, the robot needs to localize itself on this map. Fig. 5.16 shows the grid map built in the fourth experiment. Compared with the map built in the second experiment (see Fig. 5.9), the positions of the two doors are changed. Because of the open door marked with the black arrow, some free space of the new room can be perceived by the robot. The mapped person's legs are illustrated inside the dashed ellipse. The recorded start and end position of the guide-person, as well as the four *locations* indicated to the robot in the fourth experiment are marked on this map. Notice that a table about $1.60m \times 0.80m \times 0.72m$ is depicted with a blue rectangle in the corridor region, which was laid in the environment. Only the four legs of the table can be perceived by the robot, due to the limited 2D perception of the laser at a certain height. The two doors D1 and D2 were open during this experiment. The background indicating the areas not perceived by the robot is shown as well in Fig. 5.16. Note that the built maps as well as the trajectories of the guide-person are recorded for all the eight experiments. The fourth experiment is chosen as an example to show the capability of the proposed path planning approach. The same results can be equivalently achieved from any other experiment.

Since the proposed path planning approach can be performed on the simulation platform *Stage*, the robot needs to firstly localize itself in the map, and then navigate to a given goal. In this study the robot is started in the room which the robot did not enter in the fourth experiment. With the help of *GLOC* combined with the mini-exploration, the robot is stopped when it localizes itself in the map. The environment map built in the fourth experiment (see Fig. 5.16) is started on *Stage*, as shown in Fig. 5.17. The start and end pose of the robot is illustrated in this figure as well. Due to the requirement of a map started on *Stage*¹⁰, the background illustrated in Fig. 5.16 has to be removed. The foreground needs to be extracted, and thus details of the environment map are not all shown in Fig. 5.17, e.g. the legs of the table laid in the vertical aisle. However, the influence of the trivial change in the environment map can be ignored by *GLOC*. As the size of the map is consistent with the real surroundings, the robot simulated on *Stage* can perceive approximately the same as what it shall sense in the real environment.

GLOC is performed due to the detected environment map. Fig. 5.18 shows the distribution of particles recorded at three time steps. In terms of Fig. 5.17(a) the robot's start position is marked in Fig. 5.18(a). As the robot is globally uncertain in the start position, the samples are uniformly distributed through the free space of the map. Fig. 5.18(b) illustrates the distribution of particles, when the robot moves approximately 1.5 meters with respect

¹⁰See details in the web page <http://www-users.cs.york.ac.uk/~jowen/player/playerstage-manual.html>.

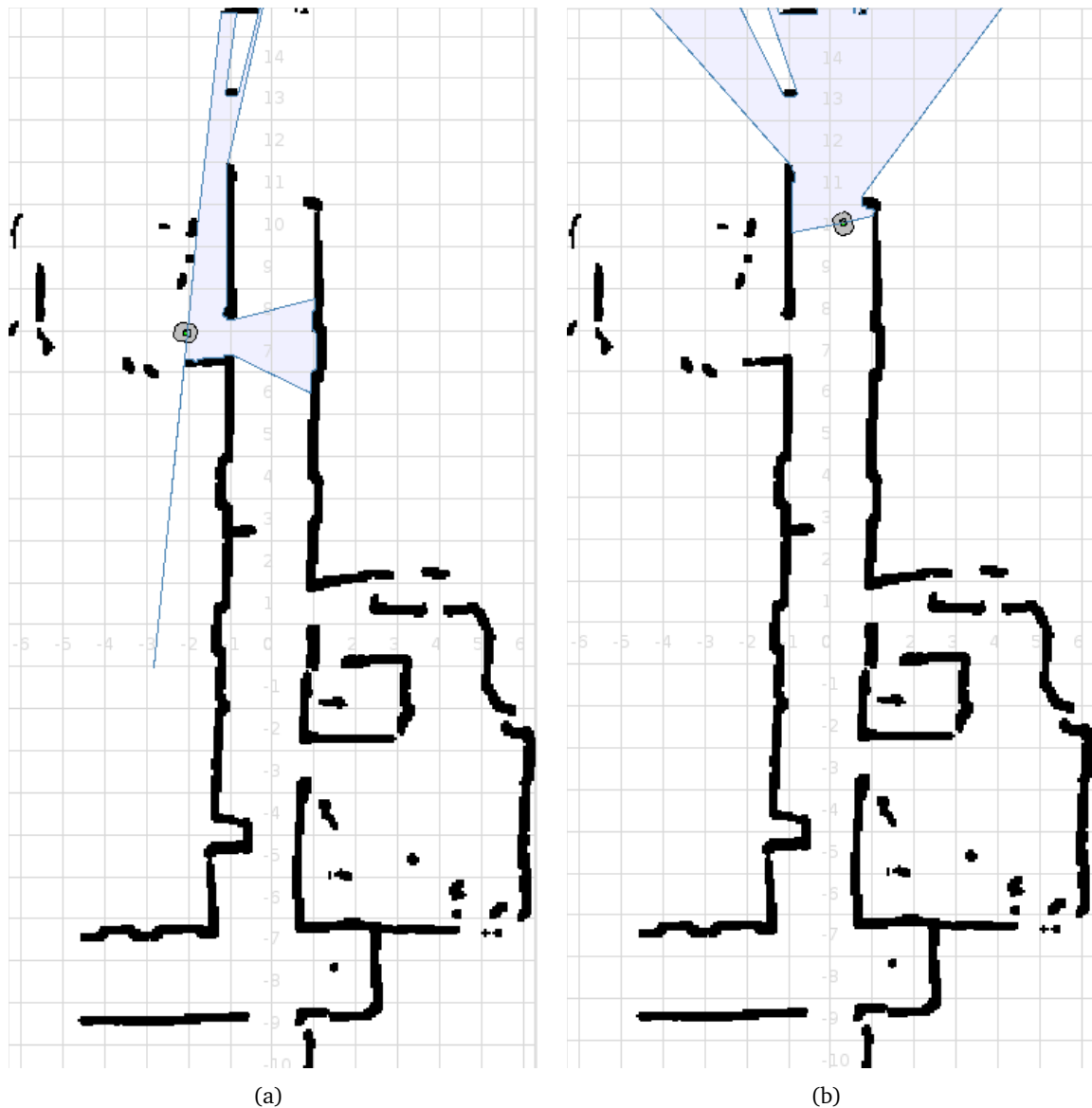


Figure 5.17.: The start and end pose of the robot localizing itself on the map built during the fourth experiment. The robot is started in the room which it did not enter during the guided tour, as shown in a). Performing the mini-exploration introduced in section 5.1.2 the robot ends at the location illustrated in b).

to the start robot's position. *GLOC* has disambiguated the robot's position according to the perceived sensor data, so that most particles center around three positions. The true and false estimated robot's positions are marked as well with the black arrows. Finally, the ambiguity is resolved after another 4 meters of robot motion in the vertical aisle. As shown in Fig. 5.18(c) the robot knows where it is. Compared to the end pose of the robot illustrated in Fig. 5.17(b), the majority of particles are now centered tightly around the correct position. Note that in the initialization of *GLOC* a large number of particles (40,000 in our case) are uniformly distributed in the environment. Nevertheless, with the help of KLD-sampling [47] a small number of samples are chosen if the density is focused on a small part of the state space. Otherwise, a large number of samples will be chosen if the state uncertainty is high. The computational complexity can be therefore drastically reduced,

once the robot's pose is disambiguated by the mini-exploration. Besides, the estimated robot's pose can be updated more frequently due to a small number of samples.

Person's trajectory-based approach vs. map-based NF1

As illustrated in Fig. 5.18(c) the robot has localized itself on the environment map. A path planning task can be performed now. In order to assess the capability of the combination of the map- and person's trajectory-based approach, the robot is driven back in close proximity to the robot's start position shown in Fig. 5.17(a) with a joystick integrated in *Stage*. As the recorded person's trajectory does not cover the room in which the robot is started, map-based path planning is necessary for the robot to navigate to a goal defined in the built map. *Location* hallway marked in Fig. 5.16 is set as the goal for the robot. The task of path planning is performed by the proposed path planning approach. The necessity of the combination of the map- and person's trajectory-based approach can be therefore shown.

As the robot is started far from the person's trajectory, map-based NF1 needs to be used. The path from the start robot's position to the hallway is shown with the purple curve in Fig. 5.19(a). Notice that the path based on the trajectory of the guide-person is adopted, as soon as a plan from the current robot's position to the set goal can be computed from the graph built with multiple layers (see section 4.2.1). In other words, the path from this robot's position to hallway is computed by the person's trajectory-based approach. The red curve in Fig. 5.19(b) illustrates this path. Fig. 5.20(a) shows the red path as well as the

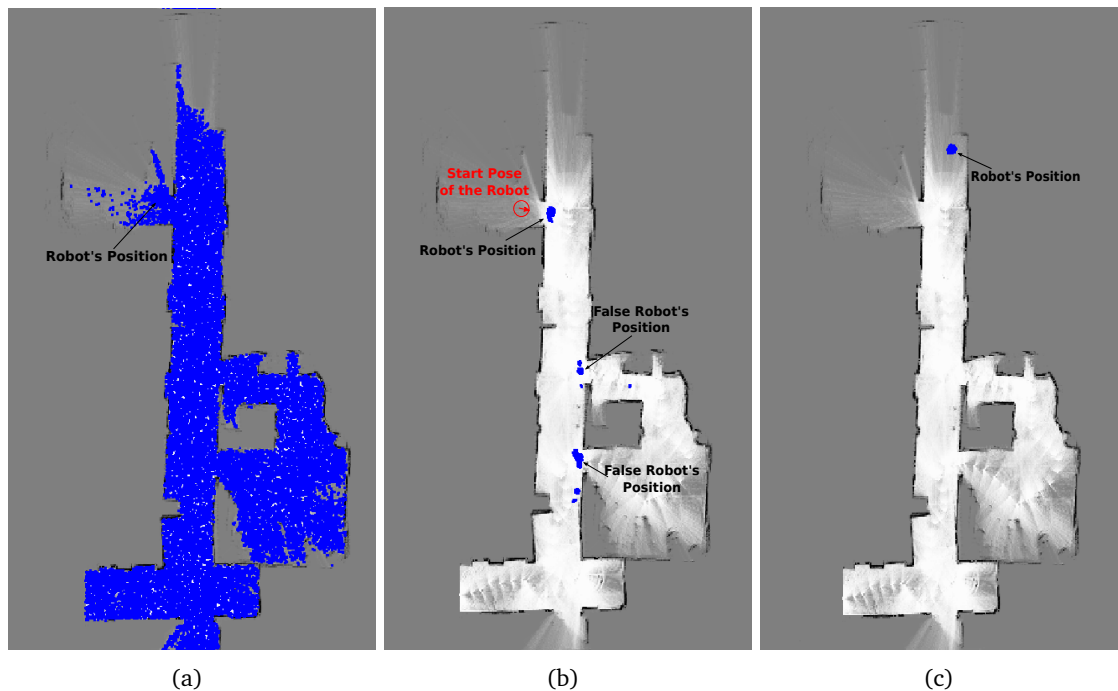


Figure 5.18.: Global localization on the map built in the fourth experiment. a) at the beginning the set of particles (40,000 samples in our case) is initialized with a uniform distribution (projected into 2D). b) Most samples are centered around three doorways after approximately 1.5 meters of robot motion with respect to its start position. c) the robot knows where it is after another 4 meters of robot motion in the vertical aisle.

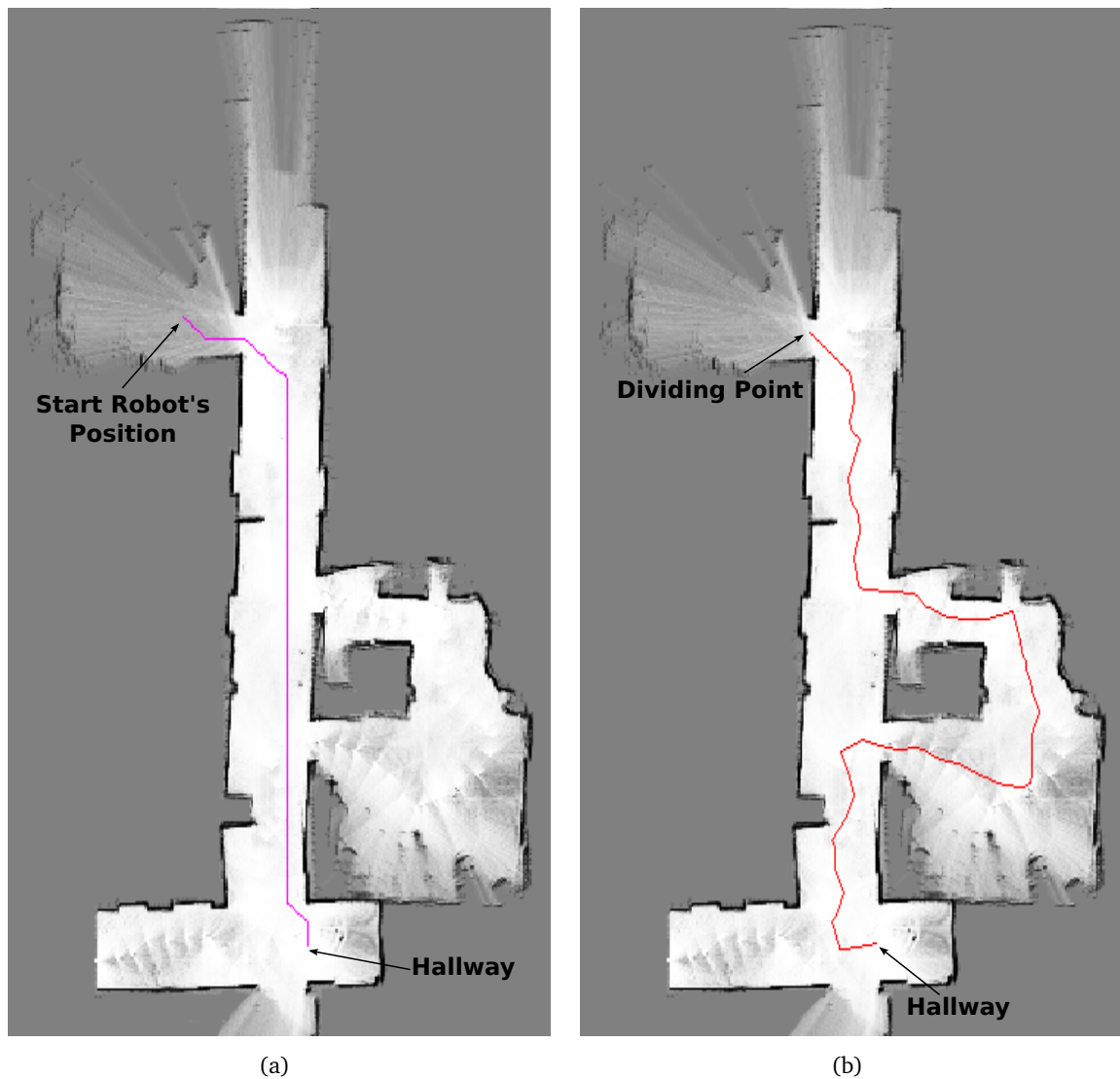


Figure 5.19.: The NF1 path and the path planned on the basis of the person's trajectory. The purple curve in a) shows the NF1 path from the robot's start position to the given goal hallway. The red curve in b) illustrates the path based on the trajectory of the guide-person. This red path found from the built graph begins at the dividing point and ends at the hallway.

graph with different layers built from the person's trajectory. As discussed in section 4.2.1, a new graph layer depicted with a different color will be created when a new location is indicated to the robot. The two graph layers beginning at the start person's position and the last location (the kitchen in our case) and ending at the first location (the hallway in our case) and the end person's position respectively are depicted as well.

Note that path P computed by the proposed approach consists of two parts. Fig. 5.20(b) illustrates the path P actually planned for the robot. The purple part indicates the NF1 path adopted by the robot at the beginning, while the red part used by the robot afterwards is planned by the person's trajectory-based approach. The robot's position at which the path to the hallway can be computed on the basis of the person's trajectory is regarded as the dividing point of the two parts of path P . Because of the discrepancy between the planned

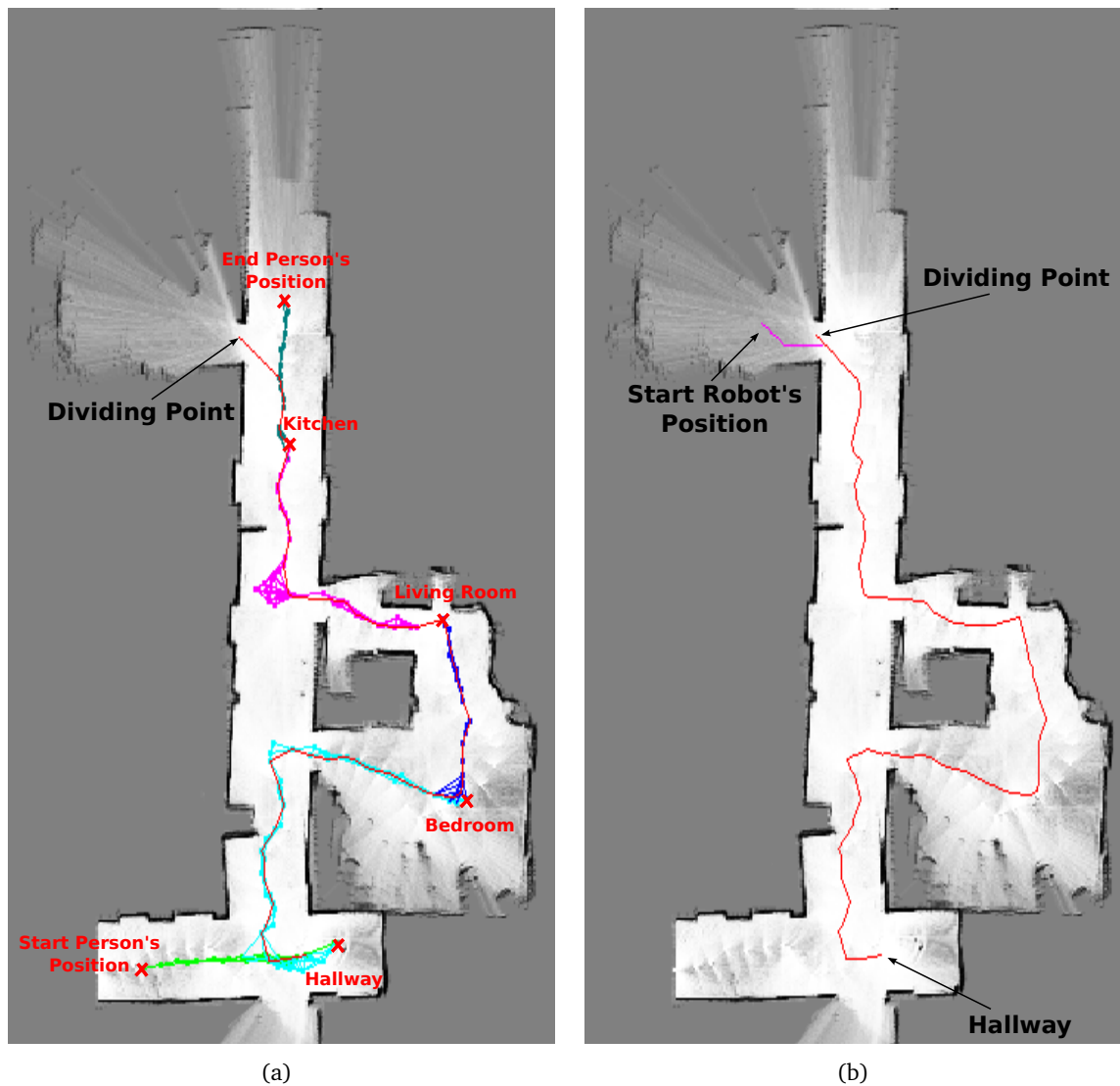


Figure 5.20.: The graph built from the person's trajectory and the actual path computed using the approach discussed in section 4.2. a) shows the created graph and the corresponding path computed according to this graph. The red path has been already illustrated in Fig. 5.19(b). The start and end position of the guide-person, as well as the four locations shown to the robot are marked. The path actually planned for the robot is illustrated in b). Two parts of this path are shown with the purple and red curve. The dividing point of the two parts is marked with the black arrow.

path and the actual robot's trajectory, the dividing point shown with the black arrow in Fig. 5.20(b) is slightly different from the end point of the purple one. The *location* hallway marked in Fig. 5.20(a) records the person's position in which the guide-person indicated the hallway to the robot. However, the goal sent to the robot is the robot's position in which the indication of a *location* was accepted (replied) by the robot¹¹. Therefore, the planned red path to the hallway ends at the position distinct from the *location* marked in

¹¹See the mini-dialog module introduced in section 5.1.2 for details.

Fig. 5.20(a)¹².

Notice that the NF1 path shown in Fig. 5.19(a) leads the robot to *location* hallway through the vertical aisle. In contrast to this NF1 path the robot has to make a detour when following the red path shown in Fig. 5.19(b). However, this detour is crucial for the robot. The necessity of the switch to the person's trajectory-based path, i.e. the red path shown in Fig. 5.19(b), is illustrated in Fig. 5.21. On the one hand, the table placed in the vertical aisle of the environment cannot be perceived by the robot except for the four legs. A crash to this table is therefore inevitable when completely using the NF1 path. On the other hand, this table has been perceived by the guide-person during the *Home-Tour*. Hence, the robot can reach the given goal when passing through the two doors D1 and D2. In other words, the NF1 path consisting mostly of straight line segments generally provides a shortest way to the goal, while navigation strategies of the guide-person, e.g. avoiding obstacles unperceivable to the robot, going through the two doors D1 and D2, are represented with the red path. The real movement trajectory of the robot should be a little different from (probably smoother than) this red path, as the robot's goals are selected from this path according to a predefined distance threshold.

Discussion

The results so far show the capability of the proposed approach in performing a path planning task under the assumption of a static environment. The necessity of executing both the map- and the person's trajectory-based approaches has been shown in a real scenario. Due to the limitation of robot perception, following the pathways of the guide-person human navigation strategies are beneficial to the robot. It is crucial for the robot especially when obstacles are not perceivable.

According to results discussed in this section, pose estimation of the robot was able to converge at the correct position when mini-exploration was applied. However, the problem of efficient localization discussed by Fox et al. [49] should be considered. By this means, the robot tries to actively navigate and sense in the environment, in order to increase the efficiency and robustness of localization. Apart from human navigation strategies, moving on *conventional* pathways of the guide-person the robot's path would be expected as well by this person. Moreover, the movement trajectory of the robot would follow a predictable pattern, as it is the path that people would like to take. Nevertheless, how to exploit person's trajectories needs to be investigated more thoroughly due to the individuality and variety of human pathways. Appropriate movement patterns adaptive to spatial context are expected to be learned from the guide-person in the interactive *Home-Tour* and exploited by the robot.

5.3. Observations from a Subject's Perspective

As mentioned at the end of section 5.1.1, the subjects walked in a forward-, a backward- or a mixing-manner during the guided tour. The five subjects who watched the robot and walked backwards during the guided tour were asked about this. Their answers were not completely same though, the reasons were consistent. As they did not know how the robot would move, they had to walk carefully and chose a backward-manner in order to see the robot clearly. Hence, the majority of the subjects (62.5% in our case) preferred a walking

¹²We choose the robot's position as a goal, as the robot has really reached that position in the guided tour.

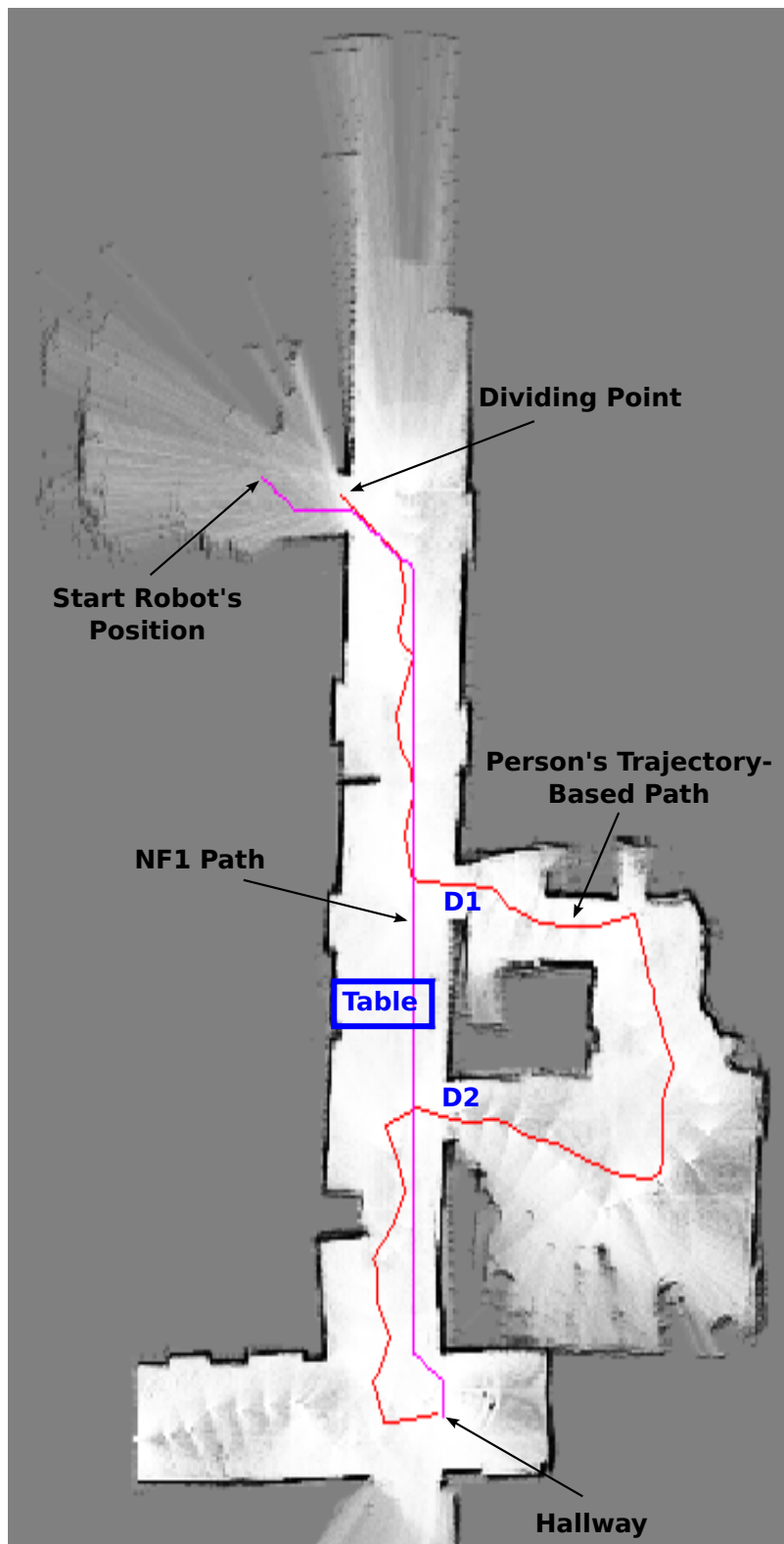


Figure 5.21.: Comparison of the map- and the person's trajectory-based path illustrated with purple and red respectively. The table and the two open doors D1, D2 are depicted as well.

manner different from the one usually adopted in a human-human interactive scenario, i.e. walking in a forward-manner. For the rest three subjects S_8 of the eighth trial run walked forwards at the beginning. As he moved much faster than the robot, he then looked at the robot and carefully went backwards. In contrast, S_2 and S_4 walked all along relatively slowly, and their speeds were comparable to the robot¹³. They both went forwards in the respective trial run except for a few necessary adjustments, e.g. S_2 paused for a moment to make the robot keep up with him, while S_4 turned her head to see the robot now and then. In summary, all subjects were able to successfully guide the robot in the environment and show *locations* given in Fig. 5.2, while paying attention to robot's movement.

5.4. Insights and Discussion

Based on the idea of *Human-Robot Joint Spatial Exploration* an integrated system that enables a mobile robot to acquire a spatial representation in an interactive scenario has been built. The assessment of the individual components presented in section 5.2 has validated that they are appropriate for the integrated system working in a real environment with real subjects. Since the mobile robot platform has been evaluated in the user study conducted by inexperienced subjects, the feasibility and practicality of the proposed system as a whole can be verified from a technical point of view.

The presented system starts from the process of adaptive context-aware person-following making the robot keep up with the guide-person, and ends at building an environment representation. Therefore, the three components constructing the core of the system interact with each other. Firstly, the person-following process influences the mapping and localization (SLAM) process, as a metric map can be built when the robot moves around in the environment. Moreover, the path planning approach is based on the trajectory of the guide-person observed in the person-following process. Next, the robot's goals sent to the person-following process as well as the person's trajectory observed by the robot are computed on the basis of robot's poses estimated in the built map. Besides, the proposed path planning approach and the integrated global localization approach *GLOC* are influenced as well by the built environment representation. Thus, insights gathered from the user study and the performance of the integrated system as a whole should be discussed in three aspects, namely map building, person-following and path planning.

Note that building an environment representation in an interactive scenario is different from a pure mapping process, as the robot is guided by a person. Robot movement and perception during the exploration of the initially unknown environment are dependent on the guide-person. Despite *Person lost* that occurred in the *Home-Tour*, the task assigned to the subjects in all trial runs was able to be accomplished with the help of the robot's responses sent by the *Mini-DLG* module. Compared with the ground truth map built by controlling the robot with a joystick, the robot was not able to perceive the new environment so thoroughly and repeatedly in a guided tour. In order to make the robot perceive the environment in detail as far as possible, the maximum speed of the robot was limited at a relatively low level. The certainty of the map shown in Fig. 5.1 is higher than the one shown in Fig. 5.9. Nevertheless, the feasibility of the map built during the *Home-Tour* has been validated to perform a path planning task.

The negative impact arising from the detected person's legs can be reduced as far as possible, when removing the false information from the built map. Therefore, learning dynamic

¹³The maximum translational and rotational speed of the robot is set to $0.4m/s$ and $25^\circ/s$ respectively.

objects in a mapping process would be beneficial to the built environment representation and pose estimation of the robot in this representation. One possible disadvantage for the mapping process should be pointed out, i.e. perception area of the robot has to be reduced because of the invalid sensor area occupied by the guide-person. However, the robot can perceive the removed area as soon as the guide-person moves.

Considering the statistics computed from the data collected in the user study, the adaptive context-aware following approach appears quite promising in the interactive (*Home-Tour*) scenario. Using the adaptive context-aware following behavior proposed in this thesis the robot was able to successfully keep up with the guide-person in all experimental runs, so that the task assigned to the subjects were all accomplished. All behavior changes of the robot can be explained in terms of the core strategies given in section 2.3.1. The capability to keep up with the guide-person and the relation between the spatial context and the selected following behaviors has been discussed in study I and study II respectively. The competence of the proposed following behavior in accomplishing the interactive *Home-Tour* has been proved by the assessment.

The core strategies of the person-following behavior are based on the spatial context currently perceived by the robot. By this means, the robot is able to change its behavior according to the most current information gained from the environment. However, the knowledge that has been obtained from the environment, i.e. the built map, may become another basis of changing the robot's behavior. Especially when the interactive *Home-Tour* continues and the robot enriches its knowledge gained from both the environment and the guide-person, a more comprehensive strategy is expected. In other words, the complete knowledge learned in the *Home-Tour* should be considered for the selection of robot's behavior.

Note that the position computed around the guide-person controls the movement of the robot in parallel-following. The movement direction of the person is implied in the direction of the vector consisting of two consecutive robot's goals. Except for the computation of robot's goals, robot speed can be computed in terms of the estimated moving speed of the guide-person. Hence, how to control both the position and the speed of the robot to perform parallel-following still needs further investigation. Besides, a number of issues remain to be investigated when taking social aspects of human-robot interaction into account. A more thorough study is therefore necessary to evaluate the acceptance and benefit of the proposed multiple person-following behavior from a user's perspective. Adaption of the person's and the robot's behavior might be evaluated as well in such an interactive person-following scenario.

As a complementary study to what have been done in section 4.2.3, the proposed path planning approach was evaluated on the map built with the person-following approach discussed in section 2.3. To perform a more general path planning task the robot was started in a region which was not entered but merely perceived by it. Although the feasibility of *GLOC* has been verified in the study, the capability of recovering from localization failures is necessary for more complex scenarios. Besides, considering dynamic obstacles generally existing in a real environment re-planning is still an open issue. As discussed in section 4.2.3 the proposed re-planning approach will be triggered when dynamic objects, such as doors can be recorded in the mapping process. Therefore, apart from a static environment representation built in the *Home-Tour* a mapping process gathering current information from the environment might be useful to deal with dynamic obstacles. Extending the perception area of the robot, e.g. integrate the Time-of-Flight camera or make use of the Microsoft Kinect mentioned in section 4.3, more potential hazards can be perceived by the robot. However, due to the limitation of robot perception, how to deal with unexpected and un-

perceivable objects outside the perception area of robot sensors in path-planning needs further investigation.

6. Conclusion and Future Considerations

The important thing is not to stop questioning.
[Albert Einstein, 1879–1955]

A long term goal of research in robotics has been to make robot that would provide services to human, thereby, improving our daily lives. For such a personal service robot, its capability to acquire an environment representation is fundamental and necessary to be developed. The ability of a robot to harmoniously coexist and work in a human-shared environment is possible only when the built representation corresponds to human spatial understanding. To acquire a spatial representation, thus to explore the environment, it is rather natural for the robot to accompany the human user while the user presents the environment to the robot. In such a human-robot joint exploration, the human spatial knowledge is incorporated in the environment representation built by the robot. Consequently, based on the idea of *Human-Robot Joint Spatial Exploration* (HRJSE) this thesis focused on an integrated solution to the problem of interactive acquisition of spatial representations.

6.1. Summary

Starting from this idea, the interactive *Home-Tour* was chosen as the basic scenario in this research to realize a human-robot joint exploration. To acquire an environment representation in the *Home-Tour*, the robot needs to keep up with the correct guide-person while learning models of the environment. Thus, the train of thought of acquiring a spatial representation formed gradually. Three integral parts, i.e. person-following, map building during the *Home-Tour* and path planning adopting human navigation strategies, were integrated in the system. By this means, a topological representation of the environment can be built incrementally. Information gained from both the surroundings and the guide-person can be incorporated in the representation as well.

The person-following behavior dealing with the problem “*where to move in the human-robot joint exploration?*” was proposed at first for the robot to complete the *Home-Tour* conducted by a guide-person. Three elementary person-following methods, namely direction-, path- and parallel-following, were designed as the components constituting a more complicated behavior, i.e. the adaptive context-aware following behavior. This hybrid following approach derived from the observation of human-human following schemes enables the robot to switch between these elementary following methods according to the current spatial context, e.g. free space, and the relative position of the interacting person. Keeping up with the guide-person is regarded as the guideline of the strategies of behavior switch, bearing the characteristics of the three elementary behaviors in mind.

Given the adaptive context-aware following behavior, the robot was assumed to be able to keep up with the guide-person and perform the *Home-Tour*. Next, building a metric map

of the environment during this guided tour was discussed. Considering the importance of mapping details and recording dynamic objects, occupancy grid map is chosen as the basic environment representation. It is meaningful, especially when the robot performs navigation tasks. The guide-person perceived by the robot in the *Home-Tour* has to be regarded as dynamic obstacles and mapped on the grid representation. In order to reduce this negative impact on the mapping process, the areas occupied by the person are detected and marked as invalid on the laser plane. Thus, the really useful information obtained from the surroundings is available for the mapping process, and the disturbance of the guide-person can be restrained.

On the basis of the metric environment representation built incrementally in the *Home-Tour*, both the robot's poses and the person's positions can be expressed in a common world coordinate system. Afterwards, a path planning approach was presented considering human navigation strategies that can be extracted from an analysis of the movement trajectory of the guide-person. The points selected from the observed human pathways form the nodes of the graph with multiple layers. New graph layer is created once a new *location* has been indicated by the guide-person. Therefore, a topological representation can be built upon the metric environment map. The environment is represented with separate *locations* forming the topological nodes. These nodes are connected by the paths found in the graph. Re-planning is triggered, when dynamic objects obstructing the path already planned for the robot are detected on the occupancy grid map. Besides, a multisensor-based motion generation enabling a 3D capable navigation for the robot was proposed. It is compatible with the current robot platform due to the same two-dimensional data input of path planning and obstacle avoidance.

The integrated solution proposed in this thesis to the problem of acquiring an environment representation was evaluated in a user study conducted by the subjects who used the robot platform for the first time. The user study was conducted in a real environment resembling an apartment with dynamic objects, such as movable doors, and passers-by. In order to prove the robustness of the system as a whole, no explanation of the hardware and software of the robot platform as well as the goal of the experiments was told to the subjects, except for the brief task introduction shown in Fig. 5.2. Various spatial arrangements were thought about in the test environment, e.g. aisles, rooms with narrow and open space, etc., so as to assess the designed person-following behavior.

All necessary data (see section. 5.2) were recorded in each experimental run. Hence, the complete user study can be replayed offline, and different environment representations built in the experiments are able to be compared with each other. A mini-dialog module was designed to facilitate the communication between the robot and the guide-person in the *Home-Tour*. With the help of *GLOC* combined with the mini-exploration strategy the robot is capable of localizing itself globally in the built map. By this means, the built spatial representation can be reused by the robot started at an arbitrary position in the free space illustrated on the environment map. The robustness and feasibility of the system has been as a whole verified by the successful performance of the task assigned to all inexperienced subjects.

6.2. Concluding Discussion and Future Considerations

The general idea of *Human-Robot Joint Spatial Exploration* provides an opportunity for a mobile robot to acquire spatial representation with the help of the human user. Especially

for a personal service robot sharing an environment with human inhabitants and performing tasks given by the user, it is necessary to take into account human spatial understanding and concepts in the acquisition process of the spatial representation. By this means, communication about the environment between the robot and user can be facilitated. In the interactive *Home-Tour* scenario the robot is guided by the user through an arbitrary indoor environment. It is able to explore the environment and learn about the surroundings. Not only the information obtained from the raw sensory data but also the spatial knowledge gained from the human user can be incorporated in the built spatial representation. Thus, direct participation of the human user into the environment exploration of the robot is regarded as a solution of map acquisition. Human spatial understanding may be provided by a human user explicitly with language, e.g. the user presents the environment to the robot, or implicitly contained in human behaviors, e.g. the movement trajectory of the human user observed by the robot.

A person-following behavior enables the robot to keep up with the guide-person during the interactive *Home-Tour* and move around in the environment. The unknown environment is therefore explored by the robot. Afterwards, a metric environment map is built, keeping in mind the negative impact of the guide-person on the mapping process. Besides that, on the basis of the built metric map an environment representation incorporating human spatial knowledge can be achieved in the *Home-Tour*. According to this environment representation navigating to a given goal can be solved as well. Hence, in this thesis a person-following behavior, a mapping process and a path planning approach were assumed as the basic aspects for a system design to realize *Human-Robot Joint Spatial Exploration*, particularly for indoor applications on the basis of the *Home-Tour* idea.

A person-following behavior A person-following behavior is generally used for a mobile robot in an interactive scenario, e.g. the *Home-Tour*. The ability of the proposed multiple person-following behavior to accomplish the interactive *Home-Tour* has been verified in the user study. Evaluation results achieved from the analysis of the data collected from all trial runs have proved the behavior switch according to spatial context. Furthermore, a relation between the selected behaviors and the arrangement of the environment can be found, as discussed in section 5.2.1.

Despite the encouraging results achieved in the user study major challenges for the adaptive context-aware following approach still lie in the following questions: (i) how to extract an appropriate description of a person-following behavior, and (ii) except for an assessment from a technical point of view, how to evaluate the performance of the designed person-following behavior from a user's perspective.

As a final evaluation of the following behavior should be given by the human user, a wide and detailed investigation is needed. By means of the further investigation, a more definite design for a person-following behavior could be obtained, and a robot's behavior preferable to the user is expected to be achieved. Apart from building an accurate theoretical model, the real performance of the robot behavior needs to be taken into account especially for a complex mobile robot platform working in a real world scenario. It directly affects the assessment of the human user. Hardware control, e.g. robot speed, the reliability of the communication between software components and robot sensors, the errors arising from the individual hardware and software components, etc., can influence the accuracy and the smoothness of the designed robot behavior. However, as summarized in section 5.3 people probably pay close attention to the robot behavior. Therefore, the approach proposed in this thesis can be regarded as a step towards the person-following behavior that is able

to facilitate human robot interaction in a real world scenario and take into account the situations in the environment.

Mapping in an interactive scenario Exciting progress has been achieved in the field of robotic mapping. A mobile robot is able to create and update a metric map of an unknown environment. The built map is generally based on the extraction of features, e.g. points, lines, objects, etc., from the environment or a free space representation, such as the occupancy grid map adopted in this thesis. However, robotic maps do not necessarily correspond to human spatial understanding and concepts. A special issue that needs to be considered in a mapping process during the *Home-Tour* is the guide-person who has to be regarded as a dynamic obstacle. The Markov assumption, sometimes referred to as static world assumption is therefore violated. Using the technique proposed in section 3.2 the negative impact on the built metric map can be reduced as far as possible. In addition, without the need of feature extraction often dependent on the assumption that an environment exhibits an known structure, the occupancy grid representation obtained directly from robot sensors shall be more robust in cluttered indoor environments. The free space representation of the environment is especially useful to plan a path for the robot.

Although the results achieved in the user study do not provide negative impact of dynamic objects on the proposed mapping process, mapping in dynamic environments is still a challenging issue. Except for the tracked guide-person, other passers-by, doors and movable furniture in an indoor environment may violate the assumption of a static world as well. Hence, learning models of moving objects is meaningful for a personal service robot to safely and successfully navigate in a human-shared environment, while effectively performing tasks assigned by human user(s).

As a step towards spatial understanding in 3D space, recently some mapping techniques have attempted to create a three-dimensional spatial model for an unconstrained environment. These approaches may be crucial especially when the environment is composed of several floors lying one above the other, or it contains non-flat structures like ramps. A thorough 3D representation is not necessarily needed for a robot moving in a 2D environment. However, objects out of the 2D robot perception area should be taken into account, since safe navigation may fail due to the unperceivable obstacles, as discussed in section 4.3. The ability of 3D perception is important for the service robot, as the objects that would be manipulated by the robot performing a task exist indeed in a three-dimensional space.

In addition, adopting exploration strategies, e.g. actively closing loops, re-entering visited places, exploration for unknown terrain, etc., may support the robot to gain further information about the environment, especially when the achieved environment models are incomplete and the environment changes over time. The results achieved in the user study have verified the practicality of the maps built in the different trial runs in performing the proposed person-following behavior and planning a path for the robot. Nevertheless, it would be necessary to use such exploration strategies to inspect errors in a mapping process and improve the quality of the map especially in large scale environments. The *Home-Tour* should not be negatively influenced, when the corresponding explanation of the robot's behavior can be conveyed to the guide-person in time.

Incorporating human spatial knowledge Integrating human spatial knowledge about the surroundings into a general robotic mapping process may enrich the metric representation including robot centric information. In the context of personal service robots an

environment representation containing information gathered from raw robot sensor data and human spatial understanding can facilitate the communication about the environment between the user and the robot. This environment representation is therefore expected to be beneficial to the robot providing services to its user.

An interactive scenario would be helpful for both the robot and its user, as on the one hand the robot may learn about the environment with the help of the user during this scenario. On the other hand, it provides an opportunity to the user to interact with the robot and know the new friend. Information conveyed by natural language, such as an introduction of various rooms, aisles, objects belonging to these introduced regions, etc., contains spatial understanding and concepts of the human user(s). Besides, human behaviors, e.g. gestures like fingertip pointing, a nod or hand touch, perceived human pathways in the interactive guided tour, etc., as well as the corresponding meaning hiding behind these behaviors should be useful for the robot to extend its knowledge about the environment.

Using the concept *location* defined in this thesis information indicated explicitly by the human user can be integrated into a robotic mapping process. Combining the analysis of the observed human movement trajectory a topological environment representation based on a metric map has been built. However, considering tremendous spatial knowledge of humans about the surroundings, how to effectively exploit the multifarious information obtained by the robot in an interactive scenario, and how to extract an appropriate environment model from these information become more challenging for the future work. The ability of interactively acquiring all possible information (language, behavior, etc.) from the user would help the robot to resolve ambiguities and effectively build a spatial representation integrating human spatial understanding. This built environment model would support the interaction with the human user about the environment, and thus facilitate the robot to accomplish tasks assigned by the user.

Future considerations Challenges mainly arising from the limited abilities of the robot in perception, interpretation and representation have been discussed in this section. However, the overall framework presented in this thesis appears promising in solving the problem of interactive acquiring a spatial representation. Research in the directions of recognizing human behaviors in an interactive scenario and questing their implications according to actual situations will hopefully help to materialize a more flexible adaptive system. Inconsistent representation of the surroundings and ambiguities in the environment representation are expected to be avoided for both the human user and the robot. Thus, a spatial representation adequate to real applications in our everyday life could be built by the robot.

A. A Brief Introduction to Particle Filters

Particle filters are a broad and popular class of Monte Carlo algorithms for the solution of optimal estimation problems in non-linear non-Gaussian scenarios. In comparison with standard approximation methods, such as the popular Extended Kalman Filter [80], the principal advantage of particle methods is that they do not rely on any local linearisation technique or any crude functional approximation. The price that must be paid for this flexibility is computational: these methods are computationally expensive. However, thanks to the availability of ever-increasing computational power, these methods are already used in real-time applications appearing in fields as diverse as chemical engineering, computer vision, financial econometrics, target tracking and robotics.

A particle filter is a nonparametric implementation of the Bayes filter and is frequently used to estimate the state of a dynamic system. The basic idea is to represent a posterior by a set S of N weighted random samples (particles), so that the discrete distribution defined by the samples approximates the desired one.

$$S = \{(s^{[i]}, \omega^{[i]}) | i = 1, \dots, N\} \quad (\text{A.1})$$

Where $s^{[i]}$ is the state vector of the i -th sample and $\omega^{[i]}$ the corresponding importance weight. Each sample represents one potential state the system might be in. The sample set approximates the probability distribution

$$f(x) \approx \sum_{i=1}^N \omega^{[i]} \cdot \delta_{s^{[i]}}(x), \quad (\text{A.2})$$

where $\delta_{s^{[i]}}(x)$ is the Dirac delta function concentrating at the state $s^{[i]}$ of the i -th sample and fulfilling the following equations

$$\delta_{x_0}(x) = \begin{cases} +\infty & \text{if } x = x_0 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.3})$$

and

$$\int_{-\infty}^{+\infty} \delta_{x_0}(x) dx = 1. \quad (\text{A.4})$$

$\omega^{[i]}$ is a positive value, and the sum over all weights is 1 as the integral of $f(x)$ over all values x is equal to 1.

$$\begin{aligned} \int_{-\infty}^{+\infty} f(x) dx &= \int_{-\infty}^{+\infty} \left(\sum_{i=1}^N \omega^{[i]} \cdot \delta_{s^{[i]}}(x) \right) dx \\ &= \int_{-\infty}^{+\infty} \omega^{[1]} \cdot \delta_{s^{[1]}}(x) dx + \dots + \int_{-\infty}^{+\infty} \omega^{[N]} \cdot \delta_{s^{[N]}}(x) dx \\ &\stackrel{\text{equation A.4}}{=} \sum_{i=1}^N \omega^{[i]} \\ &= 1 \end{aligned} \quad (\text{A.5})$$

Such a set S of samples can be used to approximate arbitrary distribution. Fig. A.1 depicts distribution $f(x)$ and its corresponding sample set. In general, the more samples that

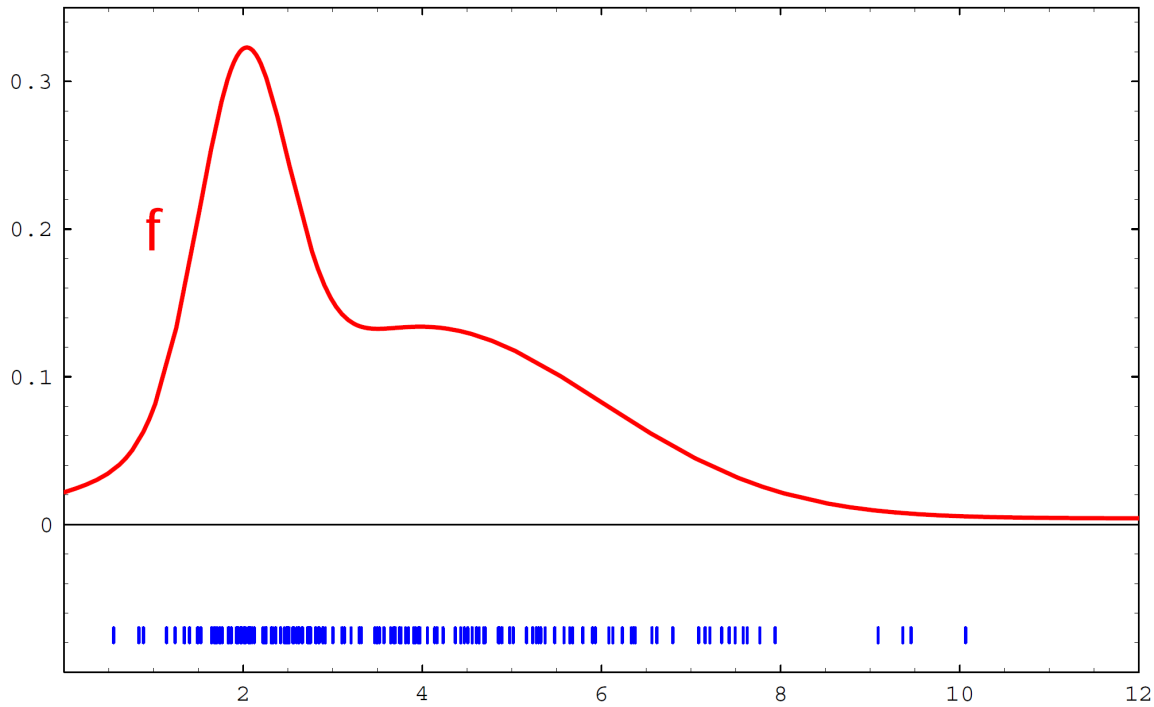


Figure A.1.: The distribution function and its approximation by samples with uniform weights. The samples are illustrated with the vertical bars under the function.

are used, the better the approximation. The problem that is needed to be dealt with is to compute the expectation that $x \in A$, where A is a region. Defining a function B which returns 1 if its argument is true and 0 otherwise, the expectation can be computed as follows:

$$\begin{aligned} E_f[B(x \in A)] &= \int_{-\infty}^{+\infty} f(x) \cdot B(x \in A) dx \\ &= \int_A f(x) dx. \end{aligned} \quad (\text{A.6})$$

Considering the sample-based representation the integral of f over A can be approximated by counting all of the particles that fall into the region A ,

$$\int_A f(x) dx \approx \frac{1}{N} \sum_{i=1}^N B(s^{[i]} \in A). \quad (\text{A.7})$$

Recalling equation A.3 and A.4 the distribution function f can be expressed by

$$f(x) \approx \frac{1}{N} \sum_{i=1}^N \cdot \delta_{s^{[i]}}(x), \quad (\text{A.8})$$

as

$$\begin{aligned}
 \int_A f(x) dx &= \int_A \frac{1}{N} \cdot \sum_{i=1}^N \delta_{s^{[i]}}(x) dx \\
 &= \frac{1}{N} \cdot \int_A \sum_{i=1}^N \delta_{s^{[i]}}(x) dx \\
 &= \frac{1}{N} \cdot \sum_{i=1}^N B(s^{[i]} \in A)
 \end{aligned} \tag{A.9}$$

and

$$\begin{aligned}
 \int_{-\infty}^{+\infty} f(x) dx &= \int_{-\infty}^{+\infty} \frac{1}{N} \cdot \sum_{i=1}^N \delta_{s^{[i]}}(x) dx \\
 &= \frac{1}{N} \cdot N \\
 &= 1.
 \end{aligned} \tag{A.10}$$

Thus, we have obtained an alternative representation of the distribution function f compared to equation A.2.

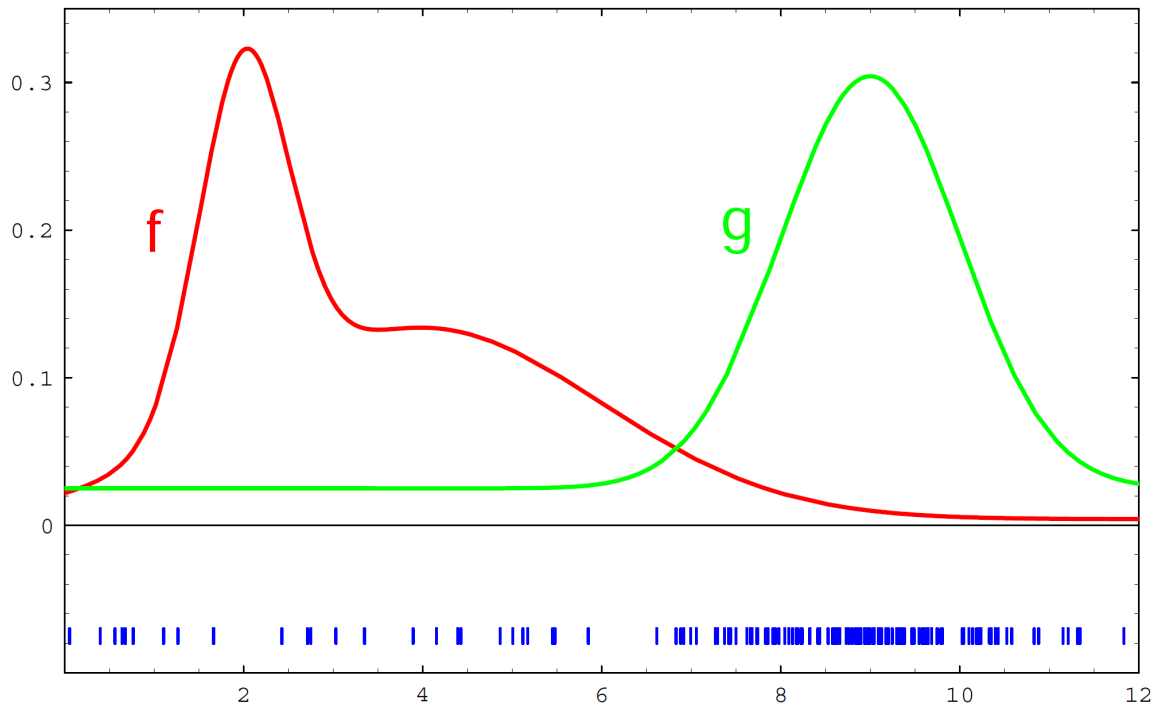


Figure A.2.: The target distribution f and the importance density g . The samples drawn from g are illustrated with vertical bars under this function.

However, the true probability distribution which the particles sample from is generally unknown or not in a suitable form for sampling. In order to deal with this problem an *importance density*¹ is introduced, so that the distribution that we want to approximate

¹Proposal density or instrumental density is used in some literature interchangeably.

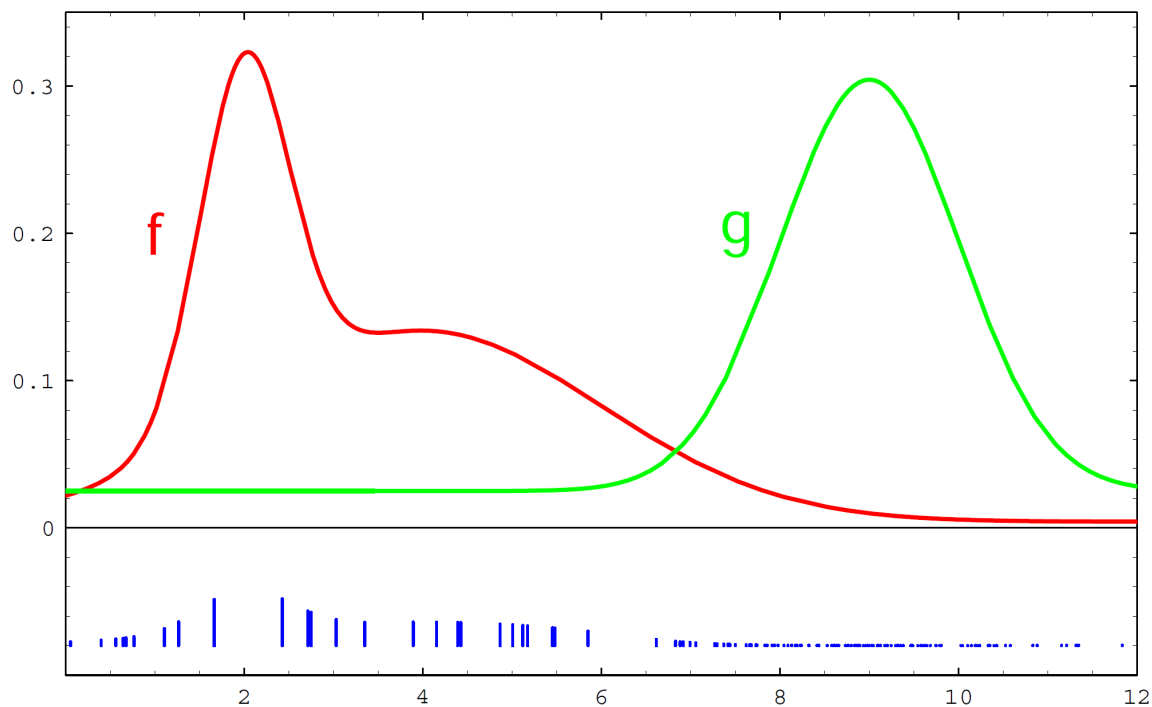


Figure A.3.: The samples drawn from the importance density g are weighted according to equation A.14 and normalized. After weighting the envelope of the resulting sample set approximates the target distribution f .

can be obtained by sampling from a different distribution, namely the *importance density*. This technique is known as *importance sampling*. It is a fundamental Monte Carlo method. Assuming that g is the importance density which can be sampled from and f is the target density, we show that it is possible to approximate f by drawing samples from g . The expectation that $x \in A$ for f can be expressed by

$$\begin{aligned}
 E_f[B(x \in A)] &= \int_A f(x) dx \\
 &= \int_{-\infty}^{+\infty} f(x) \cdot B(x \in A) dx \\
 &= \int_{-\infty}^{+\infty} \frac{f(x)}{g(x)} \cdot g(x) \cdot B(x \in A) dx \\
 &= \int_{-\infty}^{+\infty} \omega(x) \cdot g(x) \cdot B(x \in A) dx, \tag{A.11}
 \end{aligned}$$

where $\omega(x)$ is a weight function. In addition, we require that

$$f(x) > 0 \implies g(x) > 0, \tag{A.12}$$

in order to ensure that a state that might be sampled from the target distribution does not have zero probability under the importance density. According to equation A.8 $g(x)$ can be approximated by

$$g(x) \approx \frac{1}{N} \sum_{i=1}^N \delta_{s^{[i]}}(x) \tag{A.13}$$

as well. Therefore, equation A.11 can be rewritten as follows:

$$\begin{aligned}
E_f[B(x \in A)] &= \int_A f(x) dx \\
&\approx \int_{-\infty}^{+\infty} \omega(x) \cdot \frac{1}{N} \cdot \sum_{i=1}^N \delta_{s^{[i]}}(x) \cdot B(x \in A) dx \\
&= \frac{1}{N} \cdot \sum_{i=1}^N \int_{-\infty}^{+\infty} \omega(x) \cdot \delta_{s^{[i]}}(x) \cdot B(x \in A) dx \\
&= \frac{1}{N} \cdot \sum_{i=1}^N \omega(s^{[i]}) \cdot B(s^{[i]} \in A) \\
&= \frac{1}{N} \cdot \sum_{i=1}^N \omega^{[i]} \cdot B(s^{[i]} \in A). \tag{A.14}
\end{aligned}$$

Thus, *importance sampling* provides an estimate of the target distribution f . The characteristics of this estimate has been discussed in [34].

This derivation describes the basic procedure of a particle filter algorithmus. We can sample from an arbitrary distribution g to approximate the target distribution f by assigning a weight to each sample in terms of equation A.11. The sample-based representation of the importance density g and the assigned importance weights depending on the target distribution f are shown in Fig. A.2 and Fig. A.3 respectively. Although the envelope of the weighted samples approximates the target f , a normalizer is needed to sum up all weights to 1.

Notice that the weighted samples shown in Fig. A.3 do not provide samples approximately distributed according to f . A resampling step at which N samples are drawn in terms of the weights for each sample is therefore necessary in order to obtain approximate samples from the target distribution. Various resampling schemes have been introduced e.g. in [82], [25] and [34]. After resampling, the weights of all samples are set to $\frac{1}{N}$, as the weights, namely *likelihoods*, have been replaced with *frequencies* by drawing samples in terms of the weights.

Summarizing the discussion above, the particle algorithm allows us to recursively estimate the particle set S_t at the time step t on the basis of the estimation of S_{t-1} at the previous time step according to the following three basic steps:

- ▷ **Sampling:** Drawing or sampling from the importance density, so as to create the next generation of particles S'_t based on the previous particle set S_{t-1} .
- ▷ **Weighting:** Compute the weight of each particle in the set S'_t .
- ▷ **Resampling:** Resample the weighted set of particles S'_t in terms of the sample weights. Set the weights of all particles to $\frac{1}{N}$ in order to obtain the new set S_t .

Bibliography

- [1] D. Avots, E. Lim, R. Thibaux, and S. Thrun. A probabilistic technique for simultaneous localization and door state estimation with mobile robots in dynamic environments. In *Intl. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002.
- [2] S. Bahadori, A. Censi, A. Farinelli, G. Grisetti, L. Iocchi, D. Nardi, and G. D. Tipaldi. Particle based approaches for mobile robot navigation. In *Proc. of the second RoboCare Workshop*, Roma, Italy, 2005.
- [3] K. Balakrishnan, O. Bousquet, and V. Honavar. Spatial learning and localization in rodents: a computation model of the hippocampus and its implications for mobile robots. *Adaptive Behavior*, 7(2):173–216, 1999.
- [4] A. Bauer, K. Klasing, G. Lidoris, Q. Mühlbauer, F. Rohrmüller, S. Sosnowski, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss. The autonomous city explorer: Towards natural human-robot interaction in urban environments. *International Journal of Social Robotics*, 1(2):127–140, 2009.
- [5] P. Beeson, M. MacMahon, J. Modayil, A. Murarka, B. Kuipers, and B. Stankiewicz. Integrating multiple representations of spatial knowledge for mapping, navigation, and communication. In *Interaction Challenges for Intelligent Assistants, AAAI Spring Symposium Series*, 2007.
- [6] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [7] D. Beymer and K. Konolige. Tracking people from a mobile platform. In *Int. Joint Conf. on Artificial Intelligence*, 2001.
- [8] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun. Towards object mapping in non-stationary environments with mobile robots. In *Intl. Conf. on Intelligent Robots and Systems*, pages 1014–1019, Lausanne, Switzerland, 2002.
- [9] J.-L. Blanco. The mobile robot programming toolkit (mrpt) website, 2008. URL <http://babel.isa.uma.es/mrpt/>.
- [10] J.-L. Blanco, J.-A. Fernández-Madrigal, and J. Gonzalez. An entropy-based measurement of certainty in rao-blackwellized particle filter mapping. In *Intl. Conf. on Intelligent Robots and Systems*, Beijing, China, 2006.
- [11] J.-L. Blanco, J.-A. Fernández-Madrigal, and J. Gonzalez. A new approach for large-scale localization and mapping: Hybrid metric-topological slam. In *Intl. Conf. on Robotics and Automation*, pages 2061–2067, Italy, Rome, 2007.
- [12] J.-L. Blanco, J. Gonzalez, and J.-A. Fernández-Madrigal. Extending obstacle avoidance methods through multiple parameter-space transformations. *Autonomous Robots*, 24(1):29–48, January 2008.

- [13] J.-L. Blanco, J. Gonzalez, and J.-A. Fernández-Madrigal. An optimal filtering algorithm for non-parametric observation models in robot localization. In *Intl. Conf. on Robotics and Automation*, pages 461–466, Pasadena, California, USA, May 2008.
- [14] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.
- [15] J. Borenstein, H. R. Everett, and L. Feng. *Navigating Mobile Robots: Sensors and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.
- [16] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research*, 23(12):1113–1139, December 2004.
- [17] F. Bourgoult, A. Makarenko, S. Williams, B. Grocholsky, and F. Durrant-Whyte. Information based adaptive robotic exploration. In *Intl. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002.
- [18] W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, San Mateo, CA, 1997. Morgan Kaufmann.
- [19] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. MIT Press, 1998.
- [20] W. Burgard, A. Derr, D. Fox, Dirk, and A. B. Cremers. Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach. In *Intl. Conf. on Intelligent Robots and Systems*, Victoria, Canada, 1998.
- [21] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Menlo Park, August 1996. AAAI Press/MIT Press.
- [22] J. Burlet, O. Aycard, and T. Fraichard. Robust motion planning using markov decision processes and quadtree decomposition. In *Intl. Conf. on Robotics and Automation*, New Orleans, Louisiana, USA, 2004.
- [23] D. Calisi, L. Iocchi, and R. Leone. Person following through appearance models and stereo vision using a mobile robot. In *Proc. of Int. Workshop on Robot Vision*, pages 46–56, 2007.
- [24] R. Calvo, J. Cañas, and L. García-Pérez. Person following behavior generated with jde schema hierarchy. In *Int. Conf. on Informatics in Control, Automation and Robotics*, pages 463–466, Barcelona, Spain, 2005.
- [25] J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for nonlinear problems. In *Technical Report*. University of Oxford, Dept. of Statistics, 1997.
- [26] G. Casella and C. P. Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.
- [27] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardós. The spmap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–952, 1999.

- [28] Z. Chen and S. T. Birchfield. Person following with a mobile robot using binocular feature-based tracking. In *Intl. Conf. on Intelligent Robots and Systems*, San Diego, California, USA, 2007.
- [29] K. Craik. *The Nature of Explanation*. Cambridge University Press, 1943.
- [30] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2): 175–185, 2000.
- [31] M. G. Dissanayake, P. Newman, S. Clark, and H. F. Durrant-Whyte. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
- [32] J. Y. Donnat and J. A. Meyer. Learning reactive and planning rules in a motivationally autonomous animat. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 26(3):381–395, 1996.
- [33] R. Douc, O. Cappé, and E. Moulines. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69, 2005.
- [34] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2009.
- [35] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 176–183, 2000.
- [36] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.
- [37] G. Dudek and M. Jenkin. *Computational principles of mobile robotics*. Cambridge University Press, 2000.
- [38] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part I the essential algorithms. *Robotics and Automation Magazine*, 13:99–110, 2006.
- [39] P. Einramhof, S. Olufs, and M. Vincze. Experimental evaluation of state of the art 3d-sensors for mobile robot navigation. In *Workshop of the Austrian Association for Pattern Recognition*, 2007.
- [40] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, 1987.
- [41] A. I. Eliazar and R. Parr. Learning probabilistic motion models for mobile robots. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.
- [42] S. Engelson and D. McDermott. Error correction in mobile robot map learning. In *Intl. Conf. on Robotics and Automation*, pages 2555–2560, 1992.
- [43] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *Intl. Journal of Robotics Research*, 18(7):650–668, 1999.

- [44] S. Feyrer and A. Zell. Robust real-time pursuit of persons with a mobile robot using multisensor fusion. In *Intl. Conf. on Intelligent Autonomous Systems*, pages 710–715, Venice, Italy, 2000.
- [45] D. Filliat and J. A. Meyer. Map-based navigation in mobile robots I: A review of localization strategies. In R. Miikkulainen, editor, *Cognitive Systems Research*, volume 4, pages 243–282. Elsevier, 2003.
- [46] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *Int. Journal on Robotics Research*, 17(7):760–772, 1998.
- [47] D. Fox. Adapting the sample size in particle filters through kld-sampling. *International Journal of Robotics Research*, 22(12):985–1003, 2003.
- [48] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, 1997.
- [49] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.
- [50] D. Fox, W. Burgard, S. Thrun, and A. Cremers. Position estimation for mobile robots in dynamic environments. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*. AAAI, 1998.
- [51] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, Orlando, FL, 1999. AAAI.
- [52] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.
- [53] U. Frese and G. Hirzinger. Simultaneous localization and mapping – a discussion. In *Proc. of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 17–26, Seattle, WA, USA, 2001.
- [54] J. Fritsch, M. Kleinhagenbrock, S. Lang, T. Plötz, G. A. Fink, and G. Sagerer. Multimodal anchoring for human-robot-interaction. In *Robotics and Autonomous Systems, Special issue on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, volume 43, pages 133–147, 2003.
- [55] S. Fuchs and S. May. Calibration and registration for precise surface reconstruction. In *Proceedings of the DAGM Dyn3D Workshop*, Heidelberg, Germany, 2007.
- [56] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal, and J. Gonzalez. Multi-hierarchical semantic maps for mobile robotics. In *Intl. Conf. on Intelligent Robots and Systems*, pages 3492–3497, Edmonton, Canada, 2005.
- [57] D. Giuliani, M. Omologo, and P. Svaizer. Talker localization and speech recognition using a microphone array and a cross-powerspectrum phase analysis. In *Proc. Int. Conf. on Spoken Language Processing*, volume 3, pages 1243–1246, Yokohama, Japan, 1994.
- [58] R. Gockley, J. Forlizzi, and R. Simmons. Natural person-following behavior for social robots. In *Proceedings of Human-Robot Interaction*, 2007.

- [59] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.
- [60] J. E. Guivant and E. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.
- [61] J. E. Guivant and E. Nebot. Implementation of simultaneous localization and mapping in large outdoor unstructured environment. In *Advanced Robot Series*, Heidelberg, 2002. Springer-Verlag.
- [62] J. E. Guivant and E. Nebot. Improving computational and memory requirements of simultaneous localization and map building algorithms. In *Intl. Conf. on Robotics and Automation*, Washington, DC, USA, 2002.
- [63] J.-S. Gutmann and D. Fox. An experimental comparison of localization methods continued. In *Intl. Conf. on Intelligent Robots and Systems*, EPFL, Switzerland, 2002.
- [64] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, 1999.
- [65] J.-S. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proc. of Euromicro. IEEE Computer Society Press*, 1996.
- [66] A. Haasch, S. Hohenner, S. Hüwel, M. Kleinhagenbrock, S. Lang, I. Toptsis, G. Fink, J. Fritsch, B. Wrede, and G. Sagerer. Biron – the bielefeld robot companion. In *Proceedings of the Workshop on Advances in Service Robotics*, pages 27–32, Stuttgart, Germany, 2004.
- [67] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in SLAM. In *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*, Sienna, Italy, 2003. Springer.
- [68] D. Hähnel, D. Schulz, and W. Burgard. Mobile robot mapping in populated environments. *Autonomous Robots*, 17:579–598, 2003.
- [69] E. Hall. *The Hidden Dimension: Man's Use of Space in Public and Private*. The Bodley Head Ltd, London, UK, 1966.
- [70] F. Harary. *Graph Theory*. Addison-Wesley, Reading, Mass, 1969.
- [71] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [72] M. Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of Artificial Intelligence Research*, 13:33–94, 2000.
- [73] N. Hawes, M. Hanheide, J. Hargreaves, B. Page, H. Zender, and P. Jensfelt. Home alone: autonomous extension and correction of spatial representations. In *Intl. Conf. on Robotics and Automation*, Shanghai, China, 2011.

- [74] D. Helbing, I. Farkas, and T. Viscek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.
- [75] A. Helwart. Erkennung von Personen anhand ihrer Kleidung und die Anwendung auf dem Roboter BIRON. Bielefeld University, 2008.
- [76] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Proc. 12th Inter. Symposium on Experimental Robotics (ISER)*, 2010.
- [77] H. Hüttenrauch, K. Severinson-Eklundh, A. Green, and E. Topp. Investigating spatial relationships in human-robot interaction. In *Intl. Conf. on Intelligent Robots and Systems*, Beijing, China, 2006.
- [78] B. Jensen, G. Ramel, and R. Siegwart. Detecting semi-static objects with a laser scanner. In *Autonome Mobile Systeme*, 2003.
- [79] P. Jensfelt and S. Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5): 748–760, 2001.
- [80] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, 3068:182–193, 1997.
- [81] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.
- [82] G. Kitagawa. Monte carlo filter and smoother for non-gaussian non-linear state space models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.
- [83] A. Kong, J. Liu, and W. H. Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89:1032–1044, 1994.
- [84] B. Krieg-Brückner, T. Röfer, H.-O. Carmesin, and R. Müller. A taxonomy of spatial knowledge and its application to the bremen autonomous wheelchair. In C. Freksa, C. Habel, and K. Wender, editors, *Spatial Cognition, Lecture Notes in Artificial Intelligence*, volume 1404, pages 373–397. Springer, Berlin, 1998.
- [85] G. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems*, 4(2), 2007.
- [86] B. J. Kuipers. The spatial semantic hierarchy. In *Artificial Intelligence*, volume 119, pages 191–233. Elsevier, 2000.
- [87] B. J. Kuipers and Y. T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8: 47–63, 1991.
- [88] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran. Rfid in robot-assisted indoor navigation for the visually impaired. In *Intl. Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004.
- [89] C. Kwok, D. Fox, and M. Meilă. Adaptive real-time particle filters for robot localization. In *Intl. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003.

- [90] C. Kwok, D. Fox, and M. Meilă. Real-time particle filters. In *Advances in Neural Information Processing Systems*, volume 15, 2003.
- [91] S. Lang, M. Kleinhagenbrock, S. Hohenner, J. Fritsch, G. A. Fink, and G. Sagerer. Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot. In *Proc. Int. Conf. on Multimodal Interfaces*, pages 28–35, Vancouver, Canada, 2003.
- [92] J.-C. Latombe. *Robot motion planning*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1991.
- [93] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7:376–382, 1991.
- [94] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic, 1992.
- [95] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):89–96, 1992.
- [96] J. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.
- [97] M. Lohse and M. Hanheide. Evaluating a social home tour robot applying heuristics. In *Workshop Robots as Social Actors*, Munich, Germany, 2008.
- [98] A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3D rigid transformations. *Machine Vision and Applications*, 9(5):272–290, 1997.
- [99] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Intelligent and Robotic Systems*, 18(3):249–275, 1997.
- [100] M. MacMahon, B. Stankiewicz, and B. Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, Boston, MA, 2006.
- [101] A. Makarenko, S. Williams, F. Bourgault, and H. F. Durrant-Whyte. An experiment in integrated exploration. In *Intl. Conf. on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002.
- [102] R. Martinez-Cantin, N. de Freitas, and J. Castellanos. Analysis of particle methods for simultaneous robot localization and mapping and a new algorithm: Marginal-slam. In *Intl. Conf. on Robotics and Automation*, pages 2415–2420, Italy, Rome, 2007.
- [103] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312, 1992.
- [104] G. Medioni, A. R. François, M. Siddiqui, K. Kim, and H. Yoon. Robust real-time vision for a personal service robot. In *Computer Vision and Image Understanding*, volume 108, pages 196–203. Elsevier, 2007.
- [105] J. A. Meyer and D. Filliat. Map-based navigation in mobile robots II: A review of map-learning and path-planning strategies. In R. Miikkulainen, editor, *Cognitive Systems Research*, volume 4, pages 283–317. Elsevier, 2003.

- [106] H. Mine and S. Osaki. *Markov Decision Processes*. American Elsevier, Amsterdam, 1970.
- [107] J. Minguez. Nearness diagram (nd) navigation: Collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, 2004.
- [108] J. Minguez, L. Montano, and J. Santos-Victor. Abstracting vehicle shape and kinematic constraints from obstacle avoidance methods. *Autonomous Robots*, 20(1):43–59, 2006.
- [109] G. E. Monahan. A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science*, 28:1–16, 1982.
- [110] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, pages 593–598, 2002.
- [111] M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Intl. Conf. on Robotics and Automation*, Washington, DC, USA, 2002.
- [112] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence*, pages 1151–1156, 2003.
- [113] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Intl. Conf. on Robotics and Automation*, pages 116–121, St. Louis, MO, USA, 1985.
- [114] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.
- [115] P. M. Newman and K. Ho. Slam-loop closing with visually salient features. In *Intl. Conf. on Robotics and Automation*, Barcelona, Spain, 2005.
- [116] J. Peltason, F. H. Siepmann, B. W. T. P. Spexard, M. Hanheide, and E. A. Topp. Mixed-initiative in human augmented mapping. In *Intl. Conf. on Robotics and Automation*, Kobe, Japan, 2009.
- [117] R. Philippsen. *Motion Planning and Obstacle Avoidance for Mobile Robots in Highly Cluttered Dynamic Environments*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2004.
- [118] R. Philippsen and R. Siegwart. Smooth and efficient obstacle avoidance for a tour guide robot. In *Intl. Conf. on Robotics and Automation*, 2003.
- [119] M. Piaggio, P. Fornaro, A. Piombo, L. Sanna, and R. Zaccaria. An optical-flow person following behaviour. In *Proceedings of the IEEE ISIC/CIRNISAS Joint Conference*, 1998.
- [120] M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–591, 1999.
- [121] E. Prassler, D. Bank, and B. Kluge. Key technologies in robot assistants: Motion coordination between a human and a mobile robot. *Transactions on Control, Automation and Systems Engineering*, 4(1):56–61, 2002.

- [122] S. Quinlan and O. Khatib. Elastic bands: connecting path planning and control. In *Intl. Conf. on Robotics and Automation*, 1993.
- [123] S. Roweis and Z. Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2), 1999.
- [124] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation – mobile robot navigation with uncertainty in dynamic environments. In *Intl. Conf. on Robotics and Automation*, Detroit, Michigan, USA, 1999.
- [125] N. Roy, G. Gordon, and S. Thrun. Finding approximate pomdp solutions through belief compression. *jair*, 23:1–40, 2005.
- [126] C. Schlegel. Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot. In *Intl. Conf. on Intelligent Robots and Systems*, 1998.
- [127] P. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1), 1966.
- [128] M. J. Schoppers. Universal plans for reactive robots in unpredictable environments. In *Int. Joint Conf. on Artificial Intelligence*, pages 1039–1046, 1987.
- [129] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. People tracking with a mobile robot using sample-based joint probabilistic data association filters. *International Journal of Robotics Research (IJRR)*, 22(2), 2003.
- [130] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.
- [131] H. Sidenbladh, D. Kragik, and H. I. Christensen. A person following behaviour of a mobile robot. In *Intl. Conf. on Robotics and Automation*, Detroit, Michigan, USA, 1999.
- [132] R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. A Bradford Book, The MIT Press, Cambridge, Massachusetts and London, England, 2004.
- [133] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Siméon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5), October 2007.
- [134] P. Smith, I. Reid, and A. Davison. Real-time monocular slam with straight lines. In *British Machine Vision Conference*, 2006.
- [135] C. Stachniss and W. Burgard. Mapping and exploration with mobile robots using coverage maps. In *Intl. Conf. on Intelligent Robots and Systems*, pages 476–481, Las Vegas, NV, USA, 2003.
- [136] C. Stachniss, D. Hähnel, and W. Burgard. Grid-based fastslam and exploration with active loop closing. In *Online Proc. of the Dagstuhl Seminar on Robot Navigation (Dagstuhl Seminar 03501)*, Dagstuhl, Germany, 2003.
- [137] C. Stachniss, D. Hähnel, and W. Burgard. Exploration with active loop-closing for fastslam. In *Intl. Conf. on Intelligent Robots and Systems*, pages 1505–1510, Sendai, Japan, 2004.

- [138] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proceedings of Robotics: Science and Systems (RSS)*, 2005.
- [139] D. Stavens and S. Thrun. A self-supervised terrain roughness estimator for off-road autonomous driving. In *Conf. on Uncertainty in Artificial Intelligence*, 2006.
- [140] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg. A 3D laser range finder for autonomous mobile robots. In *Intl. Symposium on Robotics Research*, pages 153–158, 2001.
- [141] A. Swadzba and S. Wachsmuth. Categorizing perceptions of rooms using 3d features. In *Intl. Workshop on Statistical Techniques in Pattern Recognition*, 2008.
- [142] A. Swadzba, B. Liu, J. Penne, O. Jesorsky, and R. Kompe. A comprehensive system for 3d modeling from range images acquired from a 3D ToF sensor. In *Proc. Int. Conf. on Computer Vision Systems*, 2007.
- [143] S. Thrun. Bayesian landmark learning for mobile robot localization. In *Machine Learning*, volume 33, pages 41–76. Kluwer Academic Publishers, 1998.
- [144] S. Thrun. Learning occupancy grids with forward models. In *Intl. Conf. on Intelligent Robots and Systems*, USA, Hawaii, 2001.
- [145] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.
- [146] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [147] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 31(5):1–25, 1998.
- [148] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *Journal of Robotics Research*, 19(11):972–999, 2000.
- [149] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Intl. Conf. on Robotics and Automation*, pages 321–328, San Francisco, CA, USA, 2000.
- [150] S. Thrun, D. Fox, and W. Burgard. Monte carlo localization with mixture proposal distribution. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, Austin, TX, 2000. AAAI.
- [151] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. In *Artificial Intelligence*, volume 128, pages 99–141. Elsevier, 2001.
- [152] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [153] E. A. Topp, H. Hüttenrauch, H. I. Christensen, and K. S. Eklundh. Bringing together human and robotic environment representations - a pilot study. In *Intl. Conf. on Intelligent Robots and Systems*, Beijing, China, 2006.

- [154] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Intl. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.
- [155] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13, pages 376–380, 1991.
- [156] S. Vasudevan, S. Gachter, M. Berger, and R. Siegwart. Cognitive maps for mobile robots an object based approach. In *Proceedings of the IEEE IROS 2006 workshop - From Sensors to Human Spatial Concepts (FS2HSC 2006)*, Beijing, China, 2006.
- [157] C.-C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Intl. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003.
- [158] O. Watson and D. S. Touretzky. Navigating with the tekkotsu pilot. In *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference*, The Colony in Palm Beach, Florida, USA, 2011.
- [159] J. Weingarten, G. Gruener, and R. Siegwart. A state-of-the-art 3D sensor for robot navigation. In *Intl. Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004.
- [160] S. Williams, G. Dissanayake, and H. Durrant-Whyte. Towards terrain-aided navigation for underwater robotics. *Advanced Robotics*, 15(5), 2001.
- [161] D. F. Wolf and G. S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19:53–65, 2005.
- [162] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization for mobile robots using an image retrieval system based on invariant features. In *Intl. Conf. on Robotics and Automation*, Washington, DC, USA, 2002.
- [163] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, pages 47–53, Minneapolis, MN, USA, 1998.
- [164] F. Yuan, M. Hanheide, and G. Sagerer. Spatial context-aware person-following for a domestic robot. In *International Workshop on Cognition for Technical System*, Munich, Germany, 2008.
- [165] F. Yuan, A. Swadzba, R. Philippsen, O. Engin, M. Hanheide, and S. Wachsmuth. Laser-based navigation enhanced with 3d time-of-flight data. In *Intl. Conf. on Robotics and Automation*, pages 2844–2850, Kobe, Japan, 2009.
- [166] F. Yuan, L. Twardon, and M. Hanheide. Dynamic path planning adopting human navigation strategies for a domestic mobile robot. In *Intl. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.
- [167] H. Zender, P. Jensfelt, and G. Kruijff. Human- and situation-aware people following. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2007)*, pages 1131–1136, Jeju Island, Korea, 2007.

-
- [168] H. Zender, P. Jensfelt, Ó. M. Mozos, G.-J. M. Kruijff, and W. Burgard. An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, pages 1584–1589, Vancouver, British Columbia, Canada, 2007.
- [169] H. Zender, Ó. M. Mozos, P. Jensfelt, G.-J. M. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.
- [170] N. L. Zhang and W. Zhang. Speeding up the convergence of value iteration in partially observable markov decision processes. *Journal of Artificial Intelligence Research*, 14:29–51, 2001.