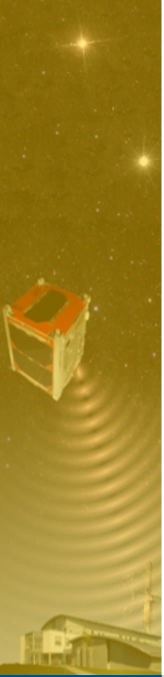


Institut für Informatik
Lehrstuhl für Robotik und Telematik
Prof. Dr. K. Schilling

Würzburger Forschungsberichte in Robotik und Telematik

Uni Wuerzburg Research Notes in Robotics and Telematics

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Band 5

Markus Sauer

Mixed-Reality for Enhanced Robot Teleoperation

Die Schriftenreihe

wird vom Lehrstuhl für Informatik VII: Robotik und Telematik der Universität Würzburg herausgegeben und präsentiert innovative Forschung aus den Bereichen der Robotik und der Telematik.

Die Kombination fortgeschrittener Informationsverarbeitungsmethoden mit Verfahren der Regelungstechnik eröffnet hier interessante Forschungs- und Anwendungsperspektiven. Es werden dabei folgende interdisziplinäre Aufgabenschwerpunkte bearbeitet:

- Robotik und Mechatronik: Kombination von Informatik, Elektronik, Mechanik, Sensorik, Regelungs- und Steuerungstechnik, um Roboter adaptiv und flexibel ihrer Arbeitsumgebung anzupassen.

- Telematik: Integration von Telekommunikation, Informatik und Steuerungstechnik, um Dienstleistungen an entfernten Standorten zu erbringen.

Anwendungsschwerpunkte sind u.a. mobile Roboter, Tele-Robotik, Raumfahrtsysteme und Medizin-Robotik.

Zitation dieser Publikation

SAUER, M. (2010). Mixed-reality for enhanced robot teleoperation. Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 5. Würzburg: Universität Würzburg.

Lehrstuhl Informatik VII
Robotik und Telematik
Am Hubland
D-97074 Wuerzburg

Tel.: +49 (0) 931 - 31 - 86678
Fax: +49 (0) 931 - 31 - 86679

schi@informatik.uni-wuerzburg.de
http://www7.informatik.uni-wuerzburg.de

Universitätsbibliothek Würzburg
Am Hubland
D-97074 Würzburg

Tel.: +49 (0) 931 - 31 - 85917
Fax: +49 (0) 931 - 31 - 85970

opus@bibliothek.uni-wuerzburg.de
http://www.opus-bayern.de/uni-wuerzburg/

ISSN (Internet): 1868-7474
ISSN (Print): 1868-7466
eISBN: 978-3-923959-67-9

Mixed-Reality for Enhanced Robot Teleoperation

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Bayerischen Julius–Maximilians–Universität Würzburg

vorgelegt von

Markus Sauer

aus

Würzburg

Würzburg 2010

Eingereicht am: 22.07.2010
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr. Klaus Schilling
2. Gutachter: Prof. Dr. Aarne Halme
Tag der mündlichen Prüfung: 01.02.2011

Danksagung

Die vorliegende Dissertation wäre nicht möglich gewesen ohne die tatkräftige Unterstützung von vielen Menschen, denen ich im Folgenden noch einmal herzlich danken möchte.

Zuerst bedanke ich mich bei meinem Doktorvater Prof. Dr. Klaus Schilling für die Möglichkeit eine Promotion unter seiner Betreuung durchzuführen, aber auch für die vielen spannenden Dinge, die ich die letzten Jahre erleben durfte, und die Erfahrungen, die ich dabei sammeln konnte. Er ermöglichte mir eine berufliche und persönliche Weiterentwicklung, die sonst bei einer Promotion nicht selbstverständlich ist. Prof. Dr. Aarne Halme einen herzlichen Dank für seine unkomplizierte Art und die zügige Erstellung des Zweitgutachtens. Ein weiterer Dank gilt meinen beiden Prüfern Prof. Dr. Phuoc Tran-Gia und Prof. Dr. Jürgen Albert. Beide habe ich schon während meines Studiums, aber auch jetzt während meiner Promotionszeit sowohl menschlich als auch fachlich sehr schätzen gelernt.

Durch die Arbeit während der Promotion am Lehrstuhl für Robotik und Telematik der Universität Würzburg und dem Zentrum für Telematik hatte ich die Gelegenheit mit einem außergewöhnlichen Team zusammenzuarbeiten, was sich auch in privaten gemeinsamen Unternehmungen außerhalb der Arbeit widerspiegelte. Danke hierfür im Besonderen an Maximilian Drentschew, Daniel Eck, Matthias Görs, Christian Herrmann, Dr. Martin Hess, Robin Heß, Dr. Peter Hokayem, Markus Krauß, Florian Leutert, Dr. Lei Ma, Marco Schmidt, Rajesh Shankar Priya, Prof. Dr. Dušan Stipanović und Lothar Stolz, sowie Heidi Frankenberger, Ulrike Walter, Eberhard von Deuster, Dieter Ziegler, Edith Reuter und Annette Zillenbiller. Bei zwei Kollegen und Freunden möchte ich mich nochmal separat besonders bedanken - Dr. Frauke Driewer und Dr. Florian Zeiger. Mit beiden habe ich in meiner Promotionszeit am engsten zusammengearbeitet und fachliche aber auch persönliche Diskussionen geführt, was ich immer sehr zu schätzen wusste. Danke auch nochmal an Dr. Florian Zeiger, dass er in unserem gemeinsamen Endspurt der Doktorarbeit in den letzten beiden Jahren egal unter welchen Umständen immer wieder den nötigen Zug aufgebaut hat und, dass er immer als Freund zur Verfügung stand. Herzlichen Dank an alle Kollegen vom Lehrstuhl III - unserem zweiten Zuhause. Die unzähligen Lehrstuhl III - Aktivitäten, bei denen wir dabei waren, haben eine bleibende Erinnerung hinterlassen, die ich nicht missen möchte. Hier nochmal einen ganz besonderen Dank an Dr. Rastin Pries und Dr. Robert Henjes für die unzähligen gemeinsamen Abende und die tolle Zeit, die wir hoffentlich in Zukunft mit allen aufrecht erhalten, bzw. immer wieder regelmäßig aufleben lassen werden. Danke auch an meinen Kommilitonen Marco Schörger für die über zehnjährige Freundschaft und seine immer positive Grundeinstellung und Art. In den letzten Jahren durfte ich im Rahmen dieser Arbeit auch mit einer Vielzahl von Studenten zusammenarbeiten. Stellvertretend für alle diese Studenten möchte ich mich insbesondere bei meinen Diplomanden Stephan Toth, Matthias Wiedemann, Manuel Göllnitz, Koami Edem Missoh, Hossein Shahrabi Farahani, Lakshminarasimhan Srinivasan und Thomas Peretzki bedanken.

Vielen Dank an meine Eltern Brigitte und Klaus und meinen Brüdern Florian und Benedikt für die uneingeschränkte Unterstützung und den Rückhalt in all den Jahren. Das Bewusstsein über diese Unterstützung und diesen Rückhalt gab mir die Sicherheit, die in vielen Situation für diese Arbeit wichtig war. Zu guter Letzt möchte ich mich bei dem wichtigsten Menschen in meinem Leben bedanken - meiner Freundin Carolin Lamprecht. In all den Jahren, die wir bisher gemeinsam gegangen sind, war sie immer meine Kraft und meine Stärke. Wann immer ich jemand brauchte, war sie für mich da, egal zu welcher Zeit, oder an welchem Ort. Wann immer ich Zweifel hatte, hat sie an mich geglaubt und mich wieder aufgebaut. Danke für alles!

Contents

1 Introduction

In the last decades research in the field of robotics has produced impressive advances and many interesting applications arose, ranging from e.g. industrial manipulators, search and rescue, up to robots on Mars. Nevertheless, robots have still not found their place in everyday life, such that today robotics faces the increasing requirement of a paradigm shift from industrial applications with fixed requirements and conditions to everyday, dynamic human environments. Robotics research has made great progress with respect to applications in such environments with a higher level of autonomy. But still, there are many tasks and applications where humans acting as coordinators or operators are needed. Reasons for this need of human support can be for instance safety issues and/or complex, dynamic real-world scenarios. The most obvious application examples are robots applied in human-robot teams in search and rescue scenarios or robots utilized in space missions. These scenarios include very complex tasks due to e.g. unknown or hazardous environments, danger for human live, or the risk of a complete mission failure resulting in costs of some million Euros. During most of these complicated missions the human is an irreplaceable part. The level of human perception capabilities and interpretation of the situation, as well as the ability to make decisions on basis of available limited information, past experience and intuition is currently not reached by artificial systems. Moreover, the human is needed for certain tasks, such as in search and rescue missions performing rescue activities or giving medical and psychological support to victims.

In order to take advantage of the superior adaptive and cognitive abilities of human in such missions it is a must to provide reasonable user interfaces to the humans fitting to the current task and the interaction role. Autonomous features of the vehicle, e.g. path-following or obstacle avoidance, relieve the operator from simple robot control tasks, and should be integrated as assistance functions to the user interface. Here, a major issue is to enable the human to both, gain and maintain situation awareness, and common ground. If common ground between the (semi-)autonomously behaving robots and the human operator is not achieved, the operator will either neglect this specific robotic system or turn off all the assistance systems. Thus, the user interface for the human operator plays one of the most significant roles for the success of any application of robots operating at remote.

Due to the nowadays available hardware providing graphical calculations, as well as computing power at reasonable costs mixed reality systems as a novel way of building effective, intuitive and integrated human-machine interfaces are increasingly of interest and realizable. Mixed reality systems combine real and computer generated objects correctly aligned in real environments and run interactively and in real-time. Dependent on the ratio of virtual and real elements the mixed reality interfaces can be further subdivided into augmented reality and augmented virtuality interfaces. Realizing this type of interfaces raises a lot of new technical challenges. However, these mixed reality concepts with the principle of integrating all necessary information the same spatial context allow for improved user interfaces in many areas of robotics. Looking at human factors theory, many of the requirements for teleoperation systems and user interface design, cannot be meet as perfect with standard methods as it is possible with mixed reality.

This work investigates the application of mixed reality technologies for teleoperation of robots. Based on careful analysis of human factors theory, literature research, and own experiments innovative technological concepts and methodologies are developed which enable to realize enhanced integrated mixed reality user interfaces in dependency on the current application, task, and corresponding interaction roles. This covers the overall teleoperation system from perception, autonomy design, communication, haptic and graphical user interface elements.

1.1 Exemplary Application: Robots in Urban Search and Rescue

Mobile robots become more and more an attractive tool to accomplish work in hazardous environments or in areas not accessible for humans. One of the most attractive application areas is the field of search and rescue, because of the often unpredictable danger for human life during an accident or a catastrophe. Typical scenarios where robots are very desirable to support the humans involved are natural disasters (e.g. earthquakes, hurricanes), fire or terrorist attacks. In these scenarios often the current state of the environment is unknown and maybe dangerous (e.g. discharging hazardous gases, materials,...). Things that might happen, when entering different areas, might be often unpredictable (e.g. danger of structural collapses). Additionally one cannot rely on previously existing infrastructure to get information about the incident area. In all these scenarios robots with appropriate equipment can support and assist the human search and rescue teams by e.g. taking sensor measurements to characterize the remote areas and perform tasks such as exploration, search or manipulation of remote objects.

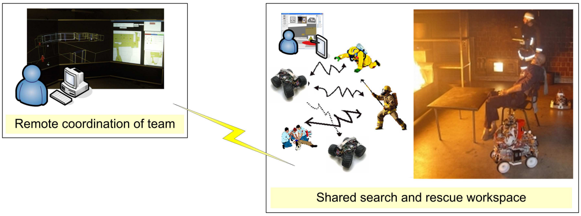

Remote coordination of team
Shared search and rescue workspace

Figure 1.1: Exemplary application scenario of human-robot teams in search and rescue.

The different roles of the team members in a typical search and rescue mission (cf. Figure 1.1) supported by robots might be formulated as follows. Humans are involved as operators outside the emergency area, taking over the role of a coordinator and supervisor (comparable to the commanding officer in current fire fighting missions). Inside the emergency area human rescue workers perform their scenario specific task (find and rescue victims, find the fire sources and extinguish fire, shut off gas valves and secure the surrounding etc.). New sensors, localization methods, and data transmission units can be integrated into existing rescue equipment [25]. Robots mainly provide efficient advanced exploration of the emergency area by sensor measurements. In this way, they help to reduce the risk for the human rescue workers and enable faster search of wide areas. The map that can be built from the collected data helps the operator to understand the situation and supports quick decision making and task distribution. Another important task for robots is to serve as a communication platform between a found victim (not immediately reachable by human rescue workers) and people outside the workspace (e.g. medical staff). Robots with more specific equipment can also take over more complex and specific manipulation task, e.g. shutting of gas valves. Humans may always take over the direct control of the robots.

Many user requirement analyses were preformed to understand what are the needs and wishes of today's rescue personnel for rescue robots. For instance Jones and Hinds [26] studied SWAT (special weapons and tactics) teams in training in order to transfer the observations made in multi-robot systems. Burke et al. [27] participated with robots in training of urban search and rescue personal as well as in real incidents [28], [29]. Adams [30] described requirements for human-robot interaction by analyzing human teams from a bomb squad and fire department.

This work is also based on a user requirement analyses [31] performed for implementation of rescue missions [32] in the context of the "PeLoTe" project. The results provided additional important guidelines for the design of robot capabilities and the user interfaces. Many ideas and suggestions of potential end-users, who were mainly from the area of fire-fighting, entered the design processes of this work. Thus, the search and rescue scenario is used as reference application for this work.

1.2 Outline

The remainder of this monograph is structured as follows. Chapter 2 gives the problem analysis. It includes a detailed analysis of related work on human-robot interaction, teleoperation and relevant human factors theory. Based on this, design elements for such teleoperation systems are derived. Design background, conditions, and demands for user interfaces and driving assistance systems are given in order to support and strengthen the understanding of the overall teleoperation system including the human. Finally, the resulting system model putting the human in the loop is presented.

Chapter 3 investigates two advanced technical elements for implementing a teleoperation system under considerations of the human operators' needs. For the core component communication of a teleoperation scenario an intelligent traffic shaping mechanism is introduced and evaluated which enables to optimize the experience of the video stream in a mixed reality user interface for a navigation task. The second part of this chapter considers the fact that the human perception in the here assumed scenarios is limited by sensors of the remote machine. Three-dimensional perception with a time-of-flight camera is investigated and evaluated with respect to calibration, parameter optimization, and pre-processing and with respect to the applicability for mixed reality user interfaces.

Chapter 4 introduces the concept which enables seamless sliding autonomy during navigation of a mobile robot by a multi-modal user interface with haptic and visual components. The introduced system is based on a robust environment perception and a task specific driving assistance system, which renders the current intention of the robot's autonomy system to a force feedback device and hereby enables the human operator to maintain situation awareness and common ground. The concept and its components are validated by extensive user studies.

Chapter 5 focuses on the developed concepts for integrated, mixed reality user interfaces. It is shown how the additional information from the robots' sensors can be used to realize advanced, fused representations of distance sensor data and camera images, which also enable a dynamic occlusion handling for an augmented reality view. In the following the mixed reality approach for graphical user interfaces for this work is discussed, and how a software framework corresponding to the needs for this kind of interfaces can be derived. Different novel, elaborated mixed reality user interfaces and corresponding evaluations are developed for team coordination but also for (assisted) direct operation of a single robot. Based on the introduced generic mixed reality user interface a new type of control of a robot with a short-term predictive user interface is proposed and analysed.

In Chapter 6 a short summary, final conclusions, and directions of future research based on this work are presented.

1.3 Innovations

In order to enable a performant mixed reality teleoperation system for robots, a human centered design of all system elements of the teleoperation chain is required. Therefore, this thesis aims to optimize the different elements of a teleoperation scenario from an overall system performance point of view including the human as design factor. It provides consistent insights in the potential, demands, and solutions for successful application of mixed reality technologies and autonomy components for increased performance and acceptance of robot teleoperation systems.

Design of Teleoperation System

The human and the actual task introduce unchangeable design factors into a teleoperation system. A detailed problem analysis taking this into consideration is provided in order to achieve a task-oriented optimization of the performance of teleoperation systems from a holistic point of view. The well-known supervisory control model [33] is applied as basis for the teleoperation model. Human factors have a significant impact on the overall performance of a teleoperation system as the human is always finally the limiting factor. For this reason design goals and demands for a user centered design of the different elements of a teleoperation system are derived including the human in the loop.

Application of Autonomy as Assistance Systems

Nowadays, autonomy can be designed and implemented very robust in defined spatial and time horizons. Taking advantage of this fact, first a sliding autonomy concept based on a multimodal mixed reality user interface is introduced and evaluated which enables seamless transition between different levels of autonomy for the robot navigation task. The user studies with the systems prove the significant increase of navigation performance and safety. The results are also strong indicators that the proposed combination of local autonomy, haptic feedback of the systems' intention, and augmented reality graphical user interface increase situation awareness and common ground, and reduce operators workload. Robust local autonomy also enables the introduction of a novel way to operate robots - a short-term predictive mixed-reality user interface. Hereby, a graphical user interface can be realized which allows for a combination of both advantageous viewpoint approaches - an exocentric viewpoint and a spatially correctly aligned augmented reality view. In addition, the system implements a partial decoupling of the teleoperation control loop such that the visibility of network effects in a mixed reality user interface can be reduced.

Three-Dimensional Perception

Various novel sensor technologies are available for robot teleoperation. We investigate novel time-of-flight cameras like the PMD camera for potential application areas in teleoperation of robots with a special focus on human-robot interaction aspects. Characteristics, calibration methods, and limitations of this novel sensor type are analyzed. Task-oriented optimization principles of sensor parameters are introduced and validated. The calibration results and the specific parameter adaptation schemes enabled to significantly improve the image quality for various applications in robotics (e.g. autonomous behaviors, driving assistance systems, graphical user interface). Building maps with this type of sensor as one application example is evaluated in a qualitative user study. The optimizations and findings have enabled further advanced elements in the graphical mixed reality user interfaces.

Graphical Mixed-Reality User Interfaces

Mixed-reality provides a technological solution for many requirements of the user-centered design of the graphical user interface for robot teleoperation but also bears a lot of challenges.

The camera image from a robot is the major element of mixed reality user interfaces for the operator during a navigation task. A traffic shaping method is introduced in order to optimize the experience of video streams for the human operator in ad-hoc networks. These ad-hoc networks are especially interesting for the reference application scenario search and rescue. The evaluation shows that the achievable increased robustness and stabilization of the video stream for a mixed reality user interface in such scenarios.

In general robots have additional distance sensors. In this work, we take advantage of this fact by a fusion of distance sensor data (e.g. from a PMD camera or laser-range finder) with two-dimensional camera images to three-dimensional environment representation for the user interface. The combined user interface element can provide a better spatial understanding of the remote environment in an integrated, natural fashion. This way of modeling a two-dimensional camera image as a three-dimensional structure based on data from distance sensors also enables to realize communication bandwidth efficient three-dimensional visualizations on any stereo device. In addition to the improved spatial perception of the remote environment, the provided approach is also a performant solution for the dynamic occlusion handling problem of augmented reality interfaces without a priori knowledge of the operating environment.

The elaborated design guidelines for the user interface are transfered to a graphical mixed reality user interface concept. A dedicated mixed reality software framework is developed which allows to quickly realize and modify mixed reality user interfaces. It also supports the usage on various stereo display devices without any further implementation effort. As the interaction role is the major design parameter for the user interface two implementations of role specific user interfaces are realized in the context of this work - one with a focus on team coordination and operation in the supervisor role, and one dedicated to the operator role for (assisted) teleoperation of a single robot. The performed evaluations of these interfaces also include the aspect of using stereo devices.

Based on the results of various conducted user studies the mixed reality component of the role specific graphical user interfaces are further developed to a generic mixed-reality user interface, which makes use of the different advanced mixed-reality technologies. Hereby it enables a better adaptation to the interaction role and the task. Together with the elaborated mixed reality framework and the three-dimensional camera image modeling it can inherently be used with stereo display devices and exploit the advantages of stereo visualizations. The generic mixed reality user interface combined with a robust autonomy component also facilitates the development of the already mentioned predictive mixed reality user interface.

2 Human in the Loop

2.1 Human-Robot Interaction (HRI)

Human-Robot Interaction (HRI) is an emerging field gaining more and more interest by researchers from many different fields of research [34]. Due to the technical developments of the recent years robots are applied to more and more tasks, and thus get closer to the human. Before, they were in most cases only better tools without the need for any real interaction with humans. Nowadays robots start to move out to everyday life and humans and robots have to interact in many ways. Goodrich and Schultz [35] define HRI as "a field of study dedicated to understanding, designing, and evaluating robotic system for use by or with humans", and to "understand and shape the interactions between one or more humans and one or more robots". This requires communication or more abstract an information exchange between the humans and the robots through an user interface. It can be realized in many ways - through graphical user interfaces in all its different shapes, force rendering systems, gestures, speech, etc.. Often a combination of different communication elements to a multimodal interfaces offer the optimum.

HRI research is inherently interdisciplinary. Contributions from fields of cognitive science, linguistics, psychology, engineering, mathematics, computer science, human factors engineering, and design are demanded in order to successfully design HRI systems [35], and hereby open up new applications for robotic systems in everyday life. The topics which need to be covered by HRI are manifold due to the different application areas for instance in home, assistive robotics, entertainment, education, industry, search and rescue, police, military, space, etc.. These applications often require very different interaction schemes between humans and robots Earlier HRI was mainly covered by the terms telerobotics and teleoperation which focused at that time on direct continuous control of a remote manipulator in industrial scenarios [33]. Robots were simply tools in a master - slave relationship. The slave robotic system reflects the movements of a master system operated by a human at a remote location. Nowadays, human-robot interaction goes much beyond direct operation of a remote system and allows for a much more interactive, intelligent behavior of the robot [36]. It is important to recognize that both, human and robot, are part and need to be part of the system and design process. It is the performance of the combined system which finally matters. In this work the focus of the presented HRI concepts is on increasing the performance in remote navigation and coordination of robots in exploration and search scenarios by applying mixed reality technologies.

2.1.1 Characteristics of HRI

HRI as new, strongly related topic can build upon a large repository with respect to theory, methodologies and technologies from Human-Computer Interaction and Human-Machine Interaction. Adams [37] points out the importance of the user centered design in HRI and the application of already available results in human factors research. Especially the design of "humane" interfaces which is responsive to human needs and frailties is important [38].

Although design of interaction between humans and robots have many comparable issues to the design of interfaces for human-computer interaction, human-robot interaction also has some major differences [39]. Fong et al. [40] differentiate between HRI, HCI and HMI as follows : "HRI (...) differs from both HCI and HMI because it concerns systems (i.e., robots) which have complex, dynamic control systems, which exhibit autonomy and cognition, and which operate in changing, real-world environments.".

Scholtz [36] identify six dimensions which need to be taken into consideration in order to successfully apply and adapt knowledge from topics like HCI and HMI:

- *Different requirements based on interaction roles.* The relationship between human and a robots can totally differ. A person can for instance supervise a robot, he/she can be a teammate to the robot, or he/she can repair or modify the robot. This leads to different requirements on the HRI system

- *Physical nature of robot.* Robots are physical entities, acting in a physical world. Thus, they need to have a certain understanding and model of their environment. This needs to be transfered to the human in order to enable him/her to understand the behavior and decisions of the robot.

- *Dynamic nature of the hardware.* HCI assumes a deterministic behavior of the computer. There is no physical change a human must track. HRI is different. Systems might change and failures might occur. It needs to be traceable by the human what changes to the system and functionality occur.
- *Environment in which the interactions occur.* The application scenario determines the conditions in which the whole interaction system must be able to work. The environment can e.g. be very dynamical, harsh, dangerous, noisy, etc..
- *Number of independent systems the user interacts with.* In HCI the human in general only interacts with one computer, even in collaborative systems. This can be totally different in HRI. The ultimate goal for HRI in team scenarios for instance is, that the human is able to interact with heterogeneous robots like in a human only team.
- *Autonomy of robots.* Most of the robots can act autonomously for specified periods of time based on their own gathered information .

In addition, typical design issues can be derived based on the six dimension of Scholtz [36] which need to be dealt with when designing human-robot interaction for a specific task and scenario:

- What are the interaction roles between human and robot?
- How is information exchanged between human and robots (e.g. graphical user interfaces, speech, gestures, etc.)? Which combination fits best for the task?
- How should autonomy be designed? Is the level of autonomy changing? Who selects the level of autonomy?
- What type of cooperation is realized? What kind of team setup is applied?
- What is the shape of the robot (e.g. humanoid, wheeled robots, walking robots, manipulators, etc.)?

- What are the relationships with respect to time and space? Is the interaction synchronous or asynchronous? Are the humans and systems collocated?

Yanco and Drury [41] offer a taxonomy in order to classify human-robot interaction. It enables to characterize and describe human-robot interaction along a set of categories like e.g. task type, team composition, level of shared interaction among team, interaction roles, time/space, autonomy level,... .

2.1.2 Interaction Roles

Robots switch over to be no longer simple stupid machines, which continuously repeat pre-programmed work sequences, but to be complex, interactive systems, which interact intelligently in various ways with humans. Thus, the different interaction roles of humans and robots need to be identified in order to optimize the interaction with respect to the requirements of the specific interaction role and the user interface accordingly .

One way to differentiate between the different ways of interaction is the model from Scholtz [36]. Scholtz's identifies five interaction roles based on the five generic supervisory's functions in the context of human interaction with automatic systems outlined by Sheridan [33]: planning what task to do and how to do it, teaching or programming the computer, monitoring the automatic action to detect failures and deviations, intervening to specify a new goal, react on emergencies or take over manual control, and learning from experience. Scholtz mainly focuses on the support for specifying actions, monitoring actions and the intervention. The assumption is made that the robot already provides a set of basic functions and the action of reprogramming as called by Sheridan is the intervention component. The five defined interaction roles in the theory from Scholtz [36] do not consider directly the learning part on human and machine side.

Based on the model from Scholtz [36] it can be distinguished between the following basic interaction roles which a human can take over when interacting with robots:

- *Supervisor Interaction*
  A human in the supervisor role is monitoring and controlling the overall

situation during a task or mission. He/she evaluates the situation based on a given goal. Accordingly he/she specifies plans on a higher level. The robot's software generates the specific actions considering its perception of its environment and the given plans by the supervisor. The supervisor can step in and specify an action or modify plans for a robot. Here the human-robot interaction focuses very much on the perception element. The supervisor can interact with a single robot, multi-robot systems or even human-robot teams. He/she is responsible for the long-term planning and the final fulfillment of the mission goal. The supervisor has to be well aware about the robots' capabilities, problems, and current state. Typically an external coordinator of a robot team has the supervisor role in a search and rescue scenario.

- *Operator Interaction*
  As operator the human will mainly act on the action level of the robot. The human will take over the operator role, when the behavior, control parameters, software, or internal models of the robot needs to be changed. According to the model longer term plans are not changed by the operator, but only by the supervisor. During direct teleoperation of a robot the human i.e. takes over the operator role.

- *Mechanic Interaction*
  The mechanic role deals with changes and adaptation of the robots' hardware. A human in mechanic role has also to take care that the applied changes to the robots have the desired effect on the behavior of the robot.

- *Peer Interaction*
  In general in the peer or teammate role the human has direct interaction with the robot. The interaction is mostly on a higher level compared to the operator role. For instance a command like "Follow me, please!" would be on this higher level. In most cases a spoken dialogue or gestures comparable to that of humans working together, would be the most natural way of interaction between human and robot for this role. The peer or teammate role most often occurs when humans and robots work together co-located as a team in order to fulfill a joint task. A typical example is again in search and rescue where humans and robots enter a incident area together as team and use the complementary abilities to reach a common goal. In

case a human e.g. wants to change the long term plans and goals of a robot he/she would switch to the supervisor role.

- *Bystander Interaction*
  In the bystander role the human has only limited or even no knowledge about the robot and the possibilities for interaction are very restricted. A bystander is in general not familiar with robots and though it is very difficult to design [42], predict and evaluate [43] interaction concepts for this role. However, for real-world scenarios it is very important to design the robot for a certain level of bystander interaction - at least to guarantee safety for people acting as bystander. A typical bystander role has a victim in a search and rescue scenario.

From the explanations above it is obvious that the boundaries between the different interaction roles are very fuzzy. It is even possible that one person takes over multiple roles, switches between the roles, or that different people take over the same role. For real-world scenarios a robot has to provide the support for the interfaces and behaviors to cope with all the five types of interaction. Nevertheless, it is very important to optimize the user interfaces especially for each interaction role in order to maximize the performance of interaction on all levels. In this work the main focus lies on the supervisor, the operator interaction role and to some extend the peer or teammate role. The human in this work most often acts as operator and supervisor at the same time. Nevertheless, many of the results can be transfered to the other interaction roles. One example is the mixed reality approach to visualize system information. This provides inherently a way to intuitively present information to bystanders or mechanics.

2.2 Teleoperation

Teleoperation can be defined as "operating a machine at a distance" [44]. In teleoperation all decision of the operator are based on what he/she can perceive and understand from the information presented to him/her through the user interface. This is limited by the perception of the robots sensors, limitations and disturbances of the communication channel and how the information is pre-processed and presented to him/her. In addition, especially for the teleoperation of mobile robots it is necessary that all information is presented in a correct spatial relationship to each other and all offered autonomy and assistance functions are transparent in behavior to the human operator. Here, key issues are localization and navigation. These characteristics and demands are especially challenging as teleoperation is in most cases applied in scenarios with unknown and unstructured areas. In these application areas full autonomy is not feasible or desired. Human operator(s) are incorporated for many tasks, including perception, cognition, decision making, or navigation support. Teleoperated robots are often used in areas where it is impossible or dangerous for human to access - such as in search and rescue [28] or planetary exploration [45]. Moreover, there are other applications for the remote scenario, such as e.g. remote learning units with real hardware [23].

In order to operate the robot at remote efficiently it is necessary that the operator understands the remote environment around the robot. This awareness of the remote environment and the robots surrounding is typically named as situation awareness [46], [47]. Sometimes also the term telepresence [48] is used with this respect. In contrast to the perception and understanding goal of situation awareness, telepresence aims for the feeling of being really present in the remote environment. This topic is mainly covered in the area of bilateral teleoperation [49] where a closely coupled master slave systems is designed for telepresence and stability.

Sheridan's [33] supervisory control offers a way to model the entire teleoperation system including the human, the remote environment the remote robot, different autonomy functionality, and the technical components involved. It also considers the nested control loops involved on the different levels of the teleoperation system. It is also possible to introduce the autonomous functionalities/behaviors of the system on both core design elements of the system: the user interface and the remote machine.

Often the supervisory control concept is only referred to systems where the operator really takes over a role comparable to the supervisor role like defined before (cf. Section 2.1.2). But the introduced supervisory control model allows for more. Adding the extension to scale the autonomy on both sides of the introduced supervisory control model enable to model different levels of autonomy ranging from full direct teleoperation to supervising an autonomous robot.

The autonomy design is also a core component and issue when designing a teleoperation system. Although full autonomy might not be reachable of feasible in many scenarios and tasks, autonomous functions included into teleoperation systems can increase the performance of the teleoperation system. There are lot of experimental results which support this e.g. [50], [33], [51], [52]. Different levels of teleoperation and autonomy are thinkable. All this levels lie somewhere between the extreme cases direct teleoperation and full autonomy. In addition there are also many ways to switch between this levels and to decide who switches between these levels e.g. [53], [54].

Another important issue is the communication link. Especially in bilateral teleoperation [49],[55], where a closely coupled master-slave system is realized over a communication link, parameters like for instance delay, jitter, and packet loss have a strong influence on the systems performance and behavior. In other teleoperation concepts where the abstraction between the user interface and the operated system is higher (e.g. mixes reality approaches [21], [19], [15]), the limitations are less but still significant. A consideration of the network properties can significantly improve the user interface quality [11], [10].

Concluding - in order to establish high performance teleoperation systems a lot of aspects need to be considered. The three major areas are the remote machine with its perception, control/autonomy systems and actuation, and the communication channels inbetween and the user interface system on the operator side.

2.2.1 Interaction between Humans and Autonomy

Schilling et al. [56] define autonomy as the capability for a spacecraft "to meet mission performance requirements for a specified period of time without external support," and "to optimize the mission product, e.g. the scientific measurements, within the given constraints". For the robot scenarios this can be adapted to: Autonomy is the capability of a system to fulfill specified tasks within given performance requirements and constraints for specified periods of time without human guidance. Autonomy of robots on any level can be achieved by combining technologies from signal processing, control theory and artificial intelligence.

Nevertheless even robots which are declared to be highly autonomous require interaction with humans. Various studies have been taken out, that analyze how humans can interact with autonomy, e.g. [57]. [58] gives a theoretical model for human interaction with automation. Supervisory control [33] allows the user to enter high-level commands for monitoring and diagnosis of the robot. Providing this type of control includes a certain level of autonomy of the different robots and makes the system capable to work even under low-bandwidth conditions or time delay in the communication link. Mixed initiative [53], adjustable autonomy [54] or collaborative control [59] [60] describe concepts of interaction between humans and autonomous behaviors of a robot. Collaborative control is based on event-driven human-robot dialogue. The robot asks questions to the human when it needs assistance for e.g. cognition or perception, i.e. the human acts as a resource for the robot. Since the robot does not need continuous attention from the operator, collaborative control is also useful for supervision of human-robot teams. Dias et al. [61] present challenges and a study for sliding autonomy in peer-to-peer human robot teams. [62] give suggestions for mode selection in HRI and summarizes observations from user studies. In particular it is mentioned, that users have problems to understand if the robot is in an autonomous mode. Goodrich et al. show in [63] observations from four experiments regarding autonomy in robot teams. They support the hypothesis that adaptive autonomy can increase the team performance. Stubbs et al. [64] provide results from a two year observational study with a science team applying a highly autonomous robot. They showed how common ground is disrupted on different levels of autonomy and the necessity of adapting the information and HRI communication dependent on the level of autonomy.

2.2.2 User Interfaces

There are different ways to realize the information exchange between the robotic systems and the human. All human senses can be considered for the user interface. Classically, the three human senses sight, hearing, and touch are addressed by HRI systems in order to realize information exchange between human and robot. Thus, communication is achieved with visual, haptic or audio elements in both directions from human to robot and vice versa.

Visual elements of the user interface can be implemented in various ways. While for remote operation this class of user interface elements is mainly represented by graphical user interfaces, for teammate or real peer to peer interaction gesture recognition and gestural cues of the robotic system provide a very natural visual interaction component. Hoffmann et al. [65] extensively use of gestural cues and gaze control in order to allow a human to naturally and collaboratively fulfill a task with a humanoid robot and the robot in a real teammate interaction role. It was possible to use this gestural cues to maintain common ground and have a collaborative task fluently solved. *Haptic or tactile* elements of the user interface can be used for instance in order to give the operator feedback of forces, surfaces or distances in the remote environment. They can also be used as input control. An operator can e.g. apply a certain force to a input device, which should be applied to an object by the robot. *Audio* elements can be used to allow a natural conversation with speech between human and HRI system. The HRI system can give feedback through spoken language or simple tones and might understand the human through speech recognition.

Dependent on the design goals and tasks assigned to a HRI system, any of these ways of communication can be combined to multimodal interfaces between human and robot like it is in human-human communication. Halme [66] shows such a concept where a human is enabled to work with a service robot in an outdoor environment on a collaborative task with various types of communication.

In this work the focus is on remote operation of robots. Thus, in the following and in more detail in Chapter 5 relevant, exemplary user interfaces for the task of remote operation and coordination of robots and human-robot teams are presented.

A wide range of user interfaces for the operation of mobile robots have been developed. For instance, [44] summarizes this in a review on several vehicle teleoperation interfaces. Conventional graphical two-dimensional user interfaces (e.g. [67], [68]) present the information as different interface components side by side without correlating them spatially. From the cognitive point of view this leads to the effect that the operator has to switch frequently between the different elements and has to correlate all the presented information in his mental model. These mental transformations may lead to significantly increased workload, hereby a decreased performance, most probably a decreased situational awareness and problems to maintain common ground between robot and humans.

More advanced graphical user interfaces try to present the information in an integrated fashion. Kadous et al. [69] describe one of the recently successful interfaces used in the Robocup USAR competition. [21] shows an approach, how to use mixed reality for teleoperation of mobile robots in the search and rescue scenario. Nielsen et al. [70] compares a three-dimensional world representation, integrating the video as projection in front of the mobile robot, against side-by-side presentation of map and video as well as two-dimensional maps together with video. An ecological interface paradigm is introduced in [52], that combines video, map and robot pose into three dimensional mixed reality display.

The existing user interfaces and its evaluations offer a good insight in the interaction process and allow to derive a certain set of basic guidelines for this thesis.

2.2.3 Studies and User Interface Guidelines

In the area of HCI a lot of studies have been performed and guidelines have been derived. Shneiderman [71] gives a summary on design guidelines for HCI. Raskin [38] points out the importance of a "humane interface", which is also very important for HRI [37]. The Apple Human Interface Guidelines[1] give an idea of the design process of the recently commercially very successful Apple user interfaces.

Experience has shown that the challenging application search and rescue is a good domain for the investigation of human-robot interaction. In this application area many studies were accomplished to determine requirements and guidelines for an efficient work between humans and robots. Navigation and solving tasks simultaneously or in joint human-robot teams assigns special requirements to the user interface.

Goodrich and Olsen [72] developed a basic list of seven principles in order to implement efficient human-robot interaction based on own experiments, experience and relevant factors from cognitive information processing:

- *Implicitly switch interfaces and autonomy modes.* Switching between different user interfaces and autonomy modes is often required in remote operation of mobile robots. This switching should happen implicitly without requiring any cognitive effort or attention by the operator. Still, it is important to give enough feedback to the operator to enable him to keep track of changes.

- *Use natural cues.* Humans have well well-calibrated and well-practiced mental models of interaction. The use of natural cues (e.g. icons, gestures, ...) allow for effective interaction as well-practiced response generation is possible.

[1] http://developer.apple.com/mac/library/documentation/UserExperience/Conceptual/AppleHIGuidelines/OSXHIGuidelines.pdf (17.09.2009)

- *Directly manipulate the world instead of the robot.* The teleoperation system should be designed with respect to the task and not with respect to the robot and interface per se. An example would be to enable the operator to point to a location in the world and the robot will do the rest autonomously.

- *Manipulate the robot-world relationship.* If directly manipulating the world is not possible, keep the focus on the information regarding the relation between robot, task and world.

- *Information is meant to be manipulated.* The operator should be enabled to directly manipulate information presented to him/her in order to perform a task.

- *Externalize memory.* The short-term memory of the operator is heavily used for remote operation of robots in order to fuse and maintain relevant information from the robot. Thus, information should be presented in an integrated fashion and important information should be made available as history in the interface.

- *Support attention management.* At lot of information is presented to the operator. If the interface has no good attention management the operator will most probably miss relevant, critical information.

Yanco et. al [73] performed a case study for the mobile robots designed for a rescue robot competition 2002. Based on their observations they derived the following guidelines for the interface:

- There should be a map allowing the operator to keep track of where the robot has been.

- Sensor information should be fused in order to lower the cognitive load of the operator.

- For multi-robot operation a single display should be used.

- Minimize the use of multiple windows.

- Provide spatial information about the robot in the environment and especially the robot’s immediate surrounding.

- Provide help on deciding which level of autonomy is most suitable.

Scholtz et al. [74] also derived guidelines for teleoperation interfaces in the field of urban search and rescue (USAR) from an analysis of the observations of the RoboCup USAR competition. User interfaces for the remote control of a mobile-robot in USAR scenarios require:

- A frame of reference to determine the position of the robot relative to the environment.
- Indicators for the health and status of the robot.
- Information from multiple sensors presented in an integrated fashion.
- Automatic presentation of contextually-appropriate information.
- As the main feedback of the robot, the camera image, might suffer from disturbances (e.g. communication problems or brightness), it is recommend to supplement the video by other sensors.
- The ability to self inspect the robot body for damage or entangled obstacles.

Burke et al. [27] participated with robots in training of urban search and rescue personal. They point out the large amount of effort required by the humans to percept and comprehend the information (>50% of the time) delivered by the robot in this exerices. Most of the studies and guidelines are based on designed experiments/tests and competitions. By contrast Murphy and Burke [29] compiled a list of four lessons learnt from the experience of applying mobile robots in urban search and rescue operations during real disasters (World Trade Center, 2001; Hurricane Charley, 2004; and the La Conchita mudslide, 2005) and nine high-fidelity field exercises:

- Building and maintaining situation awareness, not autonomous functions (e.g. autonomous navigation) is the shortage in robot operation.
- The robot should be considered as a source for information, not as something that needs to be controlled. The team members should be able to exploit it as an active information source through the provide HRI system.

- Team members use shared visual information for building shared mental models and facilitating team coordination.
- Both victims and human rescuers interact with the robot socially.

Steinfeld [75] provides a set of more technical hints based on interviews and hereby experience of six experts in robotics. Many concepts presented in this work are also based on a user requirement analyses [31] performed for implementation of rescue missions [32]. The results provided additional important guidelines for the design of robot capabilities and human interfaces. Many of the functionalities provided by the presented interfaces in this work are based on ideas and suggestions of potential end-users, which were mainly from the area of fire-fighting.

2.3 Human Factors

The overall teleoperation system performance is always limited by the human and the information which can be delivered to the human from the limited remote sensing of the robot. To optimize the presentation of the limited information from the remote environment and the robot, the user interface needs a careful consideration. It is the basis to understand the current situation and to make decision no matter what level of teleoperation is used - direct teleoperation or supervision of a robotic system with high-level commands only. In order to design an effective, efficient and usable human-robot interface it is necessary to include the user's perspective in the entire design and development process [37]. This is typically termed as user centered design. It demands to understand and incorporate the results and theories from technologies like telepresence and from human factors research, like situation awareness, common ground, operator workload, and spatial cognition in the design process of novel human-machine interfaces in order to reach maximum performance.

2.3.1 Telepresence

Steuer [48] defines telepresence "as the experience of presence in an environment by means of a communication medium". This means that the human operator receives enough information from the remote environment and the remote machine to feel physically present in the remote scenario.

A lot of work with respect to telepresence is performed in telerobotics [33]. Typically this type of telepresence in HRI is investigated with haptic systems (e.g. [76]). The challenges to keep these systems passive, robust, and transparent are especially high. The operator directly commands a master system, the generated inputs are transmitted over a communication network, rendered to a slave system, and the resulting displacements are send back to the operator. This closely coupled master-slave systems are covered by the research area of bilateral teleoperation [49].

In this work telepresence only plays a minor role as telepresence not necessarily leads to the maximum achievable task performance. In many applications the feeling of "being there" is not essential compared to situation awareness and common ground. Here, often systems which do not one-to-one render the gathered information from the remote environment achieve better results. Nevertheless, a lot of technologies and findings can be adapted to the demands of a task-oriented user interface.

2.3.2 Situation Awareness

A very precise definition from Endsley is cited in [46] - Situation awareness (SA) can be defined as "the perception of the elements in the environment within a volume of time and space the comprehension of their meaning and the projection of their status in the near future". This already indicates the three levels of the situation awareness model. The first fundamental level is the *perception* where most of the SA errors occur. The second level is the *comprehension*. This level deals with how people combine, interpret, store, retain and integrate information. The third level is *projection*. This level of SA is reached by operators which have the highest understanding of the situation and are able to project the situation in the future.

In the field of operating mobile robots, Murphy et al. [29] showed that this situation awareness is even more important than any autonomy or assistance function implemented in the robot. Especially for the application in the challenging search and rescue scenario, it emerged that situation awareness is the major bottleneck. Due to the importance, a lot of studies are performed in the area of HRI with respect to situation awareness, e.g. [30], [77], [78], [79], [80]. Adams [81] presents a concept to adapt the human situation awareness to unmanned vehicle awareness including automation aspects.

Todays systems in HRI generate a large amount of data and information. In order to generate and maintain situation awareness it is very important to filter out the information which is really needed, when it is needed and how it should be presented in an optimal way. It is generally accepted, that more data is not necessarily more information. Data overload is one aspect which can have negative effects on SA. It is very important in the user interface design process to avoid typical traps which have a negative effect on the situation awareness. Bolstad et al. [47] identify eight "SA-demons" which might have negative effects on SA: attentional tunneling; requisite memory trap; workload, anxiety, fatigue, and other stressors; data overload; misplaced salience; complexity creep; errant mental models; and out-of-the-loop syndrome.

Common Situation Awareness

Common situation awareness or also called common presence [82],[83],[84] is a technical concept to transfer task related knowledge between humans and robots. It is based on a virtual model of the environment to which all entities (humans and robots) involved in a cooperative task contribute information. This model provides the interfaces such that both robots and humans can understand the information in the same way through their specific cognition system. The concept strongly relies on localization and up-to-date map information. Thus, both humans and robots require localization systems and optionally mapping abilities. For humans this can be achieved by a personal navigation system like introduced in [84], [85]. Driewer [86] presents a related extension to the situation awareness model for human-robot team setups.

2.3.3 Common Ground

The common ground framework [87] was actually developed to understand communication and collaboration between humans. Nevertheless, it is also a key issue for HRI, as the involved human also needs to have some common ground with the robot in order to come to decisions. Common ground is needed for any coordination of a process between independent entities, what means that between the coordinating entities is mutual knowledge, mutual beliefs, and mutual assumptions. It is also necessary to continuously update the common ground. In order to properly update common ground a so called grounding process is needed in which the entities try to reach the mutual belief. Grounding in HRI is especially needed when (as in most scenarios) autonomy comes into play. The interaction system needs to support grounding processes such that the human is able to form mental models of robots how the robots collaborate. In most interaction systems there is no way of reasoning why a machine is doing what, although it is a very important issue to enable a human to collaborate with a robot comparable to human human collaboration.

Jones and Hinds [26] made a study on SWAT (special weapons and tactics) teams and how they coordinated and maintain common ground inside the team during their extreme missions. They use these results to adapt their user interfaces and show how important common ground is when the entities are not located next to each other. Stubbs et al. [88] give some examples of information which should be fed back to a human operator or supervisor, e.g. technical information about system status, status of activities requested by the human, exact information about failures and its reasons, and information about constraints under which the robot operates.

Stubbs et al. [64] present a study on teamwork with a science team, an engineering team, and a mobile robot. Their results also show that grounding is a key issue in HRI. They state different autonomy levels of the robots require different information to be exchanged in the team in order to enable the grounding process inside a human-robot team. Also the way of information exchange very much depends on this autonomy.

There is a clear overlap between situational awareness and common ground. In HRI the common ground focus on whole information exchange and collaboration process of human and robot, while situation awareness focus more on the user's information needs.

The robot's abilities for grounding are especially desirable in interactive social robots [89] in order to make the robot more natural and efficient. As it very much supports the situation awareness, the common ground and grounding framework is also very important when designing teleoperation systems. It allows for more efficient user interfaces with better performance results. In teleoperation system which support grounding in a very good way, a much better acceptance of autonomy and assistance functions of the robots can be expected.

2.3.4 Workload

Workload covers mental and cognitive workload as well as physical and temporal workload [37]. A high workload can e.g. lead to reductions in vigilance and problems to maintain accuracy. The same can be the case for very low workload because the operator is under stimulated, what might lead also to reduced vigilance, boredom, etc.. The mental processing capacity varies from human to human. This mental capacity can be influenced by many circumstances, e.g. stress, lack of sleep, environmental conditions, ... If the mental workload is too high, the operator may take wrong decisions. If the mental work load is too low, the operator may not properly monitor the robot and loose track of what is currently ongoing, such that he/she get problems to maintain situation awareness and common ground. Mental workload can be optimized by a good task distribution between operator and robot. It is essential in HRI design and development to address this topic, such that the operator is not continuously under or over loaded. In some cases additional mechanisms might be needed to adapt the task allocation according to the current operator's workload.

2.3.5 Spatial Cognition

Spatial relations and spatial cognition play a key role in teleoperation and navigation tasks for robots. Related to this there are various aspects which require consideration when designing a user interface:

- *perspective taking of the human operator.* The operator needs to be able to take over different frames of reference in HRI systems. This needs to be supported by the HRI system.

- *mental transformations between user interface elements.* The user interface itself has in general different frames of reference. Thus, a lot of mental transformations are required by the operator in order to fuse the different information in his/her mental model

When people communicate about physical spaces and tasks, perspective taking plays a substantial role. People switch perspective around 45% of the time when they talk without being interuptted [90]. For HRI systems a comparable importance can be identified. HRI system need to be able to meet perspective taking demands, in order to optimize the information exchange between human and robot. Trafton et al. [90] analyze perspective taking in HRI for a collaborative tasks between human and robot for a space scenario. HRI system should be able to solve ambiguity if it occours. Fong et al. [91] show an system architecture for solving collaborativly tasks in human-robot teams also for the space scenario, which has a dedicated component ("Spatial Reasoning Agent") responsible to cover the demands of perspective taking in the setup human-robot team. Five perspective taking frame of references can be identified when communicating in HRI systems:

- exocentric (e.g. "move in northern direction")
- egocentric (e.g. "move to my left side")
- addressee-centered (e.g. "move to your left")
- object-centered (e.g. "move to the table")
- deictic (e.g. "move over there")

Perspective taking is a major issue for real peer-to-peer human-robot interaction and in human-robot teams. For remote operation at first the different frames of reference which can occur in an user interface need to be considered. A lot of mental workload of the operator is caused by effort required for mental transformations. Mental transformations are needed to fuse the delivered information by the user interface in one mental model of the operator. This may significantly reduce the overall system performance.

Wickens et al. [92] analyses the implications for the user interface design with respect to human performance limitations related to mental transformations in order to support the operators understanding of three dimensional spatial information and motion. Six basic frames of reference for the operator can be identified in a teleoperation system:

- *World frame.* This is global frame of reference in which operation occurs.
- *Ego frame.* Typically the location of the observer/operator.
- *Head frame.* The orientation and location of the operator’s head with respect to the trunk.
- *Vehicle frame.* The frame associated with the operated vehicle.

- *Display frame.* The frame of reference of information on a display.

- *Control frame.* The frame of reference of control input devices.

Different user interface components might even introduce more frame of references in the system. Assuming at maximum six degrees of freedom (three for position and three for orientations) per frame of reference already shows the complexity for the human to match them. It even gets worse if one considers that frames of reference can also have motions in all degrees what leads to another six degrees of freedom. Thus, a key design criteria is to setup a user interface with the demanded information while minimizing the needs for mental transformations which corresponds to the task. It might not be possible or undesired to avoid all mental transformations. Thus, it is important to look at the different costs for mental rotations. Hereby the task is very important. Wickens et al. [92] provide an analysis on these costs. A key finding is that mental rotations have higher cognitive demands than translations. The costs for mental rotations are not linear. For example if an operator has two maps which have a rotational misalignment, the cognitive demands increase with a higher misalignment. Over 45 degree of misalignment these demands are significantly higher. A major design implication from the findings is that for travel or navigation tasks, the heading of the vehicle should always correspond with the upward direction of the map in contrast to fixed north-up maps. Interestingly, a three dimensional scenario and a two dimensional map where the forward direction (3D) and upward direction (2D) correspond can easily be matched by the operator. If two or more user interface elements show the same scene in different frames of reference, the frame matching of the operator is supported by applying the concept of "visual momentum". Common elements need to be visualized similar and highlighted in both representations.

The results of the different studies in the area of perspective taking and spatial reasoning/thinking clearly point out the importance to minimize the amount of different frames of reference to those absolutely necessary. The needed mental transformations by the operator need to be optimized for the task when designing user interfaces for HRI.

2.4 Evaluation of Human-Robot Interaction

In order to evaluate the performance of human-robot interaction systems both the performance parameters of the technical components as well as the performance parameters of the involved humans need to be considered. The evaluation of the human robot system from a holistic point of view is difficult and complex. It often requires a limitation to specific task-related parameters.

A lot of evaluation methods already exist in HCI. One example is the "Goals, Operators, Methods and Selection rules" (GOMS) method [93][38]. It provides an analytical way to quantitatively evaluate user interfaces without users and to hereby predict the performance of a system. Another analytical approach is the measurement of interface efficiency [38] based on information-theoretic concepts. Drury et al. [94] provide a case study on how to apply GOMS for HRI. Clarkson and Arkin [95] provide hints how heuristic evaluation from HCI can be adapted to HRI.

HRI research showed the importance of human factors for HRI [37] like for HCI. Thus, the user should be involved in the whole design process. Especially the final evaluation has to be performed with users. Due to the complexity, HRI systems are very difficult to quantitatively evaluate and compare. Thus, it would be even more important to have standards to evaluate HRI systems in order to identify interdependencies and key influencing factors. Human factors research provide a good background and models to explain different effects of systems, their elements and functionalities on human robot interaction. But often these human factors are difficult to measure directly. The NASA TLX (Task Load Index) [96] is a widely used [97] method to subjectively assess the workload of an operator through a rating of the demands on different sub-scales. The SAGAT [98] test for instance is designed to measure situation awareness directly on all three levels of situation awareness. The tested system is frozen at selected random times and the user is asked a set of questions. The answers can be compared to the real situation and provide an immediate objective feedback on situation awareness. Drury et al. [80] propose the LASSO technique to evaluate situation awareness along five awareness categories: location awareness, activity awareness, surroundings awareness, status awareness, and overall mission awareness. The major data source for the evaluation here is the comparison of utterances during the test runs (think aloud approach) with the situation at that time.

Many approaches exist to find metrics to evaluate human-robot interaction e.g. [36], [72], [73], [74]. Steinfeld et al. [99] suggests a summarized set of common metrics for HRI. Due to the incredibly large range of robot applications the focus of these metrics is on task-oriented mobile robots. These metrics are applied as experimental design background for this work. Thus, in the following these metrics are shortly introduced. Five task categories are selected. The *navigation* task covers all related to global, local navigation including obstacle encounter. The *perception* task relates to task like search, surveillance, target identification but not to perception required by other tasks. The *management* task covers the coordination and management single entities or groups of a human-robot team. For the *manipulation* task there is a direct physical interaction between robot and environment. *Social* tasks are all tasks related to social interaction between humans and robots e.g. in health care. Task metrics for these five task categories can include:

- *Navigation.* Effectiveness measures: percentage of navigation task completion, covered area, deviation from planned route, number of successfully avoided obstacles or overcome obstacles; Efficiency measures: time for task completion, time operator needs for the task and average time for obstacle extraction; Workload measures: operator interventions, average time for an intervention, effectiveness of intervention and the ratio of operator time to robot time.

- *Perception.* Passive perception (interpretation of received sensor data) - Identification measures like percentage detected, recognition accuracy; Judgment of extent measures like absolute and relative judgment of distances, sizes; Judgment of motion like absolute or relative velocity to other objects.
  Active perception: (actively obtaining sensor readings in order to e.g. solve a problem or reach confidence about a situation) - Active identification in addition includes e.g. the time or effort to confirm identification and improvements and the overall effort; Stationary search measures like e.g. accuracy of detection; Active search measures like e.g. time and effort, number of identification errors.

- *Management.* Fan out [72], Intervention response time, level of autonomy discrepancy.

- *Manipulation.* Degree of mental computation, contact errors (unintentional collisions).
- *Social.* Metrics for social robots, differ from the more performance oriented metrics for the other categories. Metrics are interaction characteristics, persuasiveness, trust, engagement, and compliance.

For this work the task specific metrics for navigation, perception and management are most relevant. The task specific metrics are complemented in [99] by the following common metrics:

- *System performance.* Quantitative performance measures effectiveness and efficiency, subjective ratings, appropriate utilization of mixed initiative.
- *Operator performance.* Situation awareness, workload, accuracy of mental models of device operation.
- *Robot performance.* Self-awareness of the robot, awareness of human, autonomy.

Task-oriented performance measurements allow for an evaluation and can be explained with the help of models from human factors in many cases. Still qualitative results from the user tests provide important aspects for the system evaluation. This can be gained through questionnaires and observations during the experiments. Thus, in this work a combination of quantitative performance measurements, subjective ratings of the test participants and qualitative observations are used to evaluate the presented concepts.

2.5 Designing the Teleoperation System

2.5.1 Model of the Teleoperation System

The supervisory control model is often used for teleoperation systems. Sheridan [33] defines that supervisory control in the strict sense means that one or more human operators intermittently programs and continuously receives information from an system which itself closes an autonomous control loop between sensors and actuators of the machine. In the less strict sense according to Sheridan the autonomous control loop is not necessarily needed. Sheridan developed the man-machine system diagram for the supervisory control model which provided the basis for the diagram in Figure 2.1. Although a great development of technologies and concept has taken place in the last years, the basic underlying conceptional model can still be applied to most teleoperation systems. While earlier manual control and direct teleoperation were really direct by means of directly interconnected inputs, which are provided to the human operator, with actuators of the operated machine, nowadays almost all systems include autonomous control loops to a certain extent.

Figure 2.1 shows the adapted model which served as basis for this work. On the one end the human interacts with input devices with the human interactive system and receives information rendered on any kind of rendering device. On the other end the task interactive system interacts with the remote environment through its actuators and gathers information about the environment and situation through its sensors. Both systems exchange information through communication. It provides also the different components, parameters and technologies which influence the elements and its nested control loops and information channels of the model. The ten basic closed control loops of such systems are shown. The level of autonomy which can be modeled with this approach can be somewhere on the continuum between direct operation and full autonomy. Current research shows that autonomy has often reached a level where the one way relationship of a human only commanding and receiving information is broken up. Concepts are available where a more bilateral relationship is reached between humans and robotic systems. Initiative can also be taken by the robotic systems [51]. Hereby, a human can also be used by the robot as information source and ask for help to solve a task. In order to use the heterogeneous capabilities in human-robot team in an optimal way and to achieve optimal team performance, this new relationship is needed.

The design constraints and limitations for a teleoperation system can be found at the top and the bottom of the list provided in Figure 2.1 - the human factors and the task related factors. These factors and constraints for a specific task or application can only be changed marginally. They determine the optimum which can be reached at maximum. Everything in-between is a matter of research and development to reach the optimal system. Each of the components has a more or less task depending influence on the system performance, which raises the need

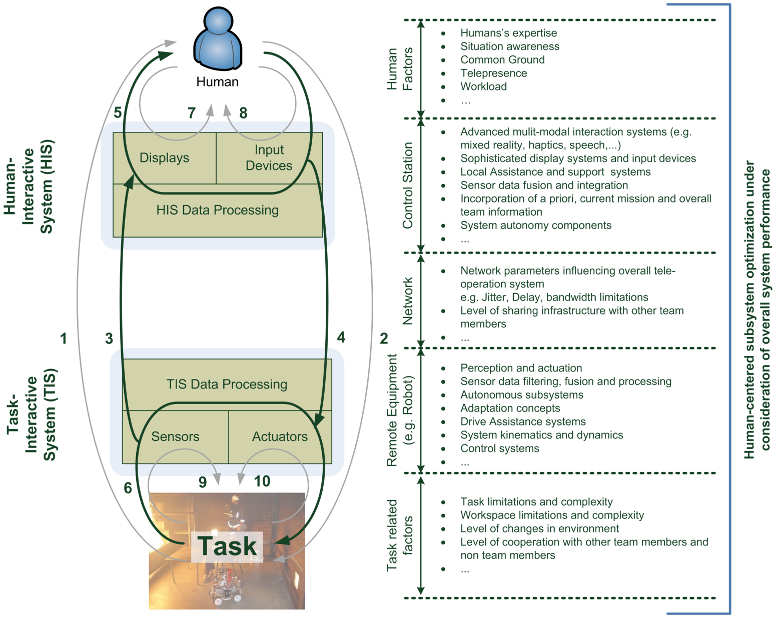

Human
5
7
8
Displays
Input Devices
HIS Data Processing
Human-Interactive System (HIS)
1
3
4
2
TIS Data Processing
Sensors
Actuators
Task-Interactive System (TIS)
6
9
10
Task
Human Factors
Humans's expertise
Situation awareness
Common Ground
Telepresence
Workload
...
Control Station
Advanced mulit-modal interaction systems (e.g. mixed reality, haptics, speech,...)
Sophisticated display systems and input devices
Local Assistance and support systems
Sensor data fusion and integration
Incorporation of a priori, current mission and overall team information
System autonomy components
...
Network
Network parameters influencing overall tele-operation system e.g. Jitter, Delay, bandwidth limitations
Level of sharing infrastructure with other team members
...
Remote Equipment (e.g. Robot)
Perception and actuation
Sensor data filtering, fusion and processing
Autonomous subsystems
Adaptation concepts
Drive Assistance systems
System kinematics and dynamics
Control systems
...
Task related factors
Task limitations and complexity
Workspace limitations and complexity
Level of changes in environment
Level of cooperation with other team members and non team members
...
Human-centered subsystem optimization under consideration of overall system performance

Figure 2.1: Control loops based on the supervisory control model (adopted from [33]) and related system elements/parameters for a teleoperation system for a mobile robot.

to look at and design such systems from the overall system perspective.

Figure 2.1 provides a teleoperation task related interaction relationship between humans and an robotic system inside a human-robot team. The extension of the supervisory control concept for entire human-robot teams which also very much influenced the work presented here can be found in [86].

Breaking it down to a technical data flow model for a specific teleoperation system inside a human-robot team, the model in Figure 2.2 can be derived. It shows where which data is generated, where information is processed and where potential disturbances can occur. Information processing and autonomous functionalities can be realized on both sides of a teleoperation system. This very much depends on the task and the corresponding constraints.

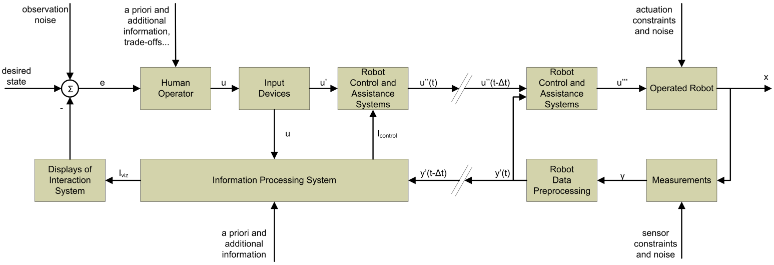

observation noise
a priori and additional information, trade-offs...
actuation constraints and noise
desired state
Σ
-
e
Human Operator
u
Input Devices
u'
Robot Control and Assistance Systems
u''(t)
u''(t-Δt)
Robot Control and Assistance Systems
u'''
Operated Robot
x
u
$I_{control}$
Displays of Interaction System
I_{viz}
Information Processing System
y'(t-Δt)
y'(t)
Robot Data Preprocessing
y
Measurements
a priori and additional information
sensor constraints and noise

Figure 2.2: Technical data flow model for the proposed teleoperation system for a mobile robot.

Concluding, when designing a human-robot interaction system, it needs to be considered that the human always is included in the control loops of the overall system. The system are designed to support humans, hence the human is always needed in the design process and the operation. As robots are getting closer to the human, HRI is a quickly growing field in recent years. The complexity of parameters influencing HRI systems and teleoperation system in particular, which can be seen in Figure 2.1 and 2.2 reveal the need for a system perspective of the system design and to put "the human in the loop".

2.5.2 Design Goals and Demands

In this thesis new concepts and components for advanced teleoperation systems are developed and evaluated. As reference task navigation of mobile robots and exploration with mobile robots is considered. In the following a brief outline of design goals and demands is given which has been elaborated based on literature and own tests and experience. It serves as guideline for a task oriented teleoperation system performance optimization in this thesis.

Design Goals for the User Interfaces

The most significant influence on the overall teleoperation performance has the interface between the human operator and the system. On the one hand the user interface provides feedback from the operated robot in the remote environment. On the other hand it enables the user to give commands and thereby control the robot through specific input devices. The feedback component can be realized by any combination of elements addressing the different senses of the human (e.g. visual, audio, haptic,...). As input devices also very different devices and their combinations are possible. The spectrum ranges from simple keyboard/joystick inputs to speech recognition, user tracking and gesture recognition systems.

The different design goals for the user interface and the most important challenges can be summarized as follows:

- *Design an integrated display.* An integrated display for the user interface should be used in order to display gathered information and sensor data which has a spatial relationship to the environment or robot, at the correct spatial location. Appropriate representations of the environment as platform for the integration of the information is needed. It should be possible that the human can reference information to certain locations through landmarks and spatial cues. The selection of viewpoints should be enabled in the integrated display. It is important to provide the operator enough information to reference the viewpoint change spatially.

- *Minimize mental transformations.* The need for mental transformations should be minimized as far as possible without disturbing natural relationships. This includes all interface components provided to the operator, e.g. graphical user interface and force feedback components and input devices. If mental transformations are needed, avoid larger mental rotations.

- *Use visualizations which are natural to the human and natural cues.* It is helpful to make use of the mental models of interactions and objects each human already has. Humans and machines work with different mental models. The user interface needs to be able to provide a seamless translation between these models.

- *Reduce user interface elements.* Minimize the different elements of the user interface to the needs of the current task and situation. Information which is not needed should be faded out. If information can be fused naturally, it should be present in a combined user interface element. If it is needed that different elements show the same information in different context (e.g. viewpoint) it is important to provide the operator matching cues between this elements.

- *Support attention management.* The operator's attention needs to be directed to crucial information. This might for instance be information which requests immediate action or information which is needed to maintain situation awareness or common ground.

- *Externalize memory.* Support the operator by enabling him/her to build up a history of information insight the interface which might be needed for situation awareness. This enables to relieves the operator's short-term memory from memorizing all necessary information in order to keep track of what happens and to fuse it his mental model.

- *Support gaining and maintaining situation awareness.* Support for gaining and maintaining situation awareness is needed in different categories. With respect to the current situation knowledge about the surrounding local environment is needed and a basic knowledge in which global environment the whole thing takes place is needed. With respect to the mission and the task, awareness about the current goals, the current task allocation and the progress of the different task is needed. If multiple entities cooperate information of the work load, work progress and allocated task might be necessary. With respect to the robots and other entities, information about status, capabilities and behavior are needed.

- *Support awareness of autonomous behaviors.* Autonomous functions are desirable with respect to task performance, i.e. robots can take over control about themselves by certain autonomous behaviors (e.g. obstacle avoidance). If the robots are not completely manual controlled, the human operator have to be properly informed about the action of the robot. This is needed to gain and maintain common ground between humans and robots. Otherwise, a frustration and mistrust might result. In the worst case the human operator will deactivate all support functions. At the best the user interface supports an optimal combination the the skills and capabilities of humans and robots.

In the context of this thesis also a preliminary study with respect to user interface design for human-robot teams in general has been performed. Details on this can be found in [18].

Demands on the Robots and Communication System

In addition to the user interface as the interface for interaction between human and the system itself in a teleoperation scenario, the other core challenge on increasing overall teleoperation performance is to optimize all the other crucial components of the teleoperation system from a system perspective and not from a single component perspective.

- *Implicit switching of autonomy modes.* Navigation of the robots can vary from direct teleoperation to fully autonomous movements. In many cases different levels of autonomy are desirable in one mission according to current situation. Thus, switching between these functionalities should happen with minimum cognitive effort for the human operator. Ideally the levels of autonomy are not discrete but continuous.

- *Carefully design autonomy and assistance functionalities.* Driving assistance system and autonomous functionalities should be designed and implemented very robust. Stability problems and failures can significantly disturb the teleoperation performance. Human operators will get confused and i.e. switch of this functionalities because of mistrust or in worst case even abort a mission.

- *Human centered environment perception and representation.* The human mainly relies on visual impression in his daily life and lives in a three-dimensional world. Thus, visual feedback from the environment and 3D representations are natural and can be very helpful. Some map representation correspond better to human cognition than others. For a human a topological description of a path might be more natural than a metric one.

- *Appropriate environment perception.* Robots need sensors which are corresponding to their working environment, the application scenario and the actual task of the robot in order to support the spatial understanding of the environment. The environmental perception sensors have to provide the needed accuracy and need to enable robustness of the implemented autonomous and driving assistance functionalities according to the current situation.

- *Optimize communication system.* All available and needed information and commands in a teleoperation system are transmitted over a communication link. Thus, significant disturbances can be induced by this component. Thus, a careful selection of the communication technologies and a careful design of protocols, communication architectures and topologies are needed.

3 3D-Perception and Communication Networks for Teleoperation Systems

In order to enable a successful and an efficient operation of mobile robots from remote (no matter of the level of autonomy) a suitable communication network, and environment perception are demanded. Only well designed communication setups, a good environment perception, and its comprehension by the mobile robots and humans enable the large application potential of mobile robots, ranging from industrial transport robots [100], to robots applied in emergency operations, and search and rescue [31], [32], and robots for space exploration [56],[101].

The application scenario and the type of interaction between humans and robots influences the requirements on the different communication layers of the ISO/OSI model [102]. For instance, if a robot replicates human movements in a master-slave constellation over a communication network, there are high constraints on this network (e.g. with respect to the tolerable delay). If the robot is teleoperated with a certain level of autonomy included in the teleoperation system, these requirements become already loose, or if robots cooperate autonomously and actively with humans in order to fulfill a joint mission goal, the communication network is only needed to send status updates, send findings and receive new data asynchronously from mission coordination center.

In addition, for the successful application of mobile robots in dynamic real-world environments, a three-dimensional perception of the robots' surroundings is a must - for both autonomous reactions as well as for assisted teleoperations. Currently, mostly nodding 2D laser range finder or cameras are used to retrieve 3D data. In the last years the new technology of time-of-flight range cameras offer here a small and light-weight camera generating 3D images, which allows for an even better environment perception with minimal required processing time.

Nevertheless, for such innovative sensor principles, further research needs to be invested to characterize performance details and to optimize parameter settings for typical application scenarios in robotics.

This chapter summarizes the contributions of this work related to two major components of teleoperation systems - communication and environment perception. The first part focuses on the implementation of teleoperation inside ad-hoc communication network setups for joint human robot teams. The adaptation of the usage of the application layer for mixed reality user interfaces dependent on the underlying IP based communication on top of a specific physical communication technology is considered. Therefore, an approach to optimize video traffic for teleoperation inside such a communication network for human-robot teams is introduced and evaluated. In the second part the sensor characteristics and application potential of a PMD camera for three-dimensional perception of the remote environment for mobile robots are analyzed. Specific aspects related to parameter optimization and limitations in measurements are discussed. In particular, range images in unstructured and changing scenes are of interest. Specific properties of mobile systems are taken into account to generate appropriate range images for navigation and mapping tasks in robotic applications. This is also incorporated into the considerations and developed concepts in Chapter 5. Compared to other related work, here the focus is on the human point of view. Thus, based on the application example of local 3D panoramas a qualitative user study was performed in order to get an idea how good humans can orient, navigate, and detect objects in this type of 3D environment representations.

3.1 Enabling Technologies

3.1.1 Communication Technologies

Nowadays, a lot of standard communication technologies are available in the consumer market, which are affordable and also well tested. Some of them already offer the basic characteristics and designs in order to be applied in systems where humans and robots work together. Setups of communication networks for human-robot systems need to be able to support different classes of communication traffic. Dependent e.g. on the tasks, the autonomy of the different systems, the team configuration, and also the type of user interface, different demands on the communication network arise. For mixed reality user interfaces the position in the virtual continuum (cf. Figure 5.1) has a significant influence on which requirements can be met. For instance an augmented reality interface where the camera image as real world component is in the focus, has high demands on the bandwidth and the jitter of the communication link compared to a pure virtual user interface which can work more or less asynchronously.

Most often, wireless communication is demanded and chosen to distribute and share information between robots or between humans and robots forming a joint team. This includes the transmissions of sensor data from the robots, observations from the humans, commands, and plans to the different team entities from the human coordinators. Compared to very specialized missions like Mars exploration with Rovers [101] or short range investigation of collapsed buildings [29], where also tethered communication can be applied, for almost all applications of mobile robots in changing environments, changing operation areas, and changing tasks this wireless communication is a must. While in the past mainly proprietary communication systems [68] and protocols were used for mobile robotics, nowadays a clear trend towards IP[1] based communication systems can be identified. The main drivers for this trend are the availability of well-tested, high performance communication hardware and the overall trend in the consumer and professional, industrial market to use IP based networks. The selection of an IP based communication systems enables a seamless integration of robots into existing networks. This is especially interesting in industrial scenarios, the area of service robotics and home automation. Figure 3.1 shows how a future robotic team can be setup for instance for the exemplary scenario in this work - a search and rescue mission. Future missions where humans and robotic systems work together to reach a common goal require communication networks with interfaces for heterogeneous communication technologies, systems with world-wide access to information and human expertise. Especially in the exemplary application search and rescue, where at least parts of the communication infrastructure might be destroyed in the local mission workspace, the systems in the local mission workspace need to be able to setup their own communication infrastructure on demand or bridge communication gaps. Heterogeneous communication technologies can supplement each other. In this work the two broadband communication technologies IEEE 802.11 wireless LAN and the mobile phone network UMTS as outlook are

[1] IP - Internet Protocol

considered as wireless interface technologies.

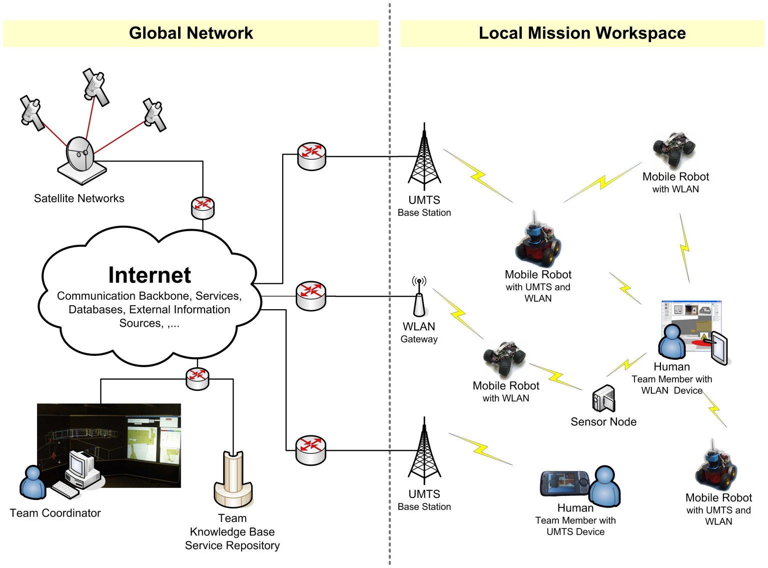

Global Network
Local Mission Workspace
Satellite Networks
Internet
Communication Backbone, Services, Databases, External Information Sources, ...
Team Coordinator
Team Knowledge Base Service Repository
UMTS Base Station
WLAN Gateway
UMTS Base Station
Mobile Robot with WLAN
Mobile Robot with UMTS and WLAN
Human Team Member with WLAN Device
Mobile Robot with WLAN
Sensor Node
Human Team Member with UMTS Device
Mobile Robot with UMTS and WLAN

Figure 3.1: Future example scenario of a heterogeneous network of mobile robots and humans working on a joint mission.

As part of this work on the demands of mixed reality teleoperation systems also a simulation setup was proposed in order to test communication setups with real communication devices and without the effort of setting up the whole human-robot team in a real environment. The communication is realized with real communication devices and the environment is simulated with a three-dimensional simulation including a good physics engine - USARSim [103][104]. Further details on this and the test runs with a distributed collaborative collision avoidance [105][106] algorithm can be found in [9].

Wireless Lan In many cases IEEE 802.11 wireless LAN (WLAN) is used as underlying technology for the wireless network in-between mobile robots, humans and other systems. Nowadays, modern telecommunication equipment with small power consumption and interfaces for easy integration are available for these wireless LAN standards. Further advantages of WLAN are the availability of a relatively high bandwidth dependent on the standard (e.g. 802.11b max. 11 MBit/s, 802.11g max. 54 MBit/s) and the high flexibility in integrating new protocols or extending features of available protocol implementations. WLAN devices can work in infrastructure mode and in ad-hoc mode.

As currently the cooperation of several vehicles is very important, challenging problems like nodes acting autonomously as communication relay, a highly dynamic and variable network topology (some network nodes may leave or join the network at any time), routing problems, and several data streams and sources with different bandwidth requirements have to be solved in order to successfully setup an appropriate communication network. Often, ad-hoc capabilities have to be present to meet this needs. The usage of IEEE 802.11 wireless LAN enables to realize an affordable wireless communication which satisfies these demands for teams of mobile robots and humans.

Wireless Ad-Hoc Networks Human-robot team scenarios demand for decentralized communication networks which allow for any to any communication. During the last years, several teams achieved remarkable results in the research area dynamic wireless communication network topologies. In 2007, Rooker and Birk presented multi-robot exploration with robots using wireless networks [107]. The University of Pennsylvania presented a mobile robot team connected via wireless network which performed localization and control tasks [108] in 2002. [109][110] placed relay nodes on demand to setup the required telecommunication infrastructure and in [111][112] mobile robots are used as relay nodes where each mobile node not only works as host but also as router for data packets of other nodes. In [113] a live audio and video data transmission via a multi-hop wireless network is demonstrated. In addition, several systems of rovers with autonomous functionalities [114], groups of unmanned aerial vehicles [115], as well as heterogeneous multi robot systems were proposed. For ground based systems Chung [116] presented a testbed for a network of mobile robots. In the field of rescue robotics [117], or for integrating UAVs into IP based ground networks [118], the use of wireless networks is quite common nowadays. With respect to unmanned aerial vehicles (UAVs), [119] presented a system using an access point running in WLAN infrastructure mode onboard the UAV. [120] presented a system for communication between a ground station and a UAV using WLAN in combination with a high-gain antenna and radio modem. These wireless ad-hoc networks offer a lot of advantages in contrast to static wireless network configurations, but also raise a lot of new challenges in the system design. Wireless communication always implies unpredictable communication delays, packet loss, or in worst case the loss of the link, which makes the provision of the required quality for teleoperation or control tasks rather complex [121].

The already mentioned ease of use, the affordable prize, small weight, and appropriate power consumption makes WLAN often a reasonable choice to set up dynamic topologies providing direct and indirect any-to-any communication of each network node. In addition, also benefits like redundant communication links in larger networks, no central administration, and a distribution of the traffic load in large networks are present. Of course, these advantages can only be used with rather complex and special routing protocols providing each node the necessary information of the network topology. The nodes itself are working as routers and must store the routing information of the complete network locally. In the field of wireless telecommunication, more than 80 different approaches for ad-hoc routing mechanisms of different types (classes) were developed. Well known classes of these protocols are pro-active, reactive, or hybrid (pro-active and reactive) protocols but also flow oriented, power aware, multicast, position based, or situation aware approaches are available. The number of implemented protocols which have reached the status to be used in lab-environments is much smaller [122] and if an appropriate real world scenario with mobile robots is considered, the alternatives are quite small. Some simulations for performance evaluations for larger scale telecommunication networks were done in the past [123][124][125] and based on [126][127][128][129]. Recently AODV, DSR, OLSR, and BATMAN and the application of multi-hop communication were analyzed with respect to mobile robot teleoperation [12][130][8]. The work published in the before listed publications served as basis for the communication of the later presented concept.

Outlook - Teleoperation over UMTS connections The upcoming high-bandwidth networks for mobile phones or mobile Internet like UMTS[2] offer another new widely used and commercially available technology with high potential for the application in mobile robot teleoperation. Up to now, the coverage of these networks has increased in a way that at least all bigger cities have access to broadband networks. This "everywhere availability" in large areas is a major advantage

[2] UMTS - Universal Mobile Telecommunications System

for any telematic application compared to a solution where infrastructure initially has to be built up and maintenance effort is necessary. Besides the operation of mobile robots, this includes applications like tele-maintenance, tele-support, or tele-monitoring of industrial facilities. These applications in general demand also these high-bandwidth communication links. Often, an additional own infrastructure might be not feasible or possible and thus, the high-bandwidth telephone networks provide there an interesting alternative. The design of this networks in a way that they should provide a seamless transition between different communication cells, the scheduling concepts for resource allocation for the different users and the ability to work indoors to a great extend are also very promising issues for robotics. In particular, the application area of service robotics can largely benefit from these characteristics.

On the other hand, the mobile phone networks like UMTS are designed for different purposes and under different constraints. Therefore, it is important to investigate the critical parameters of a communication technology like UMTS in order to adjust the possible communication parameters on the application layer in a way to realize the optimum usage of this technology for the own application, here mobile robot teleoperation. First analysis performed in the context of this thesis can be found in [7][131].

3.1.2 Network Parameters and Video Streams

Network parameters have a major influence on the overall performance of human-robot teams in scenarios where remote interaction or components exist. For teleoperation or supervisory tasks typical network parameters like communication bandwidth, delay and jitter are of major importance. In teleoperation systems in general a data flow of commands and information to the robot exist and a data flow back to the operator with sensor data and in most cases a video stream. For teleoperation different levels of automation can be applied. A higher level of automation (e.g. through autonomous obstacle avoidance) in cases where the operator is able to maintain situation awareness and common ground in general leads to a better navigation performance in time and errors. In addition, a higher level of automation leads to a higher robustness against the duration of communication delays and its jitter. The results of an investigation of the influence of latency in mobile robot operation for four different levels of automation from Luck et al. [132] and own experience with such systems supports this hypothesis.

A video stream is still one of the most important data sources for the user while operating a mobile robot at remote. The video feedback still delivers the most and richest information from the remote environment to the operator. This detailed information from the remote site is needed to increase and maintain the situation awareness (cf. Section 2.3.2) and common ground (cf. Section 2.3.3) between robot and human operator as basis for any future decisions and commands done by the human operator. Human operators have comprehensive capabilities to interpret the displayed image information, but therefore, some constraints must be fulfilled. Constant frame rates and delays below a certain threshold are a minimum requirement to use video for teleoperation. A certain delay of teleoperation systems is inevitable but a human operator can adjust to a certain extent to it. In general it is much easier for an operator to adjust to an constant delay compared to an varying delay. The selected framerate, the resolution and the quality of the video stream determine the required bandwidth from the communication links. Dependent on the human teleoperation task different characteristics of the video stream are important. If a navigation task is considered, the most important parameters are a high frame rate, low number of frame losses, and a constant inter-arrival time between the frames. Compared to these parameters the quality and resolution of the video stream is less important for navigation. On the other hand if the human has a search task (e.g. identify objects in a delivered video stream), the quality and resolution have a higher importance than the frame rate. Sheridan [33] shows the influence and importance of framerate and resolution on the performance of a telerobotic systems.

In order to compress and transmit video data a lot of standards and codecs exist. Typical examples are MPEG-1, MPEG-2, MPEG-4, DivX and MJPEG. The MPEG family and the DivX codec make use of redundancy between subsequent images in a video stream. Thus, a better bandwidth usage is possible. For MJPEG each video frame is transmitted completely and so a higher bandwidth is needed. Typically network cameras like applied in this thesis offer MPEG-4 and MJPEG video stream with different hardware compression rates. The experience with this systems showed that the selection for MJPEG is often the better choice. Although the bandwidth requirements are higher, a short communication drop out and packet loss does not cause a disturbance of a longer sequence of video frames like for the other codecs. In addition, the implemented compressions for MJPEG introduce a much lower processing delay for encoding and decoding than for MPEG-4, which is significant for the overall system delay in the teleoperation system.

3.1.3 Three-Dimensional Perception

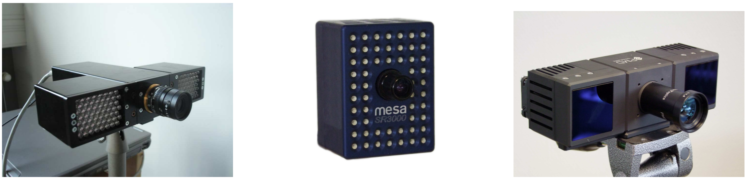
Three-dimensional perception of the environment offers significant potential to improve navigation solutions for mobile systems. Classical approaches for 3D-imaging refer to stereo-cameras, optical flow approaches and laser range scanners, while so far in industrial applications still mainly simple and reliable 2D-sensors are in use [133]. With the progress in robotics and the movement to work environments which are not completely pre-defined anymore, three dimensional perception is essential. Today, mainly three-dimensional laser range finders are used for three-dimensional perception and mapping in robotics [134][135]. As an alternative stereo cameras provide high framerates and high lateral resolution. Nevertheless, there is still high complexity on finding disparities and calculating distances from the image pairs. In addition, the applicability very much relies on the environmental conditions (e.g. lighting, structure). As three-dimensional laser range finders systems exist with nodding 2D laser scanner, as well as high accurate 3D laser scanners. The frequency of getting point clouds from this types of sensors differ from some seconds to some minutes depending on the increment of the tilt angle. In other approaches for mobile robots, two or more 2D laser scanners are mounted horizontally or vertically in order to provide a field-of-view of 360° [136]. Recently also laser range finders with a horizontal field of view of 360°, and a vertical field of view 26.8° like the Velodyne HDL-64[3] are available. It provides three-dimensional point clouds with a frequency of 10 Hz.

All these systems are quite expensive and include complex, moving mechanical components. Thus, the newly available 3D time-of-flight range cameras provide a promising alternative to these existing techniques and attract much interest [137], [138], [139]. The following Section 3.1.4 gives a more detailed overview on this new type of cameras.

3.1.4 3D-Time-of-Flight Cameras

3D time-of-flight (ToF) cameras provide instantaneously a depth image of the environment, without the requirement of any further processing of the received

[3] http://www.velodyne.com/lidar/ (24.10.2009)



(a) PMD 19k (b) Swissranger SR3000 (c) PMD CamCube 2.0

Figure 3.2: Example 3D-ToF cameras.

data in order to get distances. In addition to the depth information for every pixel of the image, an amplitude image, and for some cameras also an intensity image, which can be interpreted as a grayscale image of the environment, is received. These compact cameras can be operated at high frame rates and do not include any mechanical moving parts, being thus very suitable for mobile systems. Figure 3.2 shows two examples of ToF cameras from the leading companies PMD-Technologies[4] and Mesa Imaging[5]. The current implementations of these cameras have limitations in the field of view and the resolution. Nevertheless, these implementations already offer high potential for mobile robot applications. Table 3.1 gives important characteristics of some examples for existing implementations of 3D-ToF cameras. All this 3D-ToF cameras are based on photonic mixing device (PMD) technology. Thus, in the following also the term PMD camera is used equivalent to 3D-ToF camera.

Working Principle

In order to understand the considerations, calibration and determined characteristics of the later Sections 3.3, and 3.4 in the following the basic working principle and important parameters are summarized. The basic equations of this section and more details on electronic principle are given in [140].

A PMD camera actively sends out a modulated infrared light. In each of the pixels of the camera chip a so called photonic mixing device (PMD) or smart pixel is embedded [137, 141]. Each of these pixels allow to determine the correlation (cf. blue line in Figure 3.3) between modulated emitted reference and

[4] http://www.pmdtec.com/ (27.10.2009)

[5] http://www.mesa-imaging.ch/ (27.10.2009)

Model	Resolution	Default Ambiguity Range	Field of View
PMD 3k-S	64 × 48	7.5 m	40° × 30°
PMD 19k	160 × 120	7.5 m	40° × 30°
PMD S3	64 × 48	7.5 m	40° × 30°
PMD A-Sample	64 × 48	40 m	52° × 39°
PMD CamCube 2.0	204 × 204	7 m	40° × 40°
Swissranger SR 3100	176 × 144	7.5 m	47.5° × 39.6°
Swissranger SR 4000	176 × 144	5 m/10 m	43.6° × 34.6°

Table 3.1: Characteristics according to data sheets of exemplary 3D-ToF cameras in standard configuration.

perceived optical signal, such that the phase shift can be calculated and respectively the proportional time-of-flight and the distance.

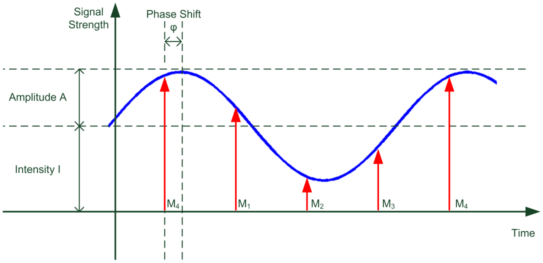


Figure 3.3: Schematic principle of phase-shift algorithm (adopted from [140]).

The correlation function is sampled four times M_1, M_2, M_3, M_4 with a phase offset of 90° in-between each sample (cf. Figure 3.3). The phase shift φ can be calculated as follows:

$$\varphi = \arctan\left(\frac{M_1 - M_3}{M_2 - M_4}\right) \tag{3.1}$$

From the phase shift φ the measured distance D can be derived:

$$D = \frac{\varphi}{2\pi} \cdot d_{max}; \qquad d_{max} = \frac{\lambda_{mod}}{2} = \frac{c}{2 \cdot f_{mod}} \tag{3.2}$$

The maximum distance without ambiguity of the measurements d_{max} is determined by the modulation frequency f_{mod} of the camera. It is half the wavelength

λ_{mod} of the modulation frequency. c denotes the speed of light. Besides the used modulation frequency of the camera also the illumination power of the camera's illumination limits the maximum distance which is measurable.

In addition to the distance two further values can be derived: the incident light intensity I (cf. Equation 3.3), and the correlation amplitude A (cf. Equation 3.4). The intensity value can also be regarded as a kind of grayscale value. The amplitude can be interpreted as kind of measure for the quality of a determined distance value.

$$I = \frac{1}{4}\sum_{i=1}^{4} M_i \tag{3.3}$$

$$A = \frac{\sqrt{(M_1 - M_3)^2 + (M_2 - M_4)^2}}{2} \tag{3.4}$$

Besides the modulation frequency the most important adjustable parameter for the PMD camera is the integration time τ. The integration time specifies the time in which incoming photons are considered for the distance measurement. It is comparable to the shutter speed of a standard camera. Tests with the camera showed that it is very important to adapt this integration time to the observed scenario and the task. A wrong selected integration time can cause either oversaturation or insufficient saturation and hereby invalid distance measurements. The desired optimal working range and the reflectivity of observed objects have influence on the the selection of the integration time. In Section 3.3.3 an algorithm is introduced to adapt this integration time automatically during runtime for the mobile-robot navigation tasks.

Quality problems, oversaturation, or false measurements can also be caused by environmental sources like natural sunlight or artificial light sources, which have an infrared component in the wavelength of the PMD camera illumination units. Thus, older camera models like the PMD 19k cannot be used in outdoor scenarios. Newer cameras (e.g. PMD 3ks, ASample, CamCube2.0) have a technology like the *Suppression of Background Illumination (SBI)* [142] integrated, which enables the cameras to work to some extend also in outdoor environments.

Applications

Many potential applications of 3D-ToF cameras in automotive, multimedia, entertainment, robotics, and other areas have been described [139]. In automotive industry, range cameras are used for crash avoidance, as a distance sensor for cruise control, traffic sign detection, park assistance, or driver tracking and monitoring, e.g. [143],[140]. In the area of multimedia and entertainment 3D-ToF cameras were used for a virtual keyboard [144] and for an interactive canvas. Another application of PMD cameras which currently raised a lot of interest is gesture and body tracking e.g. for game consoles [145]. This gesture and body tracking is also of interest for future interfaces in HRI. It might enable a more natural and intuitive peer to peer interaction with robots through higher safety [146] and new ways of commanding and teaching robots even in the industrial application area. In the context of mobile robotics [138] and [147] present ideas of potential applications. In [148] and [149] a PMD camera was used for landmark detection in order to localize a mobile robot. May et al. [150] shows an approach for 3D-pose estimation in robotics. Weingarten at al. [151] described a simple obstacle avoidance and path planning with a 3D-ToF camera. Hong et al. [152] also reports about obstacle detection for AGVs (Automated Guided Vehicle) in factory environments. Sheh et al. [134],[153] investigated 3D-ToF cameras for map-building and stepfield traversal in an artificial USAR (urban search and rescue) environment. Craighead et al. [154] tested a 3D-ToF camera for use in USAR. They especially analyzed the influence of lightening conditions and movement disturbances.

Hybrid systems of a PMD camera and a CCD video camera [155],[156],[157] or stereo camera [158],[149] have also been investigated. In order to fuse multiple depth images scale-invariant features can be used. Two approaches to retrieve such features from PMD cameras are presented in [159] and [160].

Another application of PMD cameras would be in space robotics. If the technology proves to be applicable in space, a very promising application for this type of cameras would be e.g. spacecraft tracking for rendevouz and docking maneuvers. In the context of this thesis work also contributions to this application were made. These contributions will not be covered in this monograph. Details on this work can be found in [5].

Compared to the related work described before, this thesis work investigates the application of PMD cameras for user interfaces and driving assistance systems in mobile robot teleoperation from the human point of view. Parts of the results were also published in [13].

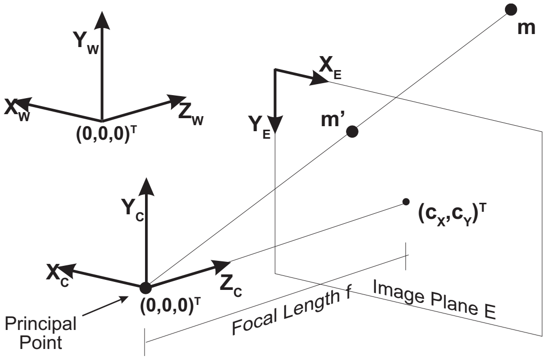


Figure 3.4: Geometry of pinhole camera model.

3.1.5 Optical Camera Calibration

Both 3D-ToF cameras and cameras for visual image capturing are optical systems which project elements of a three-dimensional world to a two-dimensional pixel array. For 3D-ToF cameras a model of this projection is required in order to transform the spherical distance values $D_s[i,j]$ for each pixel at pixel position i, j on the camera chip to metric Cartesian x, y, z coordinates relative to the camera. For the intensity and amplitude image and visual cameras the projection model is necessary in order to find the correspondence between the three-dimensional objects and the received two-dimensional images. Thus, the application of such optical camera systems in robotics requires a careful calibration in order to gather meaningful distance information (cf. Sections 3.3, 3.4, 5.2). In the following the models for projection and calibration for optical camera systems which are used in this work are introduced. This background is necessary to understand the proposed methods and concepts in the later sections.

Perspective Projection

For the projection of an optical camera typically the pinhole model is applied (cf. Figure 3.4). Three basic frames of reference exist in this model - the three-dimensional world coordinate frame L_W, the three-dimensional camera coordinate frame L_C and the two-dimensional pixel coordinate frame L_E corresponding to the pixel array of the camera chip. In order to derive the transform from world coordinates to pixel coordinates two transformations are required. The transformation T_{WC} corresponding to the extrinsic parameters of a camera calibration transforms world coordinates to camera coordinates. The transformation from camera coordinates to pixel coordinates P_C corresponds to the intrinsic or projection parameters of a camera. Equation 3.5 gives the needed mathematical model to project a point $m = (x_W, y_W, z_W, 1)^T$ in world coordinates to a point $m' = (x_s, y_s, s)^T$ on the two dimensional pixel array of the camera.

$$\begin{aligned}\begin{pmatrix} x_S \\ y_S \\ s \end{pmatrix} &= P_C \cdot T_{WC} \cdot \begin{pmatrix} x_W \\ y_W \\ z_W \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} fs_x & fs_x\alpha_c & c_x & 0 \\ 0 & fs_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{21} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_W \\ y_W \\ z_W \\ 1 \end{pmatrix}\end{aligned} \tag{3.5}$$

f denotes the focal length in metric units. s_x and s_y scale f to the unit pixels. Thus, $\frac{1}{s_x} = w_p$ and $\frac{1}{s_y} = h_p$ correspond to the physical width w_p, and height h_p of one pixel on the camera chip. During typical calibration processes as described in the later section, the focal length and the scale factors can not be derived separately. The result of an intrinsic calibration yields the combined terms $fs_x$and fs_y - the focal length f_{px} and f_{py} in pixels. However, these combined values together with the pixel array width and height allow to determine the field of view of a camera directly. No further knowledge about the physical size of the camera chip is required. By contrast the field of view calculation based only on the focal length f requires the physical size of the camera chip. With f_{px} and f_{py} the horizontal and vertical field of view FoV_h, FoV_v can be calculated as follows:

$$FoV_h = 2 \cdot \arctan\left(\frac{x_{max}}{2f_{px}}\right), \qquad FoV_v = 2 \cdot \arctan\left(\frac{y_{max}}{2f_{py}}\right) \tag{3.6}$$

x_{max} and y_{max} denote the dimensions of the camera chip in pixel.

α_c is the skew factor which is the angular relation between x and y axis of the pixel array. For available cameras this angular relation can be assumed to be orthogonal, such that α_c is equal to zero, and hereby the whole matrix element. c_x and c_y denote the pixel coordinates of the principal point on the pixel array. The matrix T_{WC} is a standard homogeneous transformation matrix with rotational and translational components.

Assuming perfect projection without distortion, the final coordinates x_n, y_n in the pixel array can be derived by normalizing the vector $(x_S, y_S, s)^T$ as follows:

$$\begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{x_S}{s} \\ \frac{y_S}{s} \\ 1 \end{pmatrix} \tag{3.7}$$

Distortion

In most optical camera systems the real lenses compared to the before listed mathematical model show distortion effects (mostly radial and slight tangential). This is mainly due to imperfectness of the real lenses. Thus, a distortion model is needed in order to compensate this for the application in augmented reality interfaces and sensor measurements. The common distortion model in Equation 3.8 was first introduced by Brown [161] and is used in many machine vision and calibration approaches (cf. Section 3.1.5).

$$\begin{aligned} x_d &= x_n \cdot (1 + k_1 r^2 + k_2 r^2 + k_3 r^6) + 2p_1 x_n y_n + p_2(r^2 + 2x_n^2) \\ y_d &= y_n \cdot (1 + k_1 r^2 + k_2 r^2 + k_3 r^6) + p_1(r^2 + 2{y_n}^2) + 2p_2 x_n y_n \\ \text{with} \qquad & r = {x_n}^2 + {y_n}^2 \end{aligned} \tag{3.8}$$

Here the coefficients k_1, k_2, k_3 correspond to the radial distortion coefficients, and p_1, p_2 to the tangential distortion coefficients. These distortion coefficients can be determined with standard calibration tools, and can be used with the distortion model to calculate a corrected, normalized image for further usage from the retrieved distorted image. For standard field of view cameras the radial distortion coefficient k_3 is neglectable and set to zero. Thus, most calibration tools do not determine k_3 by default.

Camera Calibration

Various calibration tools exist to determine the intrinsic and extrinsic parameters of a camera. Most of them are based on calibration models and concepts introduced in [162], [163], or [164]. For calibrations in this work the Matlab Camera Calibration Toolbox[6] by Jean-Yves Bouguet and the GML Camera Cal-

[6] http://www.vision.caltech.edu/bouguetj/calib_doc/ (30.10.2009)

ibration Toolbox[7] are utilized. The basic procedure of this calibration is always that a reference pattern (here a printed regular black and white chessboard) is captured with a camera from different point of views. Based on these images and the known reference pattern at first the intrinsic and distortion parameters are derived through optimization approaches. With the now known intrinsic parameters, for each of the images the relative position and orientation - the extrinsic parameters - to the reference pattern can be determined.

3.1.6 Sensor Data Fusion, Localization and Mapping

For robotic systems sensor data fusion from different data sources is often required. As example Kalman-Filter and various modifications like e.g. the extended Kalman-Filter (EKF) or analytical optimization methods are used for sensor data fusion. In particular, sensor data fusion is needed for localization and mapping in order to make use of the complementary advantages and properties of the different systems.

There are different sensor systems which allow to determine the absolute position of a robotic system. The global positioning system (GPS), Galileo, and various systems which require to setup an specific infrastructure (e.g. Metris iGPS[8], A.R.Track[9]) are examples for such systems. Relative localization is realized by dead reckoning systems, e.g. odometry systems based on wheel encoders or visual odometry system based on feature matching between subsequent images from a camera system. In general these relative localization approaches provide a good localization for short periods of time, but they have the problem of error accumulation over time. Thus, many robot systems use additional distance sensors together with an a priori known map (no matter if occupancy grid, feature map, or topological map) to reference and correct the relative localization to the environment in order to retrieve the absolute position of the robot.

Nowadays localization and mapping is often related to the simultaneous localization and mapping (SLAM) problem. Most approaches also allow to integrate a priori map information. The SLAM approaches are in particular interesting for the here mentioned exemplary application in search and rescue where in most cases no localization infrastructure is available and its unknown if map informa-

[7] http://graphics.cs.msu.ru/en/science/research/calibration (30.10.2009)

[8] http://www.metris.com/large_volume_tracking__positioning/igps/(30.10.2009)

[9] http://www.ar-tracking.de/(30.10.2009)

tion e.g. of a building is still valid. Two main directions exist to tackle the SLAM problem - probabilistic approaches and analytical optimization approaches.

For metric two dimensional occupancy grid maps, proper probabilistic methods like EKF-SLAM or FastSLAM are applied (for an overview see [165]). Here mostly two dimensional laser range finders are used. There exist also a lot of Visual SLAM approaches [166],[167]. Visual SLAM algorithms use either optical flow or robust image features. Nowadays they mostly rely on SIFT[10] [168] or SURF[11] [169] features, as these algorithms deliver currently the best results combined with a reasonable calculation time. Robust feature detection algorithms will also play an important role for future pose estimation and mapping application with PMD cameras [5].

Future robot applications, e.g. urban search and rescue, will require the extension towards 3D mapping. Because of the additional degrees of freedom in pose and orientation, the term 6D-SLAM is often used in literature. Nüchter et. al [170] for instance used the Iterative Closest Point algorithm (ICP) [171] as analytic approach to do scanmatching of point clouds, which are recorded by a 3D laser range finder. For global consistent maps and loop closing, further processing is done. On the other hand, probabilistic EKF-based 3D-Mapping was for instance done by Weingarten et al. [172].

Many of the here mentioned methods are applied in the prototype systems implemented for the research covered in this monograph. As the algorithms behind these localization and mapping approaches are not focus of this work they will only be covered very shortly if necessary.

3.2 A Concept for Intelligent Traffic Shaping in Wireless Networks

Modern multi-hop networks often use WLAN to set up ad-hoc networks of mobile nodes with each node acting as traffic source, sink, or router. Considering these networks, routes between sources and destinations might be established via several relay nodes. Thus, the utilization of intermediate nodes which are part of a route influences the overall route performance, whereas sender and receiver have no direct feedback of the overall route status. Even worse, the whole network

[10] Scale-Invariant Feature Transform

[11] Speeded-Up Robust Features

topology and behavior might change as nodes move or other nodes communicate with each other.

In case video is transmitted via wireless ad-hoc networks in a teleoperation scenario (like for the mixed reality interfaces in this thesis), the displayed video-stream for the operator might face variable frame rates, very high packet loss, and packet inter-arrival times which are not appropriate for mobile robot teleoperation.

In the following an approach using a feedback generated by the network to adapt the image quality to present communication constraints is introduced. According to the current network status, the best possible video image is provided to the operator while keeping constant frame rates and low packet loss. The concept and results can also be found in [10] and [11].

3.2.1 Scenarios

One special and very challenging application scenario for mobile robots is the exemplary application search and rescue. This scenario has very high demands on almost all aspects of mobile robots, multi-robot teams, and human-robot teams. Besides for instance mobility, localization and human-robot interaction one of the most important elements in this scenario is communication. After a disaster, the search and rescue team probably cannot (or only partially) rely on existing pre-installed communication infrastructure. Often the environment is very unstructured and cable communication systems are only applicable for short distances. Therefore, these scenarios require a wireless communication which can adapt to the current needs and constraints in a reasonable time. One idea to support this is to equip all the heterogeneous team members (humans and robots) with compatible communication devices. These team members are now available for the other team members as communication relays by implementing ad-hoc routing protocols which allows redundant communication links and higher communication distances. Another interesting chance for the application of ad-hoc networks is the use of the heterogeneity of mobile robots. On the one hand, it is possible that team entities with limited communication capabilities can use any team entity with better communication in its limited communication range to reach the rest of the team. On the other hand, the ad-hoc network offers the possibility to build subteams and to use nodes only if they are really necessary to communicate with the target. In [173], several of these scenarios were analyzed in a simulation study comparing the performance of several ad-hoc routing protocols. Figure 3.1 shows how such a typical heterogeneous team might look like.

The dual use of mobile nodes (e.g. robots) for their own communication and as communication relay for other nodes opens the possibility to extend the communication range between a control station and a teleoperated machine [111][109][112][110] or to set up a communication infrastructure in an environment containing several obstacles [118]. Figure 3.5a shows a mobile robot being teleoperated and several nodes – which could be stationary or mobile – are used to keep up the communication link. In the presented scenario a chain of nodes is established which can be considered as a kind of worst case scenario as the grade of meshing is very low (only minimum number of neighbors in range) and no redundant routes between robot and operator are available. These topological constraints have a large effect on the behavior of the used protocols and the corresponding parameter settings which was demonstrated in [174] and [130].

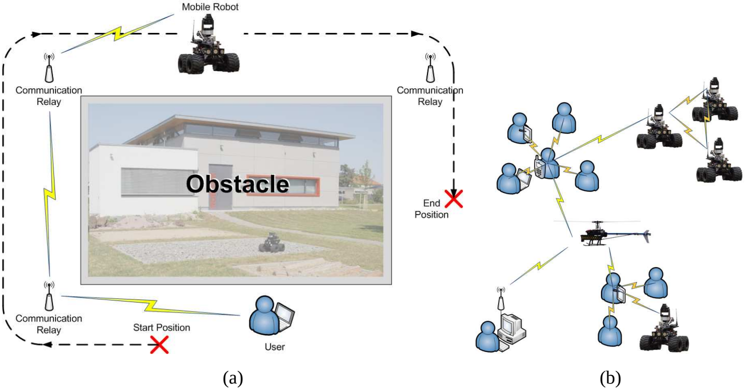


Figure 3.5: (a) Communication relay to increase the range for teleoperation of a mobile robot. (b) Communication inside and between teams.

The dynamic characteristic of ad-hoc networks allows the very flexible and efficient adaptation to the current communication needs. It is possible that nodes spatially co-located can communicate directly without involving any other node. As there is no special central node present for coordinating and forwarding the traffic, the probability of this node being a bottleneck while a larger number of nodes transfers large amounts of data (e.g. video streams) is reduced (cf. Figure

3.5b). Therefore, ad-hoc networks also very much support a distributed, decentralized communication architecture on the higher communication levels. This characteristic also supports the establishment of smaller communication subgroups on a logical or spatial level inside the whole group of nodes. The possibility to adapt the ad-hoc network to current needs for communication allows an intelligent solution to use the different wireless links more efficiently. If there are heterogeneous nodes in the ad-hoc network where some of the nodes have better communication capabilities (e.g. higher transmit power), a mobile node with only short range communication can use any of the the nodes with long-distance communication which is currently in its own communication range to transmit information to any other node in the network.

Setting up a testbed or a prototype of a multi robot system is quite easy, but nevertheless, several technical aspects which are discussed in this section must be considered to prevent unstable communication links or not suitable packet round trip times.

3.2.2 Concept

The developed concept enables to stabilize the framerate of a video stream transmitted over a multi-hop network through a variable image quality of the video stream. The quality is adjusted automatically to the current state of the wireless multi-hop network and respectively the available bandwidth of the used route for the video stream by using a feedback of the network status. Hereby, the required bandwidth for the video stream from the communication link is reduced through a dynamic higher image compression This leads to a lower image quality and to a smaller size of an image instead of taking the risk of packet loss, link congestion and complete link breakdowns. As above mentioned, the state of each single node of a route has a strong influence on the quality of the used link in terms of bandwidth, delay, and packet loss. To increase the performance of mobile robot teleoperation through a mixed reality user interface, the available frame rate at the operator user interface should be almost constant. In order to adjust the image quality according to the link, an active feedback mechanism is implemented at the application layer of each node. Thus, a feedback of the network is available for the video stream source which can be used to adapt the image quality. The proposed mechanism requires only little resources, is portable and easy to implement, and provides the operator the highest possible video quality for mobile robot teleoperation which can be guaranteed for the current network state. As it supports no traffic classes as it is known from wired IP networks, it should not be considered as a quality of service (QoS) mechanism. Anyway, available quality of service (QoS) mechanisms – e.g. integrated services (IntServ) or differentiated services (DiffServ) – are currently not applicable in ad-hoc networks of mobile robots due to very specific hardware requirements and the special solutions which are currently only available for network service providers.

3.2.3 Network Feedback and Adaptive Video Quality

The mechanism mainly consists of two parts: the network feedback, and the adaptive adjustment of the video quality. The mechanism is used for a simple admission control of the video source and intends to provide the best possible video image quality considering the current state of the link. The objective is an efficient use of the available bandwidth without overloading the route with video traffic to the operator. Thus, it is not used to increase the link quality directly but uses the available resources most efficient and reliable for the operators' video stream. In addition, it supports the avoidance of congestion of the communication link on the MAC Layer.

Network Feedback

The network feedback is responsible to transmit the status of a node to the video source. Therefore, nodes of the network host a small client program at the application layer. This client application is listening in promiscuous mode at layer 3 of the ISO/OSI model (IP-layer) and measures the utilization of the wireless link. All kinds of traffic are monitored: incoming and outgoing packets, packets for forwarding, and packets with other nodes in range as destination – basically all traffic causing the radio link of this node to be busy. The network feedback client sends very small UDP packets with an adjustable frequency (in the test setup 10 Hz) and 8 bytes as payload to the video-source if it is a used hop in the video stream route between video-source and receiving node. This payload is used to indicate the status of the corresponding node, either “normal operation” or “overload situation”. In the beginning, each node is in the “normal operation” mode. As soon as a certain utilization of the supported bandwidth is exceeded, the status of this node switches to “overload situation”. Important parameters for the network feedback clients are the feedback frequency f and the threshold for status determination d. In case f is too high, too much feedback traffic is generated which degrades the performance of the network. Even these packets are very small, too many small packets with a high sending frequency will have a very bad effect on 802.11b WLAN and will significantly decrease the throughput. Thus, the generated feedback traffic should be limited depending on the interpretation rate of the video adjustment mechanism and the selected load window for the wireless nodes. Often it is even not necessary to run a feedback client on each network node. For setting parameter d, it should be considered, that d specifies the percentage of the nominal bandwidth (e.g. for 802.11b this would be max. 11 Mbit/sec) which can be used without switching to the "overload situation" state. The feedback clients measures packets on layer 3, where the maximum available bandwidth corresponds to the "goodput" of the wireless link which is about 75% of the nominal link bandwidth (e.g. for 802.11b this would be 75% of 11 max. Mbit/sec). In order to allow a reaction on potential overload situations while providing the user a video stream with a bandwidth of 1 to 1.5 Mbit/sec for the best quality, d is set to 50%. Figure 3.6 gives an overview of the setup for the described mechanism.

As the proposed mechanism is used within a network where a link failure can occur at any time, the measurement and signaling is implemented such that link failures and link reestablishing can be monitored reliably. As the mechanism for video quality adaptation performs best with a feedback frequency of $f = 10$ Hz (according to the presented scenario an experimental setup), the generated measurement traffic has a bandwidth of less than 0.003 Mbit/sec per measurement node.

Adaptive Video Quality

The video quality in the presented system is adapted according to current state of the ad-hoc route for the video transfer. The adaption mechanism receives all status packages from the nodes between two received frames from the image source, interprets these packages and selects the quality for the next frames with a combination of previous status data and the current state. Do reduce oscillating behavior in quality switching near the selected load limit of the nodes a kind of inertia mechanism for the adaptation process was integrated. The implemented inertia mechanism guarantees not to change the image quality whenever a status

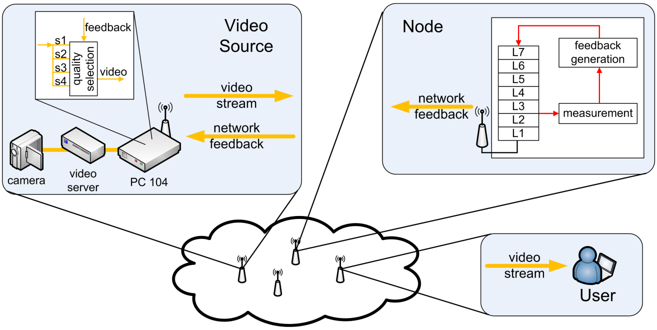


Figure 3.6: Test setup components for intelligent traffic shaping.

of a node changes. It is possible to set a certain number (cf. Algorithm 1, min/max of *inertia_counter*) of receiving same successive route load states until the quality is changed. Algorithm 1 shows the mechanism how the quality for the next frame is selected according to the received network status messages.

In the current test setup, four different video qualities are used at a frame rate of 11 frames per second each. Table 3.2 shows the average size of one image for the corresponding image quality level.

A higher number of different quality scales would also be possible. In the current test setup a minimum of -3 and a maximum of 3 are selected for the *inertia_counter*. With this value the mechanism reacts in the worst case after six frames with subsequent overload states and in average after three frames. This keeps the load caused by the video traffic on the different nodes in a certain defined window around the selected threshold for overload state. In combination with parameter d of the above described feedback mechanism, the quality adjustment intervenes as soon as a node exceeds a radio link utilization of more than approx. 78% ($\approx 50\%$ of nominal bandwidth). This prevents the node from reaching a utilization of 100% of the available maximum throughput which would result in a high packet loss rate due to an increasing number of packet collisions.

Algorithm 1: Video quality adaptation.

```
Input: video streams of different quality;
       load status messages;
foreach frame $V_q$ of current selected quality $q$ do
    if one of the nodes overloaded then
        $inertia\_counter++$;
    else
        $inertia\_counter--$;
    end
    reset node states;
    send video frame $V_q$;
    if $inertia\_counter > inertia\_counter_{max}$ then
        if $q > q_{min}$ then
            $q--$;
        end
        $inertia\_counter = 0$;
    else
        if $inertia\_counter < inertia\_counter_{min}$ then
            if $q < q_{max}$ then
                $q++$;
            end
            $inertia\_counter = 0$;
        end
    end
end
```

Table 3.2: Average size of one image per quality level.

Quality	minimum	low	medium	high
Size (kbytes)	15	26	34	47

3.2.4 Experiments and Validation

Problem Definition

In the above mentioned scenarios, the available throughput of a route via a wireless multi-hop network is a highly dynamic parameter which depends on many environmental influences and affects the quality of the application significantly. The throughput of a wireless node can be decreased due to different reasons. In case intermediate nodes of a route are also part of a route which has to transport other bandwidth intensive flows, the available bandwidth must be shared between all present routes via this node, which will reduce the available bandwidth for the video link. Furthermore, also a decreasing link quality will reduce the bandwidth and increase the packet loss probability. If the network is not reacting to traffic overload at a specific node, this will lead to unpredictable packet loss at this point and delays at the different receivers. For the teleoperation scenario the effect will be that the video stream will get randomly stuck, because packets get lost. Most probably the operator will get confused and will stop the robot.

Evaluation of Test Scenarios

In [12], difficulties of a proper setup and evaluation of scenarios with wireless ad-hoc networks are discussed. The biggest challenge is the design of a scenario which provides a base for comparable test runs. The used radio link might be disturbed by many external influences. Besides, parameters like link quality or signal strength which can be an indicator for external disturbances, there might still be an external disturbance which cannot be detected and characterized so easily. Thus, the influence of existing error sources on the test runs must be minimized or at least considered in the evaluation. In general, two methods are available for the evaluation. The first possibility is, that many test are performed in the different scenarios with the setup to be investigated. Then the settings are changed and again many test runs are performed in the same scenarios as before. Thus, a trend between the two present behaviors might be observed. The second possibility is the repetition of many test runs in one meaningful scenario. After tuning the parameters, the test runs must be repeated with the new settings. In both cases all tests runs must be performed contemporary in order to minimize environmental changes (e.g. weather). These two methods do not directly allow a quantitative evaluation, but a relative evaluation and the determination of a trend of the behavior due to changes in the parameter settings of the observed mechanisms are possible and often enough to analyze the investigated system. This work uses the second method.

Test Scenario Additional Traffic

To set up the scenario where a node is used for more than one bandwidth intensive traffic flow, four nodes are used (cf. Figure 3.7). All nodes are located such that they are in direct communication range. During the tests, defined additional UDP traffic will be generated between node 3 and node 4 while the investigated video stream is transmitted via UDP from the mobile robot to the user's PC via node 3. The generated UDP traffic is used to reach certain load levels at intermediate node 3. As in this scenario, node 3 and node 4 are in communication range to all other nodes which will also cause interferences at the physical layer.

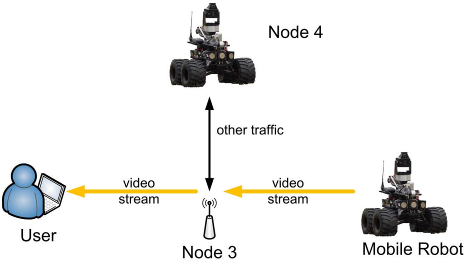


Figure 3.7: The test setup for additional traffic.

To provide best repeatability of the tests, all nodes are stationary. Only the additional traffic between node 3 and node 4 will be varied according to a defined profile. Measured categories are the packet loss and the packet inter-arrival times. These categories are measured while the amount of additionally generated traffic is increased. As reference test, video transmissions of constant target quality are used and compared to the packet loss of the transmission with adaptive quality.

The proposed mechanism was tested in a real outdoor environment with a wireless ad-hoc network of four nodes. One is the PC of the operator, one is an Outdoor MERLIN [68], and two intermediate nodes are MERLIN robots (indoor version). Figure 3.7 shows the detailed test scenario setup. All MERLIN robots have a C167 microcontroller for low-level operations and sensor data processing, as well as a PC-104 for more complex and computationally more intensive tasks. The PC-104 uses a Linux operating system and all nodes are equipped with 802.11b standard WLAN equipment (Atheros chip).

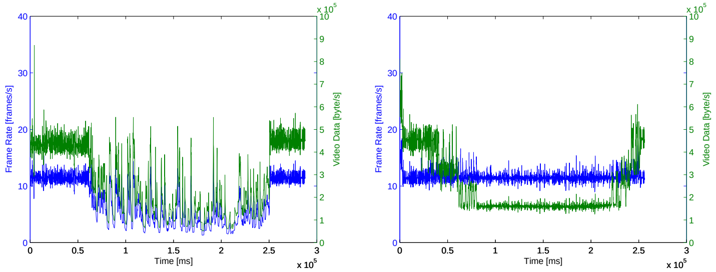
To grab the video from an analog camera (approx. 65 degree field of view) an Axis video server is used. It can grab the video from up to four cameras with a resolution of 768x576 pixels. Dependent on the configuration and connected clients, a frame rate of up to 25 images per second can be provided either as MJPEG or MPEG4 over a TCP/IP connection. For the described tests the PC-104 is connected over a cross-link cable to the Ethernet interface of the video server. As nothing else is connected to this Ethernet interface of the PC-104 it can be exclusively used for the video traffic. For the presented tests four MJPEG video streams with full resolution are established with four different compression rates. MJPEG as video compression was selected, as MPEG4 compression takes a significant longer time on the Axis server what causes a significant delay in the video stream. Secondly a loss of a packet during transmission of MPEG4 streams to the robot might lead to longer set of distorted images because compared to MJPEG not all frames of the stream contain the full image information needed. In case of the investigated scenario, the MJPEG frames are transmitted via UDP protocol over the wireless interface.

In a first step, a reference scenario was set up and measured. Therefore, no network feedback mechanism is active and a mobile robot generates a video stream which is sent to the PC of the operator as it is shown in Figure 3.7. Between node 4 and node 3, additional traffic is generated during the different test phases according to Table 3.3 to reach a defined load at intermediate node 3.

Table 3.3: Generated additional traffic.

Phase	generated additional traffic (Mbit/sec)
1	0
2	3,2
3	4
4	4,8
5	5,6
6	6,4
7	7,2
8	8
9	8,8

The results of this reference test are shown in Figure 3.8a. The x-axis shows the test time in milliseconds. The left y-axis describes the received frame rate in frames per second (fps) and the right y-axis displays the received video data rate in bytes per second (bps) at the receiving node (operator's PC).



(a) without network feedback

(b) with network feedback

Figure 3.8: Framerate and traffic without and with network feedback.

The test started with no additional traffic being present. Successively, more and more additional traffic is generated by switching to the next phase each 20 seconds according to Table 3.3. After 200 seconds of test time, the additionally generated traffic is reduced by switching back one phase each 10 seconds. In the beginning of the test – during phase 1 up to the end of phase 3 – the received frame rate is about 11 fps. After switching to phase 4 at about 60 seconds, the received video frame rate decreases significantly. The received frame rate between 100 and 200 seconds drops to $2 - 3$ fps while node 3 is overloaded. After the additionally generated traffic is reduced, the received frame rate recovered to 11 fps. Increasing the additional traffic forces node 3 to an overload situation. As the bandwidth used by the video stream cannot be adapted to the new situation, a packet loss of the video data is inevitable which is shown in Figure 3.9. The y-axis shows the number of lost packets vs. the test time on the x-axis.

Another measured category is the frame inter-arrival time of the video stream. This is a quite sensitive aspect, as a large jitter (variance of the frame inter-arrival time) is very irritating for the operator due to a very unsteady motion of the video image. Without additional traffic, the frame inter-arrival time is smaller than 100 ms with a variance close to 0 (cf. Figure 3.10a) what corresponds to the average

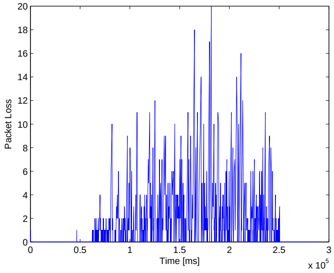
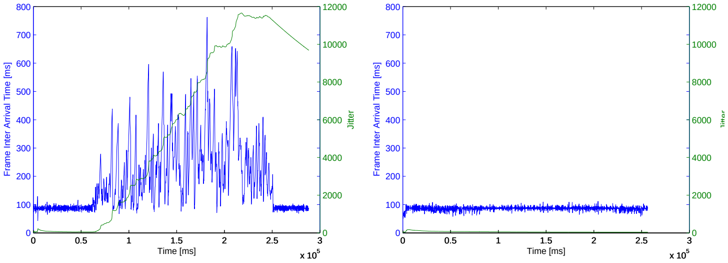


Figure 3.9: Packet loss without network feedback.

frame rate of 11 fps. After 60 seconds and an additionally generated traffic of 4.8 Mbit/sec, the frame inter arrival time increases to more than 400 ms with a variance of more than 10000 which indicates an unacceptable video for the operator.



(a) without network feedback

(b) with network feedback

Figure 3.10: Frame inter arrival time and jitter without and with network feedback.

The same test setup is used again – now with the network feedback and adaptive quality mechanism (cf. Section 3.2.3), which should improve the observed behavior. In Figure 3.8b, the frame rate and the video data rate is shown while using an adaptive video quality together with the network feedback mechanism. In the beginning, without additional traffic, the mobile robot generates a video stream of about 450000 bytes/s. During the test, the additionally generated traffic is increased similar to the test run described before. The implemented mechanism takes care that the video source reduces its generated video traffic to about 300000 bytes/s as soon as phase 3 (with an additional load of 4 Mbit/sec) is entered. Increasing the additional load at node 3 to more than 4.8 Mbit/sec results again in a reduction of the video traffic (180000 bytes/s). During the complete test run, the frame rate stays almost constantly at 11 fps as the adaptive video bandwidth reduction avoids the loss of video traffic. Also the frame inter arrival time stays constantly below 100 ms with a jitter of almost 0 (cf. Figure 3.10b).

Test Scenario Moving Robot

This scenario is set up in a way, such that several nodes of the network are included into the route between the operator's PC and the teleoperated mobile robot. Figure 3.11 shows how the different mobile robots acting as potential communication nodes (cf. 3.11 Figure mobile robot 1 to 3) were placed in a real outdoor scenario for the described tests. The three potential communication nodes are positioned around a small hill in a way that each node covers a certain reception area overlapping with the reception areas of the nearest neighbors. The small hill in the center causes a communication shadow. So a direct communication between the operator station and a node behind the hill (either mobile robot 2 or teleoperated mobile robot) and between mobile robot 1 and 3 is not possible anymore. To guarantee these test constraints the transmit power for each node was reduced to 10mW additionally. For the test the teleoperated mobile robot is controlled via joystick along the path as shown in Figure 3.11. The selected path and environment requires that the routing of the robot's communication link is changed according to the current position of the moving robot and the respective possible links to other communication nodes in the scenario. As soon as the teleoperated mobile robot enters the communication shadow of the small hill and no direct link is possible anymore, it has to communicate via one or more of the other mobile robots to reach the operator station. To achieve this dynamic routing and keep the communication link usable, ad-hoc routing protocols are applied and combined with the network feedback concept. In [174] ad-hoc routing protocols for mobile robot teleoperation are investigated. The required routing protocol parameter tuning is described in [130]. Based on the results of these two publications, B.A.T.M.A.N. is used as ad-hoc routing protocol for this scenario.

In the current scenario, only the mobile node (here: teleoperated mobile robot) generates a network feedback. There is no need to use the network feedback of the relay nodes, as all relay nodes in the scenario are stationary (link quality is not varying significantly) and the relay nodes do not create additional traffic besides the video data of the robot and the command and sensor data exchange between operator and mobile robot.

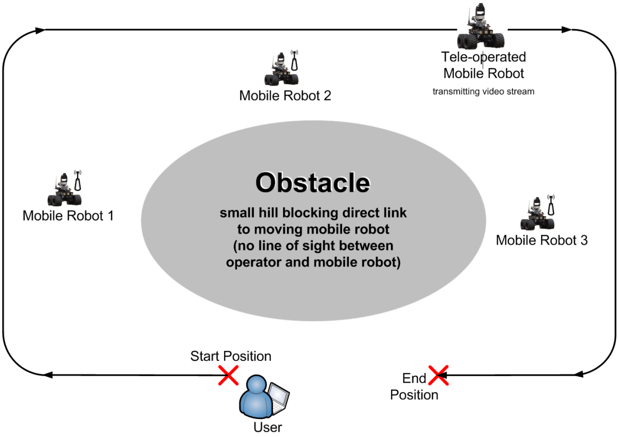


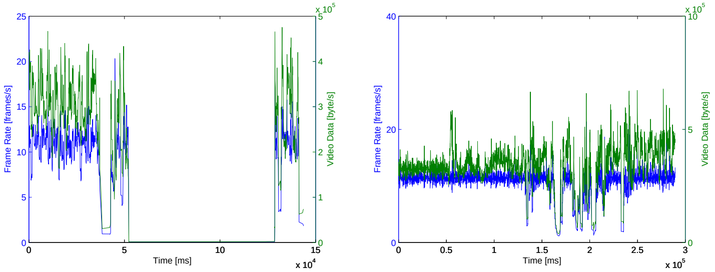
Figure 3.11: Test scenario of moving mobile robot.

While the mobile robot is moving, the network feedback mechanism is continuously monitoring the communication links and provides input for the video adaptation mechanism. With respect to the parameter settings, it has to be taken into account that several network topologies may occur which will affect the behavior of the wireless network significantly. As the communication via multiple hops implies variable bandwidth limitations due to e.g. hidden nodes, setting of parameter d (threshold for maximum link utilization) of the network feedback mechanism must be done carefully. In this work, the video source is the main traffic source of the entire network. Due to the steady flow of delay sensitive packets, a conservative setting of $d = 25\%$ is chosen as this also considers the half duplex characteristics of the WLAN links and hidden node situations of the scenario.

For this work numerous single test runs were performed. During a test run, the mobile robot is teleoperated via wireless ad-hoc network while driving around the hill. As the small hill blocks the line-of-sight communication between operator and mobile robot, several communication relay nodes will be included on-demand. During a test run, the round trip time of packets between operator and mobile robot, the frame inter arrival time, the frame rate, as well as the packet loss are measured.

While a test run is performed, several data flows are present inside the network. The operator sends control and command packets from the control PC to the mobile robot, whereas sensor data is transmitted in the opposite direction. The most bandwidth consuming data stream is the video image. All data is transmitted via the UDP protocol. In the investigated scenario, no active measurement and signaling traffic of the feedback mechanism is generated as the feedback client program is running on the mobile robot which also acts as video source. Thus, the feedback is directly sent to the quality adaptation mechanism without stressing the wireless communication. In the following paragraphs the results of one representative test run are explained exemplarily – much more tests were performed in order to draw the presented conclusions.

At first, a test run without network feedback is performed and the video stream behavior for this run is displayed in Figure 3.12a. The left y-axis of Figure 3.12a shows the video frame rate in frames per second and the right y-axis the video data in bytes. On the x-axis, the experiment time is plotted in milliseconds. In the beginning of the test, the operator is provided with a video frame rate of more than 10 frames per second. At around 40 seconds of test time, a short break down of the video link for about 5 seconds is detected. A later analysis turned out that at this time, the direct line-of-sight communication between operator PC and mobile robot was lost and a new route was set up via a relay node. Shortly after 50 seconds the video transmission failed again and could not be reestablished. A later analysis showed that an additional communication relay node was included and the route is established from the operator's PC via two relay nodes to the mobile robot. Due to the half duplex characteristics of the link, the available bandwidth decreased significantly what lead to a complete overload of the network link by the video traffic. Thus, the video link, as well as the command link broke down and the robot stopped. The observed link failure for such a long time results also from the ad-hoc routing protocol being not able to maintain the topology changes as also the signaling traffic for routing updates was lost during the overload phase. This behavior now should be avoided by the use of the network feedback mechanism.



(a) without network feedback

(b) with network feedback

Figure 3.12: Frame rate and received amount of video data at the receiver without and with network feedback.

In Figure 3.13 the round trip time (rtt) of ping packets between operator PC and mobile robot is plotted on the y-axis in milliseconds while the test is performed with active network feedback mechanism. It can be seen that at approximately 110 seconds test time the rtt increases and shows relatively high peaks at about 160 seconds of experiment time. After 200 seconds, the rtt decreases again to values below 60 milliseconds. The comparison with the routing tables for these times show, that shortly after 100 seconds experiment time, the first relay node was included into the communication link and at about 140 seconds the second relay node joined. In parallel, the packet loss is plotted for the same test run in Figure 3.14. Between 140 seconds and 240 seconds of experiment time the graph shows several short periods in which packets are lost. The majority of these packet losses occur due to the two rerouting procedures while driving the robot in areas without direct line-of-sight communication with the formerly associated communication relay node. Of course, also some of the packets might be lost due to a very high link load while communicating via multiple hops. The following figures will give more details on this. Figure 3.15 shows the frame inter-arrival time and the variance of the frame inter-arrival time (jitter) at the receiver and Figure 3.12b displays the frame rate and amount of transmitted video data at the operator's PC while the test is performed with the active network feedback mechanism. From the beginning of the test run until approximately 130

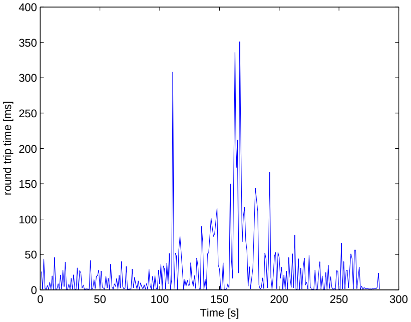


Figure 3.13: Round trip times (RTT) in milliseconds with network feedback.

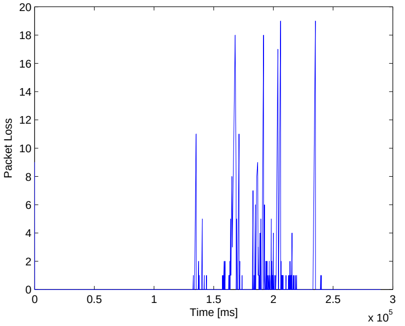


Figure 3.14: Packetloss with network feedback.

seconds of the experiment time the graph of Figure 3.15 oscillates around a value of about 90 milliseconds for the frame inter-arrival time. At 130 seconds and 160 seconds, and between 180 and 200 seconds of experiment time several peaks of up to 850 milliseconds are detected. This observation corresponds to the graph of Figure 3.12b, where the received frame rate of the video is displayed on the left y-axis and the received amount of video data is displayed on the right y-axis. From the beginning of the test until about 160 seconds of test time the video frame rate keeps constant at 11 frames per second while the transmitted amount of data varies during the rerouting at 130 seconds of experiment time. Between 160 and 240 seconds of test time, the frame rate varies due to changes in the link quality during this period of time (unstable links because of handovers). Nevertheless, the video stream did not break down (compared to the scenario without the network feedback mechanism) during the tests as now the network feedback mechanism takes care that the transmitted video data traffic does not exceed the capabilities of the complete link - also while communicating via several relay nodes. Here, the ad-hoc routing protocol is able to reconfigure the routing tables of the network nodes in time, as the video link capacity is limited before the rerouting procedure is initiated in order to reserve bandwidth for signaling and maintenance traffic. This is possible as the proposed mechanism additionally monitors the really available link capacity of the WLAN which is reduced as the link quality decreases. Thus, the network feedback mechanism prevents the network from being overloaded before a handover is started due to limiting the video traffic also in case of a low link quality to the currently associated network node.

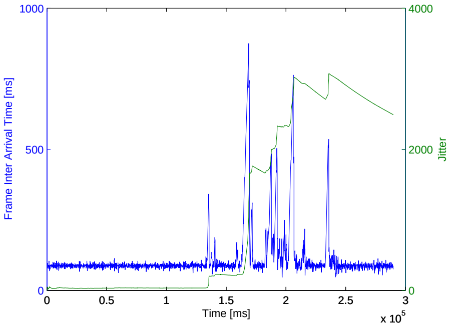


Figure 3.15: Frame inter-arrival time and jitter of video data at the receiver with network feedback.

In contrast to previous work, here, the mechanism is used while the communication link in the multi-hop network uses more than one relay node depending on the location of the mobile robot. It shows the behavior of a video stream which is used for mobile robot teleoperation in two different setups. In the first setup, the robot moves and transmits sensor and video data to the user while more and more relay nodes are included into the used route. While the number of communication relays increases, the usable bandwidth decreases due to the half duplex characteristics of the link and topology aspects. When the experiment is performed without using the intelligent shaping mechanism, the communication breaks down as soon as more than one hop is used for the communication as the available link capacity is tremendously exceeded. The second test setup uses the video shaping mechanism which as proposed here with adjusting the threshold parameter d according to the network setup as described above. The mechanism is running locally on the mobile robot which is sufficient for this scenario. This guarantees a suitable utilization of the bandwidth which is really available considering the dynamic network topology and the occurring changes in link quality and link availability. In this setup, the proposed mechanism throttles down the bandwidth required by the video stream by reducing the image quality.

3.2.5 Summary

In the first test scenario the proper functionality of the proposed adaptive quality mechanism is tested in this teleoperation scenario with real hardware under different network load situations. In situations with a very high link load due to additional other network traffic, usually the packet loss rate and the packet inter-arrival time is affected in a way that reliable and proper teleoperation is not possible anymore. By adjusting the image quality of the video stream it is possible to provide a stable video frame rate for the operator. In fact, the remaining bandwidth for the video stream is used efficiently in terms of providing a video with a stable frame rate suitable for mobile robot teleoperation.

In the second test scenario the expected behavior of the implemented mechanism was shown in a real world setup where a mobile robot is remotely controlled by an operator. It is used to navigate in an outdoor area which is not reachable via direct line-of sight communication between operator and mobile robot. Thus, other nodes have to be used as communication relays. It is shown that the network feedback mechanism in combination with an adaptive video quality keeps the video frame rate longer at a constant level and the long communication dropouts can be avoided which really increases the teleoperation capabilities.

These experiments show in a real-world ad-hoc network scenario, that the proposed mechanism is able to enhance teleoperation capabilities while a mobile robot is commanded via multiple communication relays. The easy setup and compatibility of the mechanism to existing standard protocol stacks and available standard hardware makes it a powerful tool while only a minimum of parameter tuning is necessary.

3.3 Three-Dimensional Perception with Range Cameras

The application of a 3D-ToF cameras in mobile robotics entails specific demands on the raw data pre-processing. In the following, sensor analysis, calibration procedures, data filtering, and a task specific camera parameter optimization are presented in order to process the data for autonomous behaviors, driving assistance systems, and the for user interfaces elements. For the following sections a PMD 19k cameras is exemplarily used for the analysis and tests. For other cameras these results might be slightly different and small adjustments for the calibration procedures might be required.

3.3.1 Calibration

Typically the PMD cameras deliver uncalibrated line-of-sight distances which do not represent the orthogonal distances to the image plane of the camera. Thus, at first the optical properties of the camera - the intrinsic, projection parameters - need to be determined according to the before described pin-hole camera model. Based on these parameters the received distances can be corrected and transformed to Cartesian coordinates. In addition, there exist different specific effects and characteristics of this type of camera which make it necessary to do a depth calibration. Thus, calibration of 3D-ToF cameras is a two-step process, which consists of photogrammetric calibration (like it is done for normal digital cameras) and depth calibration.

For the calibration in this work, the methods from [175, 176] are combined and a photogrammetric calibration with a chessboard was done. Kahlmann [175] used an extra array of infrared LEDs to improve corner detection of a chessboard to determine the desired camera matrix, which is not necessary here. In order to be able to use standard camera calibration tools with a printed chessboard here an upsampling for the image resolution with pixel interpolation is performed.

From calibration results gained, a significant deviation of the real field-of-view values from the data sheet values of the camera could be found. A field-of-view of 29.6°(horiz.)/22.4°(vert.) corresponding to a focal length of 12.11mm could be determined. The calibrated focal length matches the actual, correct physical focal length of the lens of the camera. The data sheet originally stated a horizontal field of view of 40°, what is incorrect.

From the PMD camera the raw distance matrix D together with information gathered by the performed photogrammetic calibration spherical coordinates $D_s(i,j)$ for the pixel indices (i,j) can be derived as follows:

$$D_s(i,j) = \begin{pmatrix} \alpha_x(i,j) \\ \alpha_y(i,j) \\ D_{i,j} \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{i-c_x}{f_{px}}\right) \\ \arctan\left(\frac{j-c_y}{f_{py}}\right) \\ D_{i,j} \end{pmatrix} \tag{3.9}$$

The spherical coordinates D_s are necessary to calculate the required Cartesian coordinates D_c. For the orientation of the coordinate frames used in this work it can be calculated as follows:

$$D_c(i,j) = \begin{pmatrix} x(i,j) \\ y(i,j) \\ z(i,j) \end{pmatrix} = \begin{pmatrix} D_{i,j} \cdot \sin(\alpha_x(i,j)) \\ D_{i,j} \cdot \cos(\alpha_x(i,j)) \cdot \sin(\alpha_y(i,j)) \\ D_{i,j} \cdot \cos(\alpha_x(i,j)) \cdot \cos(\alpha_y(i,j)) \end{pmatrix} \tag{3.10}$$

Inside the camera another small static distance error is introduced, as also the small part of the measured distance which results from the distance between chip and lens center is not constant for all pixels. For the center pixels it is the focal length. But the farer the indices of the pixels are away from the center, the larger is the resulting error. Nevertheless, for the PMD 19k and the presented applications with this camera this static error is currently neglected. For the PMD 19k the worst case error is below $1mm$ and much smaller than the actual achievable accuracy with this camera. For instance at edge pixel (160,120), assuming optimal optical center at (80,60), the distance correction value would be $12.7535mm$ compared to the used value of $12.11mm$, such that the disregard of this correction in the extreme case of an edge pixel introduces an error of $0.6435mm$. If the application or different camera setups demand for the correction of this error, the corrected distance value $D'(i,j)$ for each pixel indices (i,j) can be calculated as follows:

$$D'(i,j) = D_{i,j} - \sqrt{f^2 + ((i-c_x)\cdot w_p)^2 + ((j-c_y)\cdot h_p)^2} \tag{3.11}$$

w_p denotes the physical width of one pixel of the camera chip and h_p the corresponding height (for PMD 19K $w_p = h_p = 40\mu m$).

During photogrammetic calibration the camera also shows typical distortion parameters. As the effects for the small field of view are relatively small and the process of eliminating the distortion would cost additional processing time, for the applications in this chapter it was not considered. If another application requires a distortion correction, it can be done comparable do the approach for standard cameras presented in Section 5.2.2.

For the depth calibration at first a pixel offset can be determined which is induced by small differences of signal runtime in the signal path of the PMD camera. For standard environmental conditions it should be constant and can be compensated by a reference measurement against a white wall for the center pixels of the camera. This can be done by provided tools of the camera and it is possible to set the determined value as a global calibration shift value by the application programming interface (API) of the SDK.

In addition, cyclic effects in a characteristic curve, which were also observed by [175, 176], were detected. This effect occurs due to the fact that the modulated signal passed to the PMD chip as reference is slightly different from the signal emitted by the infrared LEDs of the PMD camera. This difference results from the fact that the LEDs in general do not emit light proportional to the voltage applied to the LED. This property of LEDs is often referenced in literature as "wiggling effect". In order to estimate and correct this resulting characteristic curve and other smaller measurement errors, in the work presented here a regression of a polynomial for each pixel is applied to the test setup (planar white wall as reference target object during depth calibration)[13, 177].

All in all, through calibration, mean accuracy, which is limited by physics [141], can be improved, and the accuracy values given in the data sheet for the central area can be reached. Unfortunately, calibration has to be repeated when parameter settings are changed. The calibration for a specific range of integration times is very time consuming, when nominal distances are calibrated by hand. Thus, precise automatic calibration equipment is desirable. Other specific calibration approaches which can improve the gathered distances are for instance presented in [178],[179]. It will depend on the targeted applications, which calibration has to chosen be concerning accuracy, effort, and calibration time processing limits. For the applications presented in this work the calibration processes described proved to be sufficient. Newer implementations of the PMD camera like the PMD CamCube 2.0 with new chip designs promise to have an improved distance measurement quality by compensating parts of the physical/electrical disturbances, like for instance the wiggling effect. The quality of these new camera implementations need to be further investigated. Nevertheless, the presented approaches might even reach better results with the raw data of higher quality from newer camera implementations.

Annotation to Modulation Frequency

As already mentioned the modulation frequency can be changed for the cameras. Hereby the range where ambiguity occurs can be modified. An increased wavelength will lead to a higher measurement range, but also do a higher principle measurement uncertainty and vice versa [141]. Tests with the PMD 19k showed, that the change is in principle possible but require a careful recalibration of the received distances. Especially interesting is, that for higher modulation frequencies than the default $20MHz$, the camera showed significant different, unpredictable behavior for changed integration times. Thus, in this work always the default modulation frequency is used.

3.3.2 Filtering

Due to environmental conditions, disadvantageous reflectivity, or large depth range inside the captured scene, erroneous distance measurements can occur, which need to be filtered out before using the data. In order to do this different standard filters like e.g. mean, median, or gauss with specific dimensions can be applied to the distances. This can be done both on spatial basis (use neighbor pixels for filtering) or temporal basis (use multiple subsequent frames). The applied filters need to be carefully selected and designed in order not to distort the targeted outcomes of the measurements with a PMD camera. With respect to filtering, the PMD cameras have the advantage that most of them also deliver an amplitude and intensity value besides the distance. These values can be used to assess the quality of the received distance measurement on a per pixel basis. In the following two approaches are introduced which were used in this work besides standard filter methods.

Bad Pixel Removal

A very simple way to filter out bad measurements, is to define a lower and upper limit for intensity or amplitude values for which pixel measurements should be eliminated or interpolated with neighbor pixels. As the amplitude represents the amplitude of the modulation signal this value is the best choice for filtering. So pixels with a too low the amplitude can be regarded as erroneous and either filtered out or the distance value for this pixel is interpolated. For the application example presented in this chapter erroneous pixels are eliminated.

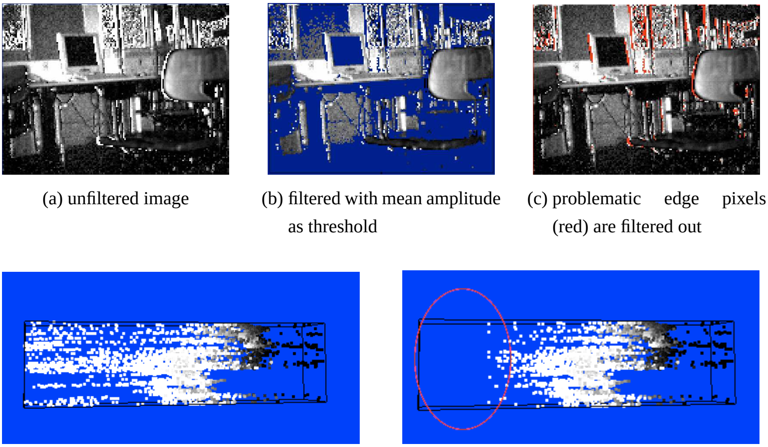
Motion Artifacts

For dynamic scenes (for a moving camera or changing environments), motion noise occurs in the stream of range images from the camera. This noise appears especially at edges of objects, such that it is difficult to determine the distance at this edge pixels (cf. Figure 3.16). In an approach analyzed by [180] these movement artifacts are detected with an additional video camera and gradient edge detection methods. In this work an approach is proposed to filter out these artifacts based only on data received from the PMD camera.

Analysis of these erroneous distance measurements due to motion showed that the problematic edge pixels often own a high intensity value, while having a low amplitude. This fact can be used to detect these erroneous pixels and to remove parts of the motion noise based only on the received data from the PMD camera. Algorithm 2 shows how the filter is realized. At first a selection value $\Delta q(i,j)$ - a weighted difference of intensity value and amplitude value - is calculated, where k is a constant weight factor. If $\Delta q(i,j)$ is larger than a defined threshold $Q_{threshold}$, the pixel at i,j is regarded as erroneous and the corresponding measurement is filtered out. During tests with the PMD 19k $k = 10$ and $Q_{threshold} = 100$ turned out to deliver good results. Figure 3.16 shows the re-

Algorithm 2: Detection of erroneous edge pixels.

```
foreach pixel indices $i, j$ do
    $\Delta q(i,j) = I_{i,j} - k \cdot A_{i,j}$;
    if $\Delta q(i,j) > Q_{threshold}$ then
        pixel $D_{i,j}$ is noisy;
        eliminate pixel $D_{i,j}$ from range image;
    else
        keep pixel $D_{i,j}$;
    end
end
```



(a) unfiltered image

(b) filtered with mean amplitude as threshold

(c) problematic edge pixels (red) are filtered out

(d) top down view of distances without filter

(e) top down view of distances with edge filter

Figure 3.16: Example for filtering. (d and e) distance measurement pixels are from top projected in 2D plane; camera is placed on the right. In (e) one can recognize that eliminated pixels lie in the back, highlighted by a red ellipse (cf. [13]).

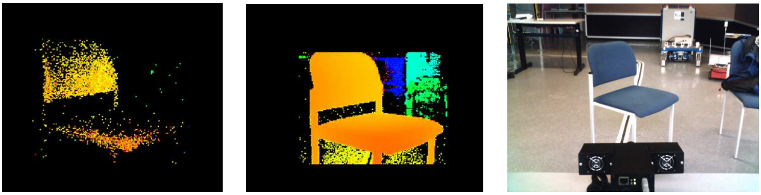
sults of filtering. Figure 3.16a is the unfiltered amplitude image. In Figure 3.16b the same scene is filtered with an amplitude threshold of the amplitude mean. In Figure 3.16c the detected problematic edge pixels (red) by Algorithm 2 are highlighted. Figure 3.16d and 3.16e shows a top down view of the measured distances of the same scene (camera captures the scene from the right). In Figure 3.16d the problematic edge pixels are not filtered. Figure 3.16e presents the result after applying the proposed filter. The erroneous pixels, which mostly lie in the back, are detected and eliminated by Algorithm 2. Large parts of the motion artifacts can be filtered out with this algorithm. Although the results of Algorithm 2 satisfies the need for this work, some of the erroneous pixels, which have an high amplitude, are not detected by the algorithm. If further filtering of these pixels is necessary, other approaches can be integrated, which most probably have to rely on information from further sensors systems in order to detect the small amount of remaining erroneous pixels.

3.3.3 Task Specific Optimization of Integration Time

Integration time is the most critical internal parameter of the PMD camera and can be changed during runtime. It describes the time period in which incoming photons are detected for one measurement cycle, in order to derive phase shift and the corresponding distance. Figure 3.17 exemplifies the influence of the integration time on the overall range image. As already mentioned if the integration time is selected too low, the amplitudes of related pixels decrease and distances for distant objects cannot be measured correctly. On the other hand, if the integration time is too high, oversaturation is observed and measurements fail, too Therefore, a major challenge in obtaining appropriate range images is to find adequate integration time values for particular scenes.

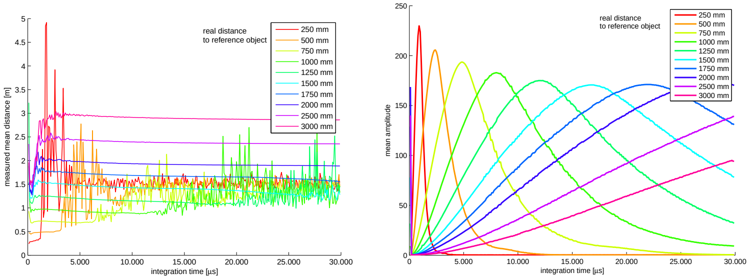
Figure 3.18 points out the relation between integration time, distance measurement, and measured amplitude. For ten reference distances the measured distance by the PMD camera with changing integration time and the measured amplitude are plotted. From this figure it can be clearly seen that for each range interval a specific integration time interval can be selected in order to receive correct distances. This integration time intervals are notably small for short distance measurements. This demands a careful adaption of integration time especially for short distances ($<$1m).

While for some camera types a kind of adaptation of the integration time is realized in hardware (e.g. SR3000), for the PMD 19k and most other PMD cam-



(a) 500 μs (b) 10.000 μs (c) observed scene

Figure 3.17: Influence of integration time, distance is coded by color.



(a) Mean distance vs. integration time.

(b) Mean amplitude vs. integration time [177].

Figure 3.18: Measured mean distances and amplitudes for an object at ten different reference distances to the PMD camera dependent on integration time.

eras it needs to be solved by software. In [181] a proportional controller for the Swissranger SR2 is proposed, which takes the mean intensity value of the image to adapt integration time to a constant preset value. They derive a constant relation between intensity and amplitude values. According to measurements with the PMD 19k for scenarios with near an far objects, which are typical for highly dynamic scenes in mobile robotics, this relation cannot be assumed (cf. Figure 3.19). Especially for near objects, the best distance measurement is achieved with a low integration time, resulting in a high amplitude and low intensity for this near object. For increasing integration time the amplitude for the close object decreases while the intensity increases. This leads to erroneous distance measurements. Figure 3.18a points out that the optimal integration time adaptation input would be the correct reference distance. As the measured distance values might be erroneous, and thus can not be applied directly for adaptation of the integration, here in contrast to the intensity average value, the amplitude mean is used to modify the integration time proportionally. Figure 3.18b in combination with Figure 3.18a support this approach, because the results indicate that a good distance measurement can be achieved for high amplitude values. In Algorithm 3 the steps of the proposed adaptation method are given. The idea is to always control the integration time, such that a constant mean amplitude $\overline{A}_d$ is reached independent of the measured distances and intensities in the observed scene. k

denotes the gain of the proportional integration time controller.

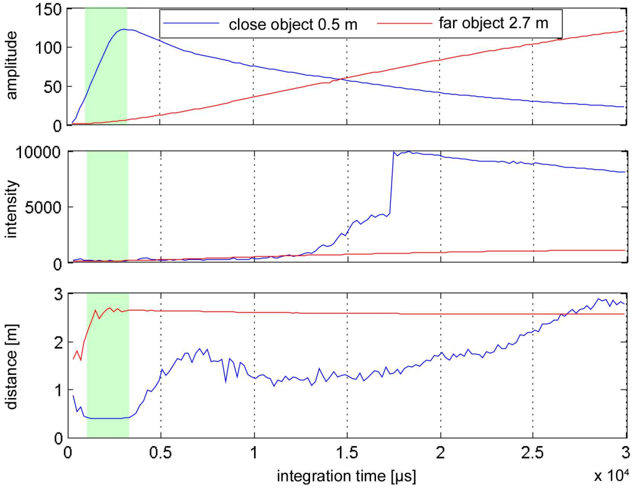


Figure 3.19: Measurement plots for two scenarios - one with a close object (blue lines) and one with a far object (red lines) at reference distances. Green background indicates correct distance for close object.

If the integration time is selected high by the algorithm in order to optimize the amplitude for objects at a higher distance, and then a much closer object moves into the field of view of the camera, instantaneously oversaturation, and hereby a lower amplitude will occur. The presented controller will first try to compensate this with an even higher integration time, which will not lead to an improvement of the amplitude due to the oversaturation. As soon as the integration time $\tau(t+1)$ is larger than a maximum τ_{max}, the integration time of the camera is reset by the algorithm to the minimum integration time τ_{min}. Hereby, the controller can adapt the integration time again in order to optimize the amplitude according to $\overline{A}_d$. In order to avoid an oscillating behavior of the controller due to oversaturation and hereby an unreachable $\overline{A}_d$, $\overline{A}_d$ has to be selected according to a desired minimum distance. The speed of adaptation of the integration time is very much influenced by the framerate. The higher the framerate is, the higher the speed of adaptation can be designed.

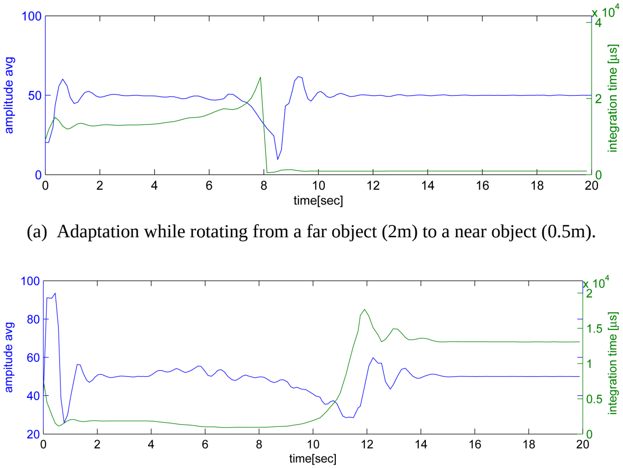
The adaptation of integration time was verified in various test-setups. Figure

Algorithm 3: Integration time update loop.

```
foreach received frame do
    // Calculate mean amplitude $\overline{A}$;
    $\overline{A} = \frac{1}{i \cdot j} \sum_i \sum_j A_{i,j}$
    // Calculate deviation $e$ from desired mean amplitude $\overline{A}_d$;
    e = ($\overline{A}_d - \overline{A}$)
    // Adapt integration time $\tau$;
    $\tau(t+1) = \tau(t) + k \cdot e$
    if $\tau(t+1) > \tau_{max}$ then
        $\tau(t+1) = \tau_{min}$;
    end
    set new integration time $\tau(t+1)$;
end
```

3.20 presents two example results for the working adaptation process. Two objects were placed orthogonal to each other, one with a near distance of $0.5m$ and one with a far distance of $2m$. For the test run the robot with the mounted PMD camera rotates with an angular velocity of 10° per second at first 90° from the far object to the near object (cf. Figure 3.20a) and then 90° degrees back from the near object to the far object (cf. Figure 3.20b). In both scenarios, moving from near to far objects, as well as from far to near objects, the amplitude converges to the selected $\overline{A}_d = 50$ and correct measurements are obtained. Figure 3.20a - the scenario from far to near - shows that the algorithm is also able to handle the problem with the instantaneous oversaturation and the corresponding lower amplitudes. It quickly resets the integration time in order to converge to the selected $\overline{A}_d = 50$.

A consequence of the controller design based on amplitude input is, that the controller will in general adapt the integration time such that for the closest objects the best distance measurements are achieved. In consequence this might even lead to a fading out of objects being far away. However, this exactly meets the demands for the application of a PMD camera for the navigation task in robotics. For this task close objects are the objects which require the most attention and accuracy as they might for instance represent obstacles or other danger for the system.



(a) Adaptation while rotating from a far object (2m) to a near object (0.5m).

(b) Adaptation while rotating from a near object (0.5m) to a a far object (2m).

Figure 3.20: Results of two exemplary test runs for the adaptation of integration time algorithm.

Dependent on the dynamic character of the application and the available framerate of the PMD camera also multiple frames with different selected integration times can be combined to a PMD image of higher quality. The selection criteria which pixel would be used for the final combined frame would be the highest received amplitude of the specific pixel indices. For current implementations of PMD cameras this is not suitable for mobile robot applications.

3.4 Application: 3D-Map User Interface Elements based on PMD Data

Mapping is a major issue in teleoperation systems in order to allow the human operator to externalize memory, in order to maintain situational awareness, and to allow for robust autonomous features for the robot. In the following as an application example first an approach is introduced to build three-dimensional 360° degree point cloud panoramas at a specific robot's position using and considering the constraints of the PMD camera. Then these panoramas are combined to a global map with an hybrid approach using PMD data and laser range finder data. Finally an informative, qualitative user study is conducted in order to find out the principle applicability of these type of visualizations for user interfaces.

3.4.1 Frame Matching

For the fusion of multiple range images from a 3D-ToF camera in this work the Iterative Closest Point (ICP) Algorithm for scanmatching is used. Other approaches e.g. probabilistic would also be possible (cf. Section 3.1.6), but ICP showed better results in current research work for 3D point cloud matching.

The ICP algorithm was first published by Besl/Mckay [171]. The principle idea of this algorithm is to find the closest point pairs for two given frames, the model frame M and data frame D, and thus search a nearest neighbor point p_{m_i} in M for every point p_{d_i} of D. After creating this correspondence C, the error function

$$E(R,t) = \sum_{i=0}^{\|C\|} \|p_{m_i} - (Rp_{d_i} + t)\|^2 \tag{3.12}$$

has to be minimized in order to find a transform (R, t) between both point clouds with rotation matrix R and translation vector t. The algorithm consists of a loop of the described process, runs iteratively, and ends on a threshold error value or time constraint. Acceleration can be reached by integrating more intelligent search methods finding nearest neighbor points, i.e. kd-tree data structures [170].

Optimization of the error function (Equation 3.12) and therefore calculating the transform matrix can be done by single value decomposition or Horn's unit quaternion based method [171][170]. The ICP algorithm will be used in the following to create 3D panoramas.

3.4.2 3D Panoramas

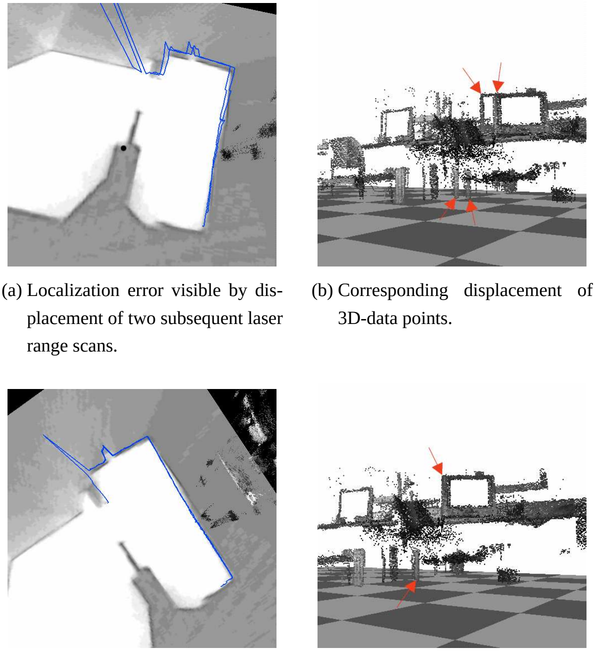
A major drawback of current implementations of 3D-ToF cameras is their limited field of view compared to other sensor due to limited camera chip resolutions and required active illumination units. Laser range finders typically have a field of view of 180° or larger, while 3D-ToF cameras have horizontal field of views in the range between 28° and 52° dependent on the applied optics and illumination units. Larger field-of-views of sensor data visualizations in the user interface can support a human operator in maintaining situation awareness during the navigation task. For the PMD cameras an extension of the field-of-view, and hereby creating 360° panoramas can be achieved by rotating the sensor and merging multiple images into one. The sensor can either be moved by a pan tilt unit or by the robot itself like done here. In contrast to panoramic image vision, for example mountain or city panoramas, in this work panoramas represent distance information colored with the gray scale image from the camera, such that the user is able to estimate distances, change viewpoints, change visualization modes and finally get a better understanding of the local surrounding of the robot.

In a first approach the iterative closest point scan matching algorithm (ICP) was implemented here for a pair-wise matching of PMD frames in order to retrieve independent 360° 3D-Panoramas at specific positions. With the application of the ICP algorithm and the usage of PMD distance data only, after a rotation of 360° the panorama could in most cases not be fully closed. This is a typical problem of pairwise matching of multiple point clouds and is referenced in literature as loop closure problem. For each pairwise matching a small scan matching error is introduced which accumulates for each additional matching. One common approach is to distribute the global matching error over the whole data set and hereby minimizing the global error [170], when loop closure is detected. In this work the closure of a 360° degree can be detected by odometry or by matching images with characteristic features, as long as there is only a little drift from rotation center point. In the test setup here, a SIFT-feature [168] comparison was used and the rotation closure was detected by observing counts of sift-feature matches between first and actual frame through maximum search. Tests showed that the rotation closure can be detected based on SIFT-features of the amplitude image. However, in this work the error distribution is not investigated in more detail as for the current work an hybrid approach with an additional laser range finder has the additional advantage to enable a referencing of the 3D panoramas to the global reference frame.

The independent local 3D panoramas described before can be integrated into a consistent global map of the environment. This supports the human operator to maintain situation awareness through the availability of a 3D representation of the entire work environment of the robot and the ability to take arbitrary viewpoints inside this 3D representation of the environment. As already mentioned the current implementations of PMD cameras have a very small field of view compared to a standard 2D laser range finder. Thus, a hybrid system combining a 2D laser range finder and PMD camera to build a 3D map of the environment is proposed here. The localization of the robot with respect to an existing two dimensional map is done based on two dimensional laser range finder data and the 3D-map data is gathered by a PMD camera. In the test setup used here the underlying localization of the robot in the environment is done by a SICK LMS 200 laser range finder and a standard monte-carlo localization algorithm. The 2D map for monte-carlo localization was built with the 2D-SLAM system of the Player Framework [182]. But also other occupancy grid maps [165] e.g. based on a footprint of a building can be used.

In order to gather these 3D-map of the environment the human operator here defines a set of waypoints where the robot should record a 3D panorama and a panorama scan angle between 90° and 360° for each of these waypoints. At these waypoints specified by the operator, the robot then records the 3D PMD data with rotational steps of 15 degree. For each frame from the PMD camera, the odometry, the gathered pose from the monte-carlo localization, the laser data, and the PMD data is stored. Based on this data for each of the waypoints a 3D-panorama is calculated, and included in a global 3D-map for the human operator based on the localization information. Compared to the approach to generate 3D panoramas only with PMD data and the ICP algorithm, here for global consistency the single PMD frames are not fused directly with the ICP algorithm. For global consistency and to speed up the system the single frames are fused based on the estimated rotation by the localization system. Although, the monte-carlo localization system seems to be very accurate to the human observer, tests showed that for the configuration used here, always a certain estimation error of some degrees exists. Figure 3.21a shows this rotational estimation error for the laser range data (blue lines) between two subsequent data sets and Figure 3.21b the corresponding error in the 3D-panorama.

Thus, since the localization is still erroneous, the ICP algorithm is applied to the laser range data for refining the localization and hereby the rotation value between two subsequent data sets. Hereby, the registration transform errors for the PMD range data is minimized. Figure 3.21a and 3.21b show the panorama registration before the ICP correction and 3.21c and 3.21d after the correction. In



(a) Localization error visible by displacement of two subsequent laser range scans.

(b) Corresponding displacement of 3D-data points.

(c) Subsequent laser range scans after ICP correction.

(d) Corresponding 3D-data points after ICP correction.

Figure 3.21: Example of correction of localization inaccuracies for two subsequent data sets. (a) and (b) without ICP correction based on laser data; (c) and (d) with ICP correction based on laser data (cf. [177]).

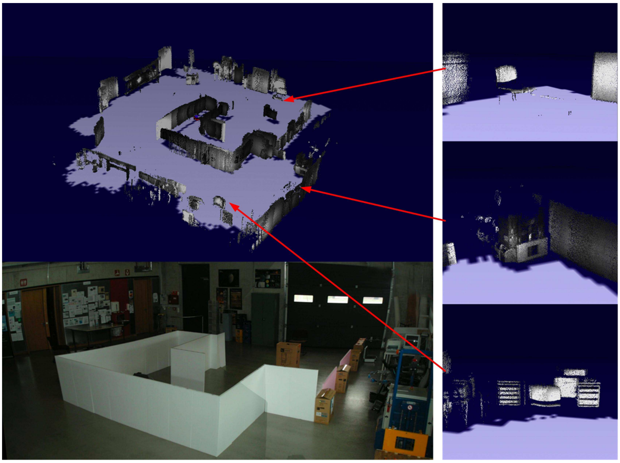


Figure 3.22: Image and generated 3D map of the test environment (left) and three different details of this map (right).

comparison running ICP on laser-range data is much faster, because a laser scan only consists of 181 points whereas a PMD range image has about 10.000 points after applying the filters.

At this point 3D scan-matching with the ICP algorithm seems to be an overhead here, because the robot and the PMD sensor only rotate in one axis. But for future work with additional degrees of freedom, for example moving the sensor in the pitch axis or driving the robot on not flat surfaces, the ICP algorithm as generic approach is advantageous and thus is selected here.

Figure 3.22 shows the environment and the generated map for the later described tests. In addition on the right of the Figure 3.22 details of the map are displayed in order to show that objects are recognizable although the resolution of the PMD 19k camera is very low compared to standard visual cameras.

After recording panoramas and integrating them in a global map, a post-processing of the pointcloud can be executed. Since a pinhole camera model was assumed when calculating Cartesian coordinates for a range image, regions exist, where the point cloud is denser then in other regions. Different methods can be used to optimize this, reduce the number of points which need to be stored, and to extract surfaces from the point datasets. With the octree cell decompo-

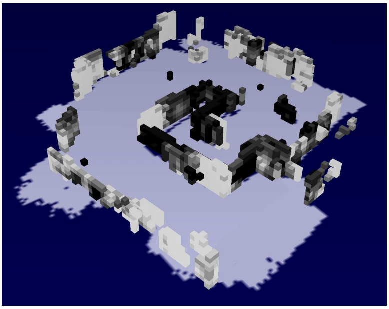


Figure 3.23: Representation of the 3D-map as octree (depth 6), side length of 18.75 cm for each cube, an a minimum of 30 points per cube.

sition for instance the whole environment cubic volume is divided recursively into eight sub-cubes up to a certain resolution depth. For an environmental cubic volume with side length $12m$ and an octree depth of $d = 6$, a side length for the leaf nodes at depth 6 of $\frac{12m}{2^6} = 18.75cm$ results. In this work an octree decomposition method is used for two tasks. On the one hand, a normalization of point density, a filtering of outliers and thus an optical correction of the point cloud is realized. This is achieved by setting a maximum depth of the octree and minimum point count for the cubes a highest tree depth. On the other hand, a decomposition with a selected maximum depth for the octree depending on the desired resolution is applied to reduce the overall number of points in the point cloud. Thus, the number of elements in the example map shown in Figure 3.22 are reduced from 934510 points to 2231 cubes, while a cube of side length 18.75 cm is defined as occupied and visualized if a minimum of 30 point of the 3D map points are inside this cube. Figure 3.23 displays this alternative representation of the point cloud 3D-map of Figure 3.22. Dependent on the application such a simplified visualization might be sufficient and would save a significant amount of memory. For the navigation task alone such an abstracted octree representation of the 3D map, might even be advantageous for the task performance.

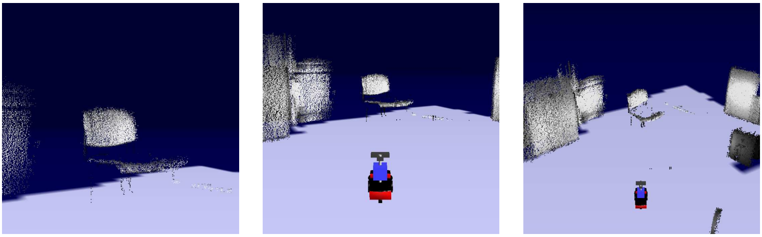
3.4.3 User Studies

In order to enable a basic assessment of how well a human can cope with such 3D maps build from PMD data a qualitative user study for the navigation and object recognition task, was performed. The ability to orient and perceive in these maps, different map representations, and viewpoint preferences are investigated. It was tested how good structures, objects, and the overall environment can be recognized and how well people were able to determine the position and orientation of the robot only based on this type of map data. The target of these tests was not the evaluation of the graphical user interface itself, and the robot control algorithms. In the following some of the test results are given.

Metrics

In the user studies as objective task-oriented metrics the primarily used metrics were, task completion yes or no, and for the object recognition tests also the level of task completion, by means of the amount of correctly identified objects. In addition, the people had to fill a questionnaire for each single test run. The questions were specific for each test run and targeted for subjective ratings of the test participants of their decisions, estimations, and preferences. In the following the important metrics will be introduced for each specific test. It is important to notice again that the tests were designed to gather qualitative results and not for quantitative comparisons.

Setup for Test Runs



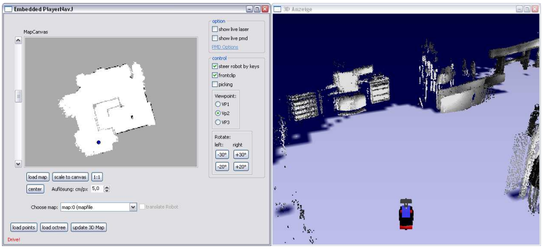
(a) Egocentric.

(b) Near Exocentric.

(c) Far Exocentric.

Figure 3.24: The three selectable viewpoints for an example scene.

As hardware setup for the test runs a mobile robot with differential drive kinematics equipped with a PMD 19K camera, and a 180° laser range finder was



□ show live laser
□ show live pmd
☑ steer robot by keys
☑ frontclip
□ picking
Viewpoint: ○ VP1 ● Vp2 ○ VP3
Auflösung: cm/px 5,0
Choose map: map:0 (mapfile
□ translate Robot

Figure 3.25: Screenshot of the graphical user interface for the tests.

utilized. For the navigation tasks the robot had a safety behavior included, which did not allow the human to operate the robot against an object. In the test environment with a size of 12 m x 14 m a small maze, and various distinctive objects were placed. The test participants sat in separate room, such that they could not see and hear the mobile robot while they were operating it. A picture of the test environment can be seen in Figure 3.22. The 3D-map also seen is this Figure was constructed from 5 panorama acquisition points and has a total of 934510 points. The octree used for one of the tests is presented in Figure 3.23 and was generated as described in the sections before. For the tests with selectable viewpoint three viewpoints were available (cf. Figure 3.24): an egocentric viewpoint, a near exocentric (1m behind, 2m above robot, tilted 11.5°) viewpoint, and a far exocentric viewpoint (3m behind, 3m above robot, tilted 28.6°). Figure 3.25 shows a screenshot of the graphical user interface for the tests.

Test Participants A group of seven test participants with various study background independently performed all the tests. Six test participants were male and one was female. The average experience with operating a robot and also with playing 3D computer games was low.

Test Procedure At first the robot, the sensors, the user interface, and possible ways of interaction were explained to the test participants. Then the test participants were given time to get familiar with the user interface, system reactions, behaviors and control modes of the robot. When the test participants were ready, they performed successively the four different test runs in the order as described in the following sections. After each test run they filled a questionnaire with specific questions to the test run and its configuration. The whole test was recorded with a video camera in order to evaluate afterwards the way people interact with the system. All tests had time limits.

Orientation Test 3D Map (Test 1)

Task The basic idea for this test was to let the human operator solve the "kidnapped robot problem". The robot was placed inside the 3D point cloud map at a specific position and the test participant should find out where the robot is located it the test environment. Therefore, he/she could rotate the robot in place in order to investigate the surrounding of the robot. Finally, he/she should mark this position inside a given 2D-map. The results of this should give an idea about how well people can abstract and reflect the 3D point map generated by the presented approach to a 2D map and how well the quality of the generated map is from the egocentric viewpoint.

Results Only two test participants could identify the robots position nearly correct. This is also reflected in the answers to the question how sure people are about their position. For a scale from 0-very unsure to 3-very sure the average rating was 1.14 with standard deviation of 0.3780. The analysis of the recorded video showed the different approaches of the test participants to tackle the problem. For instance one person made notes about about observed distances, two other participants tried to make use of hand gestures in order to match the 3D information with the 2D map. The biggest problem for the test participants seemed to be the impression of ambiguity of distances and structures in the map due to the egocentric viewpoint.

Human Controlled Navigation and Object Detection (Test 2)

Task In this test the test participants were asked to directly steer the robot from a known start position in the map to a goal position. They were provided with a 2D-map with the current robot position and 3D map with the viewpoint attached to the robot's avatar. The test participants were asked to select their preferred viewpoint relative to the robot from the three available viewpoints. The second task in this test was to find specific objects (chairs, beckrests, and freestanding boxes) in the test environment an to mark their position in a 2D map. If an object was detected, the test participants were asked to optimize the robot's position and to select the optimal viewpoint. The goal of this test was to get an idea how good the test participants can identify objects in this type of 3D map or if inconsistencies, noise, and limited resolution disturb the perception too much. Additionally the test should give an idea how far the rotational panorama generation at distinct position has negative influence on the object detection, because the objects are only seen from these points.

Result All participants could successfully complete the navigation task. In the 3D-map seven objects of the type chair, beckrest, and freestanding box could be found. In average 3.29 (std:[12] 1.2536) could be classified correctly by the test participants, 0.71 (std: 0.4880) were classified wrongly, and 3.00 (std: 1.1547) objects were missed in average. The questionnaire showed that the test participants were always very sure about the robots position and they had almost no fear of collisions while operating the robot in the 3D-map. Thus, the test participants very much trusted the 3D-map, the user interface, and the behavior of the robot. The observations during the tests also showed the importance of the additional 2D-map to reassure the global position in the environment for the human while navigating. For the navigation all test participants except one preferred the exocentric far viewpoint for navigation. Only one used the exocentric near viewpoint for navigation. For the object detection task the preferred viewpoint distribution looks different. Here for 26.1% of the objects the egocentric viewpoint, for 34.8% near exocentric, and for 39.1% the far exocentric viewpoint was prefered. Looking at the corresponding distances between robot and the object, it can be seen that for most of the egocentric viewpoint selections these distances were below 1m. This might be due to the fact that it is advantageous that the egocentric viepoint in these situations is close to the point where the PMD data acquisition took place.

Human Controlled Navigation with Octree Representation (Test 3)

Task For the navigation test with octree map representation test participants were again asked to steer the robot like in Test 2 from a given start position to a given target position. Again they were asked to select their preferred viewpoint from the available three viewpoints.

[12] std - standard deviation

Results For this type of map representation the test participants replied in the questionnaire that they mostly do not fear any collision and trust the map and the user interface. They also fully agreed that they always were aware of the robot's position in the environment. Two test participants preferred the near exocentric viewpoint. All others preferred the exocentric far viewpoint. The video recordings again showed the changing attention of the human operator between the 3D-map and the 2D-map in order to reassure the global position of the robot. The test participants were also asked in the questionnaire about their preferred configuration for the navigation task. Table 3.4 shows the answers for this question. The preference for the octree 3D-map results mainly from the fact, that this type of representation, simplifies the environment to an abstract representation as a convex hull. Thus, with this abstracted view the decisions about robot control for the human might require less cognitive effort, than a complex 3D point cloud.

Table 3.4: Preferred configuration for navigation.

	egocentric viewpoint	near exocentric viewpoint	far exocentric viewpoint
point cloud 3D-map	0 %	7.1 %	21.4 %
octree 3D-map	0 %	14.3 %	57.1 %

Orientation Test with Live PMD Data (Test 4)

Task The orientation test with live PMD data followed the same approach as test 1 with the static 3D point map. After all the test participants completed the other tests and thus knew the environment, the robot was placed at a defined position and they were asked to find out again the robot's position in the test environment only by means of rotation of the robot and mark it in the 2D map. For this test they were presented live PMD data instead of the 3D point map, and thus only the limited field of view (horiz. field of view $\approx 30^\circ$) of the camera was available. The live PMD points were color coded according to the distance, to enable a better depth perception of the test participants.

Results For this test four test participants could identify the robot's position in the environment. The other participants who marked the wrong position of the robot in the 2D map were able to recognize the objects identified in the tests before, but were not able to correctly match this with the structures of the global environment. This again validates the hypothesis that it is very difficult for the human to correlate the 3D structures from the egocentric view of a PMD camera without any other references.

Summary

A first series of user test was carried out to show the usability of the 3D map generated with the PMD camera for navigation and object recognition. In the tests, different representations were used (octree and point cloud). Furthermore, in one experiment the PMD camera was used as single live sensor feedback from the remote environment for navigation/orientation experiments. From the user tests, it could be confirmed, that navigation of a mobile robot with 3D maps built with the presented approach, and understanding of spatial relations in the environment is possible with this limited information (distance only) from the remote environment. In particular the octree representation and the wide exocentric viewpoints enhance robot control for the navigation task, while the point cloud is still that detailed, such that the human operator is able to recognize and classify objects.

During the orientation tests the test participants had to determine the robot's position in a 2D map according to a live range image stream from the remote environment or a 3D point cloud map. Many of the participants failed this task, what showed that either the the cognitive effort for this matching is too high or the 3D information is just not sufficient here. Thus, for the used PMD sensor, a user interface for navigation with PMD range images should be extended by other sensor information (e.g. standard camera) providing a larger field-of-view and a higher resolution. Besides that, range image presentation should be further processed, for example it has to be found out, what improvements are possible, when the point cloud is used to construct surfaces.

3.5 Conclusion

In this chapter contributions are presented which can improve a teleoperation system through the adaptation and optimization of the two core system components communication and environment perception.

Network parameters can significantly degrade the performance of teleoperation systems. As the control loop in such system is closed over a human, some of the issues can be compensated by the human. It is necessary to identify the important, crucial characteristics and parameters of the communication system dependent on task and mission setup. For the exemplary application scenario of this thesis, wireless communication would be optimal - at best with ad-hoc capabilities. For mixed reality and especially for augmented reality user interfaces the video image from the remote side is also the major component. In general a video stream has high demands on the bandwidth. Bandwidth is a limited resource in human-robot teams. An advantage in this case is, that systems inside such a team in general do not behave egoistic like it is the case in consumer networks. This allows for an optimization with respect to the communication of the whole human-robot team. The presented traffic shaping concept makes use of this fact, through the reduction of the requested bandwidth in order to keep the entire communication alive. The proposed concept enables to increase the teleoperation performance in an multi-hop network, as it stabilizes the framerate of a video stream and avoids stagnating video streams which might lead in worst case the operator loose the sense of orientation.

PMD-cameras offer interesting potential to capture 3D-images for mobile vehicle applications. Specific parameter adaptation schemes are proposed in order to improve image quality. This thesis addresses in particular adaptation of integration time, calibration of data, as well as filtering schemes for bad pixels targeting for the mobile robot navigation task. The proposed methods provided significant improvements of the measurement quality and hereby the three-dimensional environment perception. The application example presents how mapping algorithms can be applied with some limitations also to data from a PMD camera in order to realize assistance for a human teleoperating a mobile robot. The usability of the gathered environment representation could be validated with a informative, qualitative user study.

4 Sliding Autonomy through Haptic Guidance

When designing autonomy and assistance systems for human-robot interaction, the interaction roles of the humans involved in the systems need to be carefully considered. In general for a navigation task the human takes over the supervisor role, the operator role or both at the same time with respect to the robot (cf. Section 2.1.2). The actual role of the human very much depends on how the teleoperation interfaces are realized (cf. Section 2.2), which autonomy concepts, and which autonomy level transition concepts (cf. Section 2.2.1) can be and are applied. If for instance direct teleoperation is used, the human is in the operator action role. The overall teleoperation system is optimized for an optimal execution of the user's commands on the robot and optimal feedback from the remote environment to the user. This effects especially the design of the user interface, the dimensioning of the communication channel, low-level control features like path-tracking etc.. If a fully autonomous navigation is realized the interaction role of the human is supervisor. In this case the human will take over monitoring tasks and additional tasks like searching a received camera stream for things of interest. If the search task can also be done by e.g. computer vision algorithms, the supervisor interaction of the human with the robot is only event driven. Typical events would be e.g. a found object, or a request for a new search area [14].

In most cases the navigation task can not be realized fully autonomously in a robust way due to complex, dynamic workspaces of the robots. One the other hand often two or more humans are needed to operate a robot for a search task in a remote environment efficiently. One would navigate the robot and the other one would search e.g. a high-resolution camera images for points of interest. If one person should take over both tasks the workload would be high to operate the system. Another significant issue is the general problem of operating robots from remote - the lack of understanding what the robot is doing and why it is doing it. This aspects are addressed by the human factor terms "common ground" and "situational awareness".

The presented concept in the following enables to overcome parts of these problems by realizing sliding autonomy through haptic guidance. Haptic or force feedback for the human operator can be realized in different ways. A lot of input controllers from the commercial entertainment sector provide quite simple haptic feedback through rumbling of parts of the controller. These controllers mainly target for attention management. Other much more complex controllers like e.g. the up to six degree of freedom Phantom device developed by Sensable[1] or the discontinued two degree of freedom haptic joystick from Immersion[2] allow a much more accurate force rendering to the human operator. In robotic telepresence research (cf. Section 2.3.1) a lot of custom made systems can be found e.g. in [33] or [76]. These system are developed to research bilateral teleoperation [49] in master-slave systems. Here a direct teleoperation control loop including the human is closed over a potentially delayed communication link. Besides visual feedback the force reflecting component from the remote environment is a major component to maximize the telepresence for the human operator.

In contrast to this, in the concept proposed here, the major target is not to increase the telepresence of the human operator but the task performance. To achieve this, situational awareness and common ground need to maximized and in parallel the workload needs to be reduced in order to enable a human to fulfill both interaction roles - the supervisor role and the operator role. As the major design issue is shifted from reaching maximum telepresence to reaching maximum task performance the gathered information from the remote environment needs no longer be rendered directly to the human operator, and hereby the control loop can be decoupled to some extend, information can be modified and augmented, and assistance system can be integrated. The haptic force rendering is applied in this concept as element of a mixed reality user interface in order to make use of a second important sense of the human.

For the proposed sliding autonomy concept the human operator at first defines a certain set of points of interest he/she wants to investigate in the robot's workspace. The system takes care of planning and controlling the robot's motion including obstacle avoidance and will navigate the robot along the points of in-

[1] http://www.sensable.com/ (24.11.2009)

[2] http://www.immersion.com/ (24.11.2009)

terest through the environment fully autonomously as long as the human operator is not overriding the autonomous movements. The system's navigation intention is presented to the human operator through a directed force rendered on a haptic joystick and a graphical mixed reality user interface. As long as the operator agrees with the system's plans he/she just accepts what the systems is suggesting. This means that he/she does not counteract to the forces, the haptic joystick renders. If the human operator does not agree with the systems plan he/she either adapts the system's plans with smaller adjustment of the joysticks position or even completely overwrites the systems autonomous functions. Compared to other approaches (e.g. [183] a four level sliding autonomy for a docking task), where the switching between levels of autonomy is a discrete event and needs a certain time and dialogue between human and system, the presented concept here allows for dynamic, smooth and immediate change in the level of autonomy and control. Figure 4.1 shows how the level of autonomy is determined by the level of human intervention. If the human overrides the forces rendered by the system, he/she directly operates the robot. If he/she follows the suggestions of the system, the robot navigates fully autonomous. The following section describes in detail how this can be realized by the developed concept.

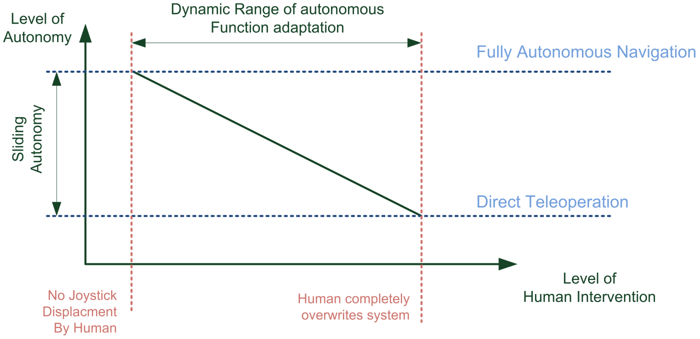


Figure 4.1: Dynamic level of autonomy determined through immediate change of level of human intervention.

4.1 Concept

The proposed concept is based on providing the operator the robot's current intention through a force feedback. In the experimental setup shown in the later section, this is realized with a force feedback joystick. The different calculated guidance forces are fused to a combined force vector. The resulting force is rendered on a two degree of freedom force feedback joystick in direction of the combined force vector. The resulting joystick position x,y represents the current planning of the robot including obstacle detection, collision prevention, and navigation towards a goal. Thereby, the force calculating system together with the joystick control is behaving like an autopilot for the mobile robot as long as the human operator is not moving away the joystick from the system's suggested position. Without this human intervention the robot would move fully autonomous along its planned path.

The generated force and its direction is composed of three different single forces - an environmental force (EF), a collision preventing force (CF) and a path guidance force (PGF). The environmental force and the collision preventing force are further developments of concepts proposed by Lee et. al [184].

The environmental force represents a force, which acts in opposite direction of the closest obstacle in the sensing range of the distance sensors of the mobile robot. Compared to the system in [184] here also sensing radii below 360° can be used.

The collision preventing force takes care that the robot always uses the safest trajectory through a narrow passage. This is realized by balancing the possible left and right turn angle without collision in a certain safety region of the robot which is also an extension of the concept of [184]. This force for instance would guide a robot, which should pass a door, through the middle of this door.

While the two already described forces represent the collision prevention and planning in the local surrounding, the third force is inter-connected with a global path-planning system and implements a force in direction to the next waypoint of the path-planning system. Figure 4.2 gives an overview how the the different elements of the system are interconnected and which data is exchanged in-between the components. An important aspect is, that the rendered force at the joystick F'_C together with the force applied by the human F_H on the joystick determine a certain position of the joystick x and y in joystick coordinates. This joystick position is then used to generate the inputs $u(t)$ for the robot which are transmitted to the robot control system through an IP-based network. The human gets feedback of the environment through the graphical user interface and the haptic feedback. The haptic feedback also presents the robot's current intention to the human operator in a very natural and intuitive way. The almost same alignment of the robot's frame of reference, the viewpoints frame of reference and the joystick alignment very much supports to maintain situation awareness, common ground, and especially to reduce the workload of the human operator. They only differ in small translational shifts such that there are no mental rotations necessary by the human operator and thus less effort from the human operator (cf. section 2.3.5).

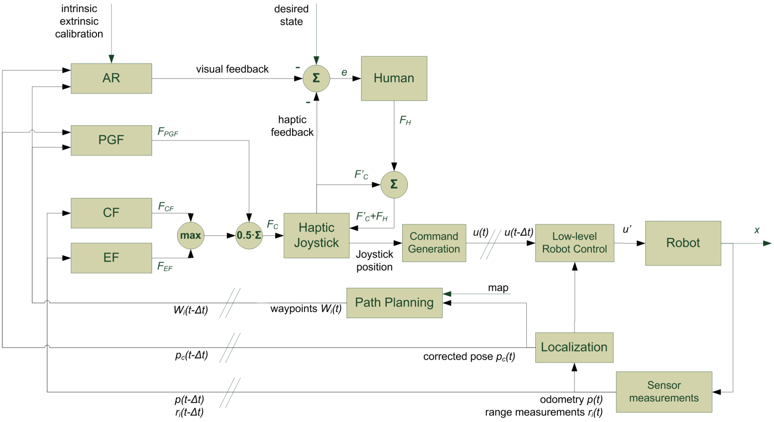


Figure 4.2: Implemented concept of haptic guidance system.

4.1.1 Command Generation

Input - Command Transformation

In order to map the inputs of the human operator through the haptic device - the force feedback joystick - to the motion control commands of the mobile robot, a transformation was introduced which is similar to the car-driving metaphor in [184].

For a better illustration in Figure 4.3 the angular positions of the joystick are projected in a 2D plane. For the transformation to motion commands this has

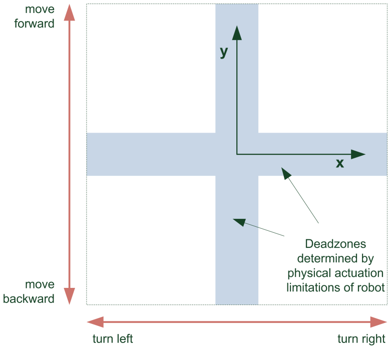

move forward
move backward
y
x
Deadzones determined by physical actuation limitations of robot
turn left
turn right

Figure 4.3: 2D-Projection of mapping joystick position x, y to motion commands velocity v and turnrate ω.

no significant relevance. The deadzones mark the areas of the joystick positions, where no movement of the robot is resulting due to physical limitation in actuation of the mobile robot. Equation 4.1 shows how the joystick positions x, y are transformed to motion commands velocity v and turnrate ω. c_v and c_ω denote the scaling constants for velocity and turnrate.

$$\begin{aligned} \text{velocity} \qquad & v = c_v \cdot y \\ \text{turnrate} \qquad & \omega = c_\omega \cdot x \end{aligned} \tag{4.1}$$

Joystick Position Control

The realization of some of the force components of the guidance system and especially the path guidance force requires, a here so called position hold force of the joystick. This is implemented with a simple position controller. For the system model of the joystick the following second order system in state-space with ξ as system state and u as system input is assumed for simplification:

$$\dot{\xi} = f(\xi, u) \quad \text{, where} \quad \xi = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \quad \text{, and} \quad u = \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} \tag{4.2}$$

The applied system input u of the controller is proportional to the force rendered by the joystick. In order to generate a position control the following controller is designed:

$$u = -K \cdot \xi$$

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = - \begin{pmatrix} k_1 & 0 & k_2 & 0 \\ 0 & k_3 & 0 & k_4 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \tag{4.3}$$

The designed controller behaves like a spring damper system on both joystick axis. As for practical reasons no detailed dynamic model is required of the haptic device used here and the sampling and control frequency are high enough (approx. 100 Hz), the constants k_1, k_2, k_3, k_4 can be determined experimentally. The joystick axis are build symmetrically, such that k_1 can be selected equally to k_3 and k_2 equal to k_4. Network effects are also not relevant for the controller as all force relevant calculations and controller are implemented directly at the haptic device controlling machine. In Equation 4.3 the equilibrium of the controller would be at zero position of the joystick. A specific setpoint (x_d,y_d) of the controller can be introduced as follows:

$$u = -K \cdot \begin{pmatrix} x - x_d \\ y - y_d \\ \dot{x} \\ \dot{y} \end{pmatrix} \tag{4.4}$$

If there is no force F_H applied to the joystick by the human, the controller presented here will take care that the joystick will always keep the desired position (x_d,y_d) and respectively the motion commands for the mobile robot are generated.

4.1.2 Guidance Forces

Environmental Force (EF)

For the environmental force F_{EF} for each distance measurement point a force vector is calculated, such that forces are directly emitted by the obstacles in the defined sensing range r_{max}. This is to some extend comparable to the repulsive

force component of the potential field approach [185].
The extension of the approach in [184] presented here can also be used with a non 360 degree distance range measurements of the robot. If less than 360 degree sensing angle is available, the standard approach would lead to the effect that the obstacle reflecting forces would move the joystick in the areas where no sensor measurements are taken. For instance if a 180 degree laser range finder mounted on the front of the robot, the environmental force components would always direct in $-y$ direction. Thus, most probably the joystick will get in a negative y position and consequently the robot will move backward (cf. Figure 4.3), what is not desirable for the system. In order to avoid this, a hold force to the joystick zero position with the joystick position controller as presented in section 4.1.1 is applied in the areas/directions where no distance measurements are taken.
From the all the single reflecting forces the maximum emitted force is taken and not the average, because for the average the range of the resulting forces would be too small and too blurred to be presented to the human operator. Figure 4.4 shows the principle and the different parameters to realize the environmental force

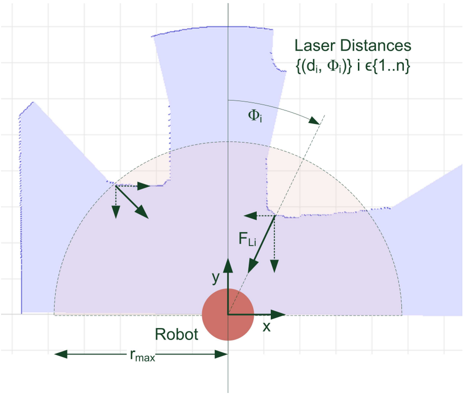


Figure 4.4: Parameters and principle of environmental force - the light red area marks the sensing area defined by r_{max} where the environmental force is considered. The blue area is the free area according to the current laser range measurement.

Each time a laser measurement with n scan points is taken, for each tuple

d_i, Φ_i with index $i \in 1..n$ of this scan an emitted force F_{l_i} is calculated. d_i is the measured distance of a laser ranges scan at index i, and Φ_i the angle of the scan point i defined clockwise from the heading vector of the robot. For a laser range finder as used in the experiments presented here with a total scan angle of 180°, Φ_i can have values in the range from -90° to $+90^\circ$. The emitted force F_{l_i} from a scan point i is determined as follows:

$$F_{L_i} = \begin{pmatrix} F_{L_i,x} \\ F_{L_i,y} \end{pmatrix} = \begin{cases} \frac{d_i - r_{max}}{r_{max}} \cdot \begin{pmatrix} \sin \Phi_i \\ \cos \Phi_i \end{pmatrix} & \text{if } \Phi_i \geq 0 \\ \frac{d_i - r_{max}}{r_{max}} \cdot \begin{pmatrix} -\sin \Phi_i \\ \cos \Phi_i \end{pmatrix} & \text{if } \Phi_i < 0 \end{cases} \tag{4.5}$$

The radius r_{max} defines a circular area around the robot in which distance measurements should be considered in order to generate an environmental force. Figure 4.4 gives an overview about the principle of the environmental force approach.

Based on the emitted forces by the scanned obstacles the x and y components F_{EF_x} and F_{EF_y} for the resulting environmental force vector F_{EF} can be determined as follows:

$$\begin{aligned} F_{EF_x} &= \begin{cases} k_4 \cdot \min_{i=0}^{n} \{0, F_{L_i,x}\} & \text{if } y \geq 0 \text{ and } \Phi_i \geq 0 \\ -k_4 \cdot \min_{i=0}^{n} \{0, F_{L_i,x}\} & \text{if } y \geq 0 \text{ and } \Phi_i < 0 \\ 0 & \text{otherwise} \end{cases} \\ F_{EF_y} &= \begin{cases} k_1 \cdot \sqrt{y} \cdot \min_{i=0}^{n} \{0, F_{L_i,y}\} & \text{if } y \geq 0 \\ -k_2 \cdot y - k_3 \cdot \dot{y} & \text{if } y < 0 \end{cases} \end{aligned} \tag{4.6}$$

As $F_{L_i,x}$ and $F_{L_i,y}$ are always smaller than 0 if the measured distance d_i at angle Φ_i is smaller than r_{max}, the minimum of all $F_{L_i,x}$ and $F_{L_i,y}$ and 0 gives the most significant value in the area of relevance defined by r_{max}. The additional 0 in the set from which the minimum is taken, makes sure that only an environmental force F_{EF} larger than zero is generated if an obstacle is in the considered area ($d_i \leq r_{max}$).

The joystick reference frame is always co-aligned with the robot reference frame such that Φ_i for the joystick force calculations is in the same reference frame and no further transformation is needed.

Collision Preventing Force (CF)

The collision preventing force is an additional guidance force component which considers also the current moving direction of the robot. It helps to keep the robot in the middle of narrow passages in order to prevent future collisions of the robot. This is achieved by keeping the maximum clock-wise turn angle Φ_{cw} and counter clock-wise turn angle Φ_{ccw} without obstacle in the defined prevention region the same size .

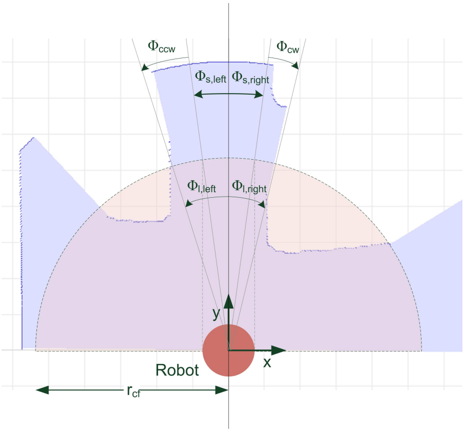


Figure 4.5: Parameters and principle of collision preventing force - the light red area marks the sensing area defined by r_{CF} where the collision preventing force is considered.The blue area is the free area according to the current laser range measurement.

In order to achieve this, the angles $\Phi_{s,left}$, $\Phi_{s,right}$, $\Phi_{l,left}$ and $\Phi_{l,right}$ are needed (cf. Figure 4.5). These angles are measured to the left and to the right from the central robot axis in forward driving direction. $\Phi_{s,left}$ and $\Phi_{s,right}$ represent the segment of the avoidance region or respectively the maximum width which is covered/needed for the robot in order to pass a narrow passage.

For the here assumed approx. circular shape of robot (cf. later described experiments) $\Phi_{s,left}$ and $\Phi_{s,right}$ are equal and can be determined as follows:

$$\Phi_{s,left} = \Phi_{s,right} = \arcsin \frac{r_{robot}}{r_{cf}} \tag{4.7}$$

r_{robot} denotes the radius of the robot and r_{cf} the radius of the region in which the collision preventing force should be applied. Any other robot shape can be applied to the concept as long as $\Phi_{s,left}$ and $\Phi_{s,right}$ represent the angular segment of the collision preventing region which is needed by the robot to pass through a narrow passage safely.

$\Phi_{l,left}$ and $\Phi_{l,right}$ are the angles to the left and to the right of the moving direction of the robot where the first time the measured distance d_i is smaller than the avoidance region radius r_{cf}.

Based on this Φ_{cw}, Φ_{ccw} and the desired new orientation of the robot Φ_{mean} can be calculated as follows:

$$\begin{aligned}
\Phi_{cw} &= \Phi_{l,right} - \Phi_{s,right} \\
\Phi_{ccw} &= \Phi_{l,left} - \Phi_{s,left} \\
\Phi_{mean} &= \frac{\Phi_{cw} + \Phi_{ccw}}{2}
\end{aligned} \tag{4.8}$$

Finally, from Φ_{mean}, Φ_{cw} and Φ_{ccw} the collision preventing force components can be calculated:

$$\begin{aligned}
F_{CF_x} &= \begin{cases} k_5 \cdot \frac{\Phi_{mean} - \Phi_{ccw}}{\Phi_{mean}} \cdot y & \text{if } y \geq 0 \\ 0 & \text{otherwise} \end{cases} \\
F_{CF_y} &= \begin{cases} k_6 \cdot \frac{d_{front} - d_{max}}{d_{max}} \cdot y & \text{if } y \geq 0 \text{ and } d_{front} < d_{max} \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{4.9}$$

The y-component of the collision preventing force F_{CF} is calculated from the current distance of the obstacle in front of the robot d_{front} and a safety distance d_{max}. It it responsible to slow down the robot in case of a near obstacle directly in front of the robot in order to enable the collision preventing x-component to act.

The collision preventing force is through its design inherently a very helpful and supporting element for the human operator in many typical navigation scenarios (e.g. drive through narrow passages, sharp turns). Through the additional consideration of obstacles which are in most cases not in the field of view or the current region of interest of the human operator it can significantly reduce the risk of collisions.

Path Guidance Force (PGF)

The environmental force and the collision preventing force are designed as reactive elements in the local, nearer environment of the mobile robot, which only occur if the robot enters an area with detectable obstacles. In contrast to these reactive elements, the path guidance force actively triggers a movement of the robot based on a global path-planning as long as waypoints to pass exist. The EF and CF hereby are local, reactive forces and PGF is global and active force with a longer time horizon.

The PGF is realized with the joystick position hold force as introduced in section 4.1.1. This hold force enables the sliding autonomy as without human intervention the robot drives the most desirable path according to the system force calculations, the sensor and navigation data. The desired position of the joystick (x_d, y_d) in order to head for current next waypoint W_i is calculated as follows:

$$x_d = k_7 \cdot \Delta\Theta \tag{4.10}$$

$$y_d = \begin{cases} k_8 & \text{if } d_{wp} > 1\text{ , } d_{wp} > d_{tol} \text{ and } \Delta\Theta < \Theta_{tol} \\ k_8 \cdot \sqrt[4]{d_{wp}} & \text{if } d_{wp} \leq 1\text{ , } d_{wp} > d_{tol} \text{ and } \Delta\Theta < \Theta_{tol} \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

The difference between the heading of the robot and the heading towards the next waypoint $\Delta\Theta$ determines the desired position x_d of the joystick in x-direction (cf. Eq. 4.10) corresponding finally to an angular velocity ω of the robot. Thus, the new ω is proportional to the relative angle between robot next planned waypoint.

The translational speed v and respectively the desired joystick position y_d is calculated from the distance to the next waypoint d_{wp} (cf. Eq. 4.11). y_d is only different from zero in case the heading of the robot is in direction of the next waypoint up to a certain tolerance ($\Delta\Theta < \Theta_{tol}$). As soon as the robot is close to the currently selected waypoint ($d_{wp} \leq 1$) the robot is slowed down in order to allow for a softer switching to the next waypoint. This is reached by using the 4th root of the distance d_{wp} in the surrounding of the waypoint. If the waypoint is reached up to a certain tolerance d_{tol} the next waypoint W_{i+1} is selected or if this has been the last waypoint from the global path planning no further F_{PGF} is generated.

From the calculated desired joystick position (x_d, y_d), the corresponding hold force F_{PGF} is calculated (cf. Equation 4.12).

$$F_{PGF} = \begin{pmatrix} F_{PGF_x} \\ F_{PGF_y} \end{pmatrix} = -K \cdot \begin{pmatrix} x - x_d \\ y - y_d \\ \dot{x} \\ \dot{y} \end{pmatrix} \tag{4.12}$$

Combined Force Rendering

In order to avoid a lot of network effects and to reach a high control frequency, the whole force calculations, command generation and closed loop joystick control is realized on the system situated on the operators side. The low-level robot control (e.g. speed controller), localization and all the sensor data pre-processing is done on the robot system (cf. Figure 4.2). Thus, the operator system receives, waypoints W_i to a certain goal pose p_{goal}, the pose corrected by the localization system p_c, the odometry pose p, range measurements from the e.g. a laser sensor r_i and combines this to a force and a visual feedback to the operator. The rendered force F'_C together with the force applied by the human F_H results in a certain joystick position which is transformed to the control input $u = \left(\begin{smallmatrix} v \\ \omega \end{smallmatrix}\right)$ and transmitted to the robot. Through this network effects can be minimized.

The different force rendering components environmental force F_{EF}, collision preventing force F_{CF} and the path guidance force F_{PGF} are combined as follows:

$$F_C = \begin{pmatrix} F_{C_x} \\ F_{C_y} \end{pmatrix} = \begin{pmatrix} \frac{F_{PGF_x} + \max(F_{CF_x}, F_{EF_x})}{2} \\ \frac{F_{PGF_y} + \max(F_{CF_y}, F_{EF_y})}{2} \end{pmatrix} \tag{4.13}$$

In Equation 4.13 the result of the maximum function is defined as the maximum of the absolute values of the arguments. From Equation 4.13 it can be seen that F_{CF} and F_{EF} are combined with component wise absolute values' maximum of the force vectors instead of using the mean. This is necessary, because the highest value from these forces also represents the most critical situation in the sensing area of the robot. If for instance in this case the mean is taken, the deflecting character of the rendered force to the human operator would have been reduced, what is not desirable for this kind of system. For the combination of the path guidance force and the force resulting from the combination of the environmental force and the collision preventing force this is different. If this forces are combined also in the same way like described before (component wise maximum) this would lead to continuous switching between the path guidance force, and the combined environmental force and collision preventing force. This would be very confusing for the human operator. Thus, the combined force F_C is calculated by the component wise mean of the path guidance force F_{PGF} and the component wise maximum of F_{CF} and F_{EF}.

An important aspect for consideration is, that the concept will not guarantee that the robot will not collide. Thus, the human operator can always achieve intentionally a collision. If this is not desired and additional security system needs to be implemented (e.g. do not allow a translational velocity if the robot is closer to an obstacle than a certain safety distance).

4.1.3 Augmented Reality User Interface

The visual component of the user interface is realized as augmented reality user interface with the framework presented in Section 5.3.2. Different interface elements can be visualized on demand overlayed on the camera image received from the mobile robot - a two-dimensional minimap, the planned path, a compass rose and the current laser data. The registration is done as suggested in Section 5.2.1 with a combination of pre-calibration of intrinsic and extrinsic camera parameters, and the continuous adjustment of the extrinsic camera parameters through the robot's localization system. Although the calibration would be sufficient for a good augmented reality overlay, a displacement in the overlay in the screen shot of the user interface (Figure 4.6) is visible. This results due to the fact, that here a viewpoint slightly above and behind the robot has been selected. Results from literature [52, 92] and own experience with such teleoperation systems showed that an exocentric viewpoint (slightly above and behind the robot) is the best choice for such type of tasks. In addition the virtual camera field of view (here: approx 70°) has been increased compared to calibrated field of view of the physical camera (here: approx 45°) in order to give the human a wider field of view of the virtual elements. This field of view increase has no effect on the overlay without the viewpoint displacement described before. Occlusion is not handled for this implementation of the graphical user interface.

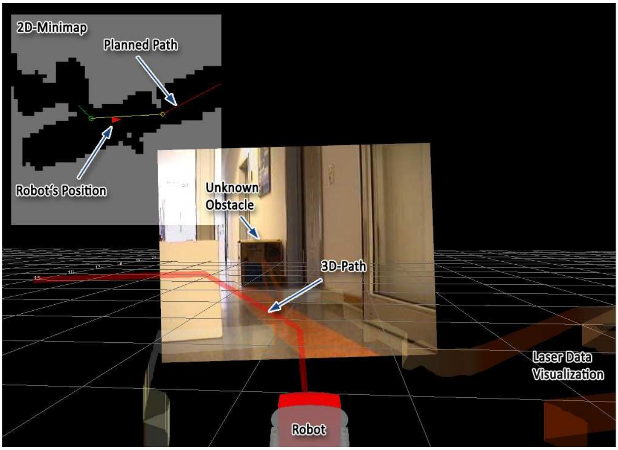


Figure 4.6: Screenshot of the realized augmented reality user interface.

4.2 User Studies

At first a set of user studies with a simulated robot and a simulated environment was performed in order to have reference tests with less uncertainties than with a real robot in a real environment. Afterwards the first gained experience and results of the simulation user experiments were applied to the system and test series with the real robot (cf. Figure 4.7) were performed. The aim of the experiments was to evaluate the developed concept in different configurations with respect to the navigation performance of the user. In order to minimize learning effects all participants were given a certain preparation time to get familiar with the system and test the system. In addition, the order of the test runs with the different modes was altered for each test participants equally distributed in order to avoid biased results due to a static order of test modes for each test participant.

4.2.1 Metrics

In order to evaluate the effects of the different assistance system and to validate the hypothesis of improving the navigation performance of a human operator with the developed concept different objective and subjective evaluation metrics were used. According to the definition of HRI metrics in [99] task-oriented metrics for

the navigation task are used to measure the system performance.
The effectiveness measures how well a the navigation task was completed in the test run. Here the following metrics were used:

- Task completed or not
- Deviation from the planned path
- Number of collisions

As measure for the efficiency of the system the following metrics were utilized:

- Total navigation time
- Velocities and velocity profiles

In addition, backward movements were recorded as special incidents.

Besides these quantitative measurable performance values all test participants had to fill a questionnaire after the experiments. The participants had to rate the different elements of the system. In addition, they had to perform a self-evaluation of the test situation for each of the test modes: They were asked for their fear of collisions, their feeling of control, the ability to recognize obstacles, how intuitive the user interface is and how good they can maintain the overview of the scenario.

All these metrics also allow for an indirect measure for the situational awareness of the human and common ground between human and robot.

For the analysis and comparison of the different performance metrics the average values and the standard deviations are given. Boxplots from descriptive statistics are used for visualization of the data. Boxplots allow to display differences between groups of data without making any assumption to the underlying statistical distribution. They enable to intuitively visualize the dispersion and skewness in the data. The red line in the boxplot shown here denotes the median. The blue box visualizes the lower and the higher quartile around the median and hereby 50% of the whole results. The whiskers extending from each end of this box marks the most extreme values within 1.5 times the interquartile range of the blue box. The red plusses indicate outliers of the maximum whisker range. For performance comparison additonal notches in the boxplots are used as visual hypothesis test for a significance level of 5% (assumption for the notches/significance level normal distribution). If the notches do not overlap it can be concluded with 95% confidence that the true medians do differ.

4.2.2 System Setup for Experiments

For the system tests and the user experiments a mobile robot Pioneer P2DX from MobileRobots[3] was used. It has differential drive kinematics and an approximate circular shape. It was equipped with a standard laptop running Linux as operating system, a SICK LMS200 laser range finder, a sonar array and a Sony network camera with approx. 45° field of view. The whole communication to the operator station was realized over standard WLAN (IEEE 802.11g). As hardware abstraction layer the Player framework [182] was integrated with a wavefront approach [186] for a global path planning and the adaptive monte-carlo localization (amcl) [187] for localization based on a partially known map. During the test runs these components proved to be more than sufficiently robust and accurate for the purposes here. The selected Player framework also allows for a seamless switching between simulation of robot/environment and the real physical robot/environment during system development and experiments. The force feedback joystick to test the presented concept was an Immersion Impulse 2000 Joystick [4].

The mixed reality user interface as already described is based on an software framework developed in the context of this work (cf. Section 5.3.2). The used framework enables to quickly realize three-dimensional graphical user interfaces in the mixed reality spectrum from virtual reality over augmented virtuality and augmented reality (AR). Here the graphical user interface component is realized

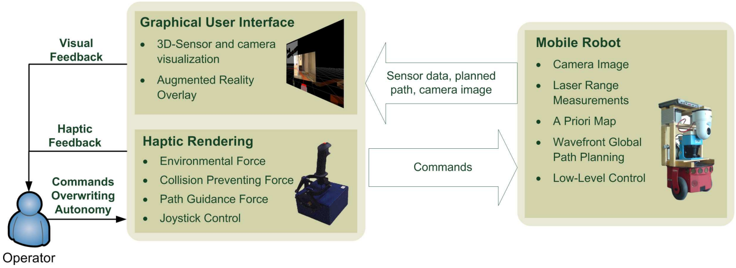


Figure 4.7: Test system setup.

[3] http://www.mobilerobots.com/ (24.11.2009)

[4] http://www.immersion.com (24.11.2009)

as AR interface with slightly modified viewpoint accepting a small AR registration error due to the misalignment of virtual and real camera viewpoint.

4.2.3 Simulation Experiments

For the first set of experiments the mobile robot and the environment were simulated with the Stage [182] simulator. Stage simulates the basic physical properties of the robot and sensors and allows to easily apply the same algorithms as for the later experiments with the real robot in the real environment. The simulated hardware was the same as the later used real hardware. For the simulation experiments no camera image from the robot was provided to the participants. Instead a small 2D Map with the robot's position and the planned path (cf. Figure 4.6) were superimposed on the 3D user interface for the operator.

Task

The task of the test participants was to navigate a robot with different assistance function configurations through a narrow maze, following a given path calculated by the global path planning module of the system. The robot's speed was limited to $1.0\frac{m}{s}$. Each participant had to operate the robot in five user interface configurations for two type of environments, such that each user had to perform ten test runs in total. The user interface configurations are the following:

- no assistive forces (NF)
- two forces (EF, CF) and 3D path
- two forces (EF, CF) and compass rose
- three forces (EF, CF, PGF)
- three forces (EF, CF, PGF) and 3D path

The two environments had the same base map. For one configuration all obstacles were mapped and for the other configuration there were some unknown obstacles, which the operator could only percept through the laser visualization or the respective forces rendered on the joystick when he/she was next to it. The second environment configuration represents more a real world scenario as there is in general only part of the map known and dynamic obstacles occur. Here the a priori unknown obstacles were placed statically at selected places. In order to avoid that the test participants get familiar with the map, different rotations and reflections of the base map were used, such that the path keeps the same without enabling the human to memorize the map.

Test Participants

Ten participants performed the experiments as described before. They were recruited on a voluntary basis from students and post graduate students. Nine participants were male and one was female. They had different study background but most of them studied computer science.

Results and Discussion

The tests with the simulated robot and simulated environment were performed to have first results in order to optimize system parameters for the real-world tests and to improve the test scenarios and procedures. The results provided a very good insight in the teleoperation system with the different interface combinations. Both test modes (fully known environment and environment with unknown obstacles) show comparable tendencies for the different configurations. For the scenarios with unknown obstacles the effect of the different user interface configurations is much higher on the performance. Here the results validate the superiority of the proposed concept for real scenarios with unknown obstacles.

The user interface configuration with the compass rose is not discussed in detail as the observation during the experiments showed, that this interface component is not suitable for this experimental setup. The test maze was very narrow, such that sharp turns at the different planned waypoints were required. Consequently there are high changes in the angle for the guidance compass near the position of the waypoints and the test participants had no chance to prepare for this change. To efficiently use this component in a narrow environment a smoothly planned path is required. Nevertheless, the performance results are also given for completeness in the figures.

Figure 4.8 shows the performance results for both test modes with and without unknown obstacles. In the following the results of the scenario with unknown obstacles are discussed as this are the relevant results for the further implementation of the real-world experiments. Table 4.1 provides the corresponding averages and standard deviations of the performance results.

All test participants completed the task successfully. Looking at the driven paths by the test participants and the deviation from the optimal path (Figures 4.8e and 4.8f), it can be observed that with the configuration with assistance components the participants keep much closer to the optimal paths. A nonparametric one-way analysis of variance (ANOVA) with the Kruskal-Wallis test of the deviation from the optimal path indicates that the configurations EF/CF+3D-Path ($p = 0.0041$), EF/CF/PGF ($p = 0.0343$), and EF/CF/PGF+3D-Path ($p = 0.0052$) are significantly different (level of significance: $p < 0.05$) from the configuration with no force. All test participants could immediately follow the optimal path very closely with the configuration EF/CF/PGF+3D-Path. This presents the robustness of the force guidance system and how intuitive the interface is. In this configuration there is only an average deviation of 7.40 percent (std: 6.70) from the optimal path compared to 76.38 percent (std: 75.95) with no assistance systems.

It could be observed that in test runs with the assistance forces the operator immediately had a better understanding of the environment, the surrounding of the robot, the robot's behavior and dynamics. Without force feedback components there was a continuous switching of attention focus between 2D-Minimap and the 3D-visualizations. In addition, there was often an unintentional heading towards obstacles what leads to large course corrections without force feedback. This can also be seen in the boxplot for back navigation (cf. Figure 4.8d). Here the Kruskal-Wallis test indicates that the force assistance configurations with 3D-path are significantly better than the configuration without forces (EF/CF+3D-Path: $p = 0.0029$; EF/CF/PGF+3d-Path: $p = 0.0011$). The configuration EF/CF/PGF with $p = 0.0559$ closely misses the 5% level of signifcance.

For the number of collisions (Figure 4.8c) the same trend can be observed. The Kruskal-Wallis test indicates that the force assistance configurations with 3D-path are significantly better than the configuration without forces (EF/CF+3D-Path: $p = 0.0003$; EF/CF/PGF+3d-Path: $p = 0.0037$). The configuration EF/CF/PGF with $p = 0.0527$ closely misses the 5% level of signifcance. The configuration with two forces and the 3D-path is in average a bit better than the three forces with 3D-path. This is mainly due to the fact, that the global path planning ignores unknown obstacles such that the path guidance force and the 3D-path misguides the operator to some extend.

One of the most interesting performance parameters is the time for task completion. For the navigation time (cf. Figure 4.8a) the configurations two forces with 3D-path and three forces with 3D-path showed the best times for task completion. For a level of significance of 5% the Kruskal-Wallis test indicates that there is no significant difference between these two test modes ($p = 0.6672$) but to the other test modes no force (EF/CF+3D-Path: $p = 0.0019$; EF/CF/PGF+3d-Path: $p = 0.0019$) and three forces (EF/CF+3D-Path: $p = 0.0172$; EF/CF/PGF+3d-Path: $p = 0.0139$). Looking at the average values for the navigation time a performance increase of more than 50 percent has been achieved comparing the configuration with no force (avg: $170.7s$; std: 60.82) and the three force and 3D-path configuration (avg: $85.1s$; std: 23.83). The results for the average speed (Figure 4.8b) is a combined value from the navigation time and the driven path. These results are presented here to have a comparison of the achieved average speeds compared to the maximum possible speed (for simulation experiments: $1.0\frac{m}{s}$).

In all configurations the 2D-minimap at the top left of the screen and the included 2D-path as user interface elements were available. Nevertheless, the 3D-path turns out to be a very important element with respect to all recorded performance parameters. The observations during the experiments supports that this element facilitates the understanding of the generated and hereby the common ground between system and operator. The 2D elements in contrast are not sufficiently supporting this as they require more mental rotations between the different frames of reference and also switching of the attention focus of the operator. By focusing on the 3D-path and hereby on the integrated display the attention focus maintains in the same frame of reference like the robot, the input device and the rendered forces.

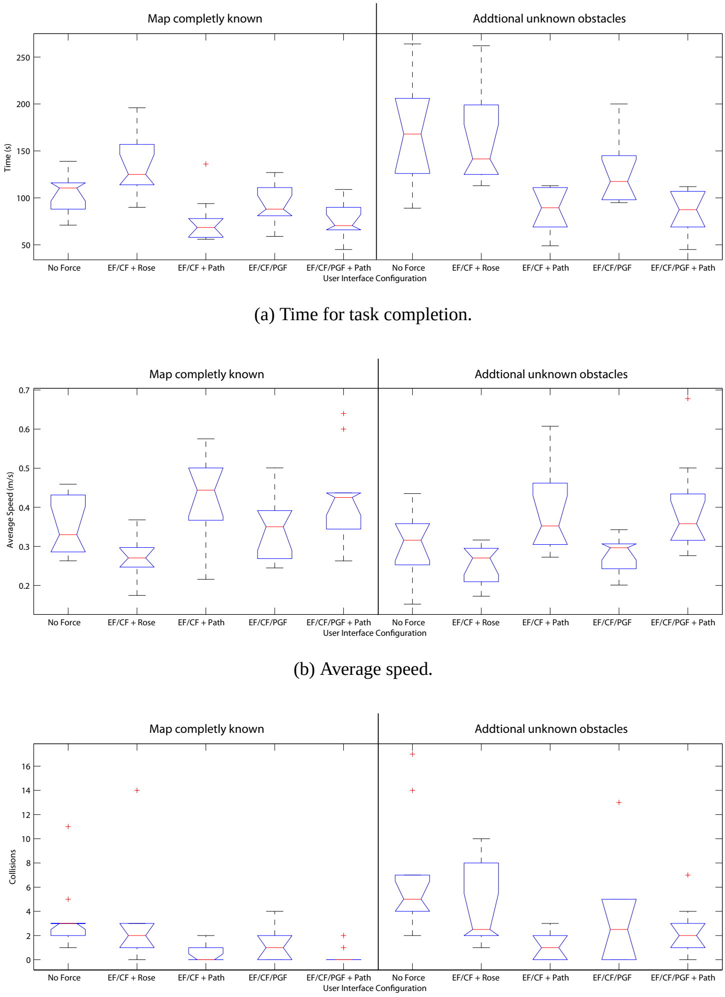
In some cases operators were even unchallenged with three forces such that they took less care and had a higher risk of collisions. A reason why the configuration with the three forces and the 3D path is better but not significantly better than the configuration with the two forces and the 3D-path is that with the EF and CF force only unknown obstacles could be better felt by the operator. Typically participants with the three force configuration decreased the speed in critical areas. Here a force feedback device able to render a higher range of forces might change the results in a positive direction for the three forces approach. Some people quickly learned how to use the three forces and its advantages in an efficient way and even amplified the speed of the system in the suggested direction. Another interesting observation during the experiments was that many of the test participants were talking much more during the test runs with three forces and 3D-path configuration. This is a good indicator for a lower workload.

Table 4.1: Average and standard deviation (std) of performance results (best result bold) for simulation tests.

	No Force	EF/CF 3D-Path	EF/CF/PGF	EF/CF/PGF 3D-Path
Time task completion [s]	170.7 (std: 60.82)	87.4 (std: 23.74)	128.1 (std: 35.05)	**85.1** (std: 23.83)
Average speed [m/s]	0.3099 (std: 0.081)	0.3850 (std: 0.1084)	0.2755 (std: 0.045)	**0.3962** (std: 0.123)
Collisions [count]	6.8 (std: 4.80)	**1.2** (std: 1.23)	3.5 (std: 3.92)	2.2 (std: 2.10)
Backward nav. [count]	5.2 (std: 3.26)	2.0 (std: 0.67)	2.9 (std: 1.29)	**1.7** (std: 0.67)
Driven path [m]	51.44 (std: 22.00)	31.52 (std: 3.25)	34.52 (std: 6.53)	**31.24** (std: 1.92)
Dev. optimal path [perc.]	76.38 (std: 75.95)	8.32 (std: 11.01)	18.58 (std: 21.81)	**7.40** (std: 6.7)

The results of the self-assessment of the test participants during the test runs are shown in 4.9. For all the asked questions the configuration with all three forces is rated best. A very interesting result is that also for the question about the feeling of control of the operator the three force configuration was rated best (NF avg: 3.6; EF/CF avg: 3.8; EF/CF/PGF avg: 4.2;), although in this configuration the actual level of control of the robot is the lowest.

In the rating of the user interface elements (cf. Figure 4.10) all elements besides the compass rose got very good rating. Although there was directly no statistically significant positive effect of the PGF measurable compared to the other force elements in the performance results, all participants judged the implemented PGF as very helpful. In the oral comments from the participants especially the combination with the 3D-path was very much appreciated. The 3D-path as user interface element was rated highest.

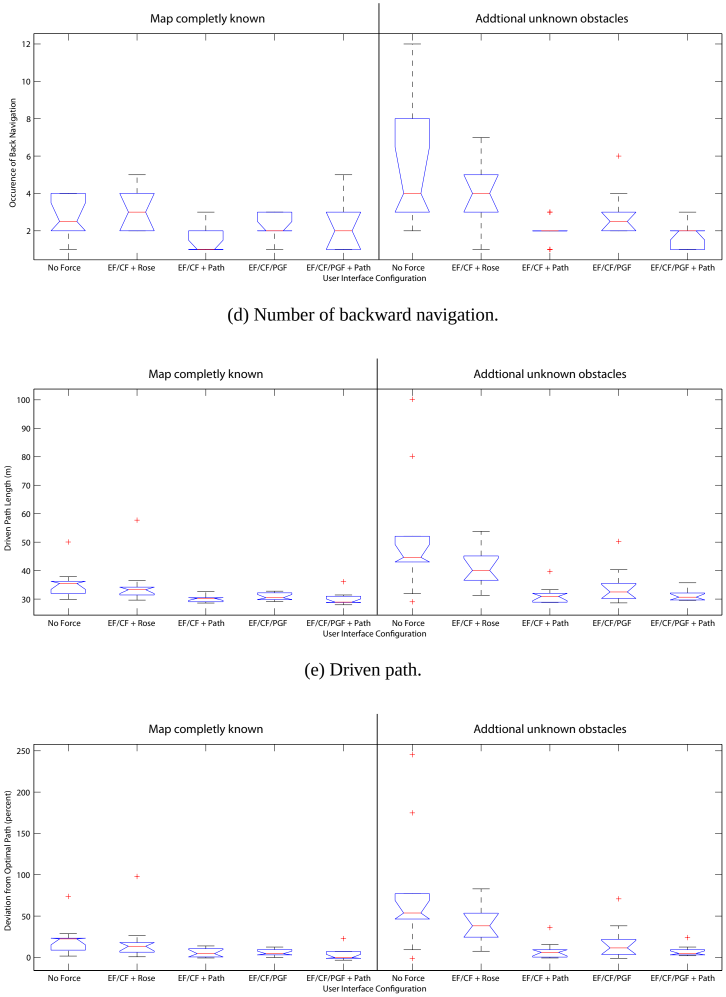


(a) Time for task completion.

(b) Average speed.

(c) Number of collisions.

Figure 4.8: Performance results simulation tests.

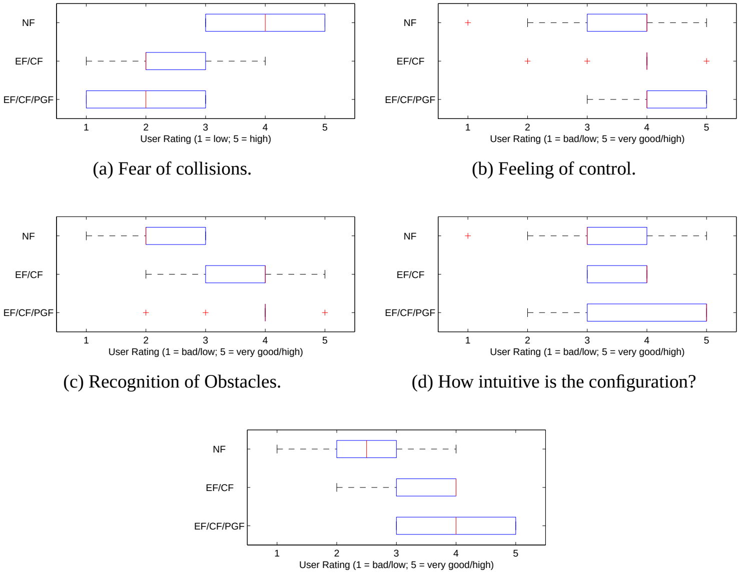


(d) Number of backward navigation.

(e) Driven path.

(f) Deviation from optimal path.

Figure 4.8: Performance results simulation tests (continued).



(a) Fear of collisions.

(b) Feeling of control.

(c) Recognition of Obstacles.

(d) How intuitive is the configuration?

(e) How good can the overview be maintained?

Figure 4.9: Self-assessment of situation during test run.

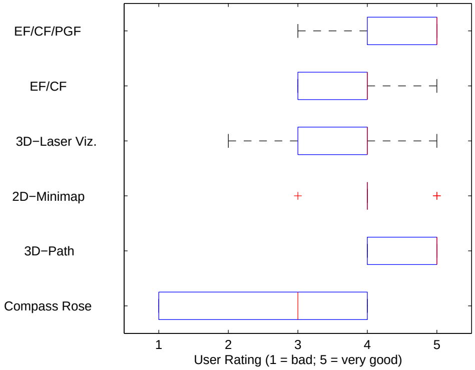


Figure 4.10: User rating of user interface elements simulation tests.

4.2.4 Real-world Experiments

For the real-world user studies the setup like described in Section 4.2.2 was used. Compared to the simulation tests in addition a camera image was projected correctly registered in the 3D user interface. The camera image delivers a very rich sensory feedback from the remote environment and hereby also supports to maintain situational awareness for the operator.

Task

The task for the real world test was again to navigate from a given start position to a given goal position. For the tests a partially known map of the environment was available and the test environment had a certain amount of unknown obstacles. The test scenario was built up in a way that the participants had to navigate through a very narrow maze section and a section where they had more space for navigation. Figure 4.11 (a) shows this map which as recorded a priori to the test and Figure 4.11 (b) shows a photo of the maze area to get an impression about the environment. For safety reasons the robot's speed limit was decreased to $0.25\frac{m}{s}$ compared to $1.0\frac{m}{s}$ in the simulation setup.

Each participant had to perform the test run in four user interface configurations. The user interface configurations the real world tests were the following:

- no assistive forces (NF)
- two forces (EF, CF) and 3D path
- three forces (EF, CF, PGF)
- three forces (EF, CF, PGF) and 3D path

The test configuration with the two forces (EF, CF) and the compass rose was left out in the real world tests as the observations in the simulation tests showed that in such narrow scenarios and with the quick, sharp turns at the given waypoints, this configuration has no real advantages. The concept with the overlayed compass rose would be advantageous in scenarios where a smooth path is given.

Participants

Ten participants performed the real-world experiments as described before. They were recruited on a voluntary basis from students and post graduate students.

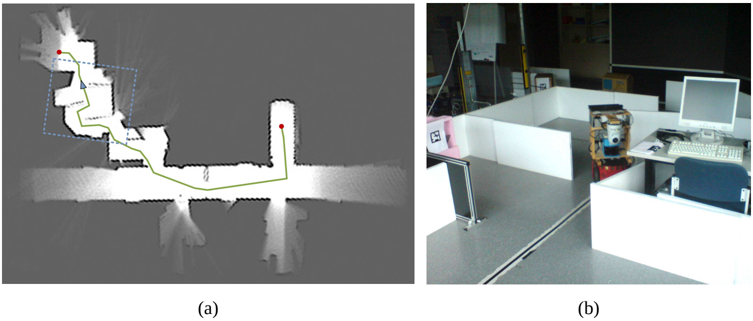


Figure 4.11: (a) Recorded map of the test environment. The red circles mark the start and goal positions, the green line a planned path by the global path planning module, the the dotted blue line the area shown on the photograph next to the map and the blue arrow the current position of the mobile robot. (b) Photograph of the maze area of the test environment (cf. [188]).

Nine participants were male and one was female. Some of them also performed the simulation experiments. They had different study background but most of them studied computer science like in the simulation tests.

Results and Discussion

For the real-world tests first results and experience from the simulation experiments were used to modify and optimize system parameters and the test procedures. Due to this fact and to the different environment the absolute values of the results of the simulation tests are not directly comparable with the results from the real-world experiments. Another aspect is that the results of these experiments have a much larger significance than simulation tests, because participants were much more aware that they are now operating a real robot in a real environment. This was an important observation during the experiments that people thought much more about the consequences of their commands compared to the simulation tests. Nevertheless, many of the gained results in the simulation experiments are directly reflected in the results of real-world experiments. The lowered maximum speed to one quarter of the maximum speed of the simulation test for safety reasons, also lead to the effect that the differences between the different configurations were scaled down. In addition, the camera image as another very rich information source from the remote environment also scales down the differences between the configurations.

In the real-world tests again all test participants completed the task successfully. For the deviation from the optimal path (Figure 4.12f) the Kruskal-Wallis test indicates that the configuration EF/CF/PGF+3D-Path is significantly better than no force ($p = 0.0191$) and EF/CF+3D-Path ($p = 0.0041$) for a level of significance of 5%. For the configuration EF/CF/PGF it closely misses the 5% level ($p = 0.0587$). In the real-world tests the PGF showed its advantagenous with respect to the deviation from the opitmal path. Some people were even better than planned (optimal) path. They smoothed the reorientation curves at the waypoints.

For the occurrence of backward navigation (Figure 4.12d) the Kruskal-Wallis Test indicates significant difference between no force and EF/CF/PGF+3D-Path ($p = 0.0373$), EF/CF+3D-Path and EF/CF/PGF ($p = 0.0498$), and EF/CF/PGF and EF/CF/PGF+3D-Path ($p = 0.0496$). The comparison between no force and EF/CF+3D-Path ($p = 0.0762$) closely misses the level of significance of 5%. The configuration EF/CF/PGF+3D-Path shows again its advantagenous with respect to this performance parameter.

The number of collisions (Figure 4.12c) is significantly reduced with the assistance systems. The Kruskal-Wallis test indicates that all force assistance configurations are significantly better than the configuration without forces (EF/CF+3D-Path: $p = 0.0014$; EF/CF/PGF $p = 0.01923$; EF/CF/PGF+3d-Path: $p = 0.0001$). An interesting aspect here is that also the absolute values of collisions is really low. Especially for the configuration EF/CF/PGF+3d-Path in more than 50 % of the test runs no collision coccured and the maximum number of collsions was one. One reason is that the test particpants in this configuration cut the curves less than for instance in the configuration EF/CF+3D-Path.

For the navigation time (Figure 4.12a) the Kruskal-Wallis test indicates that the EF/CF+3D-Path ($p = 0.02333$) and the EF/CF/PGF+3D-Path ($p = 0.0004$) are significantly better than the configuration with no forces. The configuration EF/CF/PGF+3D-Path was even significantly better than the EF/CF/PGF configuration ($p = 0.0041$). In addition, the results for the EF/CF/PGF+3D-Path configuration have a very narrow distribution. The average time for the

EF/CF/PGF+3D-Path configuration is $129.0s$ (std: 9.30) compared to $162.9s$ (std: 25.13) for the no force configuration. These differences would have been most probably even larger if the speed had not been limited to the $0.25\frac{m}{s}$ for these test. From the results for the average speed (Figure 4.12b) it can be seen that the driven speeds are very close to the maximum speed. For the EF/CF/PGF+3D-Path configuration an average speed of $0.1965\frac{m}{s}$ was reached what corresponds to 79% of the maximum speed. So in large parts of the course the test particpants drove maximum speed in these configuration.

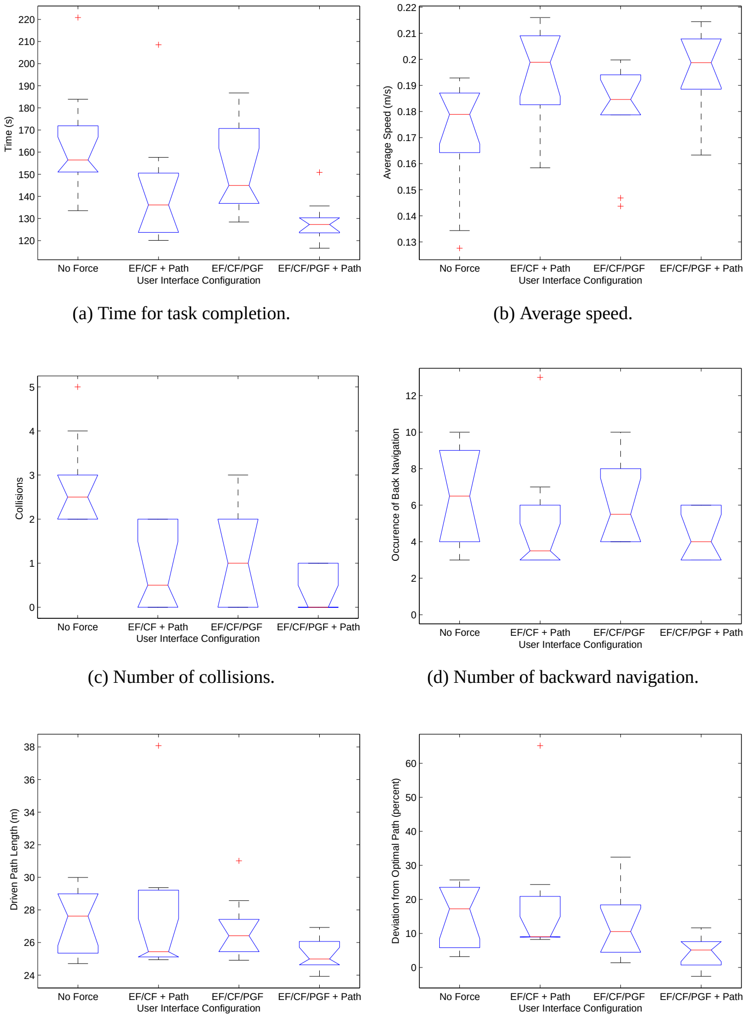
For all recorded performance parameters the configuration EF/CF/PGF+3D-Path compared to the no force configuration showed very significant improvements supported by the Kruskal-Wallis Test (time: $p = 0.0004$; average speed: $p = 0.0052$; collisions: $p = 0.0001$; backward movements: $p = 0.0373$; driven path: $p = 0.0155$; deviation from optimal path: $p = 0.0191$;) with a level of significance of 5%. This validates the approach presented by this concept. The configuration EF/CF+3D-Path is always very close to the EF/CF/PGF+3D-Path what shows the importance of this visual element by means of the performance parameters. Table 4.2 shows the averages and standard deviations of the different performance results.

The results of the self-assessment of the test participants during the test runs are shown in 4.13. For all the asked questions the two forces configuration and the three forces configuration got similar marks. Compared to the simulation experiments they lie closer together. This is mainly due to the fact that in many cases the test participants assumed that they drove worse with the three forces configuration. Although the performance values later showed the they actually drove better than with the two force configuration. Some participants also stated that these did not really prefer this PGF. But also these participants in most cases performed better than without.

In the user rating of the different elements of the user interface (Figure 4.14) again all elements got very good marks. The two force configuration and the three force configuration moved closer together like in the self assessment. The configuration with the PGF was rated a bit worse than in the simulation test. This was mainly due to the fact that the PGF was sometimes a bit irritating next to obstacles. The laser visualization was rated a bit better than in simulation.

Table 4.2: Average and standard deviation (std) of performance results (best result bold) for real-world tests.

	No Force	EF/CF 3D-Path	EF/CF/PGF	EF/CF/PGF 3D-Path
Time task completion [s]	162.9	143.0	150.9	**129.0**
	(std: 25.13)	(std: 26.50)	(std: 19.27)	(std: 9.30)
Average speed [m/s]	0.1709	0.1940	0.1799	**0.1965**
	(std: 0.023)	(std: 0.019)	(std: 0.020)	(std: 0.015)
Collisions [count]	2.8	0.9	1.2	**0.4**
	(std: 1.03)	(std: 0.99)	(std: 1.32)	(std: 0.52)
Backward nav. [count]	6.4	4.9	6.2	**4.3**
	(std: 2.41)	(std: 3.18)	(std: 2.15)	(std: 1.41)
Driven path [m]	27.38	27.48	26.86	**25.23**
	(std: 1.92)	(std: 4.10)	(std: 1.84)	(std: 0.93)
Dev. optimal path [perc.]	15.10	17.88	12.08	**4.87**
	(std: 8.60)	(std: 17.61)	(std: 9.61)	(std: 4.48)



(a) Time for task completion.

(b) Average speed.

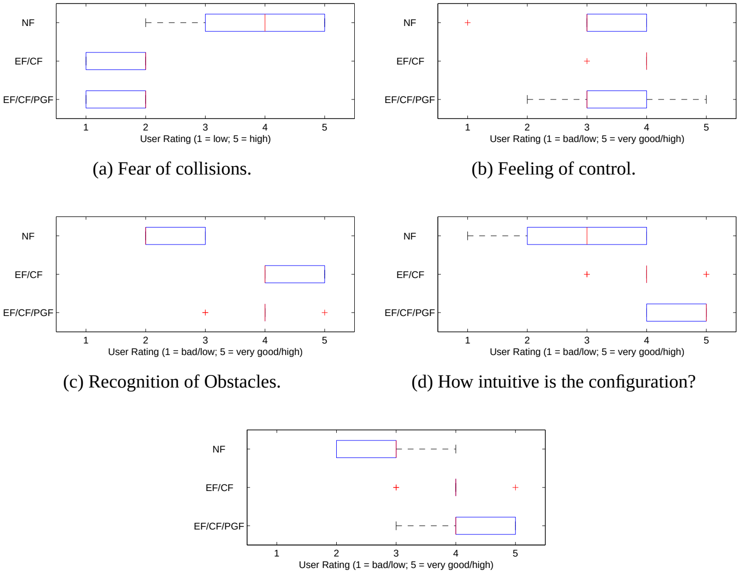
(c) Number of collisions.

(d) Number of backward navigation.

(e) Driven path.

(f) Deviation from optimal path.

Figure 4.12: Performance results real-world tests.



(a) Fear of collisions.

(b) Feeling of control.

(c) Recognition of Obstacles.

(d) How intuitive is the configuration?

(e) How good can the overview be maintained?

Figure 4.13: Self-assessment of situation during test run.

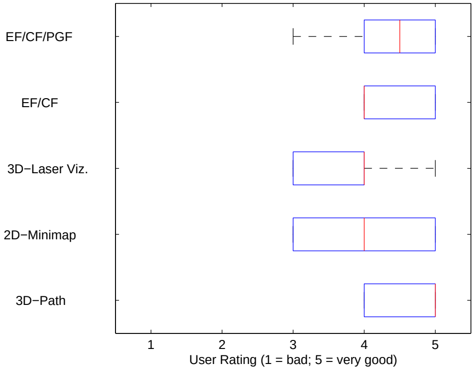


Figure 4.14: User rating of user interface elements real-world tests.

4.2.5 Experiments on Network Delay Tolerance

In order to get an impression about the human tolerance for system delays and respectively network delays as major component of the overall delays experiments with artificial network delays were performed. For this test the user interface configuration with all three assistance forces (EF, CF, PGF) and the 3D-path was selected.

Task

The task and environmental setup for this test was the same as for the real-world experiments described before. One very trained user performed the task with the real robot six times for each of the constant delays of $0ms$, $200ms$ and $500ms$ for the network traffic. For this test one test participant was sufficient as not the user interface and its components should be evaluated but the effect of the network delay. A real network delay between 1 and $2ms$ in addition to the artificial constant network delay was present with the described system setup. The robot's speed was also limited to $0.25\frac{m}{s}$ for these test runs.

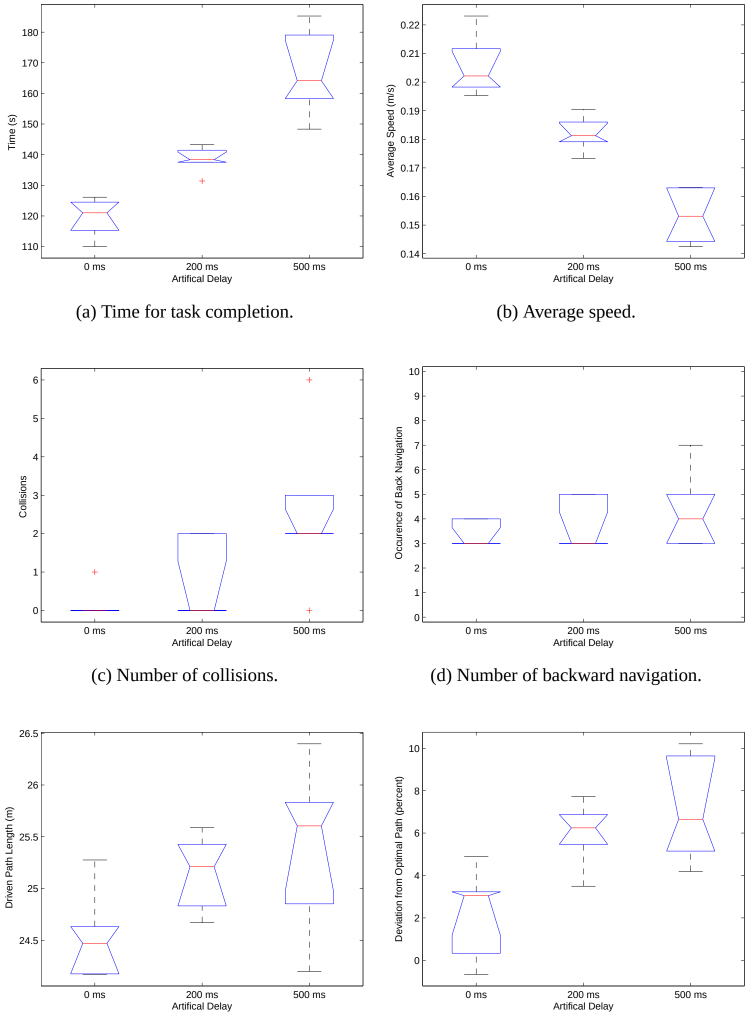
Results and Discussion

As expected by design, the overall system was stable and no negative effects of the delay on the different control loops of the system could be identified.

For each test mode with the different delays the driven trajectories look very precise. With increasing delay, the optimal path is often oversteered at the turns respectively the waypoints. Looking at the maximum speed $0.25\frac{m}{s}$ a delay of $500ms$ leads to position error in the user interface of max. $12.5cm$. This leads to the fact that with increasing delay in all narrow or otherwise complex areas, e.g. where many orientation changes are necessary the speed is significantly lowered by the operator. Straight path segments were driven without significant changes in the speed. The operator used a very typical strategy when there is delay in the feedback from the remote side. He/she steers, stops, waits for the expected system feedback such that he/she can trust the presented data again, and then gives new commands based on the stabilized information from the wait phase. The user also had the impression that he can control the robot less precise with increasing network delay and the overall feeling of safety and control is worse with the increasing network delay.

Figure 4.15 shows the task performance results of the tests with the different artificial delays. The closely correlated metrics time for task completion and average speeds show a statistically significant difference which are obvious from the notched boxplots in Figure 4.15a and 4.15b). For the other metrics a tendency of the effect of increasing network delay can be seen from the results.

This test was mainly done do check the robustness and transparency for the human operator of the concept with respect to network delays. The robustness of the system could be validated. The system proved to be robust against constant network delays. Only the typical effects of network delays on teleoperation systems like e.g. the oversteering could be seen in the results. Even these effects should be less critical when applying the proposed concept. The guidance forces and the 3D-path with its look ahead components allow for a better situational awareness on all three levels - perception, comprehension, and projection to the futures status. The very close results for the number of collision (Figure 4.15c) and the occurrence of back navigation (Figure 4.15d) for the different network delays are a good indicator for this. The increase of collision between $200ms$ and $500ms$ can be easily explained as the systems safety distance for all delays was constantly $20cm$ and the risk of a collision rose for an position error of $12.5cm$ with a speed of $0.25\frac{m}{s}$ significantly. For longer delays this safety value needs to be adapted.



(a) Time for task completion.

(b) Average speed.

(c) Number of collisions.

(d) Number of backward navigation.

(e) Driven path.

(f) Deviation from optimal path.

Figure 4.15: Performance results network delay tests.

4.3 Conclusion

Compared to other haptic systems like shown in [76] or [49], which aim for high transparency and stability of a closely coupled master-slave system in order to maximize telepresence, the concept which is introduced here pursues a different goal. Here the concept is designed task and performance oriented for the navigation task. The major design goal was not to maximize telepresence, but to optimize the overall system performance for the given task by application of results and models from human factor and cognition theory.

Based on this, a sliding autonomy concept with force feedback components and an integrated mixed reality graphical user interface is designed. The force component of the user interface renders the robots intention of further movements to the operator and allows for a dynamic override of this plans. This implements a real sliding autonomy and facilitates to maintain common ground between human and robot. The mixed reality graphical user interface integrates the received information from the remote environment in one single context and allows for an easy verification of the robots intention with respect to sensor data by the human operator. This very much supports to maintain situation awareness and the mental rotations are minimized.

The concept is evaluated with a user study. As expected the best average performance was reached with the configuration environmental force, collision preventing force, path guidance force and 3D-path. In average approximately 20 % performance increase could be reached. Sometimes even values of 100% performance increase could be achieved. The user tests confirm that the forces are not confusing and the participants attest to the forces to be very helpful.

The quantitative performance results already give a a good indicator for a better situational awareness. Especially the significant decrease in the number of collisions and backward navigations and the decreased time for task completion support this. The need for the visualization of the planned path, might be an indicator that this visual element is part of the process to maintain common ground. The principle in this work that the human controls the level of autonomy by keeping to the suggestion of the system (not changing given joystick position means full autonomy), by adjusting the joysticks position (semi-autonomy) or by completely overwriting the suggestions of the system (direct teleoperation) very much supports the hypothesis of a decreased workload, when using the system. The observations of the test participants during the performed experiments validates this. People started to talk more and some even got a bit unconcentrated. Especially participants who have already operated robots before very much appreciated the guidance force and quickly learned how to use it in an optimal way.

An interesting finding in the tests is, that the 3D-path as visual component of the user interface has a major positive effect on the performance. The observations and behavior of the test participants indicate that the operators used this as a reference to validate the forces rendered on the joystick and the robot's movements respectively. The 3D-path enables a subconscious feeling of acknowledgment of the operator's own inputs to the joystick. In addition, the positive effect of the 3D-path maybe also be achieved a bit due to the fact, that the used joystick has very limited forces (max. approx. 9 Newton). A haptic device which can render higher forces might reduce the effect of the visual feedback a bit. The experiments show that at least for the sharper turns (e.g. path guidance force switches to next waypoint) additional feedback addressing another sense (here: visual) is a must. Here also audio feedback might be an option for further testing.

The sliding autonomy concept introduced in this chapter, enables to significantly increase the teleoperation performance. The guidance force components (environmental, collision preventing and path guidance force) proved their strength in the presented user tests. The sliding autonomy concept could be validated to be implemented flawlessly. The quantitative test results and observations during the tests also provide valuable indications, that situational awareness and common ground is improved and the workload is significantly reduced. It is also shown that still the optical elements like the virtual path and the camera image itself play a key role for the humans in many aspects. In future work the introduced concept can easily be extended with minor changes to three dimensional sensor data e.g. from a PMD Camera, and to 6-DOF force feedback devices like e.g. the Phantom[5]. This allows to apply the concept also to scenarios where a real three-dimensional operation of the robot at a remote environment like e.g. an insertion task with a remote manipulator arm.

[5] http://www.sensable.com/ (24.11.2009)

5 Graphical Mixed Reality User Interface

The graphical user interface for the robot's operator has significant impact on the overall performance and successful completion of a task or a mission. Missions with complex environments and conditions like the exemplary application search and rescue often require that mobile robots are navigated either with direct or assisted teleoperation by humans. For human-robot teams humans as supervisor need to be enabled to coordinate and command the team in the remote environment. Both interaction role requires well-designed user interfaces, driving assistance and autonomous features like e.g. adapted path planning, path tracking and obstacle avoidance as support functions for the human operator.

A major challenge for the design of these user interfaces is not to overwhelm the human with a large amount of information from the mission environment presented side-by-side in various visualizations (e.g. in worst case raw data) without any direct relations. The operator in this case faces additionally to his/her actual task the difficulty that he/she has to perform many mental transformations and has to integrate all this information in a mental model about the situation at the remote place of the robot. As a result the operator often only focuses on one element of the graphical user interfaces. Often this is the camera image from the robot, because it delivers the richest information about the remote scene in an integrated and for the human natural fashion. Mixed reality, where parts of the real world (e.g. camera images) and virtual objects are merged in a combined user interface, can be beneficial for providing a good multi-sensor interface (cf. [44]) that combines video data, a priori information, sensor data and human-generated input.

The guiding idea for the presented user interface concepts is to realize an integrated user interface with advanced mixed reality technologies. Such integrated user-centered interface concepts offer a high potential in maximizing the overall teleoperation performance compared to state of the art side-by-side two-dimensional user interfaces, because they take into consideration the human as a major design parameter (cf. Chapter 2). A lot of modern display technologies exist to render these graphical user interfaces. This includes standard displays, hand-held devices like mobile phones or PDAs[1], standard and pocket projectors, see-through and closed HMDs[2], stereo displays with and without polarization classes, large-scale stereo projection systems, ... In this work standard displays, stereo displays and stereo projection systems are considered. Nevertheless any of the presented approaches can be easily transfered to any other display technology.

5.1 Enabling Technologies

5.1.1 Three-Dimensional Elements in User Interfaces

Using three-dimensional models for visualization of a remote workspace shows various advantages. First of all, the real workspace is three-dimensional and therefore, the representation of this space in 3D on the operator display is more natural for the human, such that it enables to improve the feeling of telepresence and the task performance. Often it is necessary to walk around in this 3D representation of the remote environment virtually such that the complete situation can be recognized and understood. As often the remote operation not only concerns one machine, but also the environment (e.g. when controlling robots remotely) it might be necessary to also represent changes in or with respect to the environment (e.g. movement of a mobile robot in the workspace). A large amount of data from the remote space does not necessarily improve the performance of an operator, indeed overload of the operator with data can also decrease the performance. Only adequate processing and presentation enable a user to fully understand the information content and make use of relevant data for further actions [47].

3D-models for improved user interfaces are more and more introduced into industrial processes. This includes for instance all variations of the Computer-Aided Design: the design and construction itself, component selection, physical simulation of the complete product, even Computer-Aided Manufacturing with the help of 3D-models. Other examples are e.g. the use of an augmented reality

[1] PDA - Personal Digital Assistant

[2] HMD - Head-Mounted Display

system for maintenance, [189], virtual reality for remote control and maintenance [190] or the application of 3D models for airport control [191]. These systems mainly use one 3D-model on a single system. Even if the interfaces are used on various systems each system holds and modifies its own model.

With today's available hardware and high bandwidth communication techniques, also distributed 3D-models can be realized. The most promising applications for these technologies are collaborative workspaces, where people can work together on one 3D-model and interact with this model and with each other. The Studierstube project [192] is investigating this for scientific visualizations. Multi-screen stereo projections or CAVE$^{\text{TM}}$-like systems (Cave Automatic Virtual Environment [193]) require either special multi-head graphics workstations or distributed 3D-models on workstation clusters. [194] presents the blue-C system for 3D video conferencing, where a real-time distributed scenegraph is realized with Corba and OpenGL.

Some years ago these systems could only be realized with expensive specialized hardware. But even CAVE$^{\text{TM}}$-like environments - as applied in this work, a three-plane stereo projection - can nowadays be achieved by using standard PCs and standard network equipment. This is possible due to the performance increase in commercial off-the-shelf products. It allows for using highly detailed three-dimensional models on a low-cost basis in a wide range of applications and numerous devices, as no special hardware requirements are preconditioned. An example for such systems is CaveUT [195], which is based on standard components and the UT2004 game-engine. Another low-cost CAVE$^{\text{TM}}$is presented in [196]. This development in the computer market offers new possibilities for using multi-plane projection screens for application areas of telematics, e.g. for teleoperation of mobile robots, coordination of human-robot teams, for tele-maintenance of industrial assembly lines, or for the implementation of learning units in tele-experiments.

CAVE$^{\text{TM}}$-like systems have already entered the car design process, architectural planning of shop interiors, buildings, or even complete streets, or the entertainment sector, e.g. virtual museums or city tours. Compared to these application areas, the application scenario which is targeted in this work demands additional properties. The mentioned areas normally apply a rather pre-calculated three-dimensional model, whereas the proposed telematic applications require a dynamic model, which can be changed online by the user or a sensor, e.g. updating the virtual world when teleoperating a mobile robot according to current sensor readings. Moreover, the user has to be enabled to interact with the model, i.e. switching a button in the virtual world or marking an area, which should not be visited. Finally, the three-dimensional model shall be coupled tightly to the real world, which means pre-processed sensor data has to be updated or integrated into the model and user-released interactions with the virtual model have to trigger the actuators. Systems, which handle sensor data or control the actuators, are often already available. Therefore, the ability to integrate the virtual reality based user interface with these existing systems is highly required.

On the other hand, some requirements are not as hard as they might be for other application domains like e.g. construction. Even though detailed models are necessary, very realistic virtual models are often not required for telematic scenarios. In this application area a high level of immersion into the remote scene is normally not planned. Details in the models have to be shown as long as they play a role in the technical process or give the user visual cues in order to understand the situation. Moreover, as the virtual model is often used to augment the environment model e.g. with colored sensor data, warning symbols and so on, realism is anyway not given. The synchronization requirements are also not as strong as for e.g. the video-conferencing system in [194], where all displays and 16 cameras have to be kept synchronous.

In the area of teleoperation of mobile robots, the usage of 3D-models has also proven its potential for enhancing the system performance. For instance in [70] a 3D model is used to teleoperate a single robot and the performance increase is evaluated. In [23] we demonstrate the application of 3D-models for tele-experiments with mobile robots and [21] presents the first approaches in the context of this work to operate a multi-robot team with a mixed reality user interface on a stereo projection screen.

5.1.2 Mixed Reality

When the first virtual reality (VR) systems were developed, the goal was to create an artificial world either based on real physical scenarios or fictional where the user totally feels present and immerse in. Soon, this virtual worlds were merged with real worlds. Milgram and Kishino [197] defined this as mixed reality (MR). They defined reality-virtuality continuum (cf. Figure 5.1) which has two extremes: the real environment, and the virtual environment. Every realized reality in-between can be called by the generic name mixed reality. Mixed reality can be split up into two more specific parts, augmented virtuality (AV) and augmented reality (AR). In augmented virtuality virtual environments are augmented with some real world elements, and in augmented reality the focus is on the real world representation which is augmented by virtual, computer-generated elements.

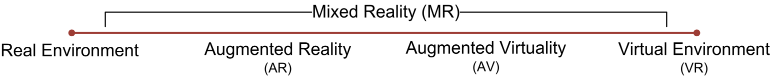


Figure 5.1: Reality-Virtuality continuum as defined by Milgram (cf. [197]).

In this work the proposed concepts can be often used as augmented virtuality as well as augmented reality. Thus, in general the mixed reality term is used here. If a section relates to special needs and application of augmented reality the more specific term augmented reality is used. Although in mixed reality systems all human senses can be addressed, in this work the focus is on visual elements and on haptic elements (cf. Chapter 4).

Azuma et al. [198], [199] give an overview of the major characteristics and challenges of an AR system and the major application areas. AR systems combine real and computer generated objects correctly aligned in real environments and run interactively and in real-time. The "correct alignment" - also called registration of real and computer generated objects in real time and the interactivity require sophisticated technologies to handle problems like the computational efficient overlay of a real world with virtual objects, management of occlusions of virtual and real objects, and tracking of position and orientation inside the real world. These technological demands are in general more crucial for augmented reality than for augmented virtuality.

Augmented and Mixed Reality in Robotics

In the field of robotics augmented reality systems are currently mainly investigated for programming, maintenance, service or training tasks. Milgram et al. [200] show an approach for telerobotic control of a manipulator arm applying augmented reality. Pettersen et al. [201] present an example, how augmented reality concepts can be utilized for a more effective way of industrial robot programming. Compared to the industrial robot applications, the requirements for augmented reality for mobile robot teleoperation differ and are often more challenging. The industrial scenarios are in most cases more static, predefined, and can be calibrated in detail, while the environment for mobile robot missions is often a priori at least partially unknown, changing over time, and the operator has to rely on the accuracy of sensor data. Thus, for mobile robot teleoperation requirements are more similar to requirements of portable mixed reality assistance systems for humans in outdoor environments, e.g. the tracking requirements in order to register virtual and real objects. The Tinmith project [202] realized at the University of South Australia is an advanced example in this area. Nevertheless, there are significant differences between the human portable and the teleoperation system, resulting from the remote control scenario. The operator who is not directly in the working environment is always limited by the capabilities of the sensors on the robot and their representations. These limitations significantly reduce the feeling of presence and the situation awareness of the operator, which are key-elements for successful mobile robot teleoperation.

In [203] an interface called "Virtual Synergy" combines a three dimensional graphical interface with physical robots, agents and human collaborators for the use in urban search and rescue. Milgram et al. [204] present an AR teleoperation interface for unstructured environments. Halme et al. [205] applied an augmented reality interface for outdoor driving tasks of a remote operated vehicle. Lawson et al. [206] show an approach to apply augmented reality for teleoperation of a robot for sewer pipe inspection. In [207] a PDA (Personal Digital Assistant) is used to operate a mobile robot, which compares an image-only, a sensor-only, and image with sensory overlay screen. [69] uses video images with sensory overlay for the control of a mobile robot in the 2005 RoboCup Rescue competition. Reitsema et al. emphasis in [208] the potential of augmented reality for space robotics applications. Green et al. [209] show an augmented reality approach to perform collaborative tasks in peer to peer interaction. Daily et al. [210] developed an augmented reality interface to give humans feedback about the navigational intention of robot swarms. The authors of [211], and [212] propose an augmented virtuality interface for the exploration task, where live video or snapshot images are embedded in a virtual model of the planetary environment. Nielsen et al. [52] suggest the developed augmented virtuality interface for their egolocial paradigm to operate mobile robots. Cooper and Goodrich [213] investigate an augmented virtuality interface for UAVs[3] with respect to enabling an operator to perform the navigation and visual search task simultaneously. Atherton et al. [214] propose an multi-perspective augmented virtuality interface for multiple robots in a Mars scenario. Wang et al. [215] propose and evaluate an interface [216] where the camera image from the robot is gravity referenced in order to increase robot status awareness in uneven terrain.

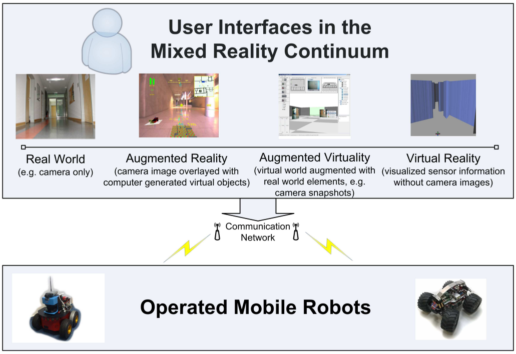

User Interfaces in the
Mixed Reality Continuum
Real World
(e.g. camera only)
Augmented Reality
(camera image overlayed with computer generated virtual objects)
Augmented Virtuality
(virtual world augmented with real world elements, e.g. camera snapshots)
Virtual Reality
(visualized sensor information without camera images)
Communication Network
Operated Mobile Robots

Figure 5.2: Reality-Virtuality continuum in the context of mobile robot teleoperation.

The presented example interfaces can be located in the reality-virtuality continuum as shown in Figure 5.2, which has been adapted to the application of mixed reality to mobile robot teleoperation. All these interfaces show the potential but also the challenges when applying mixed reality in robotics. Obviously, for the various tasks that occur in mobile robot teleoperation different needs appear. In this work a generic concept and framework is developed in order to allow the selection of the location insight the continuum dependent on the demands. The consistent, robust and efficient application of mixed reality technologies enables an increased performance of the teleoperation system. Additionally the resulting mixed reality user interfaces can be easily adapted to the actual tasks and requirements.

[3] UAV - Unmanned Aerial Vehicle

5.2 Advanced Augmented Reality Overlays

Nowadays even mobile phones have GPS for localization, and orientation sensors which enable to some extend augmented reality applications with the embedded cameras. But still most applications with accurate registration and tracking rely on marker tracking, or other computer vision approaches utilizing the video stream from the embedded cameras. In most cases the occlusion problem is not handled at all, because it would require detailed spatial knowledge of the captured scene.

Here, robots offer already more spatial information of the observed scene for the application of augmented reality in user interfaces for the operation of mobile robots. The navigation sensors needed for localization, path planning, and obstacle avoidance for instance can be used to realize advanced features for augmented reality user interfaces. In the following methods are presented how these advantageous preconditions can be used to meet the demands of dynamic registration and tracking, and how to three-dimensionally model the camera images based on distance measurements from laser range finders or time-of-flight cameras, for the specific needs of graphical user interfaces for remote operation of robots. In addition, it is shown how these basic methodologies enable directly the usage on 3D stereo systems and an accurate occlusion handling for the user interfaces.

5.2.1 Registration and Tracking

For the three-dimensional user interfaces in this work a scene graph is used as data structure. Hereby, all objects in this three-dimensional world are arranged in a tree, and each node can have its virtual representation, child nodes, etc.. In addition, it can hold its own homogeneous transformation matrix which represents the relative position and orientation to the parent node. The root node represents the basic world coordinate frame. Thus, the child nodes inherit the transformations of all the parent nodes along the branch to the root node. This type of data structure is very advantageous, for modeling robots, its sensors', and other elements. For instance a robot with a laser range finder and a camera would be represented as follows: the robot node is child node of the root node, and its transformation T_R represents position and orientation of the robot's reference point in the environment; the laser and the camera node are child nodes of the robot nodes and their transformations T_L and T_C correspond to the relative position and orientation of the sensors to the robot's reference point. This enables to apply the different tracking and calibration data available in a robotic system directly to the corresponding transformation (e.g. data from robot's localization system to T_R, extrinsic camera calibration to T_C, etc.). If all information and objects are organized spatially correct in this scenegraph structure, an augmented reality view of the scene can be achieved by setting the viewpoint of the three-dimensional rendering to the virtual representation of the origin of the physical camera, and either setting the projection matrix of the view to the calibrated intrinsic parameters (cf. Section 3.1.5) or modeling the camera image as three-dimensional object (cf. Section 5.2.2) based on the calibrated intrinsic parameters of the physical camera.

The quality of the registration for augmented reality overlays depends on the calibrated relative positions and orientations and tracking of the different objects. With respect to setting up mixed reality and specifically augmented reality user interfaces for mobile robot operation and coordination, two classes of objects which require registration and tracking, can be distinguished:

- Global objects which spatially refer to the global world coordinate frame (e.g. virtual path, way points, 3D-snapshots,...).
- Relative objects which spatially refer to the robots' reference points (e.g. visualized distance measurements, distance references, camera images,...).

These objects have different requirements on the accuracy of tracking. Global virtual elements in the augmented reality user interface do not need very high accuracy as they are only used to mark specific locations in the world, and/or coarse orientation of the operator in the environment. In addition, these objects are in general not very near, and according to the mathematical projection model the registration error in pixel gets smaller the farer the objects are from the camera. Thus, a robot localization which reaches an accuracy of some centimeters is sufficient to achieve the required registration results for augmented reality view from the robot's camera.

The local virtual elements superimposed on the camera image like for instance visualized distance measurements from a laser range finder require a very accurate calibration, as these elements are also used by the human operator for local navigation (e.g. drive around obstacles). As these sensors and the camera have static spatial relations relative to the robot their position and orientation can be calibrated very accurately with standard methods and hereby a very accurate registration for these local virtual elements can be achieved.

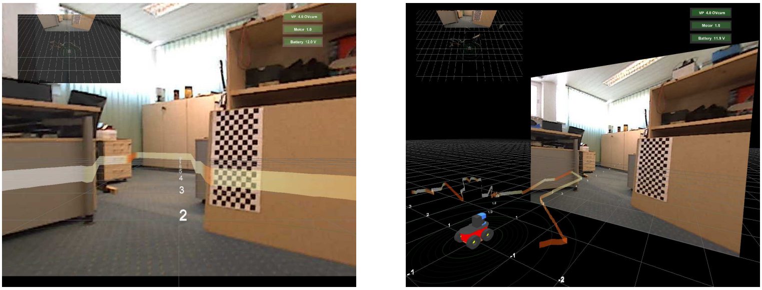
5.2.2 Modeling Camera Images

There exist various ways to implement the overlay of the real world and computer generated objects for augmented reality views. While some systems use e.g. semitransparent displays where the human can see the real world directly and objects are superimposed by this semitransparent display, most augmented reality implementations use an indirect approach, where the real environment is captured by a camera. This camera image is then presented to the human with superimposed virtual objects. As this work focuses on teleoperation, only the indirect approach via a camera is feasible for these scenarios.

Almost all camera based augmented reality system implement the overlay of camera image and virtual objects by simply setting the camera image as background on the display and rendering the virtual objects on top of this. If the projection properties of the virtual camera are set to the intrinsic calibration data of the physical camera and the virtual camera is placed and oriented like the physical camera an augmented reality view results. For some of the user interfaces developed in the context of this work, this standard approach has also been used. A significant disadvantage for remote operation tasks is, that the virtual viewpoint always needs to be attached to the physical viewpoint of the real camera as the image stays fixed in the background. For mobile robot teleoperation user interfaces this is equal to a limitation to the egocentric viewpoint. Literature, experiments and experience showed that especially for the navigation task of mobile robots a single fixed egocentric viewpoint in the user interface is not the best choice if it is avoidable.

Thus, in this work a different approach to setup an augmented reality has been developed. Based on available calibration and other sensor information a three-dimensional geometry can be calculated and the two-dimensional camera image from the real environment is reprojected on this geometry. Thereby, with a thorough application of the calibration data a texturized three-dimensional representation of the camera image results which nevertheless enables the same quality of augmented reality view like the standard approach, if the viewpoint to the virtual representation of the physical camera's position.

The first realization of this would be to use the intrinsic parameters to calculate a simple calibrated rectangular projection plane at a selectable distance. Based on the intrinsic parameters (cf. Section 3.1.5) f_{px}, f_{py}, x_{max}, y_{max}, and the selected distance of the projection plane d_{plane}, the vertices of the quad dor this projection plane can be calculated. The intrinsic parameters c_x, and c_y, together with d_{plane} can be used to shift the vertices according to the properties of the physical camera. The camera image is set as texture to the resulting quad and the whole shape is placed according to the extrinsic calibration of the camera. Figure 5.3 shows the augmented reality result of this kind of implementation and a side-view of the realized three-dimensional model. Here the data from the laser range finder is superimposed as red band.



(a) Realized augmented reality overlay.

(b) Side view of same model.

Figure 5.3: Reprojection of camera image on calibrated projection plane at $d_{plane} = 10m$.

While spatial matching of laser range finder visualization with the camera image is very intuitive and easy in the augmented reality view (Figure 5.3a), in any exocentric view (e.g. Figure 5.3b) this requires a significant effort from the human operator. Thus, the three-dimensional modeling is further extended in the following. Robots which are operated from remote are in general equipped with distance measurement sensors for navigation purposes. These distance information from the remote environment can be also used to construct geometries for projecting the camera images and hereby create a much easier understandable representation of the remote environment inside the user interface. Another advantage of this modeling is, that the user interface can be used instantaneously on a 3D steroscopic display in order to support a better depth perception of the remote environment by the human operator. Here also a stereoscopic effect is achieved with the camera image due to the three-dimensional modeling based on range sensor distance information, although it is actually a monoscopic camera image.

Figure 5.4 shows a schematic overview of the processing. Two processing sequences are realized which run independently. First the distortion for each image from the camera is corrected with the calibrated distortion parameters (cf. Section 5.2.2). The result is then written to the texture buffer. In the second sequence first the retrieved distance sensor data is pre-processed (e.g. distance and distortion correction, filtering). The result together with calibration data for the sensor and the camera are used to build a three-dimensional geometry object representing the distance profile of the remote environment. All coordinates of the vertices of this object are then transformed from the distance sensor coordinate frame to the camera coordinate frame. These coordinates and the intrinsic parameters of the camera are used to calculate new texture coordinates for each of the vertices of the geometry object (cf. Section 5.2.2). The scenegraph attached to the rendering loop holds only the references to the texture buffer and the array of texture coordinates, and hereby also runs independently of the two processing sequences. This is actually the key feature which enables this kind of three-dimensional image modeling running without any significant delays with full resolutions and high frame rates on standard computer hardware. In this work a 2D laser range finder and a 3D PMD camera are used as examples to validate the applicability of this approach.

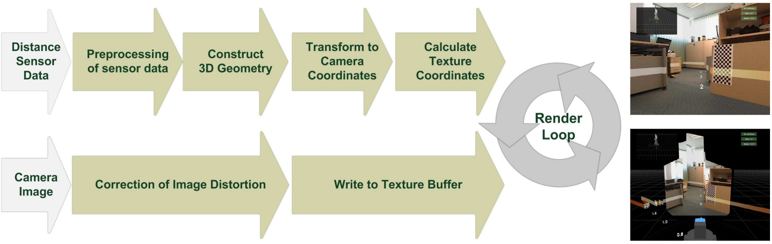
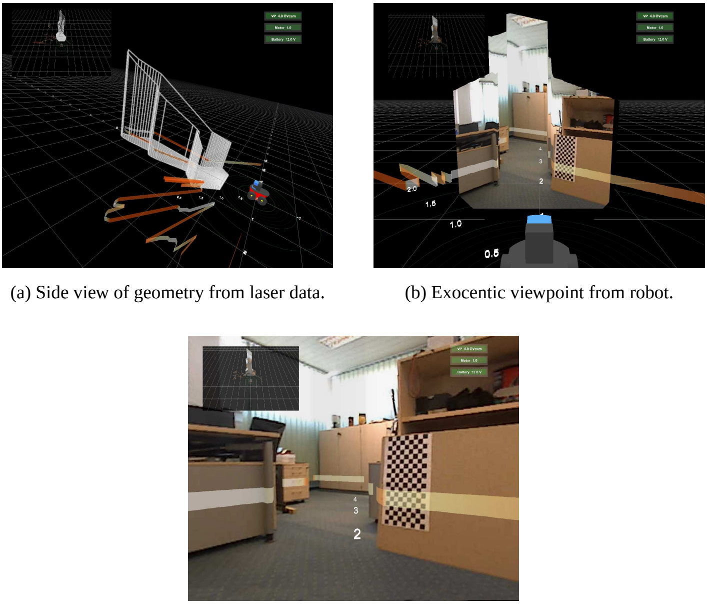

Distance Sensor Data
Preprocessing of sensor data
Construct 3D Geometry
Transform to Camera Coordinates
Calculate Texture Coordinates
Render Loop
Camera Image
Correction of Image Distortion
Write to Texture Buffer

Figure 5.4: Process of three-dimensional modeling of two-dimensional image data based on distance sensor data.

Figure 5.5 shows exemplary results of the implementation of this approach with a two dimensional laser range finder with 180° degree horizontal field of view, a resolution of 1° degree and a camera with a horizontal field of view of $\approx 68^\circ$ degree and a resolution of 640×480 pixels. The geometry (Figure 5.5a) is calculated based on calibrated position and orientation information of the laser range finder and camera, and the calibrated projection parameters of the camera. Additionally, a virtual floor is added to the geometry based on the distance data of these sensors as it better represents three-dimensional typical driving terrain for mobile robots. This is in particular important when the robot is operated in exocentric view (Figure 5.5b). Figure 5.5c shows the resulting augmented reality view when the viewpoint position of the interface is set to the virtual representation of the physical camera position and orientation.



(a) Side view of geometry from laser data.

(b) Exocentic viewpoint from robot.

(c) Augmented reality view.

Figure 5.5: Three-dimensional camera image modeling based on 2D distance data from a Sick laser range finder. The full laser range data is visualized by the red band.

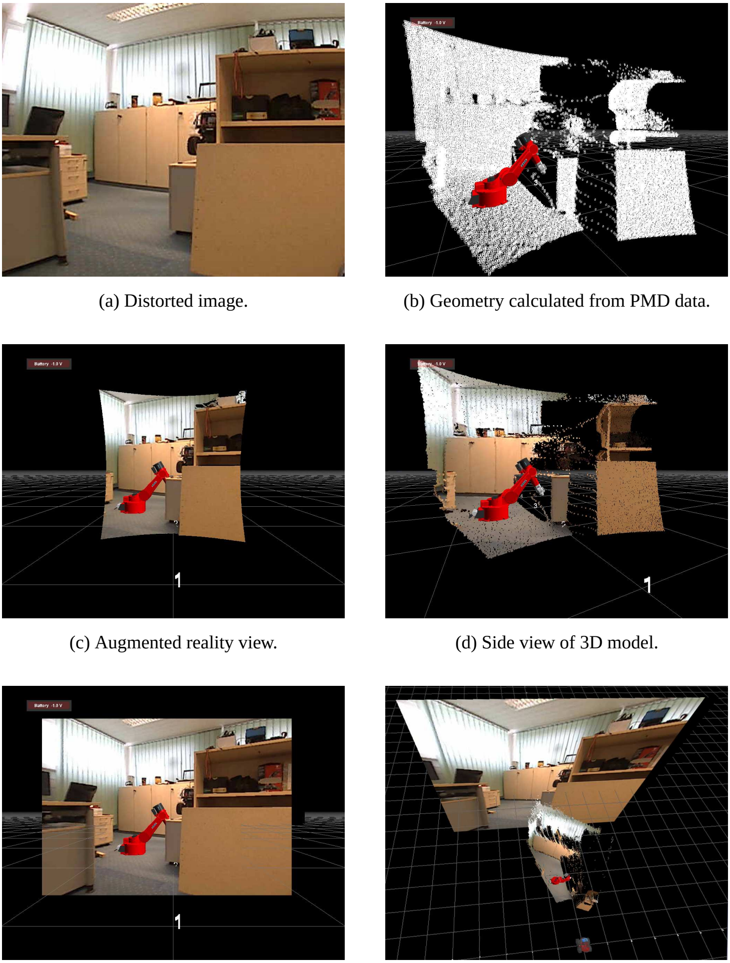
The same approach can also be used for a distance sensor which delivers a matrix of depth values. Figure 5.6 shows the exemplary results for a PMD camera with a resolution of 204×204 pixels, 40° horizontal and vertical field of view, and the same network camera as for the setup with the laser range finder. Here the integration time for the PMD camera is set to $5000\mu s$. Distances are corrected with calibrated projection parameters, distortion is compensated, and no filtering is applied. Based on the received distance information and the calibration data, a geometry object can be calculated in various ways. In Figure 5.6b for each depth value a quad is calculated with the side length dependent on the actual z value of this depth pixel and the field of view of the PMD camera. Thus, it looks like a closed surface from the cameras viewpoint. Figure 5.6d shows the result of putting the texture on this geometry object. For this setup the field of view of the PMD camera (40° degree) is smaller than the field of view of the network camera ($\approx 68^\circ$ degree). As a result image information might get lost in the augmented reality view (Figure 5.6c). Here, an additional billboard with the image in the background covering the entire field of view of the camera can be used to avoid this (Figure 5.6f). Figure 5.6e shows the resulting augmented reality view with billboard. In this figure also the robustness of the camera image modeling approach presented here is obvious, as no transitions between billboard and textured depth image geometry are visible. Instead of the billboard also combinations with 3D modeled camera images from a laser range finder for the missing areas of field of view would be applicable.

The results presented illustrate what is achievable with three-dimensional modeling of the two-dimensional camera image based on distance sensor data. The introduced approach has a lot of advantages for user interfaces for robot teleoperation:

- Better spatial representation for the human operator of the remote environment correlating distance and image information especially for any exocentric viewpoint.

- Support of different viewpoint concept for navigation (egocentric, exocentric) and any other viewpoint while still having the augmented reality view available.

- Support of a any field of view of the virtual camera, e.g. if the field of view of the camera is smaller than that of the laser range finder, the virtual



(a) Distorted image.

(b) Geometry calculated from PMD data.

(c) Augmented reality view.

(d) Side view of 3D model.

(e) Augmented reality view with billboard.

(f) Top down view of 3D model with billboard.

Figure 5.6: Three-dimensional camera image modeling based on depth data from a PMD CamCube.

field of view can be extended such that the augmented reality registration is still correct but nevertheless more distance information from the laser range finder for instance is visible.

- Direct support of stereo display devices. As additional effect, a quasi stereo image is achieved with the distance information without actually having a stereo camera. Hereby less bandwidth on the communication link and less computation power is necessary for the teleoperation system, while having almost the same result as with a real stereo camera.
- Multiple usage of transmitted distance data.
- No loss of image resolution and/or information.
- Solution of augmented reality occlusion problem without further effort (cf. Section 5.2.3).
- Dependent on mobile robot application - the concept enables to quickly realize gravity referenced displays [215] [216].

Calculation of Texture Coordinates

Textures are quadratic images with a side length of $l_{tex} = 2^n$, such that for a camera image as used for the examples with a resolution of 640×480 a texture with side length $l_{tex} = 2^{10} = 1024$ pixel is required. In order to achieve the results as described in the section before, a mathematical model is required which enables to calculate texture coordinates for each of the vertices of a given three-dimensional geometry. Usually texture coordinates are given as two-element vector (s, t) for each vertex of a geometry. s denotes the horizontal position of the texture coordinate normalized to a range from 0 to 1.0 and t the corresponding normalized vertical position. The graphics APIs then use these coordinates to allocate and interpolate the texture pixels to the surface of each primitive triangle of the geometry object. This affine texturing does not take into account any depth value of the vertices. Thus, for geometries with different z-values (cf. example Figure 5.7) affine texturing with (s, t) coordinates is not sufficient.

For such geometry affine texturing would in an undesirable result like e.g. Figure 5.8a. In order to avoid these effects perspective texturing (cf. Figure 5.8b) is required which takes into account the z-values of the vertices. For perspective

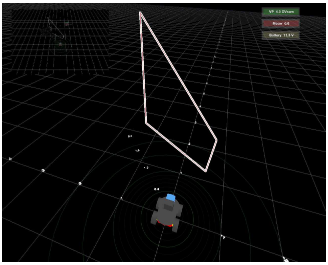
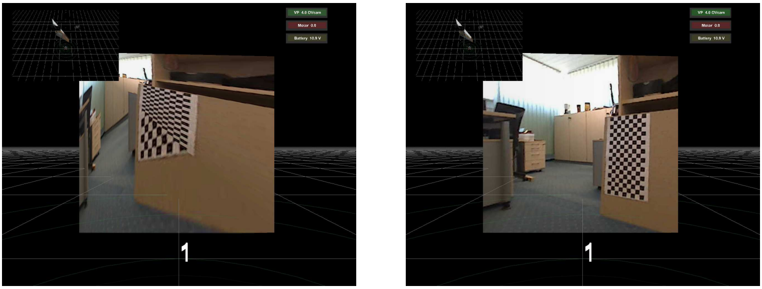


Figure 5.7: Projection quad with plane normal not parallel to camera view axis.



(a) Standard texture coordinates applied to geometry in Figure 5.7.

(b) Projective texture coordinates applied to geometry in Figure 5.7.

Figure 5.8: Application of projective texture coordinates for correct reprojection of the camera image.

texturing four-element texture coordinates (s, t, r, q) have to be used. The additional r is only required for three-dimensional textures, and not required and set to $r = 0$ for the work here as the camera image is two-dimensional. The q parameter is 0 for affine texturing. For perspective texturing it is used to incorporate the depth of the vertex in the texture mapping process. More details on the basics, the implementation, and the meaning of texture coordinates and the texture interpolation can be found in the standard openGL references [217] and details on the basics for projective texture coordinates are given in [218].

Based on the requirements and constraints of the graphic APIs and the physical fact of the sensor systems, the following equations can be derived in order to achieve the results shown in the section before. First the extrinsic calibration of the camera and the distance sensor are used to determine the transformation T_{SC} in order to transfer each of the geometry's vertex i with coordinates $(x_{S_i}, y_{S_i}, z_{S_i}, 1)^T$ from the sensor coordinate frame to the camera coordinate frame (Equation 5.1). These coordinates $(x_{C_i}, y_{C_i}, z_{C_i}, 1)^T$ are then transformed with P_{tex} to texture coordinates based on the projection properties of the camera (cf. intrinsic parameters Section 3.1.5) and the requirements for texture coordinates $(s_i, t_i, r_i, q_i)^T$. Equation 5.2 gives the resulting mathematical transformation which can be applied to any vertex of a geometry in the user interface.

$$\begin{pmatrix} x_{C_i} \\ y_{C_i} \\ z_{C_i} \\ 1 \end{pmatrix} = T_{SC} \cdot \begin{pmatrix} x_{S_i} \\ y_{S_i} \\ z_{S_i} \\ 1 \end{pmatrix} \tag{5.1}$$

$$\begin{aligned} \begin{pmatrix} s_i \\ t_i \\ r_i \\ q_i \end{pmatrix} &= P_{tex} \cdot \begin{pmatrix} x_{C_i} \\ y_{C_i} \\ z_{C_i} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{f_{px}}{l_{tex}} & 0 & \frac{c_x}{l_{tex}} & 0 \\ 0 & \frac{f_{py}}{l_{tex}} & \frac{c_y}{l_{tex}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_{C_i} \\ y_{C_i} \\ z_{C_i} \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{f_{px}}{l_{tex}} \cdot x_{C_i} + \frac{c_x}{l_{tex}} \cdot z_{C_i} \\ \frac{f_{py}}{l_{tex}} \cdot y_{C_i} + \frac{c_y}{l_{tex}} \cdot z_{C_i} \\ 0 \\ z_{C_i} \end{pmatrix} \end{aligned} \tag{5.2}$$

f_{px}, f_{py}, c_x, c_y denote the intrinsic parameters of the camera in pixel and l_{tex} the side length of the entire texture image. As the texture buffer typically is quadratic with a side-length of 2^n due to technical limitations, the camera image in general has smaller dimensions than the texture image. This needs to be considered for texture coordinate calculation. However, if the actual camera image is stored at the texture origin, the transformations given in Equation 5.2 already perform the necessary scaling and translations of the texture coordinates, such that no further processing is required here.

The described texture coordinate calculation enables a generic calculation of texture coordinates of three-dimensional objects for the teleoperation user interface, such that also other objects besides distance geometries can be texturized. These textured elements can be used to support the operator to externalize memory e.g. with textured grid/elevation maps or octree maps like described in Section 3.4.2. For storage purposes an extension with tiling and corresponding rectifications of the texture elements might be needed as extension.

Image distortion correction

As most of the visual cameras introduce a distortion on the received image, a rectification of each image is required in order to achieve good augmented reality overlays. A typical example for an distorted image is shown in Figure 5.6a. Section 3.1.5 explains these distortion effects. Applying the normalization transform to each received image would be very time-consuming and inefficient. For the proposed user interfaces here a look up table is calculated based on the distortion parameters of the calibration results at system start, because distortion parameters in general do not change during runtime. This look up table allows a direct matching between distorted pixel coordinates x_d, y_d and normalized pixel coordinates x_n, y_n. According to the distortion model given in Section 3.1.5 for each normalized pixel x_n, y_n displayed in the user interface the corresponding source pixel indices x_d, y_d from the distorted image can be determined and stored in the look up table. Hereby, a computational very fast normalization is realized. An example result of this approach is shown in Figure 5.6e with the texture image.

5.2.3 Occlusion Handling

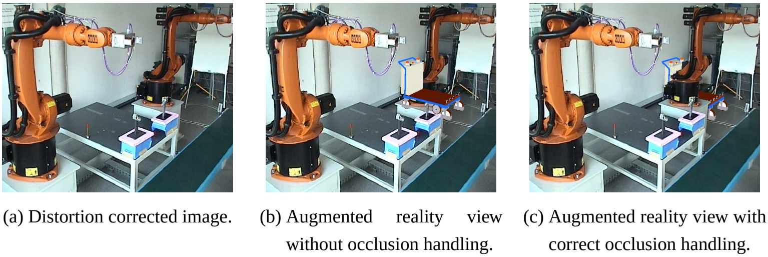
Augmented reality or more general mixed reality user interfaces allow setting up integrated user interfaces, which can reduce the cognitive requirements on a human operator through minimization of mental transformation. Nevertheless, these interfaces are very sensitive to inaccuracies in the overlay. A major problem is the correct occlusion handling between real and virtual objects. If these problems are not adequately addressed by the user interface the human operator might get confused and the advantages of the user interface, might be significantly reduced. An example for incorrect occlusion handling is shown in Figure 5.9b. The transport robot is superimposed on top of the right robotic manipulator although it is actually located behind.

In the context of this work two approaches are developed and investigated for robotic applications which enable an accurate and correct occlusion handling:

- *Masking* - Use knowledge about structures and available distance sensor data to mask the virtual objects superimposed on the camera image.
- *3D image modeling* - Realize inherent occlusion handling through three-dimensional modeling of the camera image.

Occlusion handling by masking. The masking approach has its strength in scenarios where most of the objects are well defined and can be tracked like it is e.g. typically in industrial scenarios. Here, a correct occlusion handling for known and tracked objects can be achieved based on this fact. The whole environment can be modeled as a virtual model and machine data is used to continuously update this virtual model. Figure 5.10a shows the virtual representation of the industrial demo setup in Figure 5.9a. This three-dimensional virtual model of the work environment can be rendered off-screen such that virtual objects which do not appear in the augmented reality interface are used to correctly mask other objects which should be superimposed on the camera image received from the scene. In Figure 5.10b the virtual objects which are modeled and tracked, but should not be superimposed are marked gray for illustration. These objects are rendered in a masking color (here: black) in order to create an accurate occlusion mask of the virtual scenario (cf. Figure 5.10c). Finally, the rendered virtual image of the world is mixed based on this occlusion mask with the image captured by the camera providing the real-world image from the scene. In the example each black pixel of Figure 5.10c is replaced by the corresponding pixel from the camera image, such that a augmented reality view with correct occlusion handling like in Figure 5.9c results. The masking approach in not limited to pre-defined and tracked objects in the work environment. It can easily be extended in order to incorporate additional information about the work environment. If an additional depth sensor like a PMD camera is available, the depth information can be used to create supplementary masking structures. Hereby an occlusion handling can be achieved which also covers dynamic objects in the work environment which can not be modeled and tracked fully and easily (e.g. cables, humans in the work environment) .

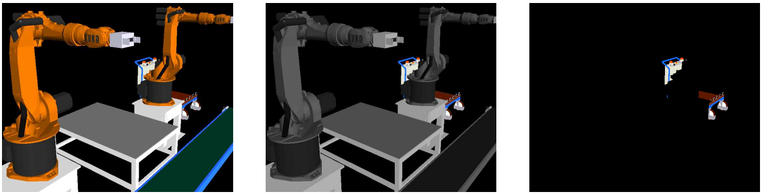
The presented masking approach fits very well in scenarios (e.g. tele-



(a) Distortion corrected image. (b) Augmented reality view without occlusion handling. (c) Augmented reality view with correct occlusion handling.

Figure 5.9: Augmentation of camera image with no occlusion handling and occlusion handling based on masking with known environment.

maintenance and tele-monitoring of industrial plants) where the location of the cameras can be placed at locations providing a good view of the desired workspace and where the camera placement opportunities do not limit the possible viewpoints for the human operator. Although it is limited to applications where the viewpoint is fixed to the virtual representation of the physical camera, it provides very robust and very good results for these scenarios with well defined environments. The application of the rendering capabilities of 3D graphics APIs allows for avoiding the application of additional specific occlusion handling algorithms.

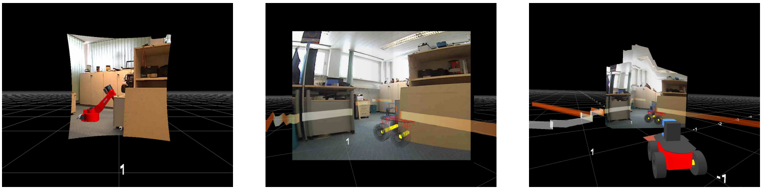


(a) Virtual model of scene. (b) Objects of the model not to superimpose marked gray. (c) Masked virtual model for augmentation.

Figure 5.10: Masking with virtual model for occlusion handling.

Occlusion handling by 3D image modeling. For mobile robot teleoperation the second approach is much more suitable which makes use of the three-dimensional modeling of the camera images based on the distance measurements of the robots' sensors (cf. Section 5.2.2). In such scenarios the work environment is in general very dynamic, unstructured, and most of the objects are unknown at the beginning of the mission. In addition localization in such applications and hereby tracking for the augmented reality registration cannot be easily realized at a comparable level like in industrial scenarios (e.g. the robot arms used for the masking example provide sub-millimeter accuracy for the tool position). Here, the three-dimensional camera modeling based on distance data provides a suitable alternative as it provides implicitly occlusion handling.

A further and even more important advantage of this approach is, that the viewpoint inside the interface can be changed. For teleoperation scenarios for mobile robots the ability to change the viewpoint can significantly improve situation awareness, which is one of the major bottlenecks of remote operation of robots. In addition, the visualization can be directly used together with a stereo display in order to enable real three-dimensional visualization. Figure 5.11a shows an example for correct occlusion handling with a camera image three-dimensionally modeled with depth data from PMD CamCube camera. The wrist of the virtual robot arm correctly disappears behind the shelf and the mobile file cabinet. The occlusion handling also works with the camera image modeled with the laser range finder (Figure 5.11b), although it cannot provide the occlusion handling quality like the PMD camera setup for all scenarios due to its two-dimensional scanning nature. Figure 5.11c shows the same scene as Figure 5.11b from an exocentric viewpoint. It illustrates very well, that also with the changed viewpoint the occlusion handling stays valid and it is visible, that the virtual robot actually stands in the shelf.



(a) Image modeled with PMD data (AR egocentric view).

(b) Image modeled with laser data (AR egocentric view).

(c) Image modeled with laser data (AR exocentric view).

Figure 5.11: Examples for dynamic occlusion handling based on distance sensor data from PMD camera and laser range finder.

5.3 Mixed Reality User Interface Approach

5.3.1 Design and Characteristics

Typically the camera image is the major element of user interfaces for teleoperation. Camera images from the remote scene provide a realistic representation and deliver a lot of spatially integrated information from the remote work environment of robots, but still require further elements for orientation. Darken et al. [219] evaluate the usefulness of video streams for a remote observer for maintaining spatial comprehension and tracking of the remote scenario. They validate the hypothesis that the use for a video stream without any other spatial reference about the camera or movements is very difficult if not impossible This is even more the case if communication limitations are present, and cause drop outs or delays like it can typically occur in robot application scenarios. Thus, an augmentation of the video stream with other spatial references and tracking information is mandatory.

Here, mixed reality technologies can be used to enable user interfaces which supplement camera images efficiently with additional information. In case the camera images are not available or have low quality, e.g. due to bandwidth problems, virtual environments based on map information and sensor data can be used as fallback. Moreover, even if the images are available, the visualization of normally non-visible sensor data, e.g. distance measurement, can augment video images and enhance the information display as demanded for spatial comprehension and tracking of the remote scenario.

The mixed reality user interface framework developed in this work for robot teleoperation enables and realizes the following key characteristics and features:

One integrated, common world model

Data from various sources (e.g. entities’ positions, camera images, maps obstacles, paths, a priori information, etc.) are modeled, and displayed to the human in an integrated fashion. Real world elements represented by video streams and sensor data are combined with virtual elements into this common model. Virtual objects are for instance built based on available map information, synthesized from sensor data, or faded in as extra information. All objects are modeled and aligned according to their physical spatial relations and properties. Dependent on gathered information all elements in the model are maintained up to date. Hereby, correct spatial relations are kept, all available information is integrated in a 3D mixed reality world, and thus the requirements for mental rotations for the human is minimized. This integrated model can be applied to both considered interaction roles in this work - supervisor as team coordinator and operator of a robot.

Natural Visualizations

Natural visualizations for the human is a key issue for efficient operation of a robot. Objects which have geometric properties should be modeled according to the representations to which the human is familiar with. On the other hand cognition behaviors of the human can be used to give intuitive, artificial guidance to the human operator. For instance if an object detected by a sensor or known a priori is an obstacle in the current moving direction of the robot, this should be intuitively visible to the human operator, e.g distance sensor data can be used to superimpose correctly aligned virtual geometries to a camera image, which visualize the limitations of the workspace in a certain area (cf. laser visualization as red band in Figure 5.5). Modeling of a 2D camera image as a 3D object based on depth sensor data (cf. Figure 5.6d) can also support this more natural cognition of the remote environment.

Support Multiple Viewpoints

The current task and the current interaction role of the human have major influence on the selection of optimal viewpoint and hereby the optimal system performance. Thus, the operator is enabled to take over arbitrary viewpoints in the integrated 3D world model. Still it is needed to provide standard viewpoints (egocentric, exocentric, top-down) for the operated systems to the human in the user interface, to make these viewpoints easily accessible, and to enable any easy reset of viewpoints. These elements give the human the possibility to undo viewpoint related actions, which is a significant design driver in HCI.

The integrated model considering spatial relations and projection properties and the support for multiple viewpoints also enables to use the user interface in the whole spectrum of mixed reality, in particular also as augmented reality user interface. It is even possible to move from an augmented reality view to an augmented virtuality view during runtime by changing the viewpoint.

Information Tracking

Different ways exist to enable the user to keep track of the information and the remote situation, and to allow for the reduction of the usage of the short-term memory of the human operator. For example, the human is supported by 3D representation of the planned, driven paths and hereby can determine the coverage of the area for exploration. A map as a global spatial reference for all the other elements is supported and can be realized in various ways. The operator can add local snapshots to the model in order to build up his/her own history of findings, landmarks, and events. These might be camera images, distance measurements (single or arrays), 3D-representations of camera images based on distance sensor data, etc.. An additional component which is important to maintain the overview of the whole scenario, mission and to plan the further actions, is a small overview map. This overview map should be attached and centered to a certain world object according to the current application and task. All different visualizations of the scenario use equal or comparable representations (e.g. color, geometry) of the specific elements keeping to the concept of "visual momentum" (Section 2.3.5).

Novel Display Devices

Many new display technologies can be applied to the user interface for robot operation. The way of realization the user interface in this work enables to make use of the capabilities of many of these devices. The concept fits well and works e.g. on stereo displays, head-mounted displays, and also on large scale stereo projection systems. Also a transfer to handheld devices is easily possible.

Adapt to the human’s interaction role

Dependent on the interaction role, the task, the mission scenario, and the available infrastructure different graphical user interface interfaces provide a optimized solution for robot operation or team coordination. The principles and characteristic suggested for the mixed reality user interfaces hold for all of these scenarios but with different adapted attributes in the final implementation. Details on the needs and challenges can be found in [20] and [16]. In the later sections different implemented interface designs are introduced, which all based on the proposed mixed reality concept. The realization of this approach is adapted according to the actual scenario requirements, and the actual role of the human in the team. Two roles are considered - a supervisor coordinating a team of robots and an operator controlling a robot inside the team.

5.3.2 Framework for Mixed Reality User Interfaces

Approach and Architecture

The design of graphical user interface for HRI requires the ability to quickly test an idea, change the operated robot(s), or change the display devices. Currently, no software framework for 3D-GUI development is available which copes with the specific requirements of robot teleoperation.

In the context of this thesis concepts and methodologies for a software framework are developed enabling a fast and easy development of mixed reality user interfaces for mobile robot operation. The software framework directly reflects the listed features in Section 5.3.1. A prototype of this software framework has been implemented as library based on Java and Java3D and is already in use for different user interface implementations. The framework enables to quickly create 3D user interfaces which can be classified anywhere inside the virtuality continuum (cf. Figure 5.1 and [197]), dependent on the application needs, the team setup, system constraints like communication and the selected robots. Virtual reality user interfaces e.g. in [23] for the applications in low-bandwidth environments like tele-laboratories [220] can be realized in the same manner like the augmented reality and augmented virtuality interfaces presented in later sections.

Usually a high level of knowledge in 3D application programming interfaces (API) is required to implement this kind of three-dimensional mixed reality user interfaces. The developed software framework enables a quick access to the implementation of three-dimensional mixed reality user interfaces for robot teleoperation. Especially for students it is advantageous, that with a smooth entry to the topic, a quick visibility of results without limitation in extending the system can be achieved. Hereby, they can quickly realize and evaluate interesting user interface systems. This could be validated by the usage of the framework for lab projects, and master/diploma thesis of students, who were able to quickly build their own representation of the 3D user interface without any knowledge of a 3D programming API. The design of the software framework also allows for the extension of defining the whole user interface as XML file, such that not even real programming knowledge is necessary.

The central element of this framework is a data structure which represents all objects (e.g. robots, sensors, cameras, maps, paths, landmarks) which should be represented in the user interface. For each physical objects a so called *model* is defined which holds all important properties and current data of this object (e.g. position/orientation, distance measurements, camera image, notes, camera calibration parameters). These models are ordered in a tree data structure, such that the parent child relationship represents their physical relation (e.g. for a robot with a laser and a camera, the robot model would be the parent node - the laser and camera model would be the child nodes of the robot model). This enables the models to inherit spatial properties. In the example it results in the fact, that if the robot moves, the attached laser and camera move accordingly.

Connector objects provide the necessary interfaces to the real hardware and additional information sources. They are initialized with connection parameters (e.g. IP addresses, serial/USB interfaces, ports) and allocate *controller* objects dependent on their available data sources. *Models* can be registered to these *controller* objects. Controller objects update the data fields of each registered model according to the update received over the connector object. The way the connector and controller objects interact enables to quickly adapt the user interface to new systems. If the same kind of information is provided by a new system only the connector object need to be changed according to the new system.

View objects define the visual representation of data inside the user interface. Two classes of view objects exists: world objects which are part of the integrated environment representation and overlay objects which always keep attached to the current viewpoint. The overlay objects are objects which should always stay in the field of view of the operator (e.g. battery status, overview map). *Models* can be registered to views. View objects are setup only with parameters considering the concrete visualization (e.g. color, transparency, filename). The view object retrieves all other necessary setup parameters directly from the model object after registration. Based on this data the virtual representation of the data is built (e.g. the intrinsic parameters of a camera model determine the projection geometry for the camera image rendering). View objects register themselves to their registered models for update notification. If the data fields of a model are updated, the model object sends a notification to each register view. On notification the view object collects its required data from the model and updates the visual representation of this information.

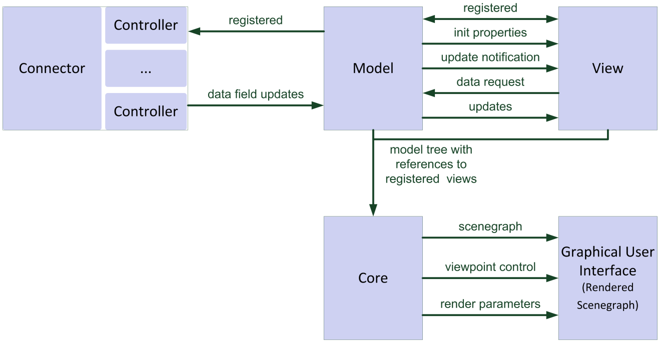


Figure 5.12: Interaction architecture for the object classes of the user interface framework.

The *core* objects manage the whole user interface. The root model and hereby the whole model tree is passed to the core object. Based on this model tree and the views attached to specific models, the whole scenegraph for the user interface is setup automatically. For each world view of a model the default viewpoints (egocentric, exocentric and top-down) are provided and the whole viewpoint control system is setup. Dependent on the selected display technology and rendering view parameters all Java3D related processes are initiated. Figure 5.12 sketches the relations between the different object classes.

The present software framework encapsulates a lot of implementation effort to realize a mixed reality user interface for robot operation. It enables to reduce the workflow to three basic steps (cf. Figure 5.13): provide the physical/spatial properties and relations, and hereby define a tree of models; connect the models to the information sources; attach visual representations inside the user interface to the models.

The software framework has been designed to be easily extensible. New models, views, connectors, and controllers can be implemented by extending the basic classes which provide the entire functionality for the software frameworks. A lot of different display devices, sensors, robots and other information sources are already supported. Examples for supported display devices are standard monitors, all stereo system which use dual head capabilities of graphics hardware, horizon-

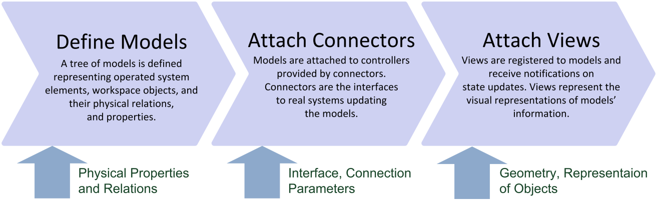

Define Models
A tree of models is defined representing operated system elements, workspace objects, and their physical relations, and properties.
Attach Connectors
Models are attached to controllers provided by connectors. Connectors are the interfaces to real systems updating the models.
Attach Views
Views are registered to models and receive notifications on state updates. Views represent the visual representations of models' information.
Physical Properties and Relations
Interface, Connection Parameters
Geometry, Representaion of Objects

Figure 5.13: Workflow for realization of an graphical user interface with the user interface framework.

tal interlaced stereo displays, 2.5D stereo displays[4], and distributed scenegraph systems like the large scale stereo projection system utilized in following sections. Currently supported robots are Merlin robots (indoor and outdoor), Pioneer robots, and all other robots supported by the Player hardware abstraction layer[182]. The following sensor connector implementations are for instance already available: MJPEG Network cameras, PMD cameras, Swissranger SR3000, Sick LMS 200 laser range finder, Hokuyo laser range finder, sonar sensors, and all standards sensors supported by the player framework.

Multi robot systems capable simulation or emulation environments are a useful and necessary tool for the user interface development in order to do first system analysis reducing setup time. The presented software framework currently provides the interfaces to two simulation frameworks - Player/Stage [182] and USARSim [103],[104]. Player is a robot device server to realize multi-robot or sensor-network systems. Stage can be used together with Player and can simulate large populations of robots in a 2D environment. USARSim is based on the Unreal Tournament 2004 game engine. It is a general purpose 3D - multi-robot simulator which provides basic physical properties of the robot and the simulated environment which closely match the real implementation of the robots and the real environment. In addition, it is also possible to simulate camera images from cameras inside the simulation, which is large advantage for user interface development. Compared to Player/Stage USARSim is only a simulation without a device server and controller concept like Player. Figure 5.14 shows a typical environment simulated with USARSim.

[4] http://www.business-sites.philips.com/3dsolutions/ (22.02.2010)

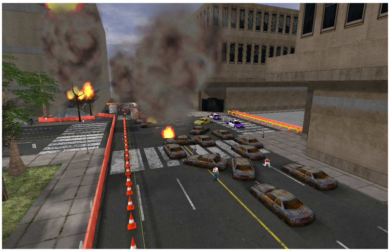


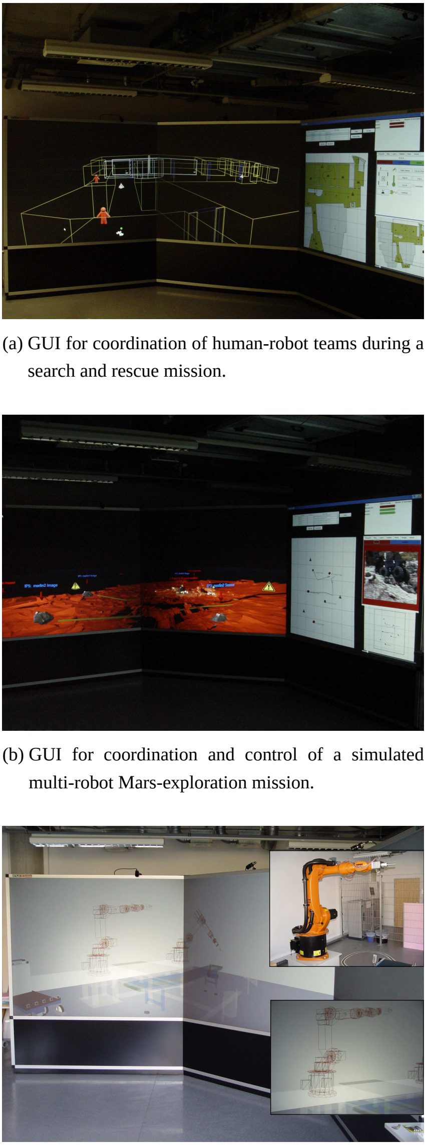
Figure 5.14: Typical environment simulated with USARSim.

Extension with distributed scenegraphs

The application (e.g. human team members share the same integrated world model) or a specific display technology (e.g. computer cluster based projection systems) might require that a three-dimensional model is shared consistently between multiple devices. This requires a flexible framework, which provides online, synchronous updates of distributed dynamic three-dimensional models. In order to enable the application of a multi-plane stereo projection for robot operation and coordination, we developed a software framework to realize a distributed scenegraph with Java and Java3D matching the requirements for such applications. In the following the basic principles are summarized. Details on the developed framework for distributed scenegraphs can be found in [20].

The used multi-plane projection stereo system consists of three projection walls, each with a width of 2 meters and a height of 1.6 meters which results in a total projection screen of 6 m x 1.6 m. The three walls are arranged with an angle of 135 degree in-between. For the projection six video projectors and a cluster composed of standard PCs are used. The pictures for the left eye and the right eye are polarized orthogonal and displayed merged on the projection screen. The users have to wear special glasses with the same polarization filters to separate the pictures for the left and right eye again and to gain the stereo effect. The glasses are light weighted and should not disturb the user in its actual task. Figure 5.15 shows a set of examples GUIs realized on our stereo projection system.

The six projection cluster computers and an additional control computer are connected to the same Ethernet network. The control workstation provides the interfaces between remote systems and human on one side and the three-



(a) GUI for coordination of human-robot teams during a search and rescue mission.

(b) GUI for coordination and control of a simulated multi-robot Mars-exploration mission.

(c) GUI for industrial monitoring and telemaintenance application.

Figure 5.15: Implemented graphical mixed reality user interfaces on large scale stereo projection system.

dimensional stereo projection on the other side. It manages changes in the three-dimensional model and broadcasts these changes to the clients. Sensors and actuators could be directly connected to this computer or can be linked via an external network.

The basic scenegraph content to be visualized is distributed over the network on system start (not limiting that new objects can be added or removed). The communication in the visualization network utilizes a multicast connection, which has several advantages over conventional point-to-point connections. Every PC which should own the common scenegraph joins an IP multicast group. That way, the control PC does not require prior knowledge of how many or which clients are used, as the router in the network manages the connection tree in real-time. Moreover, the control PC needs to send every packet only once, as the router takes care of replication and distribution to (multiple) receivers. This effectively reduces traffic on the network. With these conditions fulfilled, the presented framework enables dynamic broadcast of updates (e.g. update of positions/orientations, viewpoint manipulations, node manipulations, sensor data updates, insertion/deletion of objects, execution of pre-defined methods, etc.) for the 3D visualization. Figure 5.16 gives and overview of the system setup.

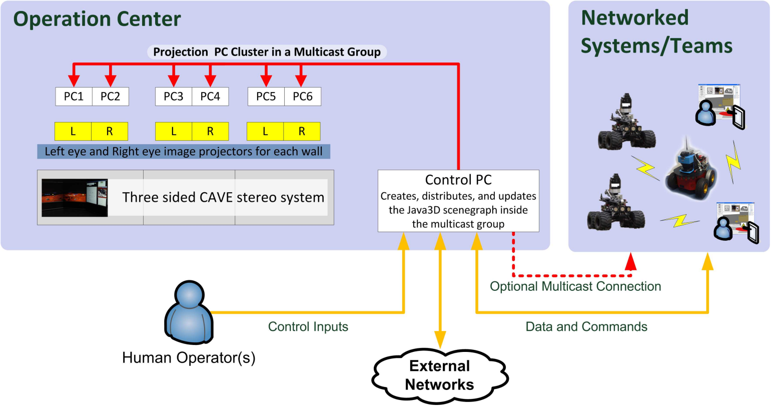

Operation Center
Projection PC Cluster in a Multicast Group
PC1
PC2
PC3
PC4
PC5
PC6
L
R
L
R
L
R
Left eye and Right eye image projectors for each wall
Three sided CAVE stereo system
Control PC
Creates, distributes, and updates the Java3D scenegraph inside the multicast group
Networked Systems/Teams
Optional Multicast Connection
Control Inputs
Data and Commands
Human Operator(s)
External Networks

Figure 5.16: Setup of three-sided projection screen for operation and coordination of remote systems.

If changes in the 3D visualization are made, nodes in the underlying Java3D scenegraph have to be manipulated. However, multiple PCs are used, each holding its own representation of the scenegraph. Thus, a direct manipulation of the graph on every PC is not possible. That means each node has to be identified in a unique way. To achieve this, two hashmaps are used on each PC. These are created at the start-up of the visualization process from the (already distributed) 3D-Scene content and are updated, when objects are added or removed. Each node in the graph is assigned a unique number, which is the same all over the network; one hashmap holds the nodes as keys, the other the numbers. So the control PC can determine the number of the node to be manipulated by entering the pointer of that node in the first hashmap, while the clients in the network can get back a pointer on that node by entering the corresponding number. This enables consistent identification and modification of nodes all over the network.

For updating of the visualized 3D scene, the control program broadcasts over the multicast network: This packet consists of a header, determining the desired operation, and the payload, i.e. all the necessary data to execute that operation. The latter consists in most cases of the number identifying the node in the scenegraph to be manipulated and the data to be changed in that node. However, the framework comprises a utility class to be used to broadcast updates, which shields the user from the technical details. Node manipulations can be broadcasted by a method call with appropriate arguments.

In the context of this work we have developed a flexible framework for 3D models that allows updating the model online, supports interaction, and integration into existing systems as it is required for the proposed scenarios. Even though the framework does not provide synchronous mechanisms, the user does not percept any asynchronism due to the applied multicast technology and a careful selection of the hardware. In the presented approach a successful trade-off is taken between flexibility, light-weight and real synchronism.

5.4 Realization of User Interfaces

Based on the considerations of Chapter 2, and user tests the mixed reality approach has been implemented in specific user interfaces for validation. In the following, first the team coordination interface is introduced. Afterwards a stereo augmented reality user interface for teleoperation of a single mobile robot is presented. The mixed reality components of both interfaces have been further integrated to a generic mixed reality user interface. There the viewpoint can be much better adjusted by the user to the current needs and many user interface elements have been further integrated to more natural representations for a human operator. This interface can be easily combined with the necessary 2D-elements of the team coordination user interface in order to enable also the supervisor operation. The presented user studies give an insight in the needs and potential of the application of augmented reality technologies for remote operation tasks.

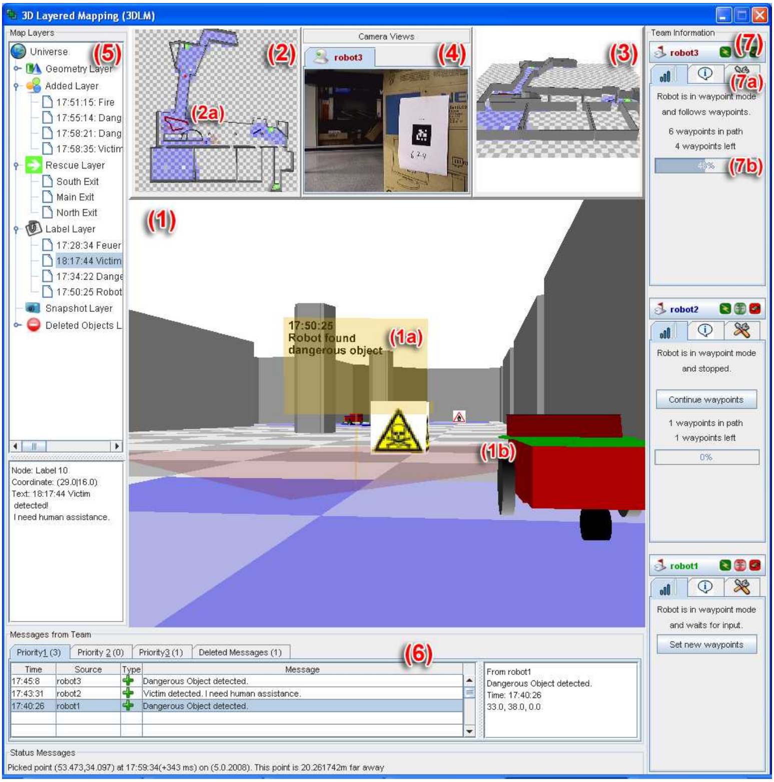
5.4.1 Team Coordination User Interface

In [18] we have analyzed the requirements for information display in the team coordinator (supervisor and operator role) and teammate (peer and operator role) user interfaces based on literature and own experiments. For task-oriented human-robot teams we used the following categories

- *Situation/environment*: overview of complete environment and knowledge of local environment of the team members.
- *Entities/team members*: actual status and capabilities of each entity, comprehension of the entities current behavior/mode and comprehension of the relations in the team (e.g. sub-teams).
- *Mission/tasks*: knowledge of the goal and the related task allocation, as well as the workload and progress of each team member.

Based on this analysis we have developed a map centric mixed reality user interface dedicated to the team coordination task. It enables a human to take over the supervisor and to some extend the operator role. This user interface is shortly introduced in the following. Detailed information on the different development steps can be found in [21], [16], [19], [18], [15], [14], and [86]. Figure 5.17 shows a screenshot of the resulting mixed reality user interface. The numbers in brackets in the following text paragraphs correspond to the label numbers in Figure 5.17.

Situation/environment. The main view for the environment is the integrated 3D model (1). Two other views of this model show an overview of the complete scene from above (2) and a side view (3). The center field shows camera images from the robots (4). The main view can be switched to different viewpoints: above



17:50:25
Robot found
dangerous object

Node: Label 10
Coordinate: (29.0|16.0)
Text: 18:17:44 Victim
detected!
I need human assistance.

Robot is in waypoint mode and follows waypoints.

6 waypoints in path
4 waypoints left

Robot is in waypoint mode and stopped.

1 waypoints in path
1 waypoints left

Robot is in waypoint mode and waits for input.

Time	Source	Type	Message
17:45:8	robot3	✚	Dangerous Object detected.
17:43:31	robot2	✚	Victim detected. I need human assistance.
17:40:26	robot1	✚	Dangerous Object detected.

From robot1
Dangerous Object detected.
Time: 17:40:26
33.0, 38.0, 0.0

Picked point (53.473,34.097) at 17:59:34(+343 ms) on (5.0.2008). This point is 20.261742m far away

Figure 5.17: Screenshot of mixed reality user interface for coordination of a human-robot team.

(overview and zoomed), entity-view (movable like a virtual robot) and attached to the real entity (following the entities movement). The user can store viewpoints and move back to these later. The environment model includes not only information about the building structure, but also task-related information as e.g. exits or dangerous areas. Therefore, the model is not only a view on the environment, but also documents the situation of the team in the remote environment. The information is arranged in layers, which are displayed in a tree-structure (5). If new information was gathered in the environment the team members can automatically update the map or send a team message. The team messages are sorted according to their priority into a table (6). The priorities are set according to the task. Here, messages that concern human life are set to highest priority (e.g. detected fires, victims), messages that concern the robots safety get medium priority (e.g. battery nearly empty) and messages for information only have lowest priority (e.g. an invalid waypoint). The supervisor can add new objects according to the team messages. He/she can also create 3D-labels (1a), which can keep more information and can be used as a memory functions for later use or for other team members.

Entities/team members. The pose of entities are visualized with avatars in the 3D model of the environment, e.g. (1b) as well as their assigned set of waypoints, e.g. (2a). The most important information of each team member is shown summarized (7): mode (waypoint or teleoperated), name, connection status, battery status, and mobility. More details are given in the tabbed panes which can be opened on demand (7a). Currently there are panes for progress of the assigned path, general information and the capabilities (sensors, equipment, special functions etc.).

Mission/tasks. For this user interface, the overall task is exploration and semantic mapping of a partly known environment, i.e. the main task of each team member is following a given set of waypoints autonomously and inform about detected fires, victims, dangers and obstacles. Maps generated with SLAM algorithms might also be included but are not considered in the presented experiments. The 3D model changes the color of the underground, such that the team coordinator gets a good overview which area was investigated already. This feature requires a relatively good self-localization of the robots; otherwise it might imply wrong assumptions to the coordinator. In cases, where the robots are not able to localize themselves correctly, this feature should be switched off or the robot should transmit a probability for the correctness of the position, which can be color-coded. Each team member has a progress pane, which shows the advance in the waypoint following (7b). The pane shows if a robot is waiting for input, if it has stopped, or if it is following the waypoints. As it can be seen in Figure 5.17, a hint is given what to do next and a related button is shown. For example, if the robot has stopped, because it found a new object, an information text is shown and a button allows the user to directly send a command to the robot to continue on its old path. The extensive visualization of the progress should give the user always best feedback about the robot's current situation, workload, and progress. This should support a reduction of the time, where there is no task assigned. The general information tab keeps the most important information of each team member as a history, such that the team coordinator can refer to this record.

5.4.2 Augmented Reality Operator User Interface

In the last section a user interface for team coordination has been introduced which focuses on the supervisor interaction role and on a high-level operator interaction. Still many application scenarios require the direct or assisted operation of a robot as part of such a team at a remote workspace.

The augmented reality user interface has been mainly designed for the task of secure and fast navigation of a mobile robot in an unknown environment during an exploration and search task. The operator should be able to understand and memorize its local surrounding fast, detect obstacles, determine the next target point, how to go there and immediately detect if the robot is approaching a dangerous situation. The basic idea for the implementation of the stereo AR user interface is to support the operator of the mobile robot during mission execution with additional information from the real world correctly registered in the image, such that the demand for mental rotations for the human operator is reduced. The integration of the additional information as virtual 3D elements in the camera image should minimize the need for changing the user focus, which is necessary in typical user interfaces where the different displays and camera image are placed side by side. Additionally, for a better depth impression and spatial understanding of the remote environment, this user interface is realized for stereo visualization.

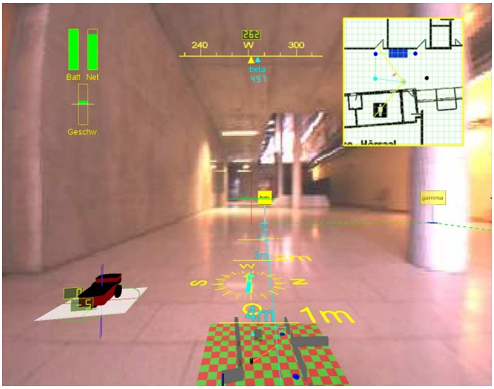


Figure 5.18: Examples for display elements of the stereo augmented reality user interface - top (from left to right): system status diagrams, HUD orientation display, 2D map; bottom: 3D pitch-roll display, 3D map and floor projection of compass rose.

The stereo augmented reality user interface here is a camera centric user interface. A stereo video stream from a mobile robot is augmented with correctly aligned virtual three-dimensional representations of sensor data received from the robot. This integrated stereo AR user-interface is present on a large-scale projection system to the robot's operator. The user interface elements are all implemented in 3D and optimized for stereo view. These elements could be aligned and displayed adjusted to the users needs. It is possible to change positions of the elements online and to hide single or all control elements. Figure 5.18 and Figure 5.19 shows some example control elements, which can be selected for the interface.

The main component of the teleoperation interface is the stereo video from the robot. It represents the work environment of the teleoperated mobile-robot. This stereo camera view is therefore the base element as background of the AR interface. All other data presented to the operator are visualized as three dimensional virtual objects. This information is integrated, and therefore correctly registered and overlaid, into the stereo camera image.

The following exemplary information elements can be shown in an integrated fashion in the user interface:

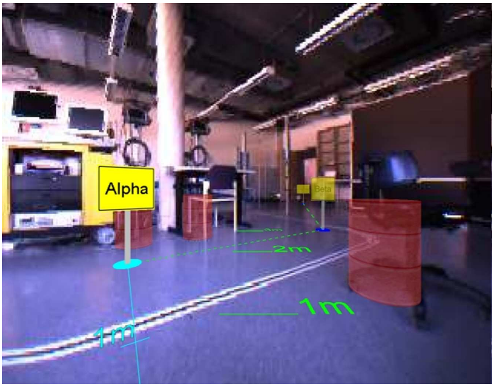


Figure 5.19: Waypoint/planned path visualizations, distance references, and highlighted obstacles (here: e.g. red barrels - cf.[221])

.

- Orientation (yaw angle) of the mobile robot
- Pitch and roll angle of the robot
- Status (e.g. battery, network link) of the robot
- Distance measurements from ultrasonic senors, corresponding to possible obstacles in the sensor field of view of the robot
- Distance reference
- Planned path and waypoints and navigation support to reach the next waypoint
- Maps as global frame of reference for the operator with robots' current positions

Waypoint system. The interface includes a waypoint management system. The user can plan a path by selecting waypoints in a 2D map. These waypoints are then placed in the correct three-dimensional position in the field of view of the robot. Additionally, the path connecting the waypoints and the distance to the next waypoint is visualized. The yellow signs in Figure 5.19 represent these waypoints. Waypoint "Alpha" for instance is the next planned waypoint.

Orientation visualization. The orientation of the robot as state element of the robot can be visualized as classical display, as Head-Up Display (HUD) element comparable to those in modern aircrafts and as compass rose projected on the floor in front of the robot (Figure 5.18). The compass displays also visualize the direction and the deviation to the direction to the next waypoint. The compass element is mainly relevant for navigation purposes.

Artificial Horizons. The artificial horizon can be visualized as classical display, as lines in the center of the screen comparable to HUDs of modern aircrafts, as a mobile robot model, and as a 3D visualization composed of two 3-sided Polygons. These visualizations of the robot's pitch and roll angle are mainly important for a safe teleoperation of the robot. They should help the user to prevent the loss of the robot by overturning.

Maps. As a frame of reference with respect to the environment a two-dimensional (Figure 5.18 top right) and a three-dimensional map (Figure 5.18 bottom center) have been implemented. They include a priori known structure, show the robot's position and field of view, range sensor information and the planned waypoints/path.

Distance/Obstacle information. A distance reference component is projected on the floor in front of the robot (Figure 5.19 center, green lines and digits). This should support distance estimations of the operator. The distance measurements from the ultrasonic sensors are utilized to highlight obstacles (Figure 5.19 red barrels) in the near surrounding of the robot.

Status information and warnings. The network, battery status, and the current velocity is presented to the user by bar charts (Figure 5.18 top left). If values reach critical states, e.g. proximity of obstacles or low battery, the corresponding warnings are shown in the center of the interface.

Figure 5.20 shows the complete user interface with selected components on the stereo projection system and the used mobile robot during tests. The system has been designed in a way that the user can fade in and out the different possible graphical components according to the current needs and own preferences. The user can even fade out all the augmentations to have the full stereo video without any occlusions.

Independent from the performance for solving the teleoperation task, the special aspects occurring when using AR and stereo techniques need to be considered. The quality of the overlay of the virtual 3D elements with the cameras or

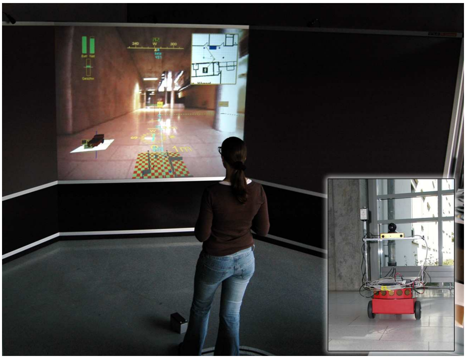


Figure 5.20: The realized stereo AR user interface on the stereo projection system and the teleoperated mobile robot.

the quality of the registration of the virtual and real world, is correlated to the performance of the overall system. The intrinsic parameters (projection parameters of camera) and the extrinsic parameters (position/orientation relative to the point of reference of the robot) need to be determined carefully. Elements like the waypoint navigation system, where the positions of the 3D elements are not relative to the robot, need a good localization of the robot in the global frame of reference. When using a large scale stereo projection system, it is also important to select the placement (especially the depth) for the different user interface elements carefully to avoid unpleasant effects (e.g. headache of the user) and disturbance of the stereo effect. More details about this interface and the different elements is given in [17]. A summary of results of user studies is also presented in Section 5.4.4.

5.4.3 Generic Mixed Reality User Interface

The two presented user interfaces in the sections before represent two specific locations in the virtual continuum. The team coordination interface is focused on the model/map of the environment and can be seen as augmented virtuality interface. By contrast the stereo augmented reality interface is video centric, and allows for a an integrated spatially correct superposition of virtual elements. Drury et al. [80] found that their map-centric interface provide a better location and status awareness while their tested videocentric interface is more effective in providing a good surrounding and activities awareness.

The listed features of this further developed generic mixed reality framework in this section allows to realize both, and to seamlessly switch between both concepts. Hereby, it enables to combine the advantages of both concepts a map-centric and video-centric interface. Although the focus of this section is on the central 3D world component and its features, it can be easily integrated with the important two dimensional elements for team coordination - the messaging system, the entities' status overviews, and structured information lists (cf. Section 5.4.1, and [14], [19]).

The graphical user interface is realized as generic mixed reality user interface. All objects and elements are spatially correct aligned in one three dimensional model representing all spatial relations, projections, and dimensions of the physical systems. Hereby, arbitrary viewpoints of the three-dimensional representation of the scene can be taken by the human operator, such that any type of augmented reality and augmented virtuality view can be realized. The integrated mixed reality user interface approach requires a careful calibration of the relative transformations, projections, and especially for the cameras a determination of the intrinsic and extrinsic parameters. These parameters of the camera are determined by standard photogrammetic methods. This also allows for combining an exocentric view with the spatial correct overlay of the real world camera image with computer generated representations of information and sensor data.

Sensor data are fused into combined three-dimensional visualizations in order to enable an increased natural perception and cognition. In the exemplary implementation (Figure 5.21) of this interface information from available laser range finders and cameras with overlapping field of view are combined to one user interface element. Based on the calibration of the camera and the laser, the camera image is projected on a geometry build from the received distances of the laser range finder according to the principles developed in Section 5.2. For the exocentric viewpoint this enables an even better understanding of the remote environment and for egocentic AR view it enables a good occlusion handling for virtual and real objects. The well calibrated three dimensional modeling of the different elements can also be seen when comparing Figure 5.21a and Figure 5.21b. Both show exactly the same 3D model - only the viewpoint is changed. In Figure 5.21b the calibrated position of the camera is chosen as viewpoint, such that a correctly aligned augmented reality view results. From this viewpoint the deformed and projected camera image looks again like a typical two dimensional rectangular image, although it is modeled three-dimensional based on the laser range data. Another advantage of this modeling is, that the user interface can be used instantaneously on a 3D stereoscopic display in order to support a better depth perception of the remote environment by the human operator. Here also a stereoscopic effect is achieved with the camera image due to the three-dimensional modeling based on range sensor distance information, although it is actually a monoscopic camera image. When using this way of combined visualization of distance and image data, tests showed that the texturing should only be active on the front side of the realized geometry and from the backside the object should be transparent. If the interface is not used in augmented reality view, standard texturing where both sides of the geometry show the camera image causes confusion to the human operator. This is due to the fact that with a non AR view, parts of the backside of the geometry might be visible. If this backside is textured with a image showing the frontside of the object, it would lead to a wrong representation of this object, because at this time nothing is known about the backside. Details on this way of three-dimensional modeling the camera image with range sensor data can be found in Section 5.2 and [3]. Virtual models which exactly match the dimensions of the real physical robots are used as avatars for the representation of the robots inside the interface. This could also be validated during the test runs where it was possible to operate the robot precisely through extremely narrow passages only based on the virtually modeled laser/camera data and the virtual representation of the robot.

By default, for each of the modeled robots and sensors the viewpoint is registered to a default viewpoint list. Thus, an immediate switching to any of these default viewpoints including all egocentic and exocentric viewpoints is available on demand. To minimize the length of the action sequences needed by the human operator a set of default viewpoints for each modeled element of the world is always on hand. These default viewpoints can be instantaneously accessed by the operator by keys associated to this type of viewpoint: egocentric (view from the element's origin), exocentric (view slightly behind and above element's origin) and a top-down view on the element. Any of this viewpoints can be selected permanently or temporarily (as long as the associated key is pressed - a quasimode [38]), can be modified, and again be reset to the default position. The feature to reset to a default viewpoint is very important, as for the human operator setting a viewpoint to a certain position/orientation needs quite an effort, and most of the arbitrary changed viewpoints are only needed for a very short time. This is in accordance with the design rule that every action should be undoable [38].

The natural cognition of the user interface for the human operator is supported by various ways. The user interface is modeled three-dimensional as the real world is three-dimensional. This can additionally be supported by stereo display technologies. Research in cognitive science showed that depth cues e.g. through illumination, shadows, sharpness are a major component for a correct three-dimensional cognition of an environment. Thus, such depth cues are also included in the user interface here, e.g. fog proportional to the distance to the actual viewpoint. Hereby, like in natural environments the farer an object is, the more it gets blurred.

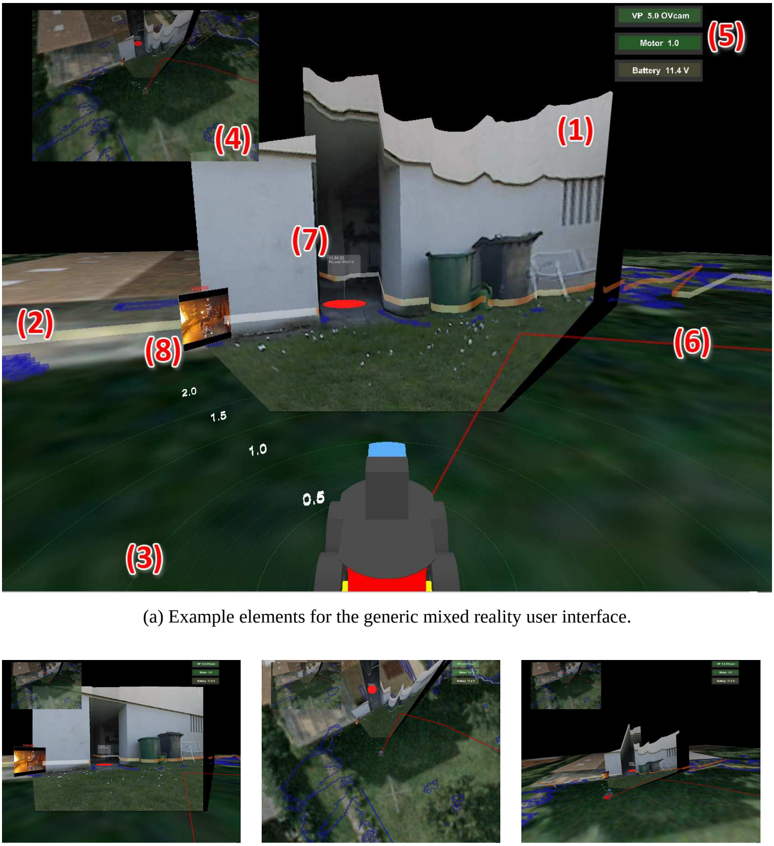
The user interface provides the communication interface between human and system. With the characteristics listed before mainly the feedback to the human is considered. But also the procedure of commanding and the input devices need to be considered for the interface appropriate control inputs are required. Besides standard devices like keyboard, mouse, and joystick also novel input devices like the Wii Controller[5] are supported. This input device based on acceleration measurements proved to be a very natural and intuitive input device for the teleoperation task, and is very much appreciated by the users. If an additional feedback device for the human operator is desired, also a force-feedback joystick can be used (cf. Chapter 4).

Figure 5.21a shows a screenshot of one implementation of the user interface with a set of example user interface elements available. The numbers in brackets in the following text paragraphs correspond to the label numbers in Figure 5.21a.

Major element is the camera image projected on the three-dimensional geometry based on sensor data from a laser range finder (1). The laser range data is additionally projected as red band (2) making use of the full field of view of the laser range finder. An additional distance reference element (3) enables an quantitative estimation of distances to objects. In combination with the robots' avatars these elements are primarily designed for robot's surrounding awareness.

Different types of maps can be visualized and faded in and out in the user interface. In the example of Figure 5.21a a satellite image is used as base 2D-map

[5] http://www.nintendo.de/ (22.02.2010)



(a) Example elements for the generic mixed reality user interface.

(b) Augmented reality egocentric view.

(c) Top-down view.

(d) Side view.

Figure 5.21: Screenshots of different views of the user interface's integrated model.

(floor in Figure 5.21) and a occupancy grid map is visualized as 3D-map (blue cubes superimposed in Figure 5.21) in order to highlight previously recorded obstacles. An additional small tilted top-down view (4) attached to the driving direction of the currently operated robot enables to maintain the overview. Hereby, mainly the location awareness can be supported.

Besides the system status information directly given by the integrated world model, specific color coded status displays (5) allow for a good status awareness of the system. The actual color of the status elements directly indicates if the status is good (green), not optimal (yellow), or critical (red).

The overall mission awareness can be supported e.g. by elements like a color-coded 3D-path (6), 3D-labels (7), or labeled snapshots (8). The color for the 3D-path can e.g. be used to indicate current progress on a planned exploration path. 3D-labels or labeled snapshots can be used by the human operator to externalize short-term memory through building his/her own geo-referenced history of interesting findings. In addition, a flashing point (7) can be used to highlight selected elements and hereby realize a component of the attention management.

The described set of interface elements of the prototype implementation show how an optimized and spatially integrated user interface can be realized.

5.4.4 User Studies

The user interfaces shown in the previous sections evolved during various smaller qualitative and larger detailed user studies. Details on published results can be found in [17], [16], [19], [18] [15], [14]. In the following some of the user studies and results are presented which focus on the operator user interfaces.

Stereo Augmented Reality

The observations and results of the user studies with the stereo augmented reality user interface (cf. Section 5.4.2) have had a major influence on the development of the other operator user interfaces. The objectives of these user studies were to test the applicability and effectiveness of augmented reality, different superimposed user interface elements, and a large scale stereo projection system for mobile robot teleoperation, and how these elements can influence the teleoperation performance as expected by theoretical consideration.

Tasks. Two tasks were performed for these studies - a combined navigation and exploration task and a navigation only task to test the superimposed virtual path. All test participants performed both tasks in separate test runs and had to fill questionnaires. The robot is equipped with a stereo camera with about 100 degree field of view, a standard web camera with about 36 degree field of view, a 3D compass, seven sonar range sensors, and a laptop mounted on the robot. The stereo projection system was the system described in Section 5.3.2. Here one wall with 100 inch diagonal was used. Three user interface configurations were available: the augmented reality interface (cf. Figure 5.18) in mono and stereo version and a standard user interface with camera image, a 2D-visualization of a map, and the heading of the robot. Before starting the test participants were given time to get familiar with the user interface elements and the controls to operate the robot while they were able to see the robot's behavior.

Exploration. For the exploration task combining navigation and perception a course with ten numbered gates was setup. Each test participant was asked to drive consecutively through these gates according to the numbers. The boundaries of the gates were placed at a distance such that it requires a certain navigation effort from the operator to pass these gates without collision. In addition, the test participants were asked to find and identify specific objects alongside this given path. The found and identified objects should be marked in a given printed map. This task had to be performed with the three interface configurations - stereo AR, mono AR, and standard. In order to reduce the influence of learning effects, three different but comparable course setups were used and the order in which each test participant had to perform the test runs with the different interface configurations was altered equally distributed.

Virtual 3D Path. Objective of the virtual 3D path experiment, was to receive subjective feedback by the test participants on the usability and effect of a 3D-path spatially correctly superimposed to the stereo camera image. Therefore, each test participant operated the mobile robot along a superimposed virtual 3D path with the interface in AR stereo configuration.

Metrics. The main element of the evaluation for this set of experiments are the subjective, qualitative results. Thus, all test participants had to fill in questionnaires after the test runs and observations during the test runs were logged. In the questionnaire people had to rate their attention to the different user interface elements, their usefulness, the augmentation, the stereo perception, and the spatial perception of the user interface. In addition, also parts of the experiments were videotaped for later evaluation.

For the virtual 3D-Path test run no further performance values were recorded. For the exploration task again task-oriented metrics for the navigation and perception task are used to measure the system performance. The following metrics were used: task completed or not, time for task completion, collisions, number of commands send by the operator, and the accuracy error of the objects drawn into the printed map by the test participant. The accuracy error represents the deviation of the position marked in the printed map from the correct position of the identified object.

Test Participants. The test participants were volunteers with an age between 21 and 31, mainly computer science students. During the test the participants were in the room with the stereo projection wall without seeing or hearing the robot. Furthermore, they were not allowed to see the robot's environment before the test. However, the test participants knew the building before it was prepared for the experiments (e.g. with obstacles, objects to find ...). All experiments were performed with nine test participants - one third female, two third male.

Results and Discussion. All test participants completed the tasks successfully. Table 5.1 shows the mean and the standard deviation of the recorded performance values. The mean values for the time for task completion are very similar for all user interface configurations. Although the stereo AR mean is 7.5% better than that of the standard user interface there is no statistically significant difference according to an ANOVA test. The observations during the test runs showed some behaviors which are not directly visible from the pure values. For the mono and stereo AR configuration the test participants had to approach the objects very close in order to identify them. Some even had to approach objects multiple times in order to see the objects from different perspectives before finally being able to identify it. For the standard user interface people could identify the objects often from a distance of some meters. This significant bias orginates mainly from the fact that the AR user interfaces had a much lower resolution per degree field of view ($3.2\frac{pixel}{degree}$) compared to the standard user interface ($8.8\frac{pixel}{degree}$). This problem in the design of the experiments was only identified after the completion of the test runs and analyzing the results. Nevertheless, it was not crucial as the major results of this experiments lay in the observations, the subjective ratings of the test participants, and the tendencies in the performance results. In addition, based on the results of these user studies much higher resolutions are used for all further developed user interfaces. For the collisions the average number of collision for the AR configurations is more than 40% lower than for the standard configuration. The statitically significant difference can also be validated with a non parametric one-way ANOVA - the Kruskal Wallis Test. With a level of significance of 5% the Kruskal-Wallis test inidicates that the collisions with standard configuration are signifcantly less than those with mono AR configuration ($p = 0.0125$) and with stereo AR configuration ($p = 0.0229$). The means for the number of issued commands by the operator are for both AR configurations more than 10% lower than for the standard configuration. Here again a bias exists due to the different pixel resolution per degree field of view and thus there is also no statistcally significant difference. The mean accuracy error of the marked objects in the printed map again is much lower for the AR configurations - upto 46%. The Kruskal-Wallis test indicates a significant difference between standard configuration and the mono AR configuration ($p = 0.0151$) and between standard configuration and the stereo AR configuration ($p = 0.0041$). The mean of the stereo AR configuration is better than that of the mono AR. Still they are quite close due to the fact that both configurations have the superimposed distance reference, which proved to be a valueable element for good distance estimations and thus has been used and further improved in the successor user interfaces (cf. Figure 5.21a).

Table 5.1: Performance results of exploration experiment - mean and standard deviation (std).

UI Configuration	Time	Collisions	Commands	Accuracy Error
	[s]	[count]	[count]	[cm]
Stereo AR	363.9 (std: 84.8)	3.6 (std: 2.4)	500.2 (std: 144.7)	87.9 (std:48.5)
Mono AR	388.4 (std: 57.4)	3.3 (std: 1.6)	486.0 (std: 143.2)	95.6 (std:27.9)
Standard	391.3 (std: 99.9)	6.2 (std: 2.4)	556.7 (std: 160.8)	158.1 (std:55.0)

In the second experiment performed by each of the test participants he/she had to follow a virtual path defined with the waypoint system of the interface. These virtual path and the waypoint symbols were augmented to the stereo video, like e.g. in Figure 5.19. All test participants appreciated very much this way of navigation support. This is also visible from the questionnaire (Figure 5.22b). Only two people did not give the highest grade for very helpful. The advantages of this element could also be validated in the sliding autonomy user studies in Chapter 4.

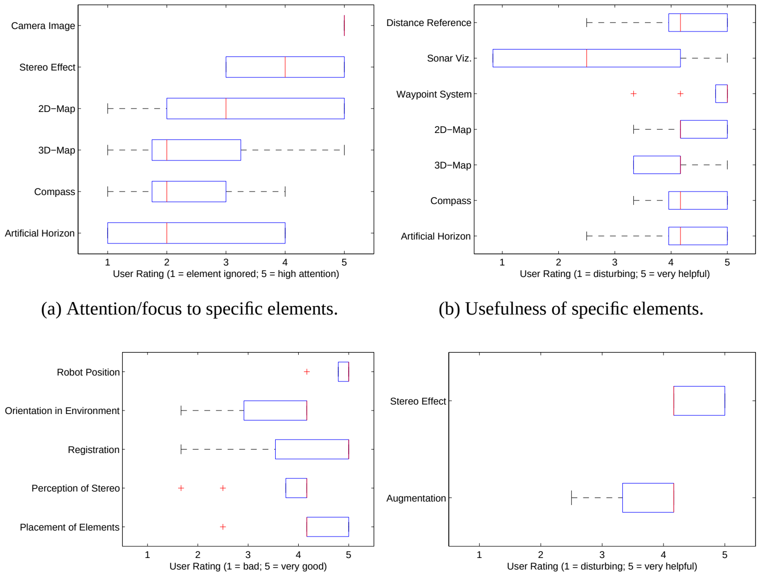
Figure 5.22 shows the boxplots (red line: median; blue box: lower and higher quartile around median; red plus: outliers of maximum whisker range 1.5 times interquartile range) of the results from the questionnaire the test participants filled after the two experiments.

One part of the questionnaire was related to the self-assessment of attention/focus to specific elements by the test participants. Figure 5.22a shows that they drew major attention to the camera image, the depth perception due to the stereo effect and the 2D-map. As expected all test participants rated the camera image as major feedback from the remote environment with high attention. The 3D-map, the compass and the artificial horizon only received little attention. Few test participants even declared that they ignored these elements completely. From these results a further integration and realization of more natural combined user interface components was supported for the development of the generic mixed reality user interface.

The test participants were also asked how useful they rate the different elements of the user interface are. Figure 5.22b shows the results of this part of the questionnaire. Almost all elements were rated as useful or very helpful, especially the waypoint system. The only element which was rated quite low was the obstacle high lightening with barrels. From the comments of the test participants two reason could be derived. At first, this type of visualization hides too much of the camera image though the barrels are partially transparent. Secondly, most of the test participants did not remember that they can fade out and in the different interface components. This obstacle high lightening was by design only thought to be used and faded in, if required, e.g. when there are bad lightening conditions. This element has been completely changed in the generic mixed reality user interface, and hereby significantly improved. The new obstacle highlighting, and object distance to robot visualization with a red limiting band and/or a texturized geometry based on the distance data, has been appreciated by everyone using this novel version of the interface as a very helpful tool for fast and safe navigation. Another observation during the test runs was that the feature to adapt the user interface to the current needs during the fulfillment of the tasks are not utilized by the participants. This result is also considered in the generic mixed reality user interface, all important changes (e.g. default viewpoints) to the user interface configuration can be directly accessed with one toggle keystroke permanently or temporarily (as long as the key is pressed) while keeping all robot control functions operational.

In order to evaluate the technical implementation of the stereo AR user interface, the test participants were asked to rate the recognizability of the robot's position, their ability to orient in the environment, the registration between camera image and virtual objects, the stereo perception, and the placement of the elements. All these questions were in average answered with good or very good grades (Figure 5.22c). This indicates that we were able to implement a high quality stereo AR user interface. Two further principal questions were if the stereo effect and the augmentation of the camera image are helpful (cf. Figure 5.22d). Although it is not visible from the performance results all test participants rated the stereo effect as helpful or even very helpful. The augmentation was also mainly rated as helpful.

In the final discussion with the test participants about the AR interface, they stated that most desirable would be a higher resolution of the stereo video or a second high resolution camera with a small field of view to inspect certain regions of interest in the large stereo video. This was also considered in the further development of this stereo AR operator user interfaces. The generic mixed reality user interface framework supports high resolution camera images with high framerates, and the use of multiple cameras in a integrated, calibrated fashion. Other suggestions like a quick access to other viewpoints which were not available in the tested version of the stereo AR interface have been also been integrated in the generic mixed reality user interface. The general opinion of the test participants about the AR interface was that is suitable for this task. Deriving from their comments most of the test participants did not take consciously advantage of the stereo presentation. The reason here might be that stereo view is natural for humans and that the experiment scenarios were too easy to really take advantage of the direct depth impression. However, this is also a good indicator for the elaborate implementation of the stereo video and the overlay for the AR interface, because there were no disturbances for the test participants, regarding stereo. A last question was which user interface configuration the test participant would



(a) Attention/focus to specific elements.

(b) Usefulness of specific elements.

(c) Spatial perception of user interface.

(d) AR and stereo assessment.

Figure 5.22: User ratings and self-assessment results from questionnaire normalized to a scale from 1 to 5.

chose for the teleoperation task. 100% of the test participants prefered the stereo configuration of the augmented reality user interface.

Technological Results. The implementation of such a stereo AR teleoperation system is very challenging, because of the large amount of technical parameters, which can significantly influence the system performance. In the following some findings of problems and solutions during the implementation of the stereo AR interface evaluated before are presented.

The first parameters to optimize are the background video characteristics: frame rate, resolution, and delay. With the given constraints (WLAN bandwidth, available computing power for compression on robot) from the setup at the time of implementation of this version of the user interface, a stereo frame rate of more than 20 fps, a resolution of 320 x 240 (for the video, the projection has

1280x1024) could be reached. Stereo camera frame rates of less than 20 fps caused discomfort to some of the people who tested the interface. The realized parameters turned out to be adequate for the navigation task. Nevertheless, if details in the environment shall be identified a higher resolution is desirable, what has been implemented in the further developments of this user interface.

The accuracy of the camera calibration, i.e. the registration of the virtual objects and the overlay, needs to be very high. The applied calibration methods (cf. Section 3.1.5) proved to be suitable.

The implementation of the AR interface as stereo system introduced further challenges and requirements. Some of these requirements only appeared after different test persons used the system. One example of these partially very subjective requirements is the accuracy of the alignment of the two camera heads. While some users could compensate several pixels of deviation in the vertical alignment of the two camera heads, others totally lost the stereo effect. Therefore, this deviation has also been measured in the calibration process and the corrected images are shown to the user. Moreover, it is difficult to see very close objects. Humans normally slightly adjust their line-of-sight for each eye dependent on the distance to the object he/she focuses on, but the parallel cameras are fixed with respect to each other. This is a limitation of the current setup as the alignment cannot be adapted to the current distance to the focused object.

Various problems which only occur when using the system in stereo could be identified. For instance, using the HUD visualization of the artificial horizon in the center of interface as it is common in HUD displays of aircrafts and in mono interfaces, significantly disturbes the stereo impression of the operator and is therefore not applicable in this type of stereo AR interface.

In addition, a function for the operator to adjust the brightness of the background stereo video to the environment conditions (e.g. contrast, light) during remote control was implemented. This very helpful feature was a result of the already gained experience with this type of interface during the implementation phase. It is now possible to adapt the brightness of the video and hereby the contrast to the virtual elements to the current requirements and environmental conditions.

Comparing Augmented Reality and Augmented Virtuality

In order to validate the advantageous of the camera centric augmented reality (AR) user interface compared to the map-centric augmented virtuality (AV) user interface for the operator role the stereo augmented reality user interface (cf. Section 5.4.2) is compared to a version of the team coordination user interface (cf. Section 5.4.1). Therefore, during the user studies for the stereo augmented reality user interface described in the section before, a test with the team coordination user interface focusing on a navigation and exploration task with a single robot was repeated with the stereo augmented reality interface. Thus, test participants were different during the test runs, but the experimental task and setup was comparable. The objective of this test was to validate the hypothesis that the stereo augmented reality interface performs better for the operator role than the team coordination interface as expected by design.

Tasks. The task for the test participants during the experiments was basically a search task. They had to navigate through an limited area covered with obstacles, find certain objects, and to mark their position in a given map. All experiments were performed with a principal test setup as described in the section before. The test participants were volunteers, mainly students with different study background. The operators were located in a remote room without seeing or hearing the robot. Furthermore, they were not allowed to see the robot's environment before the test. Part of the experiment was videotaped (operator and robot) for later evaluation. Both experiments were performed with at least eight test participants.

Results and Discussion. As main performance indicator for this experiment with the AR and the AV user interface the time until completion of the task was taken. For the AV user interface the average completion time was $1212s$ (100%) with a standard deviation of 246.3, and for the AR user interface it was $464s$ (38.2%) with a standard deviation of 70.5. The Kruskal-Wallis test verifies that the results are statistically signifcantly different ($p = 0.0003$) with a level of significance level 5%. This significant difference in the completion time indicates the expected better performance of the AR interface for an operator interaction role and leads to the assumption of a better local situational awareness for the AR interface. The AV interface was designed to coordinate multiple robots with supervisory control and the focus in the design laid on global situational awareness. A more detailed comparison on different mixed reality user interfaces can be found in [16].

5.5 Predictive Mixed Reality Display

5.5.1 Predictive Concept

Three-dimensional, integrated user interfaces [17], [52] applying mixed reality (MR) or augmented reality (AR) technologies proved their advantages compared to standard user interfaces where the elements are aligned side-by-side. But still you can typically either realize an exocentric viewpoint or an egocentric augmented reality viewpoint at the same time in the same element. In addition, for some teleoperation applications time delays can significantly disturb the human operator performance, e.g. time delays in space robotics or time-delays in system setups with ad-hoc networks where always a certain time for re-routing needs to be bridged. In space robotics one successful example to cope with the problem of longer time-delays is the virtual reality predictive display which was introduced in the ROTEX project [222] and which proved to be advantageous. A predicted virtual representation of a robot manipulator was used to operate a physical manipulator in space over a longer time delay.

With respect to the viewpoint in [223] augmented reality combined with an exocentric viewpoint is realized for mobile robot teleoperation. The authors augmented past images which are stored in a database during the robot's movement with a virtual model at the current position of their tele-operated mobile robot. The physical placement of the camera at the required position in order to achieve this exocentric view would increase the size of the robot to an undesirable size. With the system in [223] it is possible to gain this exocentric view without increasing the robot size. Nevertheless, with their approach the authors loose the important live characteristic of the camera image received from the robot. If there is a change in the environment or if something unexpected happens, this cannot be recognized because the system presents only past images.

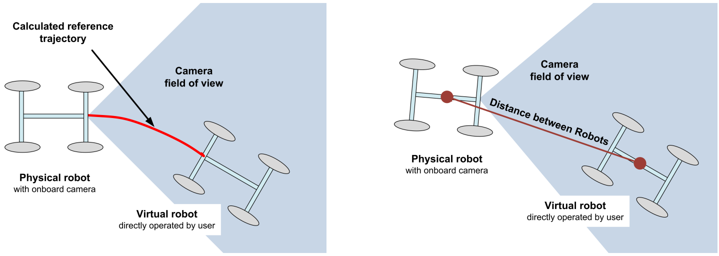
In the following the mixed reality user interface of the section before is extended to a short-term predictive mixed-reality display, which enables to tackle both of these problems. The advantages of the exocentric viewpoint are combined with the correct spatial alignment of all user interface elements at the same time in an augmented reality user interface with a live camera image. In addition, it allows for a decoupling of the human control loop and direct teleoperation loop, and hereby bridging and hiding of short time delays from the user.

The presented work introduces an approach which enables to have both important properties for the user interface - an augmented reality exocentric view. The concept makes use of the fact that nowadays a robotic system can be designed in a way that it operates very robust autonomously in a limited workspace around the robot. In this user interface the human operates a virtual robot instead of the physical real robot. The virtual robot is projected correctly aligned in the camera image from the physical robot, such that an exocentric view of the operated virtual robot and a correct AR user interface can be achieved. The physical robot with its local autonomy including obstacle avoidance follows the virtual robot with a certain distance. This can be implemented in two ways - time-based or distance based. For the time-based approach a reference trajectory is generated by commanding the virtual robot which is exactly followed by the physical robot after a certain selectable time (cf. Figure 5.23a). For the distance-based approach only a new goal position for the physical robot is defined by the virtual robot. The physical robot always moves to the virtual robot position if the distance between virtual and physical robot is above a certain threshold independent of the trajectory of the virtual robot (cf. Figure 5.23b). By this approach the human operator always receives immediate feedback on his/her control inputs even if the data from the robot is delayed. Nevertheless, he/she can always operate the robot based on the newest information from the remote environment. In addition, wrong planned paths and movements can be canceled before execution due to the predictive nature of the system and the the system gets more robust with respect to delay in the complete teleoperation chain.

In the following the concept of the distance-based short-term predictive user interface is introduced in more detail, and why the distance-based approach is advantageous compared to the time-based approach. Details on the time-based approach can be found in [4].

Timings

When designing a teleoperation system and its components the different timings and delays in the system need to be considered. Figure 5.24 gives an overview of a typical teleoperation setup. First of all there is a certain time for each sensor



(a) Physical robot exactly follows virtual robot's trajectory.

(b) Physical robot only considers virtual robot's current position as new goal.

Figure 5.23: Schematic drawing of the predictive teleoperation approach.

i for data acquisition and pre-processing (Δt_{ai}). Then there is a time which is needed to communicate and to present the data to the human operator (Δt_{com}) and finally there is a delay caused by the time a human needs to percept and react to a certain presentation of information (Δt_p). All these delays can cause significant disturbances in the closed loop system with the human as controller.

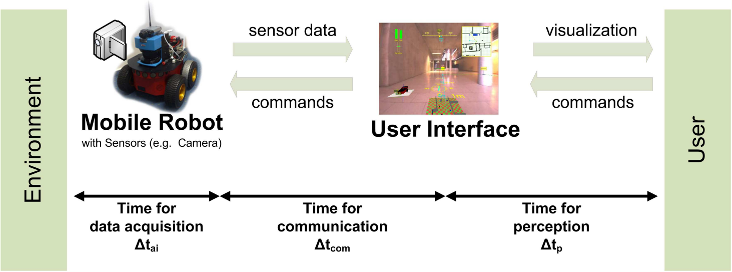

Environment
Mobile Robot
with Sensors (e.g. Camera)
sensor data
commands
User Interface
visualization
commands
User
Time for data acquisition Δt_{ai}
Time for communication Δt_{com}
Time for perception Δt_p

Figure 5.24: Teleoperation chain with most significant delays.

Thus, the proposed concept of a predictive mixed reality display enables to neglect these delays up to a certain limit. As described before, the human operator operates a virtual robot projected into the camera image which is retrieved from a physical robot with a certain delay. If the distance between virtual robot and physical robot exceeds a certain selectable limit, the physical robot autonomously moves in direction of the virtual robot (cf. Figure 5.23b) as long as the distance is higher than the selected distance threshold. The local autonomy of the physical robot is designed in a way, that the robot can plan the path to reduce the distance, that it can avoid obstacles autonomously, and that if the distance is below the threshold it always orients towards the virtual robot in order to keep it in the field of view of the camera. Figure 5.25 shows the timing of the different system events at time t. At time t the (old) sensor data (e.g. camera image, position, distance measurements) from time $t - \Delta t_1$ is presented to the human. The human generates through operation of the virtual robot a desired position for the physical robot for $t + \Delta t_3$, such that a total time difference between physical and virtual operated robot of $\Delta t_{L,F} = (\Delta t_3 + \Delta t_c + \max t_{ai})$ results. Compared to our approach presented in [4] the physical robot no longer follows after a fixed time the virtual one exactly on its defined reference trajectory, but after a certain distance and thus spatially. Hereby the prediction horizon Δt_3 and $\Delta t_{L,F}$ respectively is no longer constant. Tests with the time-based approach proposed in [4] showed, that following exactly a reference trajectory is sometimes confusing to the operator when he/she steers a lot, although the predictive approach worked very well, and was appreciated by the operators. In the time-based predictive approach each single correction step will be exactly followed by the physical robot after the total delay of $\Delta t_{L,F}$, what caused this confusion. Thus, in the new distance-based approach presented here, the predictive virtual robot and the physical robot are timely decoupled to a certain extend.

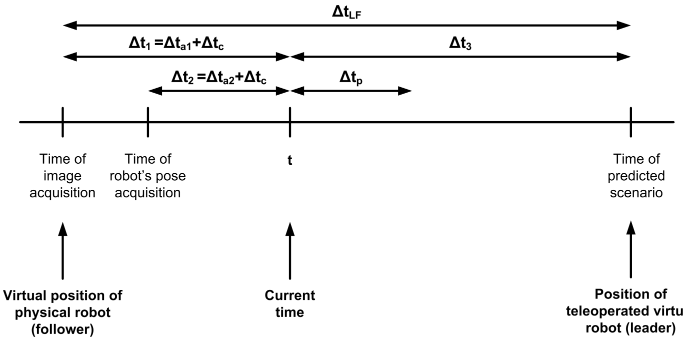

Δt_LF
Δt₁ =Δt_a1+Δt_c
Δt₃
Δt₂ =Δt_a2+Δt_c
Δt_p
Time of image acquisition
Time of robot's pose acquisition
t
Time of predicted scenario
Virtual position of physical robot (follower)
Current time
Position of teleoperated virtual robot (leader)

Figure 5.25: Timing of different system events for the predictive mixed reality system.

Setup

The different components and the data-flow in between the components to realize this extended approach are shown in Figure 5.26. On the mobile robot a local autonomy control system which works based on laser data L and position data p from the robot is implemented. The robot control system is able to steer the mobile robot to a given waypoint with obstacle avoidance. The camera image I, the laser range finder data L, and the position p are transmitted from the mobile robot to the operator station with the user interface. At the operator station a dynamic simulation model of the robot and the augmented reality interface are realized. The user inputs J to steer the robot (in the test setup a Wii Controller) are processed by the implemented dynamic model of the robot. From the dynamic model a pose $p(t+\Delta t_3)$ is calculated, which represents the position of the robot, where the operator intends the robot to move to. This pose $p(t+\Delta t_3)$ is then send to the robot as new desired goal. All other previous goals are considered as outdated by the robot, such that the robot heads and moves to the new goal. If the new goal is closer than a selected threshold distance the robot only heads to the new goal.

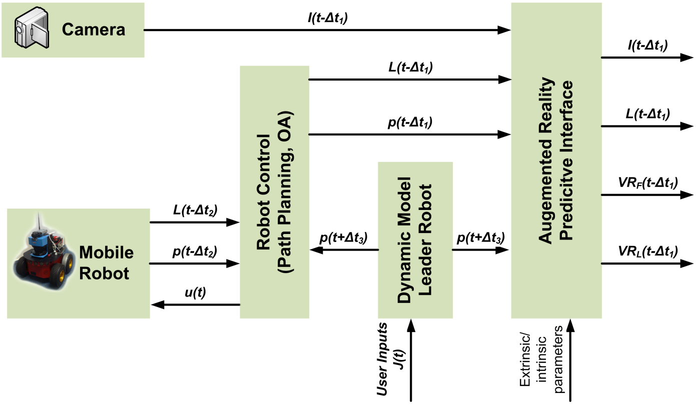


Figure 5.26: Overview of dataflow and system setup for the user interface.

In addition, the pose $p(t+\Delta t_3)$ is also processed by the graphical user interface. Together with the intrinsic and extrinsic calibration data, the integrated user interface visualization is generated based on the camera image I, the virtual operated leader robot VR_L at pose $p(t + \Delta t_3)$, the virtual representation of the physical robot VR_F at pose $p(t - \Delta t_1)$, and the virtual representation of the laser data L. Details how the graphical component of this mixed reality user interface is realized can be found in the following section.

For the implementation realized in the context of this work, a robot with skid steering and a fixed mounted camera has been chosen. The kinematic constraints of a robot of this type allow to turn the robot in place. Hereby, the physical robot is enabled to easily keep the virtual, predictive robot in the field of view of the camera and laser range finder. Alternatives for the future are a pan tilt unit for the camera and/or the laser on the robot or other robot configurations.

Graphical Realization

For the graphical user interface the generic mixed reality user interface described in Section 5.4.3 can be applied. Figure 5.27 shows the most important elements of the predictive mixed reality user interface from an exocentric viewpoint of the physical robot. The red label (1) in the figure shows the avatar for the real robot, and (2) the avatar of the operated virtual, predictive robot as wire frame.

Figure 5.28 gives two examples where the interface is used in egocentric augmented reality view. The calibrated systems enables to plan well ahead a safe way through a narrow passage (cf. Figure 5.28a). The human operator can asses easily if a robot is able to pass a narrow passage safely. The autonomy of the physical robot will then take care to actually drive through this passage safely. Figure 5.28b is an example where the systems indicates a potential collision between virtual robot and the shelves in this scene.

Model of operated Virtual Robot

In order to enable a more natural behavior of the virtual robot, a dynamic extension [224] of the standard planar kinematic model of the robot is used. In standard kinematic model x, y denote rectangular coordinates and Θ the heading angle of the vehicle in the plane defined by x, y. The control inputs are v the velocity in longitudinal direction and ω the angular turn rate. The dynamic extension of this model is realized with the introduction of the speed v as new state variable and acceleration a as new input variable leading to the following extended system

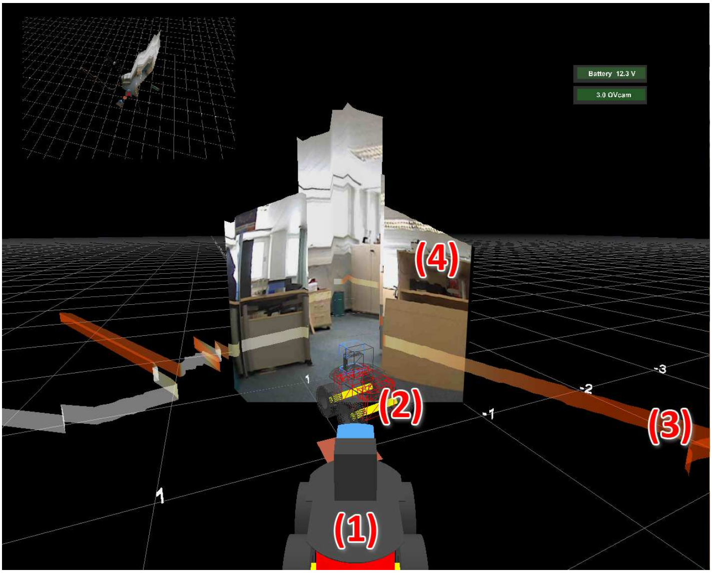
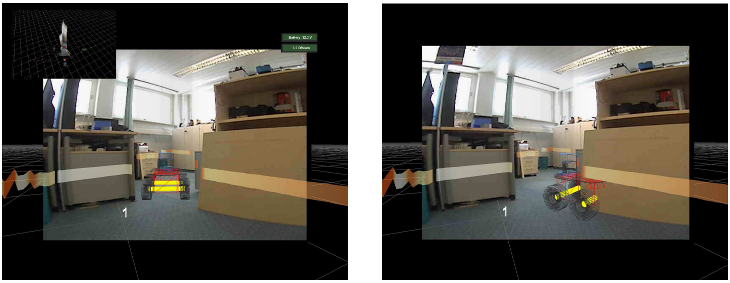


Figure 5.27: Graphical mixed reality user interface in exocentric view for the physical robot. Marked elements in the Figure: (1) avatar of the real robot; (2) operated virtual robot; (3) visualized distances from the laser range finder; (4) camera image projected on extruded laser range data.



(a) Virtual robot, passing a narrow passage.

(b) Virtual robot, showing a potential collision.

Figure 5.28: User interface with selected egocentric augmented reality view.

model: Thus, the new state vector ξ and the new input vector η can be defined as

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \Theta \\ v \end{bmatrix}, \quad \eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} a \\ \omega \end{bmatrix} \tag{5.3}$$

With the definition (Equation 5.3) the system can be written as:

$$\dot{\xi} = f(\xi) + g(\xi)\eta, \tag{5.4}$$

$$\text{where} \quad f(\xi) = \begin{bmatrix} \xi_4 \cos(\xi_3) \\ \xi_4 \sin(\xi_3) \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad g(\xi) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

This extended nonlinear model also enables an exact linearization, such that a robust multi-objective controller can be designed. This can potentially also be used for the control of the physical robot. Details on this model an the corresponding controller design can be found in [224].

5.5.2 Discussion of Results

Tests of the user interface with different people have provided interesting feedback and proved the potential of the concept. People had the impression to really operate the virtual robot in the real world and did not think anymore about the real physical robot.

The advantage of the approach is, that the teleoperation control loop is decoupled to a certain extend. This decoupling of the user input from direct action of the robot makes the overall teleoperation control much more robust against system delays and drop outs. The human operator directly sees the consequences of his/her control inputs and can even undo his/her commands as long as the virtual operated robot is not farer away from the physical robot than the selected threshold for actuation. The trials with the system showed that it is possible to hide short time-delays during the teleoperation control loop from the human operator. In one set of test runs the predictive user interface was used for teleoperation of a mobile robot in an ad-hoc network combined with an active traffic shaping.

Figure 5.29 shows the traffic profile of an exemplary test run. At first glance, throughput reductions are visible and they are really present. However, the observations during the test runs showed, that the effects of reduced image quality, delays, communication drop outs, or bandwidth limitations were not directly noticed by the user as the operation of the virtual leader robot is also possible without continuous connection. The local autonomy features on the robot take care of moving the robot towards the virtual leader robot position while taking care of obstacle avoidance. Hereby, it allows to minimize the visibility of effects like stagnant reaction of the user interface on user inputs due to the behavior of an ad-hoc network. The observations of the operators during the test runs with the predictive approach showed the reduced visibility of the network effects for the human, although the presence can be seen in the network analysis. The test with the implemented prototype system for mobile robot teleoperation proved that the approach is applicable and advantageous as expected and gave valuable insights how the human operators use such interfaces.

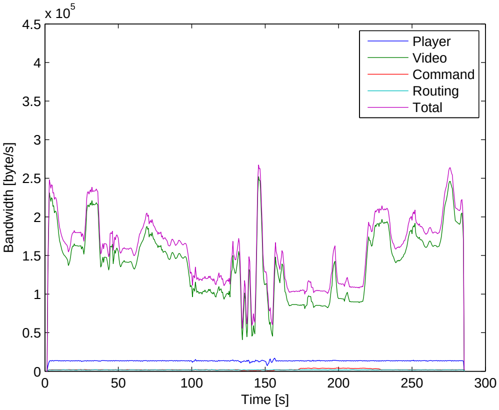


Figure 5.29: Exemplary traffic profile for operating a mobile robot with predictive user interface in an ad-hoc network.

Another result is, that for a good usability of the predictive user interfaces, a reliable short-time autonomy is a prerequisite to avoid confusion of the user. If this short-time autonomy is not robust enough or not behaving like the human operator expects the robot to behave, people start trying to outfox the autonomy instead of operating the virtual leader robot. This way, the operator tries to force the autonomy to do what he thinks being the best maneuver. As result, the performance of operating the robot is reduced. The tests during the development phase have also shown that the implemented feature to reset the virtual leader robot's position just ahead to the virtual representation of the physical robot is very important. This feature is the most comfortable way to set the virtual leader robot back to a usable position in the field of view in case the virtual leader robot was "lost" somewhere on the map. "Loosing" the virtual leader robot can easily happen if the user moves the virtual leader much faster than the local autonomy of the real robot is able to follow the virtual leader.

5.6 Summary

Remote operation and coordination of robots require a good awareness of the local surrounding of the robots and the overall situation in the remote environment. This needs to be considered when designing graphical user interfaces for the application. Based on human-factors theory and continuous tests with users an integrated and user centered approach for graphical user interfaces by application of mixed reality technologies is developed in this chapter.

The full range of the mixed reality is considered for the design of these user interfaces. In order to enable high quality augmented reality views for robot operation, specific advanced technologies are developed, set up, and analyzed. Especially the concept to efficiently project live camera images on geometries calculated from any received live distance senor data from the robot provides significant advantages for the application of remote robot operation and coordination. It allows for a better spatially integrated visualization of the remote environment to the human operator. It directly supports the easy accessible usage of different viewpoint concepts (e.g. egocentric, exocentric, augmented reality views, ...). In addition, it can make full use of the advantages of stereo display devices without significant further effort, and efficiently uses the limited resources for remote operation tasks.

Design guidelines have been developed during this work based on the detailed literature research and the results and experience gathered when users have operated the robots with various interfaces. The developed design guidelines for mixed reality graphical user interfaces are directly reflected to a software framework which enables a quick and easy setup and change of mixed reality user interface. It also provides an easy access to people who are new to this topic.

Different task and interaction role specific user interface realizations evolved iteratively during this work. A mixed reality team coordination interface is introduced which has a three-dimensional model of the remote work environment as central element. It meets the demands for information display for the actual situation, environment, involved team members, task, and mission. In addition, an advanced information management, a messaging system, and semantic maps organized in layers are included.

As interface specifically for the operator interaction role an integrated stereo augmented reality is presented, where all user interface elements are superimposed on the images provided by a stereo camera. The specific users tests carried out with the implemented augmented reality system validated the great potential of the applied augmented reality techniques. These tests promise a great potential for taking advantage of the augmented reality user interface and its specific components in future interfaces.

Based on the various smaller qualitative and larger detailed user studies, the specific implementations for the operator and supervisor interaction role have been further developed to a generic mixed reality user interface, where the viewpoint can be adjusted much better to the current needs by the user, and many user interface elements have been further integrated to more natural representations for a human operator. The generic interfaces can be easily combined (e.g. two display devices) or completely embedded into the team coordination interface, as both interface approaches are built on the same principles. Thus, it enables to have both, an optimized supervisor team coordination user interface and on demand an optimized operator user interface.

The way of spatially integrating the information in the generic user interface has also enabled an extension to a new way of robot operation - a short-term predictive mixed reality user interface combining exocentric view and augmented reality features for robot operation. It allows to decouple the teleoperation control loop over the human operator. Due to the predictive nature of the interface, an reduction of visibility of network effects to the human operator, can be achieved as the interface presents instantaneously feedback to the control inputs combined and spatially correctly aligned with the newest information from the remote environment. This way of operating the robot has also been very much appreciated during the qualitative tests with different operators. The tests also showed, that the reaction of the autonomy should match as close as possible the humans intention in order to reach maximum performance of the system. A very promising application of this predictive display concept would also be the remote operation of robotic manipulators from remote as here the overall system is much more deterministic and the autonomy or in this case path planning and tracking can be much better adapted to the humans' needs.

6 Conclusions

With the progress in robotics research and technologies the human machine interface reaches more and more the status of being the major limiting factor for the overall system performance of a teleoperation system. This work elaborates how mixed reality technologies can be applied in order to increase both, the overall system performance, and the acceptance of the user interface in the context of robot teleoperation.

First, the specific human-robot interaction and the human factors theoretical background with respect to remote operation and coordination are worked out based on a detailed literature review which already incorporated own experience with such systems. The relevant human factors are analyzed with respect to a task-oriented user interface design adapting to the interaction role of the human to the robots. Based on the supervisory control model from Sheridan [33] we sketched a system model in order to allow an optimization of teleoperation from a system perspective. Hereby the full range of autonomy in teleoperation and coordination of human-robot teams from direct teleoperation to full autonomy with supervisory control is considered, and the major relevant optimization parameters are identified. With this problem analysis background we derived a set of elementary demands and design goals for the user interface. The mixed reality technologies applied in this work match these demands and enhance performance and acceptance of the user interface. The interface design is very much influenced by the actual task and the necessary interaction roles. This work focuses on the navigation, search, and exploration tasks and on the operator and supervisor role.

In most cases the camera image is the major, central element of the user interface for teleoperation especially in mixed reality user interfaces. Network performance parameters can significantly influence the experience of the camera stream by the human operator. In the exemplary scenario urban search and rescue it is not possible to rely on infrastructure such that mobile ad-hoc networks implemented with a team of robots and humans are very desirable. The presented intelligent traffic shaping approach enables to optimize the video traffic inside a multi-hop network of non-competing nodes for a more robust navigation of robots with a mixed reality user interface. The introduced approach achieves a stabilization of the video frame rate and to keep the network link more robust through reduction of image quality. Hereby the required bandwidth is adapted dependent on the current network load situation. It has been evaluated with real hardware in different real-world scenarios. The approach works very well and it is possible to stabilize the framerate. The test users experienced much less network performance reduction and additionally the visibility of the quality reduction in some frames was very little.

3D-perception of the environment promises various advantages for user interfaces dedicated to teleoperation of robots is natural, unstructured environments. In this context novel time-of-flight cameras like PMD cameras are investigated with this respect to the remote operation of robots. Time of flight cameras provide immediately a depth image of the captured scene. Specific issues of calibration, filtering schemes, and the dynamic application oriented adaptation for camera parameters are presented. By application of the developed methods significant improvements of the three-dimensional perception can be achieved. An application example for user-centered mapping purposes is presented and the usability is validated in a informative, qualitative user study.

Different ways exist to communicate the gathered knowledge of the robots' sensors to the human operator(s). A sliding autonomy concept is introduced combining force and visual augmented reality feedback. The force feedback component allows to render the robot's current navigation intention to the human operator. If the human is not overriding the currently rendered guidance forces, the robot navigates fully autonomous, such that a real sliding autonomy with seamless transitions is achieved. The level of autonomy only depends on the current level of human intervention. The integrated graphical mixed reality user interface allows the human operator to validate the robot's current intention based on the sensor data. The sliding autonomy proved in user studies to be advantageous with increases of more than 20 % for the different performance parameters. The user studies also confirm the reached high quality and robustness of the system as test participants very much appreciated the implemented assistance features. The results are strong indicators that the application of the concept supports for a significantly higher situation awareness and common ground. An important result is also the additionally improved performance when further visual hints (here: superimposed 3D-path) about the systems intention are provided to the human operator in order to reach common ground between system and human.

The sliding autonomy concept was successfully validated with a two-dimensional laser range finder as distance sensor. However the concept allows for an easy extension with a three-dimensional range sensor like the PMD camera for the application in three-dimensional, unstructured environments in future. If the concept would be adapted to other applications for instance the pick and place task or insertion tasks with a robot manipulator arm, also other force feedback devices which enable a force-rendering with more degrees of freedom can easily be integrated.

The graphical user interface used for the sliding autonomy concept bases on the mixed reality approach for graphical user interfaces elaborated in this work. The way of spatially integrating information in the correct context allows to realize any kind of specific interfaces from the mixed reality continuum including high quality augmented reality views. This approach is supported by different developed, innovative technologies which make use of available information from the robots. An example is the realization of the live projection of two-dimensional camera images on geometries generated from robots' distance sensor data like e.g. PMD cameras or laser-range finders. This delivers much more realistic and natural representations of the remote scene and very much supports the integrated spatial understanding of the situation by the human operator. Furthermore this enables also an occlusion handling for the augmented reality view without a priori knowledge about the captured scene besides the proposed static occlusion handling. The major characteristics of the mixed reality approach are directly reflected in the developed mixed reality software framework. It provides a flexible and easy access to realize innovative mixed reality user interfaces and supports different stereo devices which can be advantageous for remote operation. The different implementations of the mixed reality approach in graphical user interfaces specifically for the supervisor role and the operator are presented and evaluated. Based on the evaluations and experience with these interfaces the mixed reality component has been further developed to a generic mixed reality user interface which further integrates information, provides more natural information representations, and an optimized viewpoint system for the users' needs.

The generic mixed reality user interface enables a further extension of the teleoperation system to a short-term predictive mixed reality user interface. With this operation concept, it is for instance possible to have both an exocentric view and an augmented reality view, to reduce the visibility of system delays for the human operator, and to cancel wrongly given commands to a certain extend. It also turned out that the matching of the autonomy component to the humans plan is very important factor for the system setup. The qualitative experiments with the predictive mixed reality user interface validated the great potential of the developed interaction concept. Many other application areas exist where this way of interaction with a short term predictive component might be very advantageous. It directly visualizes to the human operator the consequences of his/her input commands in the correct environmental, spatial context. The remote operation of a robot manipulator arm is one example where the available system information enables a quick and high-quality implementation of such an approach. The operator would command a virtual representation of the robot arm superimposed on the camera image from a camera mounted to the wrist or tool of the real, physical robot arm. This allows for a very intuitive and precise commanding of the robot without any actual risk resulting from the current applied command. It also supports views in narrow operation scenarios of the tool, which can currently sometimes not even be achieved if the robot and the human are next to each other. Thus, the investigation of the application of the short-term predictive mixed-reality user interface in the robot manipulator arm scenarios would be an interesting topic for future research.

Future extensions to the predictive user interface would also be the design and evaluation of predictive operation assistance systems. It is for instance possible to integrate an obstacle avoidance for the virtual operated robot in combination with distance measurement sensors on the physical, real robot. If a possible collision of the virtual robot is detected according to the range data, the execution of the delayed commands can be stopped well in advance or the operator has to explicitly override such assistance systems if they are realized as operation blocking systems.

In this monograph we showed how mixed reality technologies can be applied in order to enhance robot teleoperation. Concepts, technologies, and frameworks have been developed and evaluated which enable for novel user-centered approaches to the design of mixed-reality user interface for the described task area. The results enable to increase the performance of a teleoperation system by applying mixed reality technologies on the different levels of a teleoperation system. Both the technological requirements and the human factors are considered to achieve a consistent system design. In addition, a lot of potential future activities for very promising future applications could be elaborated. Future application areas of the results range from the exemplary application scenario urban search and rescue, space robotics, advanced user interfaces for operating and programming industrial robot manipulators, to areas of service robotics. In the area of service robotics a very interesting application area of the results would be intuitive support systems for elderly people. The mixed reality concepts elaborated in this work enable to support the elderly with spatially integrated and natural user interfaces e.g. for driving assistance systems on wheel-chairs or scooters.

Bibliography of the Author

[1] M. Sauer, F. Zeiger, and K. Schilling, “Mixed reality user interface for mobile robot teleoperation in ad-hoc networks,” in *Proceedings 2nd IFAC Symposium on Telematics Applications*, 2010.

[2] M. Sauer, F. Leutert, and K. Schilling, “An augmented reality supported control system for remote operation and monitoring of an industrial work cell,” in *Proceedings 2nd IFAC Symposium on Telematics Applications*, 2010.

[3] M. Sauer, F. Leutert, and K. Schilling, “Occlusion handling in augmented reality user interfaces for robotic systems,” in *Proceedings of the 41st International Symposium on Robotics (ISR 2010) and 6th German Conference on Robotics (Robotik 2010)*, 2010.

[4] M. Sauer, M. Hess, and K. Schilling, “Towards a predictive mixed reality user interface for mobile robot teleoperation,” in *Proceedings of the IFAC Workshop on Networked Robotics (Netrob 2009)*, 2009.

[5] T. Tschichholz, L. Ma, M. Sauer, and K. Schilling, “Model-based spacecraft pose estimation and motion prediction using photonic mixer devices,” in *Proceedings 60th International Astronautical Congress (IAC’09)*, 2009.

[6] F. Zeiger, M. Sauer, L. Stolz, and K. Schilling, “Integrating teams of mobile robots into wireless ad-hoc networks,” in *Proceedings of the IFAC Workshop on Networked Robotics (Netrob 2009)*, 2009.

[7] F. Zeiger, M. Sauer, L. Stolz, and K. Schilling, “Teleoperation of a mobile robot via umts link,” in *5th International Conference on Informatics, Automation and Robotics (ICINCO 2009)*, pp. 121–126, 2009.

[8] F. Zeiger, N. Krämer, M. Sauer, and K. Schilling, *Springer Lecture Notes in Electrical Engineering - Informatics in Control, Automation and Robotics*, vol. 37, ch. Mobile Robot Teleoperation via Wireless Multihop Networks - Parameter Tuning of Protocols and Real World Application Scenarios, pp. 139–152. Springer, 2009.

[9] M. Sauer, F. Zeiger, and K. Schilling, “A simulation setup for communication hardware in the loop experiments,” in *5th International Conference on Informatics, Automation and Robotics (ICINCO 2009)*, pp. 312–317, 2009.

[10] F. Zeiger, M. Sauer, and K. Schilling, “Intelligent shaping of a video stream for mobile robot teleoperation via multihop networks in real-world scenarios,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008) Workshop on Network Robot Systems: human concepts of space and activity, integration and applications*, 2008.

[11] F. Zeiger, M. Sauer, and K. Schilling, “Video transmission with adaptive quality based on network feedback for mobile robot teleoperation in wireless multi-hop networks,” in *5th International Conference on Informatics, Automation and Robotics (ICINCO 2008)*, 2008.

[12] F. Zeiger, N. Kraemer, M. Sauer, and K. Schilling, “Challenges in realizing ad-hoc networks based on wireless lan with mobile robots,” in *First Workshop on Wireless Multihop Communications in Networked Robotics (WMCNR 2008)*, 2008.

[13] M. Wiedemann, M. Sauer, F. Driewer, and K. Schilling, “Analysis and characterization of the pmd camera for application in mobile robotics,” in *17th IFAC World Congress*, 2008.

[14] F. Driewer, M. Sauer, and K. Schilling, “Design and evaluation of an user interface for the coordination of a group of mobile robots,” in *The 17th International Symposium on Robot and Human Interactive Communication (RO-MAN 2008)*, 2008.

[15] F. Driewer, M. Sauer, and K. Schilling, “Mensch-roboter interaktion in rettungseinsätzen (human-robot interaction in rescue operations),” *Schw-*

erpunktheft der i-com zu Mensch-Maschine-Interaktion in sicherheitskritischen Systemen, vol. 7, no. 1, pp. 5–11, 2008.

[16] M. Sauer, F. Driewer, K. E. Missoh, M. Göllnitz, and K. Schilling, "Approaches to mixed reality user interfaces for teleoperation of mobile robots," in *13th IASTED International Conference on Robotics and Applications (RA 2007)*, 2007.

[17] M. Sauer, F. Driewer, M. Göllnitz, and K. Schilling, "Potential and challenges of stereo augmented reality for mobile robot teleoperation," in *10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (HMS 2007)*, 2007.

[18] F. Driewer, M. Sauer, and K. Schilling, "Discussion of challenges for user interfaces in human-robot teams," in *3rd European Conference on Mobile Robots (ECMR 2007)*, 2007.

[19] F. Driewer, M. Sauer, and K. Schilling, "Design and evaluation of a teleoperation interface for heterogeneous human-robot teams," in *10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (HMS 2007)*, 2007.

[20] F. Driewer, M. Sauer, F. Leutert, and K. Schilling, "A flexible framework for distributed three-dimensional models in telematic applications," in *IASTED International Conference on Telematics (TA 2007)*, 2007.

[21] F. Driewer, M. Sauer, and K. Schilling, "Mixed reality for teleoperation of mobile robots in search and rescue scenarios," in *12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2006)*, 2006.

[22] G. Zysko, F. Zeiger, K. Schilling, and M. Sauer, "Remote laboratory experiments addressing path planning for mobile robots," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO 2005)*, pp. 431 – 434, 2005.

[23] M. Sauer, F. Zeiger, F. Driewer, and K. Schilling, "Remote control on mobile robots in low bandwidth environments," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO 2005)*, pp. 163 – 168, 2005.

[24] M. Sauer, D. Eck, and M. Schmidt, “Augmented reality user interface for controlling a mobile robot with onboard camera,” in *Mechatronics & Robotics Conference (Mechrob 2004)*, 2004.

General References

[25] J. Saarinen, S. Heikkila, M. Elomaa, J. Suomela, and A. Halme, “Rescue personnel localization system,” in *Proceedings IEEE International Safety, Security and Rescue Robotics Workshop*, pp. 218–223, 2005.

[26] H. Jones and P. Hinds, “Extreme work teams: using swat teams as a model for coordinating distributed robots,” in *CSCW ’02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, (New York, NY, USA), pp. 372–381, ACM, 2002.

[27] J. L. Burke, R. R. Murphy, M. D. Coovert, and D. L. Riddle, “Moonlight in miami: a field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise,” *Human-Computer Interaction*, vol. 19, no. 1, pp. 85–116, 2004.

[28] R. Murphy, “Human-robot interaction in rescue robotics,” *IEEE Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 34, no. 2, pp. 138–153, 2004.

[29] R. Murphy and J. L. Burke, “Up from the Rubble: Lessons Learned about HRI from Search and Rescue,” in *Proceedings of the 49th Annual Meetings of the Human Factors and Ergonomics Society*, 2005.

[30] J. A. Adams, “Human-robot interaction design: Understanding user needs and requirements,” in *Proceedings of the 2005 Human Factors and Ergonomics Society 49th Annual Meeting*, (Orlando, FL), pp. 447–451, 2005.

[31] F. Driewer, H. Baier, and K. Schilling, “Robot-human rescue teams: a user requirements analysis,” *Advanced Robotics*, vol. 19, no. 8, pp. 819–838, 2005.

[32] K. Schilling and F. Driewer, “Remote control of mobile robots for emergencies,” in *16th IFAC World Congress, Prague, Czech Republic*, 2005.

[33] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. The MIT Press, 1992.

[34] J. A. Adams and M. Skubic, “Introduction to the special issue on human-robotic interaction,” *IEEE Transactions on Systems, Man and Cybernetics - Part A, Special Issue on Human-Robotic Interaction*, vol. 35, no. 4, pp. 433 – 437, 2005.

[35] M. A. Goodrich and A. C. Schultz, “Human-robot interaction: A survey,” *Foundation and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[36] J. Scholtz, “Theory and evaluation of human robot interactions,” in *36th Annual Hawaii International Conference on System Sciences*, 2003.

[37] J. A. Adams, “Critical considerations for human-robot interface development,” in *AAAI Fall Symposium on Human-Robot Interaction, AAAI Technical Report, FS-02-03*, (Cape Cod, MA), pp. 1–8, 2002.

[38] J. Raskin, *The humane interface : new directions for designing interactive systems*. Reading, Mass.: Addison-Wesley, 2. print. ed., 2000.

[39] C. Breazeal, “Social interactions in hri: the robot view,” *IEEE Transactions on Systems, Man, and Cybernetics Part C: Application & Reviews*, vol. 34, no. 2, pp. 181–186, 2004.

[40] T. W. Fong, C. Thorpe, and C. Baur, “Collaboration, dialogue, and human-robot interaction,” in *Proceedings of the 10th International Symposium of Robotics Research, Lorne, Victoria, Australia*, (London), Springer-Verlag, 2001.

[41] H. Yanco and J. Drury, “Classifying human-robot interaction: an updated taxonomy,” in *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2841–2846, 2004.

[42] C. L. Bethel and R. R. Murphy, "Non-facial/non-verbal methods of affective expression as applied to robot-assisted victim assessment," in *HRI '07: Proceedings of the ACM/IEEE international conference on Human-robot interaction*, (New York, NY, USA), pp. 287–294, ACM, 2007.

[43] J. Scholtz and S. Bahrami, "Human-robot interaction: development of an evaluation methodology for the bystander role of interaction," in *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3212–3217, 2003.

[44] T. Fong and C. Thorpe, "Vehicle teleoperation interfaces," *Autonomous Robots*, vol. 11, no. 1, pp. 9–18, 2001.

[45] T. W. Fong and I. Nourbakhsh, "Interaction challenges in human-robot space exploration," *Interactions*, vol. 12, no. 1, pp. 42–45, 2005.

[46] M. R. Endsley, "Theoretical underpinnings of situation awareness: A critical review," in *Situation awareness analysis and measurement* (M. R. Endsley and D. J. Garland, eds.), ch. 1, pp. 3–26, Lawrence Erlbaum Associates, 2000.

[47] C. Bolstad, A. Costello, and M. Endsley, "Bad situation awareness designs: What went wrong and why," in *Proceedings of the International Ergonomics Association 16th World Congress*, 2006.

[48] J. Steuer, "Defining virtual reality: Dimensions determining telepresence," *Journal of Communication*, vol. 42, no. 4, pp. 73–93, 1992.

[49] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[50] E. Krotkov, R. Simmons, F. Cozman, and S. Koenig, "Safeguarded teleoperation for lunar rovers: From human factors to field trials," in *Proceedings IEEE Planetary Rover Technology and Systems Workshop*, 1996.

[51] D. Few, D. Bruemmer, and M. Walton, "Improved human-robot teaming through facilitated initiative," in *Proceedings 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2006)*, pp. 171–176, 2006.

[52] C. Nielsen, M. Goodrich, and R. Ricks, "Ecological interfaces for improving mobile robot teleoperation," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 927–941, 2007.

[53] D. Bruemmer, J. Marble, D. Dudenhoeffer, M. Anderson, and M. McKay, "Mixed-initiative control for remote characterization of hazardous environments," in *36th Annual Hawaii International Conference on System Sciences*, 2003.

[54] J. Crandall and M. Goodrich, "Experiments in adjustable autonomy," in *Proceedings IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 1624–1629, 2001.

[55] L. Ma and K. Schilling, "Survey on bilateral teleoperation of mobile robots," in *13th IASTED International Conference on Robotics and Applications*, pp. 489–494, 2007.

[56] K. Schilling, J. de Lafontaine, and H. Roth, "Autonomy capabilities of European deep space probes.," *Autonomous Robots*, vol. 3, pp. 19–30, 1996.

[57] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita, "Interactive humanoid robots for a science museum," *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 25–32, 2007.

[58] R. Parasuraman, T. Sheridan, and C. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 30, no. 3, pp. 286–297, 2000.

[59] T. W. Fong, C. Thorpe, and C. Baur, "Robot as partner: vehicle teleoperation with collaborative control," in *Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*, 2002.

[60] T. Fong, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 699–704, 2003.

[61] M. B. Dias, B. Kannan, B. Browning, E. Jones, B. Argall, M. F. Dias, M. B. Zinck, M. M. Veloso, , and A. T. Stentz, "Sliding autonomy for peer-to-peer human-robot teams," in *Proceedings 10th International Conference on Intelligent Autonomous Systems*, 2008.

[62] M. Baker and H. Yanco, “Autonomy mode suggestions for improving human-robot interaction,” in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2948–2953, 2004.

[63] M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall, “Managing autonomy in robot teams: observations from four experiments,” in *HRI ’07: Proceedings of the ACM/IEEE international conference on Human-robot interaction*, (New York, NY, USA), pp. 25–32, ACM, 2007.

[64] K. Stubbs, P. J. Hinds, and D. Wettergreen, “Autonomy and common ground in human-robot interaction: A field study,” *IEEE Intelligent Systems*, vol. 22, no. 2, pp. 42–50, 2007.

[65] G. Hoffman and C. Breazeal, “Collaboration in human-robot teams,” in *Proceedings of the AIAA 1st Intelligent Systems Technical Conference*, 2004.

[66] A. Halme, *Human - robot interfacing by the aid of cognition based interaction*. Vienna: IN-TECH, 2008.

[67] R. Olivares, C. Zhou, J. A. Adams, and R. Bodenheimer., “Interface evaluation for mobile robot teleoperation,” in *Proceedings of the ACM Southeast Conference (ACMSE03)*, (Savannah, GA), pp. 112–118, 2003.

[68] D. Eck, M. Stahl, and K. Schilling, “The small outdoor rover MERLIN and its assistance system for tele-operations,” in *Proceedings of International Conference on Field and Service Robotics (FSR 2007), Chamonix (France)*, 2007.

[69] M. W. Kadous, R. K.-M. Sheh, and C. Sammut, “Effective user interface design for rescue robotics,” in *HRI ’06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, (New York, NY, USA), pp. 250–257, ACM, 2006.

[70] C. W. Nielsen and M. A. Goodrich, “Comparing the usefulness of video and map information in navigation tasks,” in *HRI ’06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, (New York, NY, USA), pp. 95–101, ACM, 2006.

[71] B. Shneiderman, *Designing the user interface : strategies for effectivehuman-computer interaction*. Reading, Mass. [u.a.]: Addison-Wesley, 2. ed. ed., 1992.

[72] M. A. Goodrich and D. R. Olsen, “Seven principles of efficient human robot interaction,” in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 4, pp. 3942–3948, 2003.

[73] H. A. Yanco, J. L. Drury, and J. Scholtz, “Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition,” *Human-Computer Interaction*, vol. 19, no. 1, pp. 117–149, 2004.

[74] J. Scholtz, J. Young, J. Drury, and H. Yanco, “Evaluation of human-robot interaction awareness in search and rescue,” in *Proceedings IEEE International Conference on Robotics and Automation (ICRA ’04)*, vol. 3, pp. 2327–2332, 2004.

[75] A. Steinfeld, “Interface lessons for fully and semi-autonomous mobile robots,” in *Proceedings IEEE International Conference on Robotics and Automation (ICRA ’04)*, vol. 3, pp. 2752–2757, 2004.

[76] P. Hinterseer, S. Hirche, S. Chaudhuri, E. Steinbach, and M. Buss, “Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems,” *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 588–597, 2008.

[77] H. Yanco and J. Drury, “"where am i?" acquiring situation awareness using a remote robot platform,” in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2835–2840, 2004.

[78] J. Scholtz, B. Antonishek, and J. Young, “Evaluation of a human-robot interface: development of a situational awareness methodology,” in *37th Annual Hawaii International Conference on System Sciences*, 2004.

[79] J. Drury, J. Scholtz, and H. Yanco, “Awareness in human-robot interactions,” in *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 912–918, 2003.

[80] J. L. Drury, B. Keyes, and H. A. Yanco, “Lassoing hri: analyzing situation awareness in map-centric and video-centric interfaces,” in *HRI ’07: Proceedings of the ACM/IEEE international conference on Human-robot interaction*, (New York, NY, USA), pp. 279–286, ACM, 2007.

[81] J. A. Adams, “Unmanned vehicle situation awareness: A path forward,” in *Proceedings of the 2007 Human Systems Integration Symposium*, (Annapolis, Maryland, USA), 2007.

[82] A. Halme, “Common situation awareness as basis for human-robot interface,” in *Climbing and Walking Robots* (M. Tokhi, G. Virk, and M. Hossain, eds.), pp. 3–18, Springer Berlin Heidelberg, 2006.

[83] J. Suomela, J. Saarinen, and A. Halme, “Creating common presence for a multientity rescue team,” in *Proceedings of the 16th IFAC World Conference*, 2005.

[84] J. Suomela, J. Saarinen, A. Halme, and P. Harmo, “Online interactive building of presence,” *Field and Service Robotics*, vol. 24, pp. 395–404, 2006.

[85] J. Saarinen, *A Sensor-Based Personal Navigation System and its Application for Incorporating Humans into a Human-Robot Team*. PhD thesis, Helsinki University of Technology Automation Technology, 2009.

[86] F. Driewer, *Teleoperation Interfaces in Human-Robot Teams*. PhD thesis, Universität Würzburg, Am Hubland, 97074 Würzburg, 2008.

[87] H. H. Clark and S. A. Brennan, “Grounding in communication,” in *Perspectives on socially shared cognition* (L. B. Resnick, J. M. Levine, and S. D. Teasley, eds.), 1991.

[88] K. Stubbs, P. Hinds, and D. Wettergreen, “Challenges to grounding in human-robot interaction: Sources of errors and miscommunications in remote exploration robotics,” in *HRI ’06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, (New York, NY, USA), pp. 357–358, ACM, 2006.

[89] S. Kiesler, “Fostering common ground in human-robot interaction,” in *Proceedings IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN 2005)*, pp. 729–734, 2005.

[90] J. Trafton, N. Cassimatis, M. Bugajska, D. Brock, F. Mintz, and A. Schultz, “Enabling effective human-robot interaction using perspective-taking in robots,” *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 35, no. 4, pp. 460–470, 2005.

[91] T. Fong, J. Scholtz, J. Shah, L. Fluckiger, C. Kunz, D. Lees, J. Schreiner, M. Siegel, L. Hiatt, I. Nourbakhsh, R. Simmons, R. Ambrose, R. Burridge, B. Antonishek, M. Bugajska, A. Schultz, and J. Trafton, “A preliminary study of peer-to-peer human-robot interaction,” in *Proceedings IEEE International Conference on Systems, Man and Cybernetics (SMC ’06)*, vol. 4, pp. 3198–3203, 2006.

[92] C. D. Wickens, M. Vincow, and M. Yeh, *Handbook of visual spatial thinking*, ch. Design Applications of Visual Spatial Thinking: The Importance of Frame of Reference, pp. 383 – 425. Cambridge University Press, 2005.

[93] S. K. Card, T. P. Moran, and A. Newell, *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, 1 ed., 1983.

[94] J. L. Drury, J. Scholtz, and D. Kieras, “Adapting goms to model human-robot interaction,” in *HRI ’07: Proceedings of the ACM/IEEE international conference on Human-robot interaction*, (New York, NY, USA), pp. 41–48, ACM, 2007.

[95] E. Clarkson and R. C. Arkin, “Applying heuristic evaluation to human-robot interaction systems,” Tech. Rep. GIT-GVU-06-08, Georgia Institute of Technology, 2006.

[96] S. G. Hart and L. E. Staveland, *Human Mental Workload*, ch. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research., pp. 139–183. Amsterdam: North Holland: Elsevier, 1988.

[97] S. G. Hart, “Nasa-task load index (nasa-tlx); 20 years later,” *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 50, pp. 904–908(5), 2006.

[98] M. R. Endsley, *Situation Awareness Analysis and Measurement*, ch. Direct measurement of situation awareness: validity and use of SAGAT, pp. 149–174. Lawrence Erlbaum Associates, 2000.

[99] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, “Common metrics for human-robot interaction,” in *HRI ’06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, (New York, NY, USA), pp. 33–40, ACM, 2006.

[100] K. Schilling, M. Mellado, P. Lopez, and J. Remm, “Free navigating autonomous transport robots for flexible industrial production,” in *Changing the Ways We Work* (N. Mårtensson, R. Mackay, and S. Björgvinsson, eds.), pp. 674–684, IOS Press, 1998.

[101] K. Schilling, “Control of planetary rovers,” *Special Issue of Control Engineering Practice*, vol. 5, p. 823ff, 1997.

[102] *ISO/IEC 7498-1:1994(E): Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*. International Organization for Standardization(ISO), Geneva, Switzerland.

[103] J. Wang, M. Lewis, and J. Gennari, “Usar: A game-based simulation for teleoperation,” in *47th Annual Meeting of the Human Factors and Ergonomics Society*, 2003.

[104] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, “Usarsim: a robot simulator for research and education,” in *2007 IEEE International Conference on Robotics and Automation (ICRA’07)*, 2007.

[105] D. M. Stipanović, P. F. Hokayem, M. W. Spong, and D. D. Šiljak, “Cooperative avoidance control for multiagent systems,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 699–707, 2007.

[106] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Princeton Series in Applied Mathematics, Princeton, NJ: Princeton University Press, 2008.

[107] M. N. Rooker and A. Birk, “Multi-robot exploration under the constraints of wireless networking,” *Control Engineering Practice*, vol. 15, no. 4, pp. 435–445, 2007.

[108] A. Das, J. Spletzer, V. Kumar, and C. Taylor, “Ad hoc networks for localization and control,” in *Proceedings of the 41st IEEE Conference on Decision and Control, (CDC 2002)*, vol. 3, pp. 2978–2983, 2002.

[109] H. G. Nguyen, N. Pezeshkian, A. Gupta, and N. Farrington, “Maintaining communication link for a robot operating in a hazardous environment,” in *Proceedings of ANS 10th International Conference on Robotics and Remote Systems for Hazardous Environments, Gainesville, Florida, March 28-31*, 2004.

[110] N. Pezeshkian, H. G. Nguyen, and A. Burmeister, “Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations,” in *Proceedings of the 13th IASTED International Conference on Robotics and Applications, Wuerzburg (Germany)*, 2007.

[111] H. G. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J. M. Spector, “Autonomous communication relays for tactical robots,” in *The 11th International Conference on Advanced Robotics, Proceedings of ICAR 2003, Coimbra, Portugal, June 30 - July 3*, 2003.

[112] N. Pezeshkian, H. G. Nguyen, and A. Burmeister, “Unmanned ground vehicle non-line-of-sight operations using relaying radios,” in *Proceedings of the 12th IASTED International Conference on Robotics and Applications, Honolulu, Hawaii (USA)*, 2006.

[113] Y.-C. Hu and D. Johnson, “Design and demonstration of live audio and video over multihop wireless ad hoc networks,” in *Proceedings MILCOM 2002*, vol. 2, pp. 1211–1216, 2002.

[114] L. Parker, “Alliance: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots,” in *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems ’94. ’Advanced Robotic Systems and the Real World’ IROS ’94*, vol. 2, pp. 776–783, 1994.

[115] A. Ollero, G. Hommel, J. Gancet, L.-G. Gutierrez, D. Viegas, P.-E. Forssen, and M. Gonzalez, “COMETS: A multiple heterogeneous UAV system,” in *IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR 2004)*, 2004.

[116] T. Chung, L. Cremean, W. B. Dunbar, Z. Jin, D. Moore, A. Tiwari, D. V. Gogh, and S. Waydo, "A platform for cooperative and coordinated control of multiple vehicles: The caltech multi-vehicle wireless testbed," in *Proceedings 3rd Conference on Cooperative Control and Optimization*, 2002.

[117] M. N. Rooker and A. Birk, "Combining exploration and ad-hoc networking in robocup rescue," in *RoboCup 2004: Robot Soccer World Cup VIII* (D. Nardi, M. Riedmiller, and C. Sammut, eds.), vol. 3276 of *Lecture Notes in Artificial Intelligence (LNAI)*, pp. 236–246, Springer, 2005.

[118] F. Zeiger, C. Selbach, B. Ruderisch, and K. Schilling, "An application protocol to integrate a small size helicopter into an ip based ad-hoc network," in *RoboComm '07: Proceedings of the 1st international conference on Robot communication and coordination*, IEEE Press, 2007.

[119] A. Ollero, J. Alcazar, F. Cuesta, F. Lopez-Pichaco, and C. Nogales, "Helicopter Teleoperation for Aerial Monitoring in the COMETS Multi-UAV System," in *3rd IARP Workshop on Service, Assistive and Personal Robots (IARP 2003), Madrid (Spain)*, 2003.

[120] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, 2002.

[121] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Talyor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1, pp. 111–131, 2008.

[122] W. Kiess and M. Mauve, "A survey on real-world implementations of mobile ad-hoc networks," *Ad Hoc Netw.*, vol. 5, no. 3, pp. 324–339, 2007.

[123] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 85–97, ACM, 1998.

[124] C. Perkins, E. Royer, S. Das, and M. Marina, “Performance comparison of two on-demand routing protocols for ad hoc networks,” *IEEE Personal Communications*, vol. 8, no. 1, pp. 16–28, 2001.

[125] T. D. Dyer and R. V. Boppana, “A comparison of tcp performance over three routing protocols for mobile ad hoc networks,” in *MobiHoc ’01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, (New York, NY, USA), pp. 56–66, ACM, 2001.

[126] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing*, pp. 153–181, Kluwer Academic Publishers, 1996.

[127] J. Redi and R. Welsh, “Energy-conservation for tactical mobile robots,” in *Proceedings of Military Communications Conference, MILCOM*, 1999.

[128] S. Das, C. E. Perkins, and E. M. Belding-Royer, “Ad hoc On-Demand Distance Vector (AODV) Routing.” IETF RFC 3561, July 2003.

[129] I. Chakeres and E. Belding-Royer, “Aodv routing protocol implementation design,” in *24th International Conference on Distributed Computing Systems Workshops*, pp. 698–703, 2004.

[130] F. Zeiger, N. Kraemer, and K. Schilling, “Tuning parameters of routing protocols to improve the performance of mobile robot teleoperation via wireless ad-hoc networks,” in *International Conference in Informatics, Control, Automation and Robotics (ICINCO 2008)*, 2008.

[131] F. Zeiger, F. Kempf, and K. Schilling, “UMTS one way delay characterization for mobile robot teleoperation,” in *Proceedings of the 41st International Symposium on Robotics (ISR 2010) and 6th German Conference on Robotics (Robotik 2010)*, 2010.

[132] J. P. Luck, P. L. McDermott, L. Allender, and D. C. Russell, “An investigation of real world control of robotic assets under communication latency,” in *HRI ’06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, (New York, NY, USA), pp. 202–209, ACM, 2006.

[133] H. Everett, *Sensors for mobile robots*. AK Peters, 1995.

[134] R. Sheh, N. Jamali, M. W. Kadous, and C. Sammut, “A low-cost, compact, lightweight 3d range sensor,” in *Australasian Conference on Robotics and Automation*, 2006.

[135] A. Nüchter, K. Lingemann, and J. Hertzberg, “Kurt3d – a mobile robot for 3d mapping of environments,” *ELROB Technical Paper, Hammelburg, Germany*, 2006.

[136] O. Wulf and B. Wagner, “Fast 3d scanning methods for laser measurement systems,” in *International Conference on Control Systems and Computer Science (CSCS14)*, July 2003.

[137] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck, “Smart pixel - photonic mixer device (pmd),” in *M2VIP ´98 - International Conference on Mechatronics and MachineVision in Practice*, pp. 259–264, 1998.

[138] A. Prusak, H. Roth, and R. Schwarte, “Application of 3d-pmd video cameras for tasks in the autonomous mobile robotics,” in *16th IFAC World Congress*, 2005.

[139] T. Oggier, B. Büttgen, and F. Lustenberger, “Swissanger sr3000 and first experiences based on miniaturized 3d-tof cameras,” in *Proceedings of the First Range Imaging Research Day at ETH Zurich*, 2005.

[140] T. Ringbeck, “A 3d time of flight camera for object detection,” in *8th Conference on Optical 3-D Measurement Techniques*, 2007.

[141] T. Möller, H. Kraft, J. Frey, M. Albrecht, and R. Lange, “Robust 3d measurement with pmd sensors,” in *Proceedings of the First Range Imaging Research Day at ETH Zurich*, 2005.

[142] T. Ringbeck, T. Möller, and B. Hagebeuker, “Multidimensional measurement by using 3-d pmd sensors,” *Advances in Radio Science*, vol. 5, pp. 135–146, 2007.

[143] U. Scheunert, B. Fardi, N. Mattern, G. Wanielik, and N. Keppeler, “Free space determination for parking slots using a 3d pmd sensor,” in *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 154–159, 2007.

[144] H. Du, T. Oggier, F. Lustenberger, and E. Charbon, “A Virtual Keyboard Based on True-3D Optical Ranging,” in *British Machine Vision Conference 2005*, vol. 1, pp. 220 – 229, 2005.

[145] M. Profittlich, “Minority report,” *INSPECT*, vol. 6-7, 2009.

[146] M. Fischer and D. Henrich, “3d collision detection for industrial robots and unknown obstacles using multiple depth images,” in *Proceedings of German Workshop on Robotics 2009 (GWR 2009)*, 2009.

[147] C. Joochim and H. Roth, “Development of a 3d mapping using 2d/3d sensors for mobile robot locomotion,” in *Proceedings IEEE International Conference on Technologies for Practical Robot Applications (TePRA 2008)*, pp. 100–105, 2008.

[148] N. Ruangpayoongsak, *Development of autonomous features and indoor localization techniques for car-like mobile robots*. PhD thesis, University of Siegen, 2006.

[149] C. Netramai, O. Melnychuk, J. Chanin, and H. Roth, “Combining pmd and stereo camera for motion estimation of a mobile robot,” in *The 17th IFAC World Congress*, 2008.

[150] S. May, D. Droeschel, D. Holz, C. Wiesen, and S. Fuchs, “3D Pose Estimation and Mapping with Time-of-Flight Cameras,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on 3D Mapping*, (Nice, France), 2008.

[151] J. Weingarten, G. Gruener, and R. Siegwart, “A state-of-the-art 3d sensor for robot navigation,” in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, vol. 3, pp. 2155–2160, 2004.

[152] T. Hong, R. Bostelman, and R. Madhavan, “Obstacle Detection using a TOF Range Camera for Indoor AGV Navigation,” in *Proceedings Performance Metrics for Intelligent Systems Workshop (PerMIS) 2004*, 2004.

[153] R. Sheh, M. Kadous, C. Sammut, and B. Hengst, “Extracting terrain features from range images for autonomous random stepfield traversal,” in *Proceedings IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR 2007)*, pp. 1–6, 2007.

[154] J. Craighead, B. Day, and R. Murphy, “Evaluation of canesta’s range sensor technology for urban search and rescue and robot navigation,” in *Proceedings of the 2006 International Workshop for Safety, Security and Rescue Robotics (SSRR 2006)*, 2006.

[155] K.-D. Kuhnert and M. Stommel, “Fusion of stereo-camera and pmd-camera data for real-time suited precise 3d environment reconstruction,” in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4780–4785, 2006.

[156] M. Lindner and A. Kolb, “Data-Fusion of PMD-Based Distance-Information and High-Resolution RGB-Images,” in *Proceedings of the International IEEE Symposium on Signals, Circuits & Systems (ISSCS)*, vol. 1, pp. 121–124, 2007.

[157] A. Kolb, E. Barth, and R. Koch, “ToF-Sensors: New Dimensions for Realism and Interactivity,” in *Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on ToF Camera based Computer Vision (TOF-CV)*, pp. 1–6, 2008.

[158] T. Prasad, K. Hartmann, W. Wolfgang, S. Ghobadi, and A. Sluiter, “First steps in enhancing 3d vision technique using 2d/3d sensors,” in *11. Computer Vision Winter Workshop 2006* (V. Chum, O.Franc, ed.), pp. 82–86, Czech Society for Cybernetics and Informatics, 2006.

[159] K. Koser and R. Koch, “Perspectively invariant normal features,” in *Proceedings IEEE 11th International Conference on Computer Vision ICCV 2007*, pp. 1–8, 2007.

[160] M. Haker, M. Böhme, T. Martinetz, and E. Barth, “Scale-invariant range features for time-of-flight camera applications,” in *CVPR 2008 Workshop on Time-of-Flight-based Computer Vision (TOF-CV)*, pp. 1–6, IEEE Computer Society, 2008.

[161] D. Brown, “Close-range camera calibration,” *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.

[162] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *Proceedings Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 666–673, 1999.

[163] R. Y. Tsai, “A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses,” *Radiometry*, pp. 221–244, 1992.

[164] J. Heikkila and O. Silven, “A four-step camera calibration procedure with implicit image correction,” in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, 1997.

[165] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* The MIT Press, September 2005.

[166] A. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1403–1410, 2003.

[167] T. Barfoot, “Online visual motion estimation using fastslam with sift features,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 579–585, 2005.

[168] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[169] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (surf),” *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[170] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, “6d slam with approximate data association,” in *Proceedings of the International Conference on Advanced Robotics ICAR ’05*, pp. 242–249, 2005.

[171] P. J. Besl and N. D. Mckay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[172] J. Weingarten and R. Siegwart, “Ekf-based 3d slam for structured environment reconstruction,” in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3834–3839, 2005.

[173] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-based performance analysis of routing protocols for mobile ad-hoc networks,” in *MobiCom ’99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 195–206, ACM, 1999.

[174] F. Zeiger, N. Kraemer, and K. Schilling, “Commanding mobile robots via wireless ad-hoc networks - a comparison of four ad-hoc routing protocol implementations,” in *Proceedings IEEE International Conference on Robotics and Automation ICRA 2008*, pp. 590–595, 2008.

[175] T. Kahlmann, F. Remondino, and H. Ingensand, “Calibration for increased accuracy of the range imaging camera swissranger$^{\mathrm{TM}}$,” in *ISPRS Commission V Symposium ’Image Engineering and Vision Metrology’*, 2006.

[176] M. Lindner and A. Kolb, “Lateral and depth calibration of pmd-distance sensors,” in *International Symposium on Visual Computing (ISVC06)*, vol. 2, pp. 524–533, Springer, 2006.

[177] M. Wiedemann, “Einsatzmöglichkeiten einer PMD-Kamera zur Navigation eines mobilen Roboters,” Diplomarbeit, University of Würzburg - Department of Computer Science, 2007.

[178] M. Lindner and A. Kolb, “Calibration of the intensity-related distance error of the pmd tof-camera,” in *Intelligent Robots and Computer Vision XXV: Algorithms, Techniques, and Active Vision* (D. P. Casasent, E. L. Hall, and J. Röning, eds.), SPIE, 2007.

[179] M. Lindner, A. Kolb, and T. Ringbeck, “New Insights into the Calibration of TOF Sensors,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Workshop on ToF Camera based Computer Vision (TOF-CV)*, pp. 1–5, 2008.

[180] O. Lottner, A. Sluiter, K. Hartmann, and W. Weihs, “Movement artefacts in range images of time-of-flight cameras,” in *Proceedings International*

Symposium on Signals, Circuits and Systems (ISSCS 2007), vol. 1, pp. 1–4, 2007.

[181] S. May, B. Werner, H. Surmann, and K. Pervolz, “3d time-of-flight cameras for mobile robotics,” in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 790–795, 2006.

[182] B. Gerkey, R. T. Vaughan, and A. Howard, “The player/stage project: Tools for multi-robot and distributed sensor systems,” in *11th International Conference on Advanced Robotics (ICAR)*, (Coimbra, Portugal), pp. 317–323, 2003.

[183] F. W. Heger, L. M. Hiatt, B. Sellner, R. Simmons, and S. Singh, “Results in sliding autonomy for multi-robot spatial assembly,” in *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pp. 5–8, 2005.

[184] S. Lee, G. S. Sukhatme, G. J. Kim, and C.-M. Park, “Haptic teleoperation of a mobile robot: A user study,” *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 3, pp. 345–365, 2005.

[185] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *International Journal of Robotic Research*, vol. 5, no. 1, pp. 90–98, 1986.

[186] H. Choset, K. M. Lynch, G. A. K. Seth Hutchinson, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, June 2005.

[187] D. Fox, “Adapting the sample size in particle filters through kld-sampling,” *International Journal of Robotics Research*, vol. 22, 2003.

[188] S. Toth, “Implementierung von haptischen und Augmented Reality Navigationsassistenzfunktionen für mobile Roboter,” Diplomarbeit, University of Würzburg - Department of Computer Science, 2008.

[189] C. Beuthel and P. Jonsson, “Maintenance support through wearable computer and augmented reality technology,” in *Proceedings 1st IFAC Conference on Telematics Applications in Automation and Robotics*, pp. 511–516, 2001.

[190] P. Harmo, A. Halme, P. Virekoski, M. Halinen, and H. Pitkänen, “Virtual reality assisted telepresence system for remote control and maintenance,” in *1st IFAC-Conference on Mechatronic Systems*, pp. 1075–1080, 2000.

[191] C. Beuthel, F. Dai, E. Kolb, and E. Kruse, “3d visualisation for the monitoring and control of airport operation,” in *Proceedings 1st IFAC Conference on Telematics Applications in Automation and Robotics*, pp. 523–528, 2001.

[192] D. Schmalstieg, G. Reitmayr, and G. Hesina, “Distributed applications for collaborative three-dimensional workspaces,” *Presence: Teleoperators & Virtual Environments*, vol. 12, no. 1, pp. 52–67, 2003.

[193] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, “Surround-screen projection-based virtual reality: the design and implementation of the cave,” in *SIGGRAPH ’93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 135–142, ACM, 1993.

[194] E. Lamboray, S. Wurmlin, and M. Gross, “Real-time streaming of point-based 3d video,” in *Proceedings IEEE Virtual Reality*, pp. 91–281, 2004.

[195] J. Jacobson, M. L. Renard, J.-L. Lugrin, and M. Cavazza, “The caveut system: immersive entertainment based on a game engine,” in *ACE ’05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, (New York, NY, USA), pp. 184–187, ACM, 2005.

[196] D. Pratsch, “Mixed reality for efficient engineering collaboration: The areo-cave extension,” in *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing Vol II*, 2006.

[197] P. Milgram and F. Kishino, “A taxonomy of mixed reality visual displays,” *IEICE Transactions on Information Systems*, vol. E77-D, no. 12, pp. 1321–1329, 1994.

[198] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, “Recent advances in augmented reality,” *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, 2001.

[199] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, pp. 355 – 385, August 1997.

[200] P. Milgram, A. Rastogi, and J. Grodski, "Telerobotic control using augmented reality," in *IEEE International Workshop on Robot and Human Communication RO-MAN'95*, pp. 21–29, 1995.

[201] T. Pettersen, J. Pretlove, C. Skourup, T. Engedal, and T. Lokstad, "Augmented reality for programming industrial robots," in *Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 319–320, 2003.

[202] W. Piekarski, *Interactive 3D Modelling in Outdoor Augmented Reality Worlds*. PhD thesis, University of South Australia, Adelaide, 2004.

[203] S. Tejada, A. Cristina, P. Goodwyne, E. Normand, R. O'Hara, and S. Tarapore, "Virtual synergy: A human-robot interface for urban search and rescue," in *AAAI Mobile Robot Competition*, pp. 13–19, 2003.

[204] P. Milgram, S. Yin, and J. J. Grodski, "An augmented reality based teleoperation interface for unstructured environments," in *Proceedings American Nuclear Society (ANS) 7th Topical Meeting on Robotics and Remote Systems*, (Augusta, Georgia, USA), pp. 966–973, 1997.

[205] A. Halme, J. Suomela, and M. Savela, "Applying telepresence and augmented reality to teleoperate field robots," *Robotics and Autonomous Systems*, vol. 26, no. 2-3, pp. 117–125, 1999.

[206] S. W. Lawson, J. R. G. Pretlove, A. C. Wheeler, and G. A. Parker, "Augmented reality as a tool to aid the telerobotic exploration and characterization of remote environments," *Presence: Teleoperators & Virtual Environments*, vol. 11, no. 4, pp. 352–367, 2002.

[207] H. Keskinpala, J. Adams, and K. Kawamura, "Pda-based human-robotic interface," in *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3931–3936, 2003.

[208] J. Reitsema, W. Chun, T. W. Fong, and R. Stiles, "Team-centered virtual interactive presence for adjustable autonomy," in *AIAA Space 2005*, 2005.

[209] S. A. Green, X. Chen, M. Billinghurst, and J. G. Chase, “Collaborating with a mobile robot: An augmented reality multimodal interface,” in *17th IFAC World Congress*, 2008.

[210] M. Daily, Y. Cho, K. Martin, and D. Payton, “World embedded interfaces for human-robot interaction,” in *Proceedings 36th Annual Hawaii International Conference on System Sciences*, 2003.

[211] C. Nielsen, B. Ricks, M. Goodrich, D. Bruemmer, D. Few, and M. Few, “Snapshots for semantic maps,” in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2853–2858, 2004.

[212] D. Bruemmer, D. Few, M. Walton, R. Boring, J. Marble, C. Nielsen, and J. Garner, “"turn off the television!": Real-world robotic exploration experiments with a virtual 3-d display,” in *38th Annual Hawaii International Conference on System Sciences (HICSS ’05)*, 2005.

[213] J. Cooper and M. A. Goodrich, “Towards combining uav and sensor operator roles in uav-enabled visual search,” in *HRI ’08: Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, (New York, NY, USA), pp. 351–358, ACM, 2008.

[214] J. A. Atherton, B. Hardin, and M. A. Goodrich, “Coordinating a multi-agent team using a mulitple perspective interface paradigm,” in *Proceedings of the AAAI 2006 Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, 2006.

[215] J. Wang, M. Lewis, and S. Hughes, “Gravity-referenced attitude display for teleoperation of mobile robots,” in *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 48, pp. 2662–2666(5), 2004.

[216] M. Lewis and J. Wang, “Gravity-referenced attitude display for mobile robots: Making sense of what we see,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 1, pp. 94–105, 2007.

[217] Opengl, D. Shreiner, M. Woo, J. Neider, and T. Davis, *OpenGL(R) Programming Guide : The Official Guide to Learning OpenGL(R), Version 2 (5th Edition)*. Addison-Wesley Professional, August 2005.

[218] M. Segal, C. Korobkin, R. van Widenfelt, J. Foran, and P. Haeberli, "Fast shadows and lighting effects using texture mapping," in *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 249–252, ACM, 1992.

[219] R. P. Darken, K. Kempster, M. K. Kempster, and B. Peterson, "Effects of streaming video quality of service on spatial comprehension in a reconnaisance task," in *Proceedings of the Meeting of I/ITSEC*, 2001.

[220] F. Zeiger, M. Schmidt, and K. Schilling, "Remote experiments with mobile robot hardware via internet at limited link capacity," *IEEE Transactions on Industrial Electronics*, 2009.

[221] B. Ricks, C. Nielsen, and M. Goodrich, "Ecological displays for robot interaction: a new perspective," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, vol. 3, pp. 2855–2860, 2004.

[222] G. Hirzinger, K. Landzettel, and C. Fagerer, "Telerobotics with large time delays-the rotex experience," in *Proceedings IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94. (IROS '94)*, vol. 1, pp. 571–578, 1994.

[223] M. Sugimoto, G. Kagotani, H. Nii, N. Shiroma, F. Matsuno, and M. Inami, "Time follower's vision: a teleoperation interface with past images," *IEEE Computer Graphics and Applications*, vol. 25, no. 1, pp. 54–63, 2005.

[224] D. M. Stipanovic, G. Inalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.