A Holistic Approach for Highly Versatile Supervised Autonomous Urban Search and Rescue Robots

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vom Fachbereich Informatik der
Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

von

Dipl.-Inform. Stefan Kohlbrecher
(geboren in Hannover)

Referent:	Prof. Dr. Oskar von Stryk
Korreferent:	Prof. Dr. Daniele Nardi (Sapienza Università di Roma, Italien)
Tag der Einreichung:	20.10.2015
Tag der mündlichen Prüfung:	02.12.2015

D17
Darmstadt 2016

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AHRS	Attitude and Heading Reference System
API	Application Programming Interface
BDI	Boston Dynamics Incorporated
BICA	Best in Class Autonomy
BRICS	Best Practice in Robotics
CAD	Computer Aided Design
CPU	Central Processing Unit
CRASAR	Center for Robot-Assisted Search and Rescue
DARPA	Defense Advanced Research Projects Agency
DOF	Degrees of Freedom
DRC	DARPA Robotics Challenge
ESCHER	Electric Series Compliant Humanoid for Emergency Response
FlexBE	Flexible Behavior Engine
FOG	Fiber Optic Gyroscope
GFE	Government Furnished Equipment
GIS	Geographical Information System
GPU	Graphics Processing Unit
HECTOR	Heterogenous Cooperating Team of Robots
HRI	Human Robot Interaction
ICP	Iterative Closest Point
IDE	Integrated Development Environment
IK	Inverse Kinematics
IMU	Inertial Measurement Unit
LCM	Lightweight Communications and Marshalling
LIDAR	Light Detection and Ranging
LVDT	Linear Variable Differential Transformer
MAGIC	Multi Autonomous Ground-robotic International Challenge
MAV	Micro Air Vehicle
MEMS	Microelectromechanical Systems
NDT	Normal Distribution Transform
NIH	Not Invented Here
NIST	National Institute of Standards and Technology
OCS	Operator Control Station
ODE	Open Dynamics Engine
ORB	Oriented FAST and Oriented BRIEF
OSRF	Open Source Robotics Foundation
PCL	Point Cloud Library
PSM	Polar Scan Matching
PTAM	Parallel Tracking and Mapping
RANSAC	Random Sample Consensus
RFID	Radio-Frequency Identification

ROI	Region of Interest
ROS	Robot Operating System
RRL	Real Robot League
RSSI	Received Signal Strength Indicator
SA	Situation Awareness
SLAM	Simultaneous Localization and Mapping
slerp	Spherical Linear Interpolation
TEPCO	Tokyo Electric Power Company
ToF	Time of Flight
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UGV	Unmanned Ground Vehicle
UKF	Unscented Kalman Filter
URDF	Universal Robotic Description Format
USAR	Urban Search and Rescue
USB	Universal Serial Bus
USV	Unmanned Surface Vehicle
VRC	Virtual Robotics Challenge

Abstract

The use of robots for search and rescue tasks has tremendous potential to mitigate disasters and save lives of both of disaster victims and first responders. Moreover, robots actually deployed for disaster response are neither highly intelligent (i.e. autonomous) nor do they intelligently make best use of a human supervisor in the loop nor do multiple heterogeneous robots work together in an intelligent manner.

Within this thesis, a holistic systems-oriented approach together with a number of developments of key functionalities to increase robot autonomy for rescue robots systems are presented and evaluated. These are usable for a wide range of robotic systems and operation modes, from unmanned ground vehicles for exploration and surveillance to complex high degree of freedom humanoid robotic systems that can be used for remote manipulation in disaster environments and thus, may potentially serve as "avatars" for human response forces supervising them. Importantly, cooperation between one or more human supervisors and such robotic systems is demonstrated, increasing overall reliability and capability.

The presented approaches are evaluated in simulated and real world robot experiments and presented within the scope of some of the most competitive international competitions for rescue and disaster response robots in the world, the the DARPA Robotics Challenge competition and the RoboCup Rescue Robot League, demonstrating their performance beyond laboratory environments and representing an important milestone towards their future use in real disaster environments.

Kurzdarstellung

Die Nutzung von Robotern für Such- und Rettungsaufgaben hat großes Potential zum Schutz des Lebens von Opfern und Rettungskräften sowie zur Schadensminimierung im Katastrophenfall. Roboter, die bislang in Katastrophen eingesetzt werden, besitzen jedoch keine intelligenten (autonomen) Fähigkeiten. Bei ihnen wird darüber hinaus keine intelligente Zusammenarbeit zwischen Roboter und menschlichem Überwacher eingesetzt.

In dieser Arbeit wird ein systemorientierter ganzheitlicher Ansatz für verbesserte autonome Fähigkeiten von Rettungsrobotern vorgestellt und evaluiert. Teil hiervon sind mehrere entwickelte modulare funktionale Komponenten für die Lokalisierung und Kartierung, Suche nach Opfern und komplexe Manipulationsaufgaben. Sie sind nutzbar für ein breites Spektrum von Robotersystemen und Betriebsarten, von unbemannten Bodenfahrzeugen für Erkundung und Überwachung bis zu komplexen humanoiden Robotersystemen, die für Fernmanipulation im Katastrophenfall geeignet sind. Es wird gezeigt, dass die Kooperation zwischen einem oder mehreren menschlichen Überwachern und dem Roboter die Zuverlässigkeit und Fähigkeiten des Gesamtsystems verbessert.

Der dargestellte Ansatz wird in simulierten und realen Experimenten evaluiert. Hierbei wird auf zwei der konkurrenzstärksten Robotikwettbewerbe der Welt zurückgegriffen, die DARPA Robotics Challenge und den RoboCup Rescue Robot League Wettbewerb. Hiermit wird die Anwendbarkeit über Laborbedingungen hinaus gezeigt, was ein wichtiger Meilenstein auf dem Weg zur Anwendung in realen Katastrophen ist.

1 Introduction

1.1 Motivation

Robots are increasingly being used for dull, dirty and dangerous tasks outside the structured automation and laboratory environments that have been the predominant applications in the past. A highly relevant and most challenging application of robotic systems is their use for disaster response tasks characterized by a high degree of degradation and unstructuredness.

Disaster response spans a wide range of scenarios, including exploration of a disaster scene or manipulation of objects in the environment as well as underwater, ground and air applications and areas from hundreds of square kilometers to few square meters in size. In this thesis, ground robotic systems that enable remote mobile observation, exploration and manipulation are considered.

The application domain considered in this thesis is the use of robots for Urban Search and Rescue (USAR) and related tasks. Commonly defined as "the process of locating, extricating, and providing for the immediate medical treatment of casualties in collapsed structures" [55] USAR is a challenging application domain. The contributions presented in this thesis are enabling robots to be remotely and autonomously controlled and are as well relevant for other applications, like space and deep sea robotics.

An example for a complex and challenging disaster response application are the disaster mitigation operations at the Fukushima Daiichi nuclear plant after the tsunami in 2011. Here, robots had to negotiate irregular terrain while at the same time providing situational awareness to the operator(s). Mobile platforms and manipulators have been controlled via teleoperation. Experience from these operations as reported by robot operators showed that training effort slowed down the start of operations, while high workload and low situational awareness hindered and slowed ongoing operations with current robotic technology. Moreover, significant research and development needs for advanced intelligent robot technologies and flexible levels of interaction with remote human operators became obvious which, among others, lead to the initiation of the DARPA Robotics Challenge [145], [118].

The most notable difference between the usually well structured industrial or service settings, in which today's robots are mostly deployed and a USAR scenario is the much higher degree of uncertainty about the environment in the latter. Knowledge and detailed models about the environment and tasks to be solved are often available a priori in the first mentioned settings, making long-term planning feasible. However, in disaster scenarios, this is often not the case. Both the high degree of uncertainty about the environment and the complex nature of it thus make the USAR scenario a most challenging one.

Current state of the art control approaches for USAR robotic systems rely on teleoperation [134]. In contrast to this, improved assistance and autonomous capabilities promise to decrease remote human operator workload and increase the reliability of USAR robotic systems. Ideally, one or more human supervisor(s) should only be required to give provide task- or mission level goals to the robotic system, like pointing out areas to explore, or specifying objects to pick up, with the robotic system then performing most of the specified task autonomously.

In this thesis, the human operator or operators interacting with the robot are thus called supervisors to reflect the intended shift from pure teleoperation to supervisory control and interaction where possible.

These capabilities, however, will only be leveraged by remote human supervisors if they know and trust the capabilities of their robots. It is thus of high importance that these capabilities are determined with objectivity by testing and evaluation and not exaggerated. Operators have to be introduced to the robot capabilities and trained in their use so they are confident to use them. A failure to do so will result in users not trusting the robotic system and, in the worst case, refusal to use it at all.

Thus, it is apparent that robotic systems for USAR and related tasks need more intelligence and autonomy to either support the remote operator or even operate completely autonomously (for instance when a communication connection has been lost and the operator is unable to communicate with the robot).

When suitable autonomous capabilities are provided, the robot can increasingly be viewed a team member able to take over tasks that would otherwise be performed by a human operator. The aspect of teaming between remote humans and the robot thus becomes highly relevant with increasing autonomous abilities of robots. By viewing the remote supervisor and robot as a team working towards a common goal, new opportunities and chances arise for leveraging synergies, distributing workload and adaptively reacting to unfolding situations.

Being able to leverage the potential of such kind of human-robot teaming requires a holistic approach to USAR robotic systems. Once the robot is no more viewed as a "tool" that is used by the operator, but an agent that provides the capability to take over tasks, this has to be reflected in system design. Interfaces need to make capabilities readily accessible to both remote human supervisors and also onboard systems, ensuring all involved parties and components are notified of who is in control.

Current state of the art USAR robotic systems provide specialized capabilities for certain tasks. They are limited to a pre-defined mode of control (mostly teleoperation) and often require specialized infrastructure such as proprietary communication systems. In research, often highly impressive results for isolated problems are being obtained, but how and if integration into a complex and reliably performing robotic system would affect overall system performance for complex tasks is frequently not evaluated. This enforces the need for taking a holistic view and approach to the problem of supervised autonomous USAR robots.

Multiple different dimensions of capabilities determine performance of a USAR robotic system and determine the suitability for use in different USAR application scenarios:

- Consideration of the type and level of **physical interaction with the environment**. There are two distinct classes of such interaction: Interaction for the purpose of **locomotion** of the robot in the environment and potential interaction for the purpose of **manipulation** of the environment. While locomotion is common to all USAR robots and may be performed using wheeled, tracked, bipedal or other approaches, manipulation capabilities (and the need for it) vary significantly across robotic systems.

  Although often no explicit need for manipulation is included in system design for exploration tasks, manipulation may be mandatory to fulfill an exploration mission, e.g., to access rooms through opening doors or by removing debris blocking doors or stairways.

Complex manipulation abilities are needed to perform advanced disaster recovery tasks, e.g., for closing or opening valves, for attaching fire hoses, or for using tools for making repairs.

- Consideration of **perception** and leveraging synergies between robotic system and operator. For complex unstructured environments such as those prevalent in many USAR tasks, best performance can be achieved by distributing tasks among the team consisting of robot and operators. Thus, it is necessary to enable **perception and semantic scene understanding by the operator** by ensuring the necessary information is made available in a clearly understandable fashion. At the same time, **perception onboard the robot** should support and augment the operator as to reach highest performance and reliability.

- Consideration of **planning and navigation** capabilities. Depending on the nature of the environment as well as sensing and computing capabilities, the robotic system might be capable of automatically planning actions needed for locomotion and manipulation tasks.

- A **flexible level of robot autonomy** that can be adapted as needed in real-time during the mission. The complexity of USAR tasks means that full autonomy often is not feasible, while full teleoperation can be extremely demanding and error-prone for the operator.

- Consideration of **communication constraints**. USAR robots and operator control stations operating over wireless links have to be resilient to connectivity degradation or loss and provide graceful handling in the event that the quality of connectivity between operator and base station changes.

It should be noted that there are dependencies between these different dimensions, making it crucial to take a holistic approach to the overall system design. For instance, a direct teleoperation control approach can only be used reliably if the communication channel between robot and operator allows for it (by providing sufficient bandwidth and low latency).

In pure teleoperation mode, the operator is solely in charge of control, whereas in full autonomy mode the robot is fully in charge. The shared control modes in between these extremes are in many ways the most promising ones providing the potential to achieve the best performance, as synergies between human and robot capabilities can potentially be best leveraged. In such a case, the remote supervisor and robot effectively form a supervisor-robot team and cooperate in different modes of operation and corresponding levels of robot autonomy.

When taking the view of a hybrid control architecture [5], overall performance depends on the distribution of tasks among the human-robot team. The reactive layers on a short time scale are naturally better suited for robot (onboard) control, while deliberative layers for planning and task specification can benefit from help via human input the most.

A notably related challenge is the translation of human intent to robot action. Even when it is straightforward for a human operator to make sense of a situation and think about or say what a robot is supposed to perform, transferring this intent to the robot in a reliable and intuitive manner can be surprisingly hard, as complex motions in cartesian space might be required and motion constraints such as for balance control and collision avoidance have to be obeyed.

Figure 1.1 provides an overview of what solving USAR tasks entails. Depending on the actual status of the environment, tasks ranging from exploration or victim search to the required removal of debris might have to be solved to fulfill the overall goal of locating and extricating victims from a disaster site.

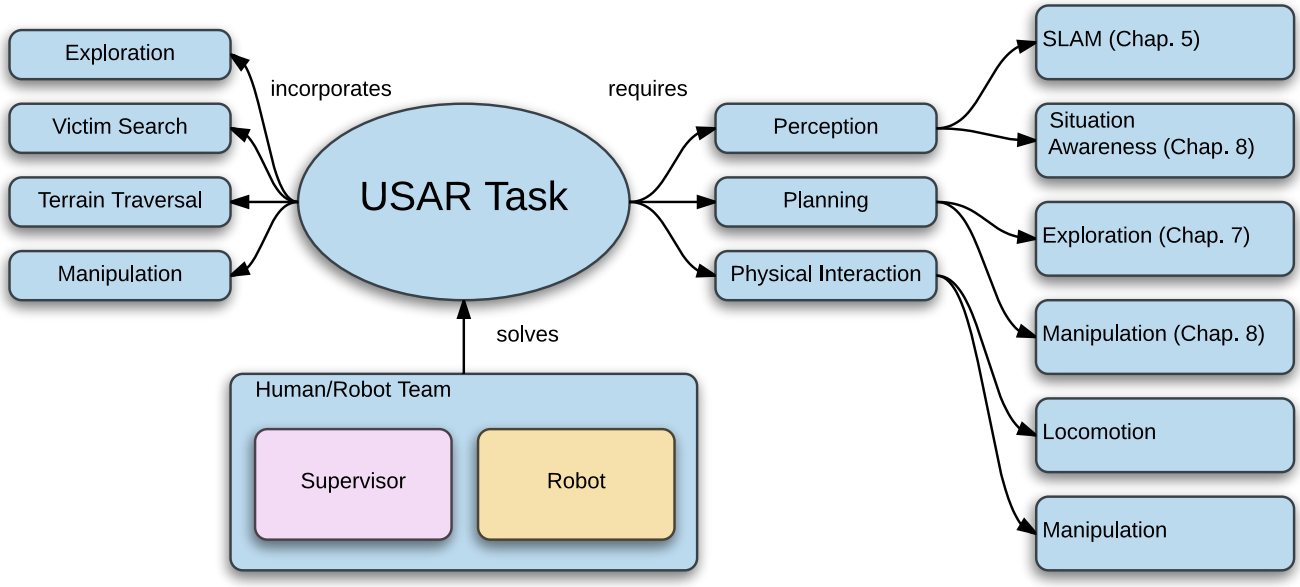


Figure 1.1: The USAR task (middle) entails a multitude of required subtasks (left). To be solved by the team of remote human supervisor and robot (bottom), the team has to provide multiple diverse capabilities (right).

To enable a robotic system to perform such versatile tasks, a wide range of capabilities is required as shown on the right of Figure 1.1. Physical interaction can only be performed by the robot, but others can be taken over or supported by a remote human supervisor. A more detailed discussion of this distribution of responsibility follows in Chapter 4.

An important aspect of sustainable advancement of robot capabilities and enabling validation of experimental results by other researchers is making research results available for replication of results as well as for the advancement of the field in general. For this reason, several contributions described in this thesis have been made available as open source software and have already found widespread use in the research community (like the methods described in Chapter 6).

1.2 Outline of this Work

As discussed in the preceding section, a holistic approach for intelligent USAR robotics promises to increase performance, versatility, and reliability. Gaps in available robotic technologies mean that the ability to leverage synergies between supervisor and robot is limited. This thesis contributes to filling gaps in several robot capabilities and investigates experimental evaluations with a holistic robotic systems approach.

For this purposes, it is focussed on three key aspects:

First, robot autonomous capabilities on the one hand and capabilities of the operator to supervise and support the robotic system on the other hand are strengthened; the goal is to support the full range of operating modes from teleoperation to full autonomy to increase versatility and robustness of the human-robot team.

Second, a wide range of robotic platforms and technologies can be used for USAR applications and depending on the task, some platforms might be better suited than others; the focus is thus

on supporting a wide range of different platforms to be able to adapt also to the needs of future applications.

Third, with increased autonomous capabilities, the resulting complexity of system design increases enormously and must be mastered; system architecture considerations are thus another focus of this thesis.

In Chapter 2 an overview of the start of the art and research in disaster response robots is provided, discussing current capabilities and capability gaps both from a field and a research perspective. Both applications in Benchmarking and evaluation scenarios for real disasters as well as efforts for simulating them for development purposes are discussed.

The highly diverse and versatile robots used in this thesis are discussed in Chapter 3. Both intelligent ground robotic systems focussing on autonomous exploration tasks and highly complex humanoid robots focussing on complex manipulation tasks are investigated, representing different ends of the potential spectrum of USAR and related disaster response tasks.

Overall system design considerations and the resulting architecture are discussed in Chapter 4. This topic is often not in the focus of academic research, but it is a key factor for robot performance in realistic applications when considering complex tasks.

Simulation of USAR robots and environments is discussed in Chapter 5. Simulation plays a very important role in the development of algorithms and software for complex robotic systems, accelerating research and development, improving reliability and performance and reducing cost through evaluating real robot software on simulated robot hardware. To this end, simulation tools for USAR environments based on the open source Gazebo simulator have been developed and are presented in Chapter 5. They allow comprehensive simulation of USAR scenarios and robots in simulation, including all sensor data and the same environments as used in real world experiments [99]. The contributions towards simulation allow for reproduction of USAR experiments using the Gazebo simulation framework.

State estimation for USAR robots with a focus on Simultaneous Localization and Mapping (SLAM) is discussed in Chapter 6. The ability to estimate the state of the robot and the environment is a pre-requisite for many higher level autonomous capabilities, as only reliable localization allows leveraging deliberative approaches beyond purely reactive behaviors. As the environment in USAR scenarios can differ significantly from a potentially previously known pre-disaster state, performing SLAM as opposed to pure localization is required. Performing SLAM also potentially increases system robustness due to the system being able to update the internal map state representations in case of unforeseen changes. The *hector_slam* [86] system has been developed to provide the required capability. Unlike prior approaches, it provides highly reliable LIDAR-based localization and mapping capabilities without requiring odometry data. It has been made available as open source and has become one of the most frequently used SLAM systems within the open source robotics community.

Improved assisted and autonomous navigation and victim search capabilities for harsh USAR environments are discussed in Chapter 7. For navigation in harsh environments such as those encountered after disasters, reliable methods are required that minimize the risk of robots getting stuck or otherwise failing during navigation. Contributions that improve the reliability of both autonomous and operator-guided navigation, as well as search for victims, are presented in Chapter 7 [83] [98] [82]. The contributions aim to provide more reliable and exhaustive navigation and victim search both in simulated and real scenarios.

Contributions towards enabling the use of humanoid robots in USAR environments under bandwidth constraints for complex tasks are discussed in Chapter 8. For tasks beyond monitor-

ing and exploration, the use of full-size humanoid robots as avatars for human response forces appears both highly promising and challenging. Described also in [81] and [85], approaches for versatile perception and manipulation through human-robot teaming are discussed. Unlike other approaches, reusability has already been shown through the use of components by multiple teams during the DARPA Robotics Challenge (DRC).

In Chapter 9, the contributions made are demonstrated and evaluated in experiments with a focus on comprehensive scenarios that provide several of the challenges also encountered in real disaster response.

In Chapter 10, the contributions made in this work are summarized and discussed. They are then discussed in the conclusions section. An outlook addresses some of the directions of further research and development towards creating highly versatile and intelligent USAR robots.

2 Background

With the contributions of this work focussed on USAR and disaster response robots, first, an overview of robotic disaster response is provided. This encompasses a review of the state of the art as well as of internationally acknowledged competitions that (among others) have been introduced to advance the state of the art.

2.1 Robots for Disaster Response

In real-world disaster response, robots are mainly used for monitoring purposes, with fixed wing or rotary wing Unmanned Aerial Vehicle (UAV) systems used for large scale reconnaissance and small Unmanned Ground Vehicle (UGV) systems for the exploration small spaces in more localized scenarios like building collapses. Sea-borne Unmanned Surface Vehicle (USV) systems and scenarios are beyond the scope of this work. A comprehensive overview of the field is available in Chapter 50 of [143] and in [106]. Considering the taxonomy presented there, this work concentrates on *ground* rescue robots systems for *USAR* tasks that are characterized by relatively small but highly challenging environments containing victims in voids and potentially requiring manipulation capabilities for tasks such as clearing debris or operating tools.

UGV systems are nowadays mostly teleoperated via onboard cameras, which puts tremendous stress [133] on operators and makes them the most important factor for success or failure of robot operations. There is a gap between approaches used within research and those used in real disaster scenarios. Understanding the reasons for this significant discrepancy is very important.

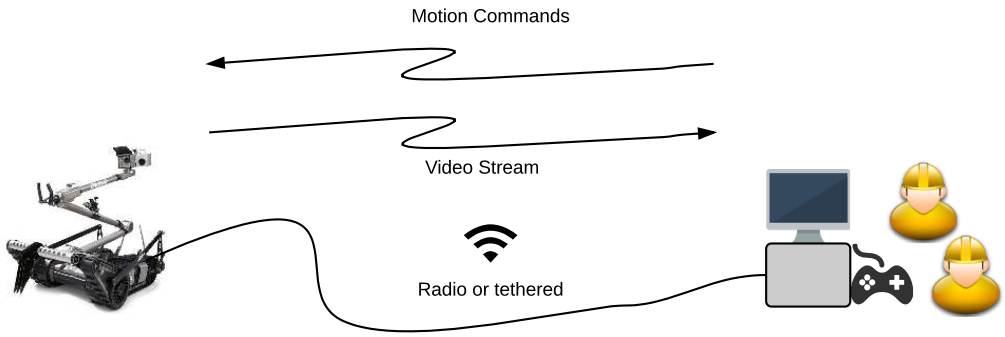
As disasters are chaotic events, very little prior knowledge of the state of the environment can be assumed after a disaster struck. This is in stark contrast to applications where semantic knowledge about the environment allows the realization of autonomy based on prior information. Service robots operating in kitchens or factories are examples where prior domain knowledge can be leveraged to achieve highly autonomous behaviors [14].

In USAR applications, systems have to provide very high reliability and robustness while being challenged by harsh terrain and environmental factors such as humidity, rain, dust, and others. In research applications, robots are often operated by (robot) domain experts. These domain experts frequently have a good grasp of robot capabilities, although over- or underestimation of those can inhibit performance. In contrast to research applications, robots are nearly always operated by first responders or more generally, disaster domain experts when used in real-world disasters. This makes it very important that the employed robotic systems are easy and intuitive to use.

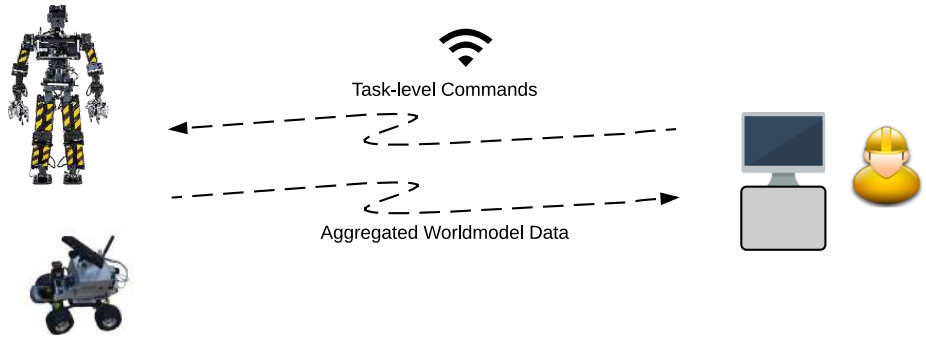
2.1.1 State of the Art

A distinction has to be made between the state of the art for use of robots for real disaster mitigation and their use in research. Use of robots in real disaster response has been limited and so far while providing a useful tool for teleoperated exploration of disaster sites, no human lives have been saved directly through the use of robotic disaster response to the author's knowledge.

In Figure 2.1a), the typical state of the art field deployment of robots for USAR is shown. There are multiple operators responsible for robot operation and the system is teleoperated via



(a)



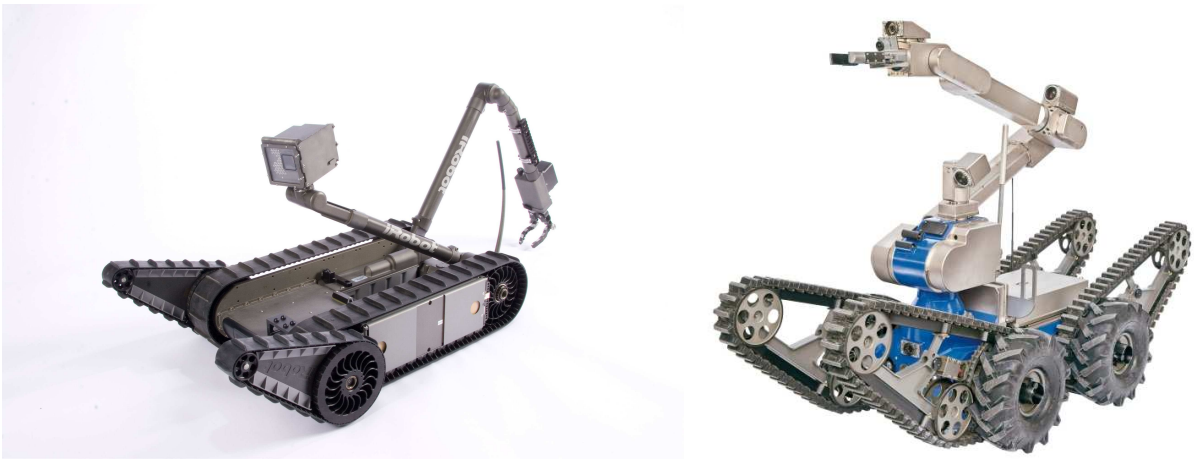
(b)

Figure 2.1: Current state of the art and future goal for interaction between operators and robots: (a): Typical state of the art approach with multiple humands teleoperating a single robot. (b): Future approaches might focus on task-level supervision. Leveraging autonomy, communication might be intermittent (as indicated by the dashed arrows).

a radio or tether link, with motion commands passed to the robot and video streamed from the robot to the operators (who cannot be called supervisors in this case). In contrast to this, an approach allowing for task-level supervision as shown in 2.1b) is a topic of research, but not a fielded capability so far.

A particular challenge is decreasing the size and increasing the reliability of systems that have been developed by the research community so they are useful in highly challenging real world environments that exhibit environmental factors such as small openings, loose and compliant ground, dust, smoke, humidity and others.

It can be expected that advances in manufacturing and the resulting miniaturization of components will solve some size and weight issues in the future, but robustness to environmental factors for tasks like sensor data processing or locomotion remains challenging software problem despite advances in technology.



(a) (b)

Figure 2.2: Typical response robots: (a): iRobot Packbot (Source: iRobot) (b): Telerob Telemax (Source: Cobham Mission Systems)

2.1.2 Unmanned Ground Vehicle Platforms

For USAR and response robot applications, tracked vehicles in the 30kg - 100kg range are nowadays the most commonly used platforms. Figure 2.2 shows two examples. With the ability to turn in place, low ground pressure and high mobility, vehicles of this class can be controlled and teleoperated with relative ease. For increased mobility, these tracked platforms are often capable of varying their track geometry using so-called flippers, small tracks that can be rotated to allow adjustments of the track geometry according to environment conditions. For the ability to climb stairs, using flippers is often required.

Examples of other, less conventional platforms include the RHex robot using six rotating "legs" [1]. While demonstrated to have high mobility, it was never adopted beyond research applications. There are multiple examples of other non-conventional ground robots showing promising mobility, but it appears to be difficult to develop response robots to maturity and convince response forces to use them.

2.1.3 Constrained Communication

Communication conditions in USAR can vary tremendously depending on the situation encountered and can range from very high bandwidth and low latency to conditions where no wireless communication is possible at all. Examples of such conditions are mining accidents or the Fukushima Daiichi disaster in described in detail in the following section. Response robots are frequently fitted with analog or IP camera systems that provide imagery also in adverse connectivity conditions, albeit with greatly reduced quality.

An alternative to wireless communication is using tethers and routing communication through cable. Tethering can be advantageous for the communication capability it provides, as well as the ability to potentially retrieve the robot (if it is small). However, tethers can severely limit the mobility of robots as entanglement in the environment has to be avoided.

2.1.4 Real Disaster Response Examples

General Disaster Response

The Center for Robot-Assisted Search and Rescue (CRASAR) used their rescue robots for operations in multiple real disasters. Documented in multiple publications, these operations provide unique insight in the requirements for successful use of robots in disasters. This includes disasters such 9/11 [33], [104], the La Conchita mudslide [109] and lessons learned from other responses [107] A small size of robots is identified as an important property in many applications as real disasters such as building collapses often only provide small voids for robots to traverse. The use of tethers is considered highly useful, providing both means to transmit video data and the ability to retrieve robots if they become stuck. Due to dust build-up and low lighting, the recognition of buried victims is very hard to perform even for trained human operators. The use of robots in partial or total building collapses thus is extremely challenging, as the small required size of robots and extremely challenging environment conditions make it difficult to achieve situational awareness for operators. Sensors are directly affected by dust and can only be mounted low due to vehicle size.

Fukushima Daiichi Response

The Fukushima nuclear plant disaster provided unique insight into current capabilities of robots used for mitigation of man-made high-risk disasters. Robots could not be used for the disaster *rescue* phase of disaster response, as no systems were deployable in the required short time. In the *recovery* phase, multiple systems were used with success. Initially, response robots made available by irobot were used, weeks after the disaster incident. After multiple months, Quince robots developed at Tohoku university also in the scope of the RoboCup Rescue competition were used to explore the interior of the reactor buildings [164]. While providing highly useful information about the internal state of the damaged reactor buildings, preparation of robotic systems took a long time. Preparation time accounted for both soft- and hardware modifications, but also for the required training of Tokyo Electric Power Company (TEPCO) operators. Only operators from TEPCO were allowed to operate the systems inside the reactor building, so significant time was and had to be invested in this training.

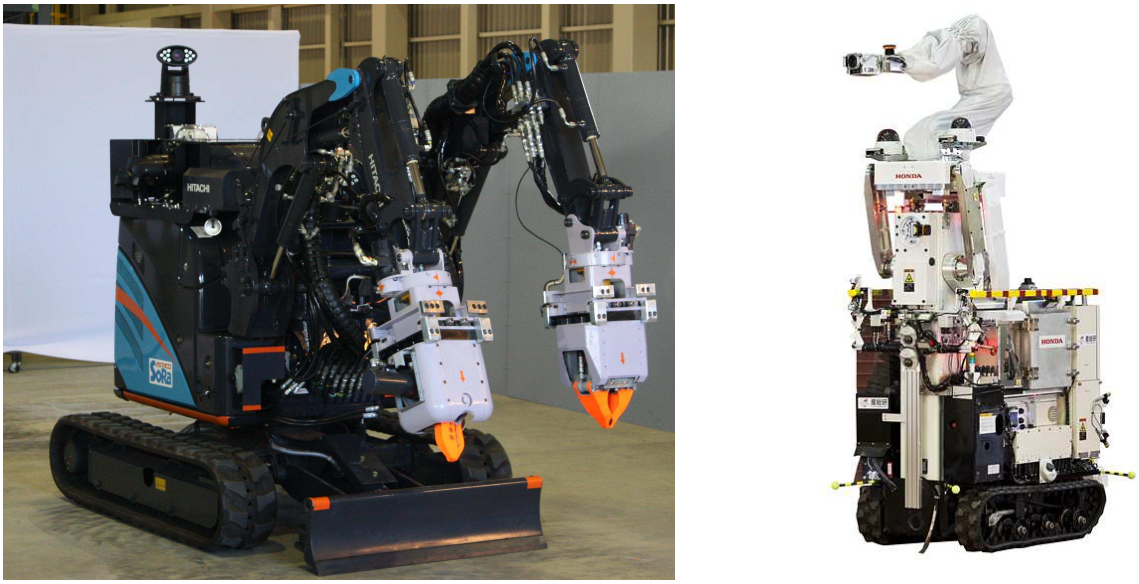
The requirements of post-disaster clean-up of the nuclear plant also prompted the development of multiple robotic systems designed for this task by Japanese companies. These systems are specialized designs such as the Hitachi ASTACO-SoRa system for debris removal or the Honda High-Access survey Robot that is used for inspection and manipulation of high access areas.

Mirandola Earthquake Damage Assessment

At earthquake-hit town Mirandola, robotic systems developed within the NIFTi project [1] were used for damage assessment of a partially collapsed church [88]. While not a USAR application, this operation demonstrated that advanced sensing capabilities beyond the use of analog video often can be very beneficial, with the used ground robot providing LIDAR-based data of the inside of the partially collapsed church.

The Mirandola response is an example of an operation that provided a glimpse into the future in the sense that advanced mapping and planning techniques were used to create maps of the

[1] http://www.nifti.eu/mission



(a) (b)

Figure 2.3: Specialized robots used in Fukushima: (a): Hitachi ASTACO_SoRa robot for debris removal (Source: TEPCO) (b): Honda High-Access survey Robot for inspection and manipulation in high access areas (Source: Honda).

explored environments. The situation at the site with no human victims in danger allowed for testing research systems without major risk.

2.2 Benchmarking

The ability to benchmark performance is highly important for USAR robots. Benchmarking them is highly challenging, as realistic response scenarios can only be emulated with significantly more effort than many other robotic application scenarios. For instance, for testing service robots, often a kitchen and a few household items are required. In contrast to this, simulating a USAR scenario involves rough terrain, simulated victims, constrained communications and potentially other factors such as obscurants.

2.2.1 NIST Standard Test Methods for Response Robots

The National Institute of Standards and Technology (NIST) in the US leads the most comprehensive initiative for benchmarking response robot performance in the world [97]. Based on well documented standardized methods, response robots are tested for specific capabilities such as mobility or inspection. Test setups are designed with a focus on reproducibility and ease of manufacturing, enabling easy re-creation of test setups in different locations. Based on technology advances and emerging requirements the test setups are refined and updated and continuously reviewed [103], [105].

(a) (b)

Figure 2.4: Kobe earthquake building collapses: (a): Total collapse of residential building (b): Partial collapse of commercial building

2.2.2 RoboCup Rescue

Motivated by the 1996 Kobe earthquake (Figure 2.4), the RoboCup Rescue competition aims to foster research in robotic systems for Urban Search and Rescue.

Multiple sub-leagues focus on different aspects of disaster response, with the simulation leagues simulating large-scale *strategic* disaster [78] response and *tactical* multi-robot response and the Real Robot League (RRL) providing a simulated disaster scenario for real robots.

The RoboCup Real Robot Rescue League is supported by NIST and uses Standard Test Methods developed for response robots for repeatable testing of robot capabilities. Often, test methods are first introduced in the RoboCup competition and then evolve to become standard test methods [70], [140]. A recent overview of the state of the league is available in [114].

It should be noted that arena design represents a trade-off between the unpredictability of real disaster situations and the desired repeatability of test results. Using 1.2m side length tiles, the RoboCup Rescue arena focusses on medium scale scenario and not the small search space scenarios that are prevalent in building collapses. A "ant farm" scenario simulating such collapses is available but has not been the focus of the competition so far.

While robots used in the Rescue Robot League are research platforms, there is significant impact beyond the competition and results in academia. Quoting Satoshi Tadokoro, one of the original founders of the competition:

> It had significant contribution to advancement of robotics for disaster response and recovery. For example, Quince would not have been able to survey the nuclear reactor buildings of Fukushima-Daiichi Nuclear Power Plant, if our team had not participated in the worldwide competitions.

In [108] potential tasks for the RoboCup Rescue Robot League competition based on experience with real disasters are presented. The proposed tasks are *Reconnaissance and Site Assessment*, *Rescuer Safety*, *Victim Detection* and *Mapping and Characterizing Structure*. All four tasks have been and continue to be part of the RoboCup Rescue competition.

(a) (b)

Figure 2.5: Example of a RoboCup test scenario modeled after real disaster lessons learned: (a): View from Quince robot blocked by debris on ladder from further exploration of reactor building at Fukushima (Source: TEPCO) (b): Emulation of a similar obstacle in the RoboCup Rescue arena at RoboCup 2013.

Scenario Description

The RoboCup Rescue competition aims at advancing the field in relative to orthogonal directions like robot mobility and autonomy. For this reason, both autonomous and teleoperated systems are allowed in the competition. The increased difficulty of autonomous operation is considered in the rules and there are different arena types used in the competition, which differ in scoring. The most important types are:

- *Yellow Arena.* Here, robots can score points for finding victims only in fully autonomous operation. The terrain features roll/pitch ramps but is generally not as challenging as the other arena parts
- *Orange Arena.* This part of the arena can only be reached by driving through most of the yellow arena. Victims can be scored using teleoperation here and the terrain features more difficult terrain like crossed roll/pitch ramps
- *Red Arena.* This part features the most difficult terrain, including stairs, step fields, and steep ramps. Teleoperation is allowed here, too.
- *Blue Arena.* This part features tasks that require manipulation, such as picking up and placing objects or turning a valve. Teleoperation is allowed here, too.

2.2.3 DARPA Robotics Challenge

The most cited motivation for the DRC is the Fukushima Daiichi nuclear disaster. It showed that robotic systems with sophisticated manipulation and locomotion capabilities could have

significant impact in man-made disasters, as they could explore and perform manipulation even in conditions that are too dangerous or impossible to reach for human responders. As detailed above, robots with these capabilities were not available shortly after the Fukushima disaster hower. This prompted the Defense Advanced Research Projects Agency (DARPA) to initiate the DRC program, with the goal of accelerating the development of highly capable robots for disaster response. Locomotion and manipulation capabilities in degraded environments originally intended for humans were identified as a major challenge for Fukushima-type scenarios, so the DRC focussed on robots that can operate in such scenarios. While not absolutely required, an anthropomorphic shape is advantageous, as tools made for humans can be used easily.

2.2.4 euRathlon

The euRathlon competition is based on an initiative by the EU for improving Search and Rescue robots [135]. Similar to the DRC a scenario based on the Fukushima disaster is used for the euRathlon "Grand Challenge". The focus is less on manipulation and more on search and observation, however. Marine, land and aerial vehicles might be used by teams, also simultaneously.

3 Robots Used in this Thesis

This chapter provides an overview of the robotic systems that have been used to obtain the results presented in this work.

3.1 Hector Unmanned Ground Vehicles

Team Heterogenous Cooperating Team of Robots (HECTOR) is part of research training group GRK1362 "Cooperative and Adaptive Monitoring in Mixed-Mode Environments". Developing robotic systems for the use in disasters as mobile sensor nodes, the team uses the RoboCup Rescue competition as a benchmarking scenario. Multiple robotic systems with complementary abilities are used and described in the following.

Overview

Both wheeled and tracked platforms are used for the work presented in this thesis. The vehicles share a common computing and sensor setup that can be mounted on either a wheeled ("Hector UGV") or tracked ("Obelix UGV", Taurob Tracker) platforms.

Sensors

Wheel/Track Encoders:: To measure the translational and rotational speed of vehicles, all vehicles are equipped with encoders measuring wheel or track motion. Internally, this odometry data is used for low level (velocity) control.

Laser Scanner:: The vehicle is equipped with a Hokuyo UTM30-LX Light Detection and Ranging (LIDAR) system. It is mounted on a roll/tilt unit at the front of the autonomy box and is mainly used for 2D mapping. The LIDAR system can be stabilized to stay close to the intended scan plane regardless of vehicle attitude.
Optionally, a Hokuyo URG-04LX or SICK TiM300 LIDAR can be mounted on the back of the vehicle. Pointing towards the ceiling, this LIDAR allows the acquisition of additional 3D data.

RGB-D Camera:: An RGB-D camera is used for environment perception tasks like traversable terrain detection, 3D mapping and also for victim verification. The sensor is mounted on the pan/tilt unit that is also used for the thermal camera. Originally, a Microsoft Kinect camera was used, but it has been replaced with an Asus Xtion Pro Live sensor for the smaller size and lower mass.

Ultrasound Range Finders:: Additionally to the LIDAR, a set of ultrasound range finders mounted at the rear of the vehicle optionally allows for reactive autonomous collision avoidance when moving backward, as the LIDAR only covers a 270 degrees field of view.

Inertial Measurement Unit:: To measure the attitude of the platform, vehicles are equipped with a 6 Degrees of Freedom (DOF) Analog Devices ADIS16350 Inertial Measurement Unit (IMU) or similar IMU capabale of measuring accelerations and angular rates.

Vehicle Platforms

Hector UGV: The Hector UGV (Figure 3.1, right) is the first platform used by Team Hector. It is based on a Ackermann steered Kyosho Twin-Force RC car. For the same reasons laid

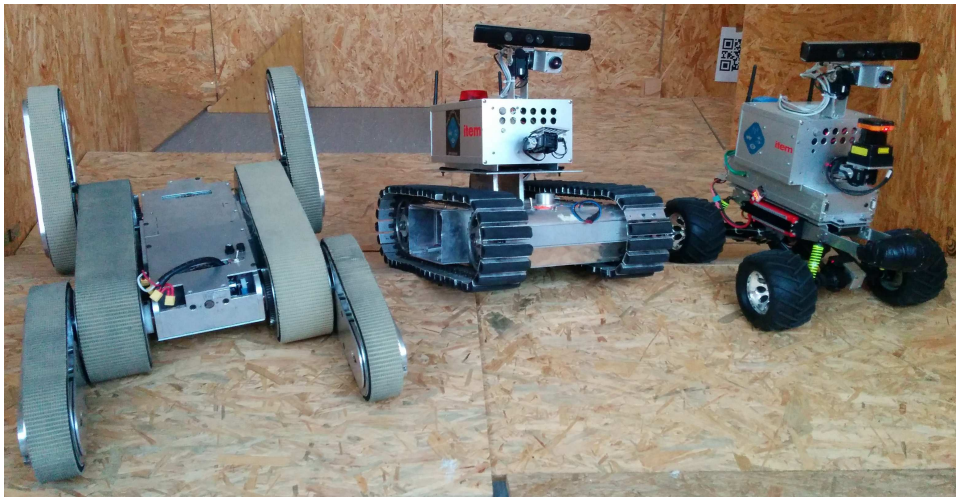


Figure 3.1: Unmanned ground vehicles used by Team Hector: Highly mobile tracked platform with Flippers, tracked platform with adapted autonomy box, Hector UGV.

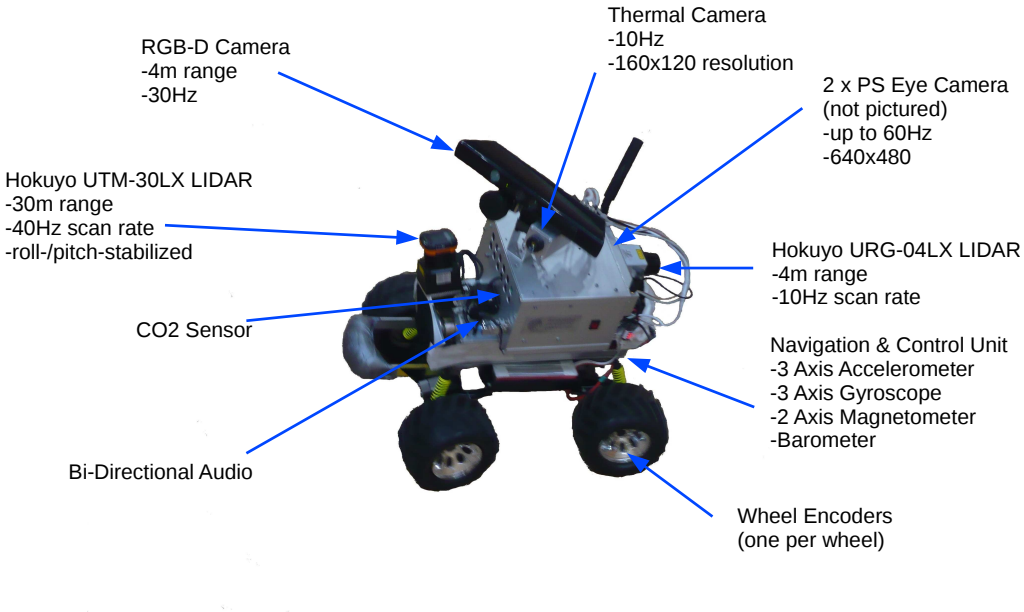


Figure 3.2: Sensors used shown on the Hector UGV system. The autonomy box on the robot is modular and the same sensor suite can thus easily be used also on the tracked robots used within this thesis.

UGV Chassis	Hector	Obelix	Tracker
Mass	8kg	60kg	35kg
Actuation	DC motors	DC motors	Brushless motors
Steering	Ackermann, four-wheel steering	Differential (tracked)	Differential (tracked)
Maximum velocity	1 km/h	0.5 km/h	3 km/h
Variable geometry	N/A	Flipper front/back	Flipper front
Payload	4kg	20kg	40 kg

Table 3.1: Comparison of UGV platforms. The Hector UGV platform is by far the most lightweight vehicle. It is easily man-packable and has good mobility in terrain not exceeding steps of 5cm height. For more challenging terrain, the two tracked platforms are better suited. With variable geometry, they are capable of negotiating steep inclines and ramps.

out in [121], the system is based on this commercially available "toy" platform, as this significantly reduces cost while providing good robustness and reliability. The system was used for participation in the RRL since 2009 and continuously upgraded based on lessons learned during evaluations and competitions. For instance, to increase maneuverability, all-wheel steering based on stronger servos was added.

The Hector UGV is well suited for performing exploration tasks in terrain with steps no higher than 6cm. The car-like suspension system and low weight provide for good mobility and make the platform a capable autonomous system, as long as terrain is not too harsh.

Obelix UGV: The "Obelix" UGV (nicknamed so because of the larger size and weight compared to robots previously used by the team) is a custom built platform developed and built in cooperation with Rajamangala University of Technology Phra Nakhon (RMUTP). Equipped with front and back flipper arms, the platform provides improved mobility capabilities compared to the Hector UGV used previously.

Taurob Tracker: The Taurob Tracker is a tracked platform by Austrian company Taurob [1] [17] . Already in use with the fire department of Vienna, it is a mature platform that can be ATEX certified and provides excellent control and mobility.

Hardware Modularity

The complete hardware structure of the Hector vehicle setup is shown in Figure 3.3. To achieve interoperability between different vehicle platforms, the autonomy box is connected to the vehicle base using a single Universal Serial Bus (USB) connection (or possibly Ethernet in the future). The sensor and software setup can thus be largely (if not completely) agnostic to the underlying platform employed for locomotion. The autonomy box is equipped with a state-of-the-art Intel Core i7 mobile Central Processing Unit (CPU) and an optional high-performance Graphics Processing Unit (GPU). It serves as the main control system.

The separation of vehicle base and autonomy box, even on the hardware layer, simplifies independent testing and offers a high degree of flexibility. The perception system can easily be mounted on other robots or used as a separate instrument for the evaluation of test scenarios.

[1] http://taurob.com/

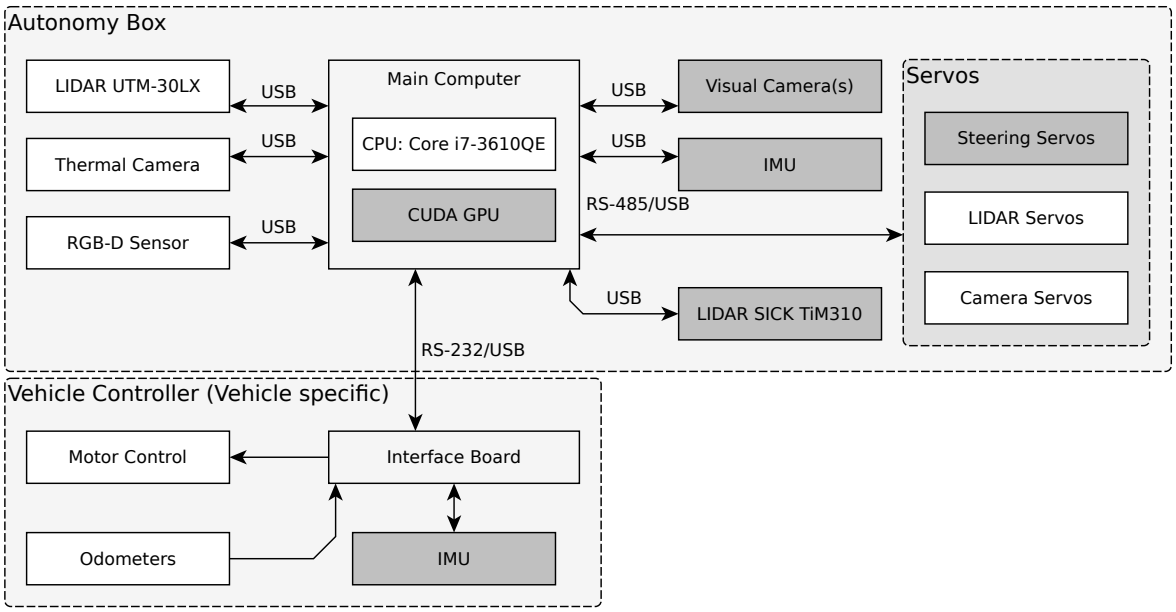


Figure 3.3: Structure of hardware components. Dark grey boxes indicate optional components with the capability to quickly add/remove them as needed.

Both autonomy box and the wheeled robot base can and have been used in various indoor and outdoor scenarios as a flexible and lightweight research platform.

3.2 Atlas

Atlas is an anthropomorphic robotic system developed by Boston Dynamics Incorporated (BDI). The robot is hydraulically actuated and has 28 DOF. Depending on the version it weighs 150kg to 180kg and can be considered one of the most advanced anthropomorphic robotic systems in the world. The robot was designed and built by BDI as the Government Furnished Equipment (GFE) robot for the DRC. After the Virtual Robotics Challenge part of the DRC, six Atlas robots were provided to the six highest scoring teams in the competition. Some of the contributions presented in this thesis have been investigated while working with DRC Team ViGIR and the Atlas robot the team received as a result of scoring the sixth place in the Virtual Robotics Challenge (VRC) competition.

Unlike most full-size humanoid systems used in research, Atlas is hydraulically actuated. While this actuation principle provides high bandwidth both in achievable torque and velocities, hydraulics control poses significant challenges due to friction and stiction effects [19] in the hydraulic joints. The high performance of the robot also means that much stricter safety precautions than with most other systems have to be observed. For instance, persons are not allowed to come closer than 10 feet when the hydraulic pump is running.

Over the course of the DRC competition, the Atlas robot was modified by BDI based on evolving requirements and lessons learned during operating the robot. Initially equipped with 6DOF arms that provided only poor manipulability in the relevant workspace in front of the robot, the final version of Atlas is equipped with 7 DOF arms that provide much-improved manipulation capabilities compared to prior versions. For these new arms, the four joints farthest away from the end effector are still hydraulically actuated, while the three lower joints are electrically ac-

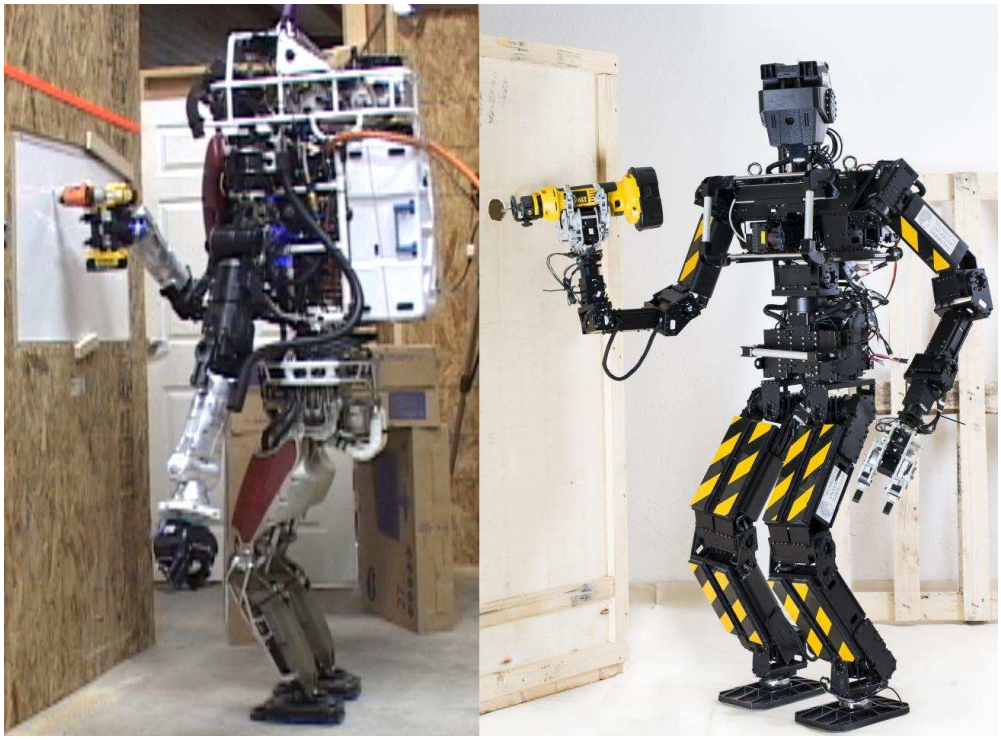


Figure 3.4: Two of the robotic systems used: The Boston Dynamics Atlas robot and the Robotis THOR-MANG robot

tuated. With the requirement for untethered operation, the robot also received a new hydraulic pump and battery pack to allow for at least 1 hour of untethered operation.

The robot is equipped with a high-performance KVH 1750 IMU using Fiber Optic Gyroscope (FOG) technology. The data from this IMU system can be fused with leg kinematics to estimate robot odometry with high accuracy.

For external sensing, Atlas is equipped with a Multisense SL sensor head, featuring both a stereo camera and a continuously spinning LIDAR. This sensor is supplemented with two additional fisheye cameras looking sideways for providing situation awareness to the operator(s).

3.3 THOR-MANG

THOR-MANG is an anthropomorphic robotic system developed by Robotis Co. in South Korea. The robot has 30 DOF and is electrically actuated. It is smaller and weighs significantly less than Atlas, while otherwise providing similar capabilities in that it can walk on two legs in 3D terrain, has two 7 DOF arms and a sensor suite consisting of RGB and LIDAR sensors.

Dynamixel Pro type motors are used as actuators for all joints. These servo motors provide high torque and for instance allow the robot to kneel down and get back up again, a task that is notoriously difficult to achieve for full-size humanoid systems. They feature built-in position control with highly accurate position sensing, but only allow for current-based torque control. This puts a natural limit on the quality of force estimation and control. The robot is thus primarily position-controlled.

The robot is equipped with 6 DOF force/torque sensors in the wrist and ankles. This allows the use of admittance control-based compliant control approaches despite using position control for the joints.

Robot Type	Atlas	THOR-MANG
Mass	150kg	50kg
Height	1.88m	1.5m
Number of DOF	28	30
Control Bus	CAN	RS-485
Control API bandwidth	1 kHz	125Hz
Control internal bandwidth	1 kHz	>2kHz
Actuation	Hydraulic	Electric
Control Interfaces	Position Velocity Torque (pressure diff based)	Position Velocity Torque (current based)
Force Sensing	6DOF wrist 3DOF ankle Pressure diff per joint	6DOF wrist 6DOF ankle Current per joint
Onboard Computation	3 Intel Core i7 PCs QNX RTOS based control PC	2 AMD 1.6GHz
Connectivity	10 Gbps fiber optic Ethernet	1 Gbps Ethernet WiFi 802.11n
External Sensing	Hokuyo UTM-30LX LIDAR FPGA stereo system 2 Fisheye cameras	2 Hokuyo UTM-30LX LIDAR Monocular camera
Walk Engine	QP based capture point	QP based capture point

Table 3.2: Comparison table of Atlas and THOR-MANG robotic systems. Both systems are anthropomorphic humanoid robots. While Atlas has very high performance, this is paid for by high weight, cost and operational footprint. For many applications, THOR-MANG provides similar capabilities while being less involved to handle.

The robot has been demonstrated to be capable of performing the qualification tasks required for entering the DARPA Robotics Challenge Finals by multiple participating teams.

The robot is equipped with a Microstrain IMU used as an Attitude and Heading Reference System (AHRS). Depending on configuration, the robot is equipped with up to two Hokuyo UTM-30LX EW LIDAR sensors and an RGB camera. It is thus capable of providing similar, if not the same type of data that the sensors on Atlas provide.

3.4 Hector Centaur

Hector Centaur is a prototype robot used for the demonstration of contributions within this thesis in simulation. It uses the previously presented "Obelix" robot UGV platform and the upper body of the THOR-MANG robot. Additionally, it is equipped with a stabilized LIDAR system for performing SLAM and a thermal camera to enable autonomous search for simulated victims.

Combining the robustness of tracked locomotion and navigation with the versatility of a humanoid upper body allows demonstrating the whole range of contributions of this thesis within a single robotic system.

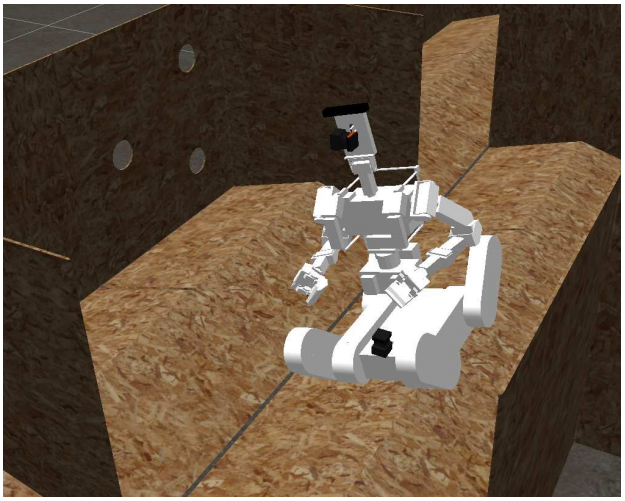


Figure 3.5: The Hector Centaur vehicle used for demonstration purposes in Gazebo simulation.

So far used only for experiments in simulation, the vehicle might serve as an important evaluation platform towards the development of a new USAR robotic system combining some the advantages of both tracked and humanoid systems.

4 System Architecture

System architecture considerations are often not discussed in detail in academic research and publications as the focus is on novel approaches for specific subproblems and selected robot capabilities. As previously discussed, the complexity of supervised autonomous USAR robots warrants a closer look at this topic. Application to complex tasks, as those discussed in this thesis, can only be demonstrated and evaluated using comprehensive robotic systems. The requirements, architecture and design for such systems are discussed in this chapter.

As noted in [35], "Architecture is design but not all design is architectural.". There is a distinction between architectural design, which is concerned with software architecture, and detailed design, which is concerned with the design of components; a detailed discussion is available in [45].

For overall system performance, the design aspect is as important as the components making up the system. Conversely, good design cannot overcome weak performance of system components. Architectural aspects are thus discussed in this chapter, while later chapters provide further insight into the design of components and functionality that is required for enabling supervisor-robot teaming in complex USAR scenarios using different types of robots.

4.1 Requirements for a Holistic Approach

The holistic approach towards supervised autonomous robots for USAR in this thesis incorporates three major aspects as illustrated in Figure 4.1.

Flexible Level of Interaction

Autonomous capabilities can reduce the workload of human operators and provide assistance functionality that can increase reliability and effectiveness of the human-robot team. In some cases, autonomy might even be required if connectivity between supervisor and robot cannot be guaranteed.

However, autonomy can not yet be guaranteed work reliably under all circumstances and for highly complex tasks. Recent experience such as from the DRC suggests that the ability to fall back to more human supervisor-centric control modalities can increase overall system reliability significantly.

Support for Heterogeneous Robot Types

Exact requirements for future USAR situations and other disasters cannot be predicted accurately. Additionally, advances in research and technology mean that research has to be adjusted to leverage new capabilities. For this reason, a wide range of robots with very heterogeneous capabilities should be considered for USAR and related scenarios. These capabilities can range from exploration, observation and victim search to highly complex manipulation.

System Architecture

The system architecture has to support heterogeneous robot types and flexible levels of interaction. To facilitate dissemination and reproduction of results and approaches by other

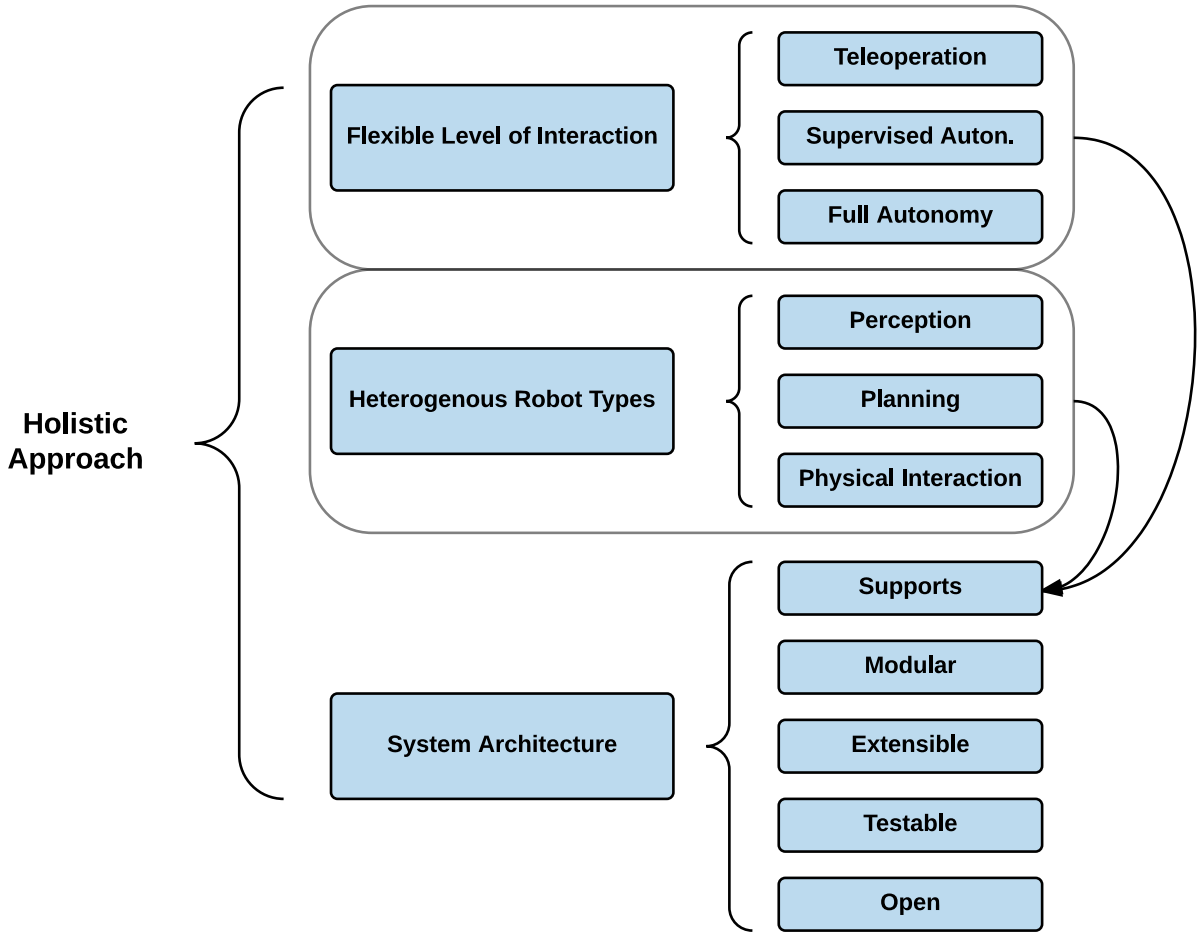


Figure 4.1: Three pillars of the holistic approach and their relationship.

researchers, modularity, openness and extensibility are desirable. To be able to verify system capabilities, the means to test system functionality has to be provided.

Only some of these requirements are satisfied by existing approaches, but a holistic approach incorporating all aspects as shown in Figure 4.2 for USAR applications has not been investigated much so far.

Figure 4.2 depicts the required capabilities and the required interaction between robot and operator in detail. Some capabilities always have to provided by the robot (such as the capability for physical interaction), but for most capabilities, a human supervisor can interact with the robot to increase flexibility and performance.

A requirement that is not as relevant for mature commercialized robotic systems ready to be employed by end users, but highly relevant for those in research, is the ability to leverage existing developments and software, allowing to avoid the duplication of efforts and spending time for re-implementing approaches.

At the same time, relying on standard software for functional system components allows reproducing results and using components provided by other researchers. This is a major driver for accelerating research in robotics and thus also a key factor for accelerating the development of USAR robots towards supervised autonomous systems that are deployable in real disaster situations.

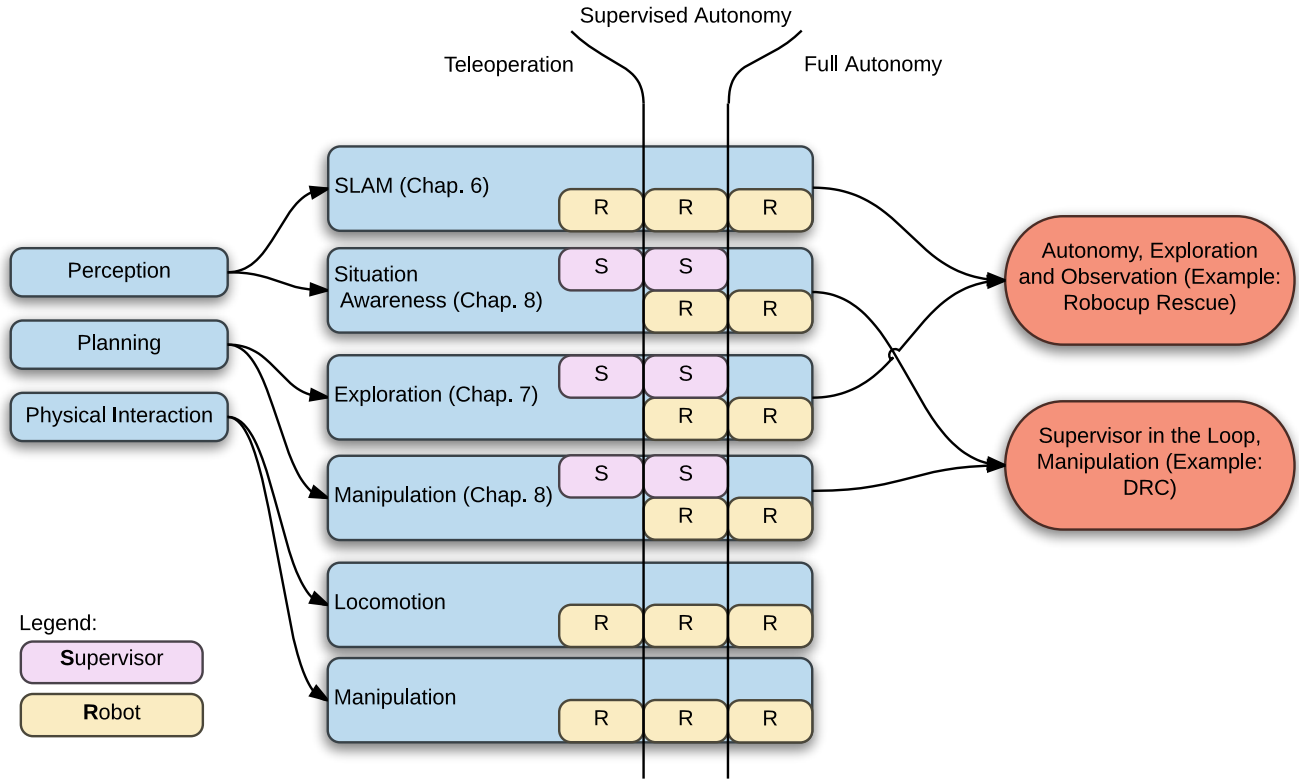


Figure 4.2: Required capabilities and associated control modalities.

The achievable complexity of robotic system architectures is limited unless the architectural design allows a transparent exchange of functional components (e.g. for SLAM or path planning) and also can be extended by additional functional components. Modularity, re-usability, and extensibility are key properties of the architectural design needed to enable sustainable robotic system development.

While robots can be considered expendable in the sense that a loss is acceptable (in contrast to human responders), high reliability and resilience are important aspects that USAR robotic systems have to provide. Failures in disaster situations can have grave consequences, for instance when a robot gets stuck or otherwise unresponsive. It can then block access for responders, or tie up efforts of response forces that would be required elsewhere.

As communications in disaster environments can be degraded, the possibility of delayed, reduced bandwidth, intermittent or even completely absent communication has to be considered in the design and appropriate measures have to be taken to be tolerant to variations in communication link quality. This also motivates the need for autonomous capabilities: Autonomous performance under ideal (communications) conditions might actually be inferior to a human expert using teleoperation. However, under constrained communication conditions with outages or very high latencies, teleoperation might become impossible to use; in that case, leveraging autonomous functionality, for instance for motion planning and control, is the only possible way to proceed.

4.2 Related Work

In this section, work related to the requirements stated in the previous section is discussed.

4.2.1 Flexible Level of Interaction

The incorporation of interaction capabilities between a remote supervisor and a robot is a major challenge for the system design of a USAR robotic system. A related discussion of remote human-robot interaction is available in [163]. Here, multiple taxonomies for interaction between human operators and robots are discussed. The focus is on multiple operators and robots and the resulting combinations of control options. However, the concept of a human-robot team as partners and the explicit consideration of this kind of closely coupled and interdependent teaming is not discussed in detail.

The capability to establish a dialogue between robot and human is the underlying idea behind the *collaborative control* concept introduced in [52]. With the human viewed as a resource for the robot, it becomes possible to distribute tasks in both directions.

Extending the idea of *collaborative control*, the *coactive design* concept does not focus on autonomy, but on the interdependence between involved parties. The aim is not to take the human out of the loop in ideal conditions but to embrace interdependence between actors and leverage interaction between them for higher reliability of the overall system.

Within the DRC, multiple approaches for performing complex manipulation using humanoid robots have been developed by competing teams. With a focus on complex tasks that require human supervision, most approaches provide only limited autonomous capabilities and are optimized for use with humanoid robots performing manipulation tasks.

There is a wealth of research in task allocation and sliding autonomy between robots and robot teams [37]. A collaborative control concept between a human and a remote robot team has been developed in [115]. It is based on an extension of situation awareness to the concept of situation overview and an event-based communication concept [116].

In this thesis, however, the focus is not on higher numbers of human or robot teammates. Instead, the scope of possible interactions between a supervisor and robot should be expanded. For robots performing exploration in USAR environments, this scope is from teleoperation to full autonomy. For the accomplishment of complex manipulation tasks, however, full autonomy is not feasible in most cases. Therefore such complex tasks have to be solved in close interaction with human supervisors, leveraging their cognitive abilities and expertise.

For the vision of this supervisor-robot teaming to work, it is essential to provide the human operator with the required Situation Awareness (SA). It is defined in [46] 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.”

SA thus enables the supervisor to make well-informed decisions. Likewise, the robotic system also has to obtain SA to operate safely in challenging environments. The ability of a robot to plan motions that avoid obstacles, for instance, requires that the robot has SA through a model of the surrounding environment geometry.

4.2.2 Heterogeneous Robot Types

The support of heterogenous robot capabilities is limited in the existing body of work on USAR robots in real applications. Most fielded systems are teleoperated, as previously discussed in Chapter 2.1.

The Fukushima Daiichi nuclear accident resulted in renewed interest in robots for complex manipulation in disaster response. One of the most notable activities resulting from this renewed interest was the DRC. The focus here was on complex manipulation und human supervision. Autonomy was not required and participants naturally leveraged the fact that communication, albeit limited, was possible between supervisors and robot.

While limiting complex manipulation capabilities to require a human supervisor in the loop appears to be the only practical way to achieve such tasks in the foreseeable future, there are scenarios in USAR in which increased, or even full autonomy is desirable and realizable in the short term. An example of this are tasks like autonomous exploration and victim search. In contrast to complex manipulation, for solving such tasks, highly autonomous capabilities appear more adequate.

It is thus a desirable property for robots to provide the full range of capabilities from autonomous exploration to supervised manipulation in a single system, allowing for flexible use and adaptation to the situation encountered.

4.2.3 System Architecture

Middleware

For general system design considerations, there has been significant effort in research to formalize, standardize and harmonize robotics research and development. The use of standardized middleware has gained significant support in recent years.

There are multiple definitions of the term middleware. In [7], Middleware is defined as "a class of software technologies designed to help manage the complexity and heterogeneity inherent in distributed systems. It is defined as a layer of software above the operating system but below the application program that provides a common programming abstraction across a distributed system."

Middleware is employed in complex software systems to allow for horizontal integration of components. Instead of interconnecting components directly, they use well-defined interfaces via a middleware for communication, decoupling components from each other.

In the past, a lack of standardization in robotics meant that many research groups developed their own middleware approaches. The survey from 2008 provided in [100] provides a good overview of the fragmentation of the research community with regards to robotics middleware in 2008. Without common middleware, a lot of duplicate implementation effort hindered progress, as for instance, low-level drivers had to be created by each researcher group for each middleware independently.

With the Best Practice in Robotics (BRICS) project [18], a wide range best practices were investigated, including a dedicated Integrated Development Environment (IDE) and component model for robotics research and development. Also discussed in [73], the adoption of model-driven development methods has not been widespread so far, however.

Instead, Robot Operating System (ROS) [1] has become the most used and de-facto standard middleware framework within the robotics research community in recent years. ROS does not provide a well-defined component model specification or model driven design tools, instead relying more on conventions and contracts between users. Discussions of possible reasons why

[1] `http://www.ros.org/`

the adoption of model-driven approaches is limited despite potential advantages are available in [61] and [158].

Open Source

The typical Not Invented Here (NIH) syndrome [76] can significantly impede progress in research due to researchers preferring to use their own solutions to problems instead of drawing on to the developments of others. However, being able to overcome this requires suitable standardization enabling a modular exchange of components.

With the wide adoption of ROS and other standard tools, this barrier got smaller and a reproduction of results requires less effort than a few years ago thanks to standardization.

However, this requires that researchers do not only publish findings based on the research they perform, but also make the actual software itself available for others. In this spirit, most of the work presented in this thesis is made available as open source software.

As noted in [120], following the open source paradigm can lead to much-improved robustness of software. With availability and use, shortcomings and bugs get detected faster, leading to more reliable and useful software.

This also has been observed with the software presented in this thesis. Numerous issue reports and improvements have been filed by users for the *hector_slam* system described in Chapter 6.

4.3 Design of System Architecture

To satisfy the research-level requirements on reproducibility and modularity, ROS is chosen as the underlying middleware. Some of the software described in this work was originally developed using the RoboFrame [149] framework, but once ROS emerged as a standard middleware, the architecture was migrated.

The nowadays nearly ubiquitous proliferation of ROS in the research community allows for using established standard interfaces and the ROS architecture allows for the development of highly modular code. With a large user base, the barrier of entry for other researchers to use open source developments is much lower, which is highly advantageous considering the goal of advancing research for challenging applications like intelligent USAR robots which also are not the main interest of commercial robotic technology developments.

While ROS provides transparent capability for distributing components over different machines by means of the network based TCP/IP-based transport, communication constraints can impose additional challenges that make using ROS standard transports not feasible in some highly constrained scenarios. For those, specialized communication bridge tools need to be used, separating the ROS networks of the onboard and Operator Control Station (OCS) sides. Such software has been developed by partners at Team ViGIR during participation in the DRC [85], but is not the focus of this thesis.

To achieve high reliability, satisfying the requirement for observability, predictability, and directability of the robotic system as laid out for the coactive design concept [71] is required. When considering the human supervisor and robot as a team, the members thus have to allow each other to understand the state of the other side (observability). They also have to be able to predict and understand the intent of the other side (predictability). Lastly, team members have to be able to communicate meaningful and accurate commands (directability).

The capability to inform the operator about the robot state using appropriate visualization thus has to be part of the system design. Also through proper visualization, predictability can be

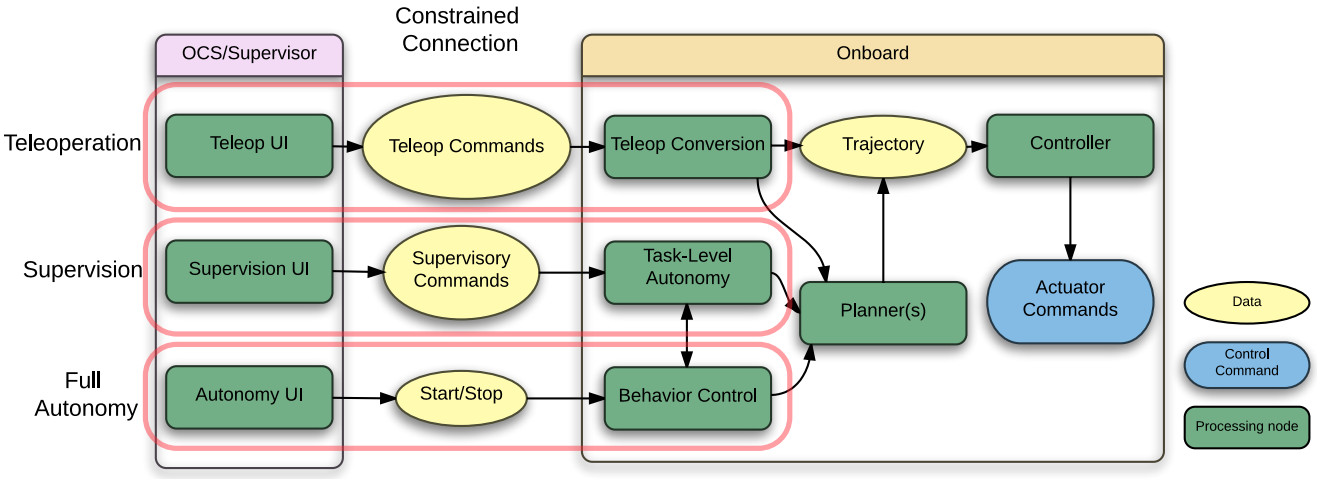


Figure 4.3: Supporting multiple levels of interaction with the robotic system using a common backend. Control approaches from teleoperation, supervised autonomy to full autonomy have to be supported for maximum flexibility and robustness to error.

achieved. Achieving directability requires interfaces that allow for efficient and reliable interaction. These concepts will be revisited in following chapters in the respective sections (Sections 6.3, 7.3 and 8.3).

As noted previously, to achieve high reliability and flexibility, the capability to flexibly change control and interaction modes between autonomous and teleoperated operation is crucial. While autonomous and assistive functions promise to reduce the workload of operators and in some cases higher reliability, they can be brittle in real-world scenarios, where unexpected situations and failures can foil proceeding with the mission. In such cases, the capability of flexible switching between modes can significantly improve the reliability of the system, as the human supervisor has a "toolbox" of options at her disposal and can dynamically switch between them, adapting to the situation.

Figure 4.3 provides an overview of the different levels of interaction and how they possibly cause interaction within onboard systems. As suggested by the size of ellipses representing data sent to the robot, the communication requirements generally are lower both in bandwidth and latency with increased autonomy.

As the lowest level of interaction between operator and robot, teleoperation should always be available. Bypassing autonomous functions, this interaction mode shifts the burden to the operator. Importantly, connectivity between robot and operator has to be sufficient in both directions, otherwise, teleoperation becomes slow, unsafe or even impossible.

With currently fielded robotic systems, these good communications conditions have to be met, as otherwise, the robot becomes inoperable. Once autonomous assistance functionality is in more widespread use, the capability to fall back to teleoperation can be allowed to be impeded by communication constraints, allowing for new applications. As teleoperation is the last fallback mode in case autonomous components fail, availability of it is important for overall reliability and ability to recover.

In supervised autonomy mode, the operator provides task-level goals to the robot that are then followed autonomously using onboard systems. This reduces reliance on good connectivity and low latency communication, as the robotic system can follow task-level goals even when

communication is intermittent. Such an approach requires sophisticated sensing and planning capabilities for onboard systems, however.

Using full autonomy, the human operator only specifies the mission and associated tasks and provides a start command, monitors data provided by the robot to maintain SA and either reacts to requests from the robot or switches to a lower autonomy mode on her own discretion. The clear advantage of full autonomy is that there is no need for communications as long as everything works well. The onboard autonomy system leverages the capabilities for task-solving used in the assisted autonomy mode and also makes use of planning capabilities, either directly, or via the task-level autonomy functionality.

It is crucial that when using a flexible level of interaction, the system stays in a well-defined state. For instance, when teleoperation commands are sent, autonomous control components have to be notified of the switch in interaction level as to not cause undefined behavior when commands both from the operator and autonomous executive are executed at the same time.

5 Simulation of Robots and Scenarios for Development and Testing

To achieve high reliability of complex autonomous systems, the capability to perform comprehensive testing of the complete system in scenarios with high similarity to intended use cases is highly important. Setting up a real robot for such testing is both time and effort consuming, especially when considering scenarios that cannot easily be provided physically. Often, this is the case because real-world constraints do not allow for arbitrarily scaling an evaluation scenario.

To provide means for both development and evaluation of the comprehensive systems used within this thesis, they are simulated using *Gazebo*[1]. This simulator provides dynamics simulation using the Open Dynamics Engine (ODE) physics simulation library and others (in most recent versions). Actuation, sensors, and scenarios can be simulated. This enables testing of robotic systems with negligible setup effort over a wide range of scenarios without the need of setting up real hardware.

To enable meaningful simulation for the approaches described in this work, additional contributions had to be made, as not all functionality required for USAR simulation were directly available in Gazebo. The developed packages are also widely used in the RoboCup and ROS communities.

5.1 Related Work

USARsim [31] has been the dominant simulation framework for USAR applications for a long time. It is based on the Unreal Tournament game engine and has been demonstrated to provide plausible simulation for USAR scenarios [32]. It also has been shown to provide useful capabilities for Human Robot Interaction (HRI) research [156].

While delivering these promising capabilities, further development of USARsim slowed in recent years and ROS integration was missing for a long time. This integration was added only recently [9], but the research community largely moved on towards the use of Gazebo for simulation as it provides higher modularity, integration with the ROS ecosystem and is maintained by the Open Source Robotics Foundation (OSRF), among other advantages.

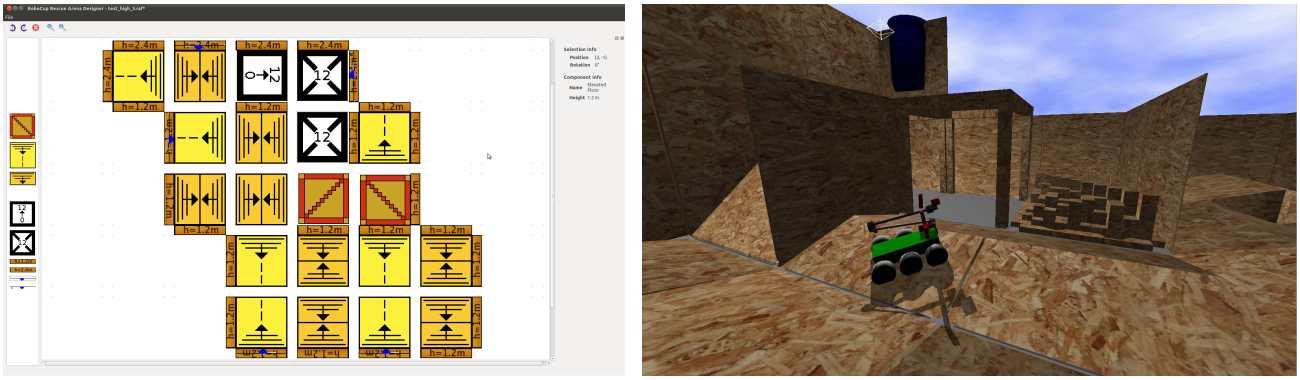
The RoboCup Rescue Robot Simulation League, a long time user of USARsim, is also in the process of switching to Gazebo [141] as their principle simulation framework. It, however, has to be noted that USARsim has frequently been used for multi-robot simulation in the past [10], [153] and equivalent capability has not been demonstrated using Gazebo so far.

V-REP [125] is another versatile simulation framework that for a long time did not provide ROS integration. MORSE [44] is a simulation framework based on the Blender 3D modeling software and was released in 2013 and has not found widespread adoption.

With the announcement of the DRC, it became clear that Gazebo would be used as the principle simulator for the VRC. The funding and additional resources associated with this decision are other aspects in favor of using Gazebo as principle simulation tool.

Due to the described factors, Gazebo was selected as the simulation tool for the USAR robotics research within teams Hector and ViGIR and also for the research performed for this thesis.

[1] `http://gazebosim.org/`



(a) (b)

Figure 5.1: USAR scenario simulation: (a): Screenshot of the GUI tool for creating USAR scenarios showing an example scenario (b): The same scenario as simulated in *gazebo* simulation

5.2 USAR Scenario Simulation

Simulated RoboCup Rescue arena scenarios serve to evaluate the robot capabilities for USAR applications. The *hector_nist_arena_designer* ROS package permits the fast and intuitive creation of user defined test scenarios. Figure 5.1 shows a screenshot of the arena designer and of the resulting scenario in Gazebo. Examples of different components that can be used for creating scenarios are shown in Figure 5.2.

Having been developed as what can be considered a toolkit for the evaluation of response robots, the NIST arena building blocks allow for quickly creating a variety of different scenarios both in the physical world, but also in simulation.

5.3 Thermal Imaging

There is no built-in simulation for thermal imaging available for Gazebo, precluding the use of thermal cameras in simulation. This sensor type is among the most important for the detection of simulated victims, however. For this reason, a simulated thermal sensor was developed as part of providing a comprehensive USAR simulation evaluation and testing system.

As multispectral simulation currently is not supported by Gazebo, as a workaround certain colors that otherwise do not appear in the simulated environment are used to mark hot regions. The simulated thermal camera then maps those to hot pixels, while all other parts of the camera image are artificially reduced in intensity.

This approach does not provide high fidelity simulated thermal imagery with all of the effects observed on a real system, like reflections off the ground, but allows for using the same thermal processing backend as on the real system. Figure 5.3 shows an example of a scene with both visible light image and simulated thermal image of the same scene displayed.

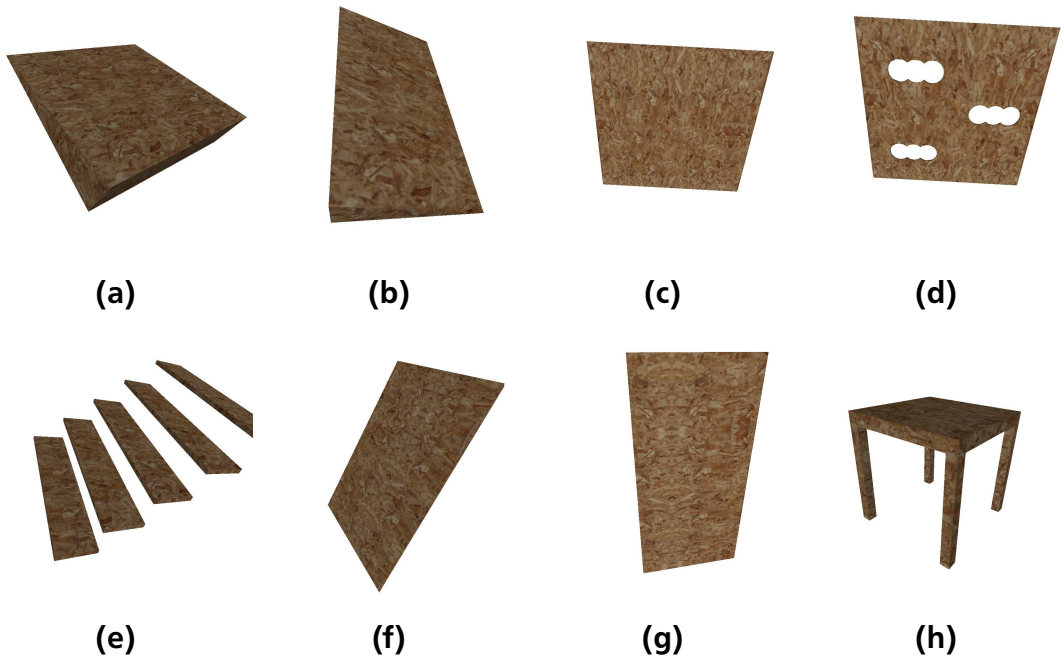


Figure 5.2: DHS-NIST-ASTM International Standard Test Method Elements for Simulation: (a): Full ramp (b): Half ramp (c): Wall (d): Wall with victim holes (e): Stairs (f): 45 degree ramp (g): High wall (h): Elevated floor

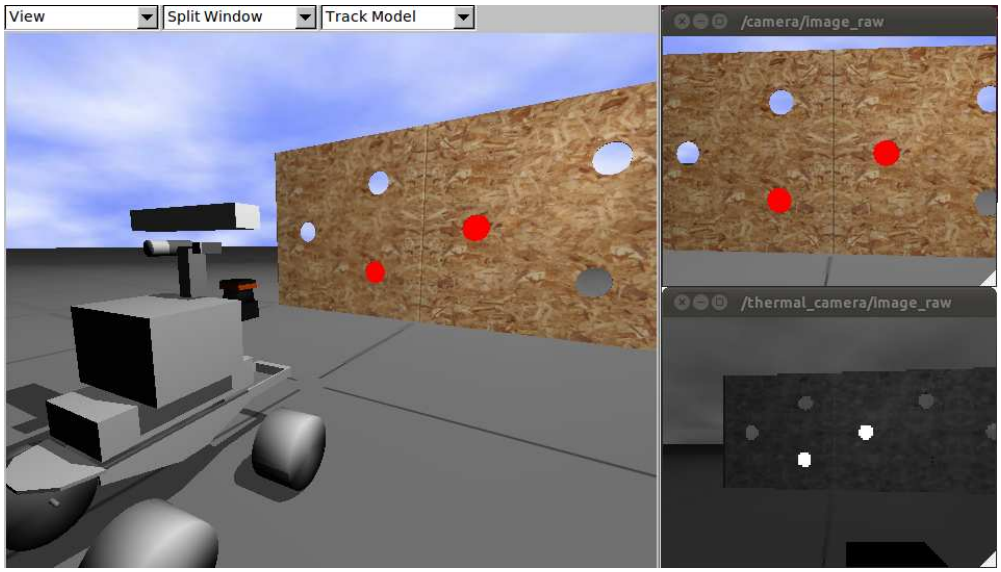


Figure 5.3: Thermal Camera Simulation: A thermal camera is simulated through a Gazebo plugin

5.4 Tracked Vehicles

Simulation of tracked vehicles is not available for Gazebo. There are multiple approaches that are feasible, representing trade-offs between simulation fidelity and computational complexity. The perhaps most straightforward approach is the approximation of tracks by adding differential drive wheels to the model. This approach only works reliably on flat ground, however.

Another approach is simulation based on a comprehensive dynamics model, modeling single track links and their interaction with the ground similar to the approach described in [128]. While promising high fidelity, such an approach is hard to adjust for close to real-time simulation and can consume significant computational resources.

To achieve sufficient fidelity tracked robot simulation that allows for climbing stairs and negotiating ramps, an approach directly applying a wrench to the robotic system was thus developed. With this approach, the tracks of the system are modeled as having no friction. The robot thus slides over the ground. Based on commanded velocities for the robot, a wrench is applied at a reference point, which leads to robot motion. While this approach is of limited fidelity, it allows for traversing the intended scenarios and provides plausible performance mics model, modeling single track links and their interaction with the ground for many applications.

5.5 Quadrotor Simulation

UAV and Micro Air Vehicle (MAV) systems are among the type of robotic systems for which simulation is most advantageous as unlike ground vehicles, they generally cannot roll to a stop on failure. It was thus quite surprising that simulation of such vehicles was not available for simulation with Gazebo for a long time. To close this gap, the the *hector_quadrotor* stack for ROS was developed [99], allowing to simulate quadrotor drones in Gazebo.

Unlike specialized quadrotor simulators focussing on single aspects such as aerodynamics, Gazebo integration allows for comprehensive simulation of whole systems including arbitrary sensors that are available within Gazebo. Figure 5.4 shows an example of using the Parallel Tracking and Mapping (PTAM) visual SLAM system [79] modified for use with UAVs [157] in simulation. As shown there, the simulation of sensors allows simulating the whole workflow including calibration of camera sensors.

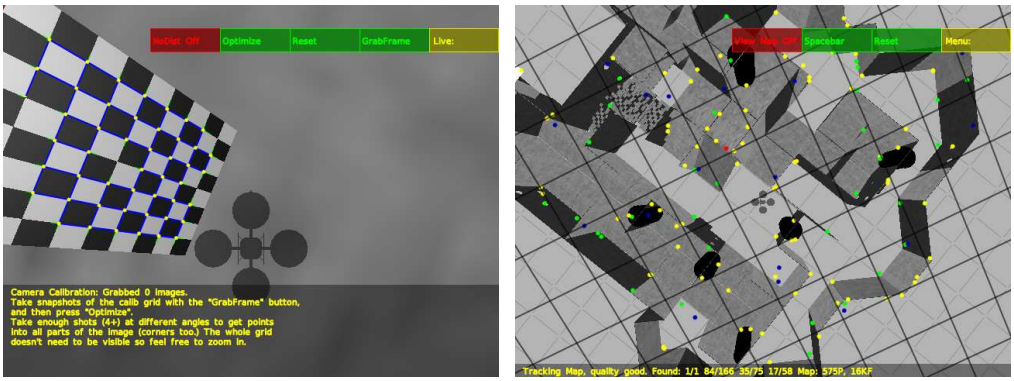
5.6 Other Applications

In the ARGOS Challenge [2], robots have to perform inspection tasks on oil and gas sites. While not a USAR scenario during normal operating conditions, some of the challenges encountered are quite similar: Robots have to navigate in narrow environments and the state of objects in the environment has to be reliably determined based on robot onboard sensors.

To achieve this also in simulation, a high fidelity model of the environment based on Computer Aided Design (CAD) data has been created. It allows for the simulation of complete inspection missions in simulation and also for the simulation of disaster scenarios as demonstrated in Figure 5.5b, where structure collapsing onto a walkway is simulated.

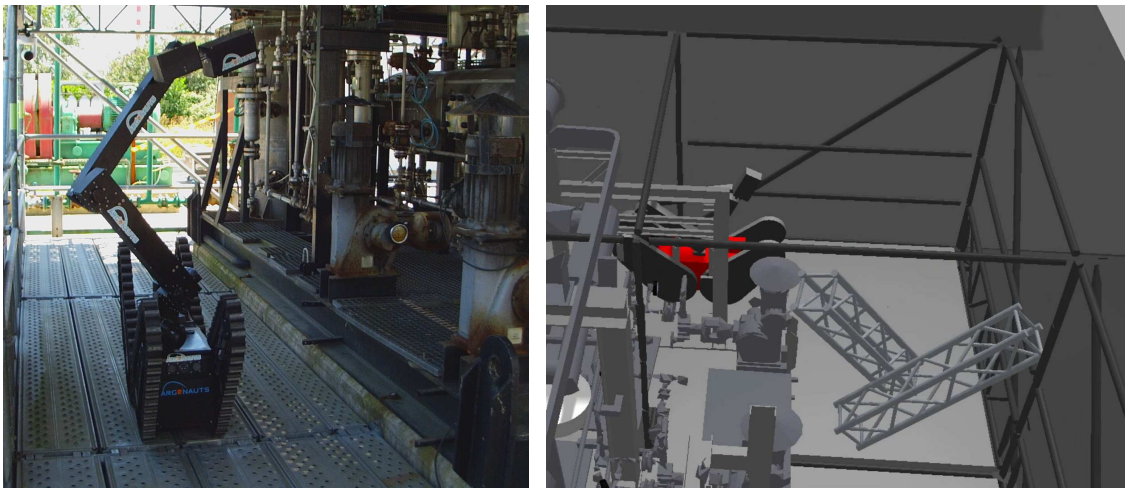
Using the simulated site, navigation and exploration capabilities of UGVs were verified in simulation first and afterward tested on the real site.

[2] `http://www.argos-challenge.com/`



(a) **(b)**

Figure 5.4: Visual SLAM simulation using a quadrotor UAV: (a): Calibration of camera system in simulation. (b): Screenshot of PTAM being used for visual SLAM on a quadrotor hovering above a simulated NIST standard arena for response robots.



(a) **(b)**

Figure 5.5: Industrial inspection simulation: (a): Real scenario with robot inspecting a object of interest (b): Simulated scenario. Using scripting, dynamic events like trusses falling on the walkway during a mission can be simulated.

6 Simultaneous Localization and Mapping

To enable reliable operation in USAR scenarios, it is either highly desirable (in case of robot teleoperation) or required (in case of autonomous operation) that robots are capable of learning a map of the environment and simultaneously localizing within it. This chapter is based on work that previously has been published in [86].

The approach has been published open source in 2011 for ROS, named *hector_slam*. It has since become one of the most frequently used SLAM solutions for performing SLAM on mobile robotic systems with LIDAR sensors. Example applications are described at the end of this chapter.

6.1 Related Work

There has been a wealth of research into the SLAM problem in recent years, with reliably working solutions for typical office-like indoor scenarios using Rao-Blackwellized particle filters [57] like Gmapping [56] being available as open source software. However, these solutions work best in planar environments, rely on available, sufficiently accurate odometry and do not leverage the high update rate provided by modern LIDAR systems. For unstructured environments, that lead to significant roll and pitch motion of the carrier, or implementation on aerial platforms such systems are not applicable or have to be modified significantly.

A useful distinction in the literature is between SLAM front-end and back-end systems. While SLAM front-ends are used to estimate robot movement online in real-time, the back-end is used to perform optimization of the pose graph given constraints between poses that have been generated using the front-end. The approach presented in this work serves as a SLAM front-end and does not provide pose graph optimization like approaches presented in [74] and [87]. However, we show that in many scenarios such optimizations are not needed under real world conditions as the approach is sufficiently accurate for robots to perform their mission.

Multiple indoor navigation systems based on laser scan matching have been presented for the use on Quadrotor UAVs [6, 58, 41]. Here, a two-stage approach is employed, with a front-end fast scan alignment step for pose estimation and a slower back-end mapping step running in the background or on a remote computer. The pose estimates from scan alignment are not directly incorporated into the vehicle control loop and thus it moves at a low speed only.

In [80] and [66] other front-end systems used on mobile robots are described. In contrast to system presented in this work they do not provide a full 6DOF pose estimate and are not available as open source software.

Work on localization using scan matching started with the Iterative Closest Point (ICP) [166] algorithm which originated as a general approach for registering 3D point clouds. The main drawback of many ICP-based methods is the expensive search for point correspondences, which has to be performed in every iteration. Polar Scan Matching (PSM) [38] avoids the correspondence search by taking advantage of the natural polar coordinate system of laser scans to estimate a match between them. Scans have to be preprocessed to be used in the polar scan matcher. The real-time correlative scan matching approach [111] uses an exhaustive sampling based approach for scan matching. Using several optimizations this approach is capable of real-

time application. Normal Distribution Transform (NDT) [16] based scan matching aligns scans to a mixture of normal distributions representing preceding scans.

For littoral water scenarios there has been research into using expensive multi-sensor scanners [62], but to the author's knowledge, there is no single emitter LIDAR-based SLAM approach available that was tested under real world conditions.

With the availability of low-cost RGB-D sensors like the Microsoft Kinect, SLAM and odometry approaches using these sensors became available. Impressive results are reported for many of these approaches, but testing in simulated USAR environments revealed that they are not well suited for applications with sudden sensor motion. These motions cannot be prevented in USAR applications, however, making available RGB-D approaches infeasible to use.

6.2 Contribution

The contribution described in this chapter is a flexible and robust SLAM approach that is significantly more robust than other state of the art systems. It allows for learning maps of environments despite challenging motion of the platform carrying the used sensors. Available open source, it is used widely used by researchers and used for diverse applications such as simulated USAR tasks, for mapping with quadrotor UAVs or with new low-cost LIDAR sensors. Related publications are [86] and [136].

Johannes Meyer contributed the 6DOF state estimation system described in [86]. Philipp Scholl contributed the RSSI fingerprinting code and experiments described in [136].

6.3 Human-Robot Interaction and Supervision Aspects

While the SLAM system runs self-contained entirely on the robotic system and as such interaction with human operators is very limited, an important aspect that might not be immediately obvious becomes clear when viewing the system from the user (i.e. this case responder) perspective:

The generation of easily readable environment maps is the main deliverable responders are interested in. Taking this perspective, the performance and usefulness of SLAM software is most importantly measured by the quality of the human-readable maps that it provides.

On the other hand from a research perspective, the ability to compare results in a standardized manner is perhaps the most important aspect.

To satisfy both requirements, the *hector_geotiff* package that is part of *hector_slam* provides the ability to generate maps using the standardized GeoTIFF [123] format. As discussed in [137] and [113], this format allows for providing geo-referenced data, facilitating systematic performance evaluation.

The format is easily human-readable, as the background grid has 1 m^2 tiles, making scale immediately visible. Most grid maps shown in Figures in this thesis are shown in their GeoTIFF representation for this reason.

6.4 Approach

To be able to represent arbitrary environments an occupancy grid map representation is used, which is a proven approach for mobile robot localization using LIDARs in real-world environments [150]. As the LIDAR platform might exhibit 6DOF motion, the scan has to be transformed

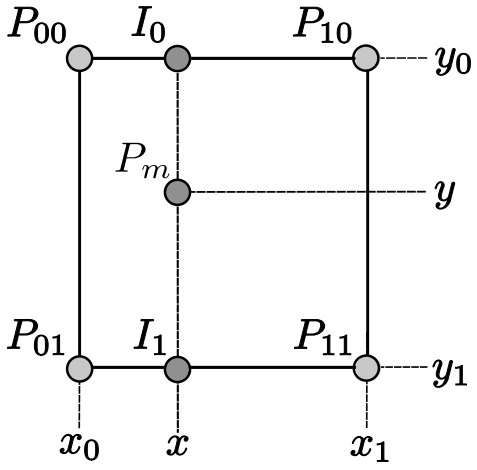


Figure 6.1: Bilinear filtering of the occupancy grid map. Point P_m is the point whose value shall be interpolated.

into a local stabilized coordinate frame using the estimated attitude of the LIDAR system. Using the estimated platform orientation and joint values, the scan is converted into a point cloud of scan endpoints. Depending on the scenario, this point cloud can be preprocessed, for example by downsampling the number of points or removal of outliers. For the presented approach, only filtering based on the endpoint z coordinate is used, so that only endpoints within a threshold of the intended scan plane are used in the scan matching process.

In the following, steps are detailed that in combination allow robust and computationally highly efficient scan-matching.

6.4.1 Map Access

The discrete nature of occupancy grid maps limits the precision that can be achieved and also does not allow the direct computation of interpolated values or spatial derivatives. For this reason, an interpolation scheme allowing sub-grid cell accuracy through bilinear filtering is employed for both estimating occupancy probabilities and spatial derivatives. Intuitively, the grid map cell values can be viewed as samples of an underlying continuous probability distribution.

Given a continuous map coordinate P_m, the occupancy value $M(P_m)$ as well as the gradient $\nabla M(P_m) = \left(\frac{\partial M}{\partial x}(P_m), \frac{\partial M}{\partial y}(P_m)\right)$ can be approximated by using the four closest integer coordinates $P_{00..11}$ as depicted in Figure 6.1. Linear interpolation along the x- and y-axis then yields

$$\begin{aligned} M(P_m) \approx & \frac{y - y_0}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{11}) + \frac{x_1 - x}{x_1 - x_0} M(P_{01}) \right) \\ & + \frac{y_1 - y}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{10}) + \frac{x_1 - x}{x_1 - x_0} M(P_{00}) \right) \end{aligned} \tag{6.1}$$

(a) (b) (c)

Figure 6.2: Occupancy grid and spatial derivatives: (a): Occupancy grid map (b): Smoothed derivative $\frac{\partial M}{\partial x}(P_m)$ (c): Smoothed derivative $\frac{\partial M}{\partial y}(P_m)$

The derivatives can be approximated by:

$$\begin{aligned}\frac{\partial M}{\partial x}(P_m) \approx & \frac{y - y_0}{y_1 - y_0}(M(P_{11}) - M(P_{01})) \\ & + \frac{y_1 - y}{y_1 - y_0}(M(P_{10}) - M(P_{00})) \end{aligned} \tag{6.2}$$

$$\begin{aligned}\frac{\partial M}{\partial y}(P_m) \approx & \frac{x - x_0}{x_1 - x_0}(M(P_{11}) - M(P_{10})) \\ & + \frac{x_1 - x}{x_1 - x_0}(M(P_{01}) - M(P_{00})) \end{aligned} \tag{6.3}$$

It should be noted that the sample points/grid cells of the map are situated on a regular grid with distance 1 (in map coordinates) from each other, which simplifies the presented equations for the gradient approximation.

6.4.2 Scan Matching

Scan matching is the process of aligning laser scans with each other or with an existing map. Modern laser scanners have low distance measurement noise and high scan rates. A method for registering scans might yield very accurate results for this reason. For many robotic systems, the accuracy and precision of the laser scanner are much higher than that of odometry data, if odometry is available at all. There are many scenarios for which odometry data is either not available or potentially erroneous. Examples of such applications are provided in Chapter 6.6.

Our approach is based on optimization of the alignment of beam endpoints with the map learned so far. The basic idea of using a Gauss-Newton approach is inspired by work in computer vision [93]. Using this approach, there is no need for a data association search between beam endpoints or an exhaustive pose search as in other approaches. As scans get aligned with the existing map, the matching is implicitly performed with all preceding scans.

We seek to find the rigid transformation $\xi = (p_x, p_y, \psi)^{\mathrm{T}}$ that minimizes

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{n} \big[1 - M(\mathbf{S}_i(\xi))\big]^2 \tag{6.4}$$

that is, we want to find the transformation that gives the best alignment of the laser scan with the map. Here, $\mathbf{S}_i(\xi)$ are the world coordinates of scan endpoint $\mathbf{s}_i = (s_{i,x}, s_{i,y})^{\mathrm{T}}$. They are a function of ξ, the pose of the robot in world coordinates:

$$\mathbf{S}_i(\xi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & cos(\psi) \end{pmatrix} \begin{pmatrix} s_{i,x} \\ s_{i,y} \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix} \tag{6.5}$$

The function $M(\mathbf{S}_i(\xi))$ returns the map value at the coordinates given by $\mathbf{S}_i(\xi)$. Given some starting estimate of ξ, we want to estimate $\Delta\xi$ which optimizes the error measure according to

$$\sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\xi + \Delta\xi))\right]^2 \to 0\,. \tag{6.6}$$

By first order Taylor expansion of $M(\mathbf{S}_i(\xi + \Delta\xi))$ we get:

$$\sum_{i=1}^{n} \left[1 - M(\mathbf{S}_i(\xi)) - \nabla M(\mathbf{S}_i(\xi)) \frac{\partial \mathbf{S}_i(\xi)}{\partial \xi} \Delta\xi)\right]^2 \to 0\,. \tag{6.7}$$

This equation is minimized by setting the partial derivative with respect to $\Delta\xi$ to zero:

$$2 \sum_{i=1}^{n} \left[\nabla M(\mathbf{S}_i(\xi)) \frac{\partial \mathbf{S}_i(\xi)}{\partial \xi}\right]^T \left[1 - M(\mathbf{S}_i(\xi)) - \nabla M(\mathbf{S}_i(\xi)) \frac{\partial \mathbf{S}_i(\xi)}{\partial \xi} \Delta\xi)\right] \quad = \quad 0 \tag{6.8}$$

Solving for $\Delta\xi$ yields the Gauss-Newton equation for the minimization problem:

$$\Delta\xi = \mathbf{H}^{-1} \sum_{i=1}^{n} \left[\nabla M(\mathbf{S}_i(\xi)) \frac{\partial \mathbf{S}_i(\xi)}{\partial \xi}\right]^T \left[1 - M(\mathbf{S}_i(\xi))\right] \tag{6.9}$$

with

$$\mathbf{H} = \left[\nabla M(\mathbf{S}_i(\xi)) \frac{\partial \mathbf{S}_i(\xi)}{\partial \xi}\right]^T \left[\nabla M(\mathbf{S}_i(\xi)) \frac{\partial \mathbf{S}_i(\xi)}{\partial \xi}\right] \tag{6.10}$$

An approximation for the map gradient $\nabla M(\mathbf{S}_i(\xi))$ is provided in Section 6.4.1. With equation (6.5) we get

$$\frac{\partial S_i(\xi)}{\partial \xi} = \begin{pmatrix} 1 & 0 & -\sin(\psi) s_{i,x} - \cos(\psi) s_{i,y} \\ 0 & 1 & \cos(\psi) s_{i,x} - \sin(\psi) s_{i,y} \end{pmatrix} \tag{6.11}$$

Using $\nabla M(\mathbf{S}_i(\xi))$ and $\frac{\partial S_i(\xi)}{\partial \xi}$, the Gauss-Newton equation (6.9) can now be evaluated, yielding a step $\Delta\xi$ towards the minimum. It is important to note that the algorithm works on non-smooth linear approximations of the map gradient $\nabla M(\mathbf{S}_i(\xi))$, meaning that local quadratic convergence towards a minimum cannot be guaranteed. Nevertheless, the algorithm works with sufficient accuracy in practice.

For many applications a Gaussian approximation of the match uncertainty is desirable. Examples are updates of parametric filters as well as pose constraints for use with graph optimization

Figure 6.3: Multiresolution representation of the map: (a): 20cm grid cell length (b) 10 cm grid cell length (c) 5cm grid cell length

SLAM backends. One approach is to use a sampling based covariance estimate, sampling different pose estimates close to the scan matching pose and computing a covariance from those. This is similar to the sigma points approach in the Unscented Kalman Filter (UKF) [155]. A second method is the use of the approximate Hessian matrix to arrive at a covariance estimate. Here, the covariance matrix is approximated by

$$\mathbf{R} = \mathrm{Var}\{\xi\} = \sigma^2 \cdot \mathbf{H}^{-1} \tag{6.12}$$

where σ is a scaling factor dependent on the properties of the laser scanner device. A comprehensive derivation is available in [25].

6.4.3 Multi-Resolution Map Representation

Any hill climbing/gradient based approach has the inherent risk of getting stuck in local minima. As the presented approach is based on gradient ascent, it also is potentially prone to get stuck in local minima. The problem is mitigated by using a multi-resolution map representation similar to image pyramid approaches used in computer vision.

In our approach, we optionally use multiple occupancy grid maps with each coarser map having half the resolution of the preceding one. However, the multiple map levels are not generated from a single high-resolution map by applying Gaussian filtering and downsampling as is commonly done in image processing. Instead, different maps are kept in memory and simultaneously updated using the pose estimates generated by the alignment process.

This generative approach ensures that maps are consistent across scales while at the same time avoiding costly downsampling operations. The scan alignment process is started at the coarsest map level, with the resulting estimated pose getting used as the start estimate for the next level, similar to the approach presented in [59]. A positive side-effect is the immediate availability of coarse-grained maps which can, for example, be used for path planning.

6.5 Experimental Evaluation

In this section, the performance of *hector_slam* is discussed based on multiple experiments. Most datasets for the provided experiments are available online [1] as ROS bag files for reproduction by other researchers.

As the focus of the presented SLAM approach is on providing state estimates and learning maps while the carrying platform experiences challenging motion, prior datasets recorded using large mobile robotic systems moving on flat ground as discussed in detail in [89] are not suitable for evaluation. As also described there, for evaluating the performance of SLAM systems, datasets incorporating loop closures are frequently used. The same approach is followed here and performance for multiple challenging datasets is shown. Further discussion on measuring map quality is available in [8].

Videos of the experiments discussed below are available online [2].

6.5.1 RoboCup Rescue

The Hector SLAM system has been used by Team Hector Darmstadt in the challenging environments of the RoboCup Rescue Robot League competition since 2010 and was crucial to winning multiple competitions as further detailed also in Chapter 9.4.

In Figure 6.4a) a map generated using the Hector UGV system during the RoboCup 2012 final mission is shown. Despite encountering significant rough terrain in the rescue arena, the error on loop closure is small and does not affect autonomous navigation. Further examples of maps generated during the RoboCup competition are shown in Figures 9.11 and 9.12.

The presented system has become the most widely used one by competitors in the RoboCup Rescue competition. Given instructions provided via tutorials online, even teams otherwise concentrating on other USAR robot aspects could easily add mapping capabilities to their robotic systems, resulting in accelerated progress towards autonomous systems in the league.

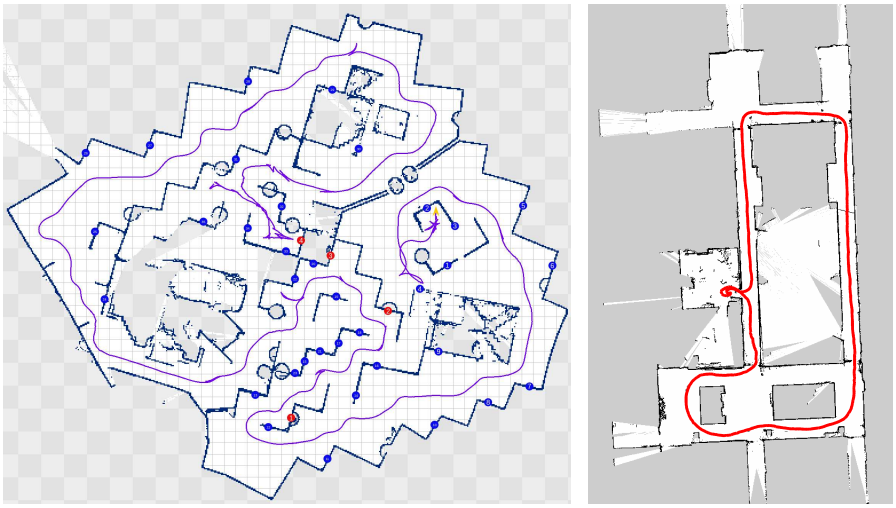
6.5.2 Handheld Mapping System

An embedded mapping system is shown in Figure 6.5. It consists of a Hokuyo UTM-30LX LIDAR system, an Intel Atom Z530 based CPU board as well as a small low-cost Microelectromechanical Systems (MEMS) IMU. The system thus provides all necessary sensors for the application of the presented SLAM approach. It can easily be mounted on unmanned vehicles or carried by hand to learn maps of the environment.

Figure 6.4b) shows a map generated of an office floor at TU Darmstadt using the handheld mapping system, demonstrating consistent mapping despite a large loop. Figure 6.6a shows a map learned by walking through the RoboCup 2011 Rescue Arena with the system in hand overlaid over a ground truth map and Figure 6.6b shows the system used for mapping the new building at Dagstuhl castle, Germany. As it is visible from paths and maps, the system is sufficiently accurate to close loops in these scenarios without the use of an approach for explicit loop closure, keeping computational requirements low and preventing changes to the estimated

[1] https://code.google.com/p/tu-darmstadt-ros-pkg/downloads/list

[2] https://www.youtube.com/playlist?list=PL0E462904E5D35E29



(a) (b)

Figure 6.4: Example maps with loops: (a): Map generated using the Hector UGV system in the final mission of the RoboCup RRL 2012 competition. Despite closing traversing multiple roll and pitch ramps, the offset on loop closure is < 10cm, not affecting autonomous robot operation (b): Map generated using the embedded mapping system also demonstrating consistent map learning despite closing a large loop.

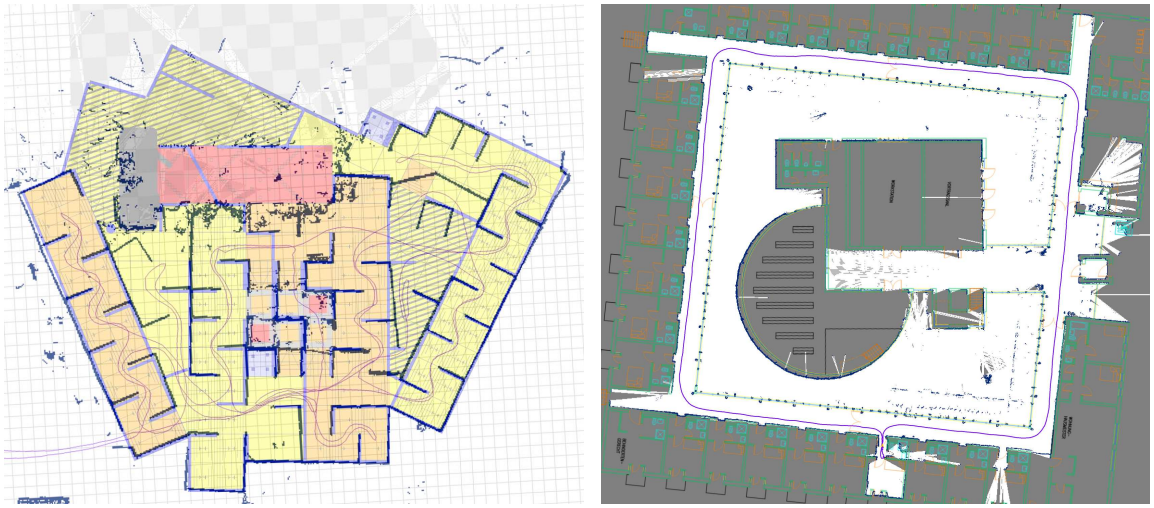
Figure 6.5: Handheld mapping system

map during runtime. Videos and ROS bag files of the experiments are available online[3]. The logged sensor data can be played back to the SLAM system at 3x real-time speed on the Atom Z530 CPU without loss of map quality. The SLAM system thus consumes less than half of overall computational resources of the embedded mapping system with the used settings.

6.5.3 Third Party Evaluation

With the described system available as open-source software, there are publications comparing it to other approaches. In [94], it is compared to other 2D SLAM approaches available for ROS. The top approaches in the evaluation show comparable results. The evaluation uses only a small, flat ground scenario, however, which is less challenging than the USAR and handheld mapping scenarios mentioned previously.

[3] `http://www.gkmm.tu-darmstadt.de/rescue`



(a) (b)

Figure 6.6: Maps learned using the handheld mapping system overlaid with ground truth data: (a): RoboCup 2011 Rescue Arena with multiple small loops. (b): New building at Schloss Dagstuhl exhibiting a large loop.

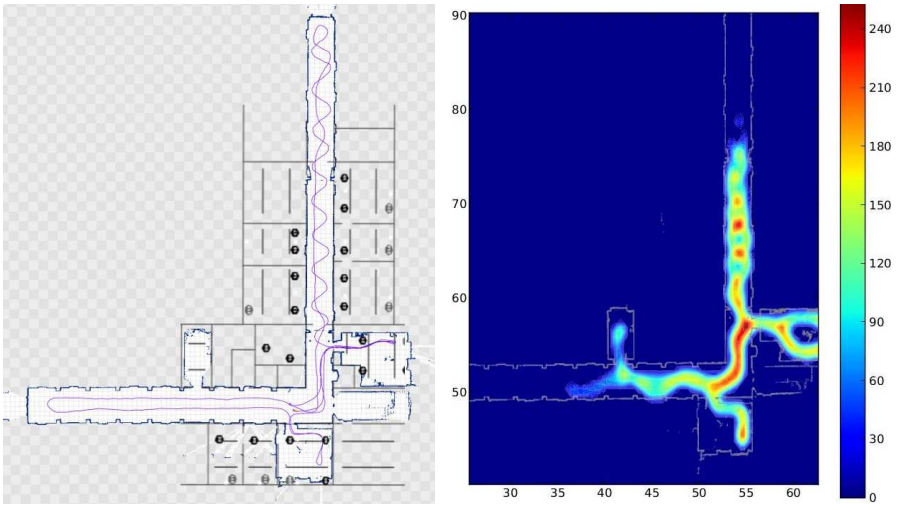
6.6 Applications

A reliable SLAM solution is of great interest both for disaster robotics, but also in other fields. In the following, the use of the *hector_slam* system for different applications is discussed. While originally developed for the USAR scenario, the versatility and easy setup allowed for the system to be used in a wide range of applications. The provided examples are not an exhaustive list and there are known commercial users of the system that are not mentioned here.

6.6.1 Fast Radio Map Building

For indoor localization, WiFi-based localization approaches based on fingerprinting show promising results. They are based on pre-recording the Received Signal Strength Indicator (RSSI) at known positions in the environment and using these "fingerprints" for localization afterward. As noted in [139], The setup for such approaches however often is cumbersome. It requires manual georeferencing of RSSI samples based on a given floor plan of the environment. This approach takes a long time and is error-prone as a lot of tedious manual annotation is involved. A floor plan also has to be available or has to be generated manually in the worst case.

Described in detail in [136], the handheld mapping system was combined with a WiFi USB-stick to provide the capability to localize in an indoor environment, while simultaneously recording data about received RSSI. This approach allows building radio maps for WiFi-based fingerprinting localization significantly faster and with higher reliability compared to many approaches that require manual annotation. Figure 6.7 shows an example map generated at TU Darmstadt. The system has also been used by for experiments with Bayesian compressed sensing [110].



(a) (b)

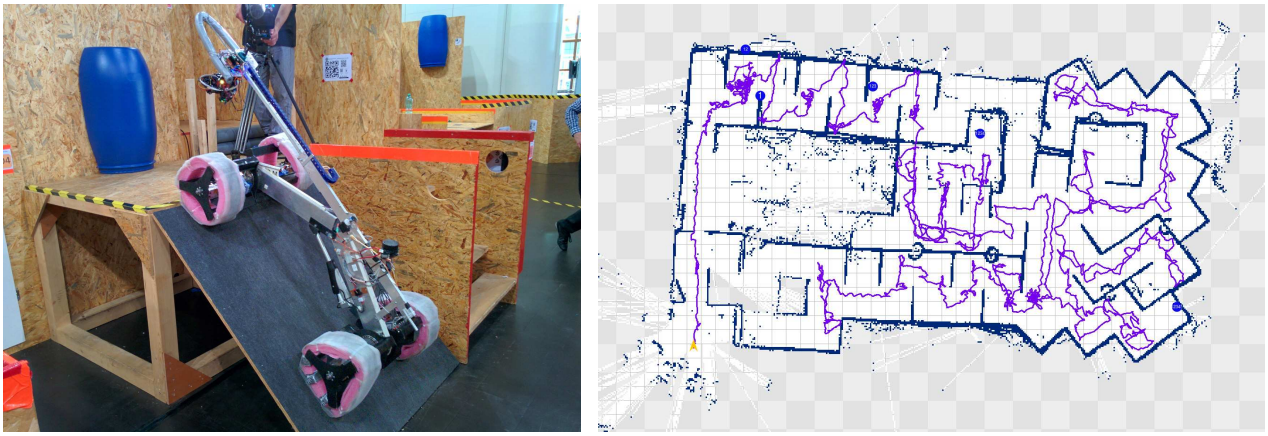
Figure 6.7: Radio map building: (a): Map overlaid over ground truth data. Sensor nodes visible as gray and black circles. (b): Visualization of recorded RSSI data. Higher values indicate better distinguishable RSSI fingerprints.

6.6.2 Littoral Mapping

The feasibility of using the system for larger scale littoral mapping was demonstrated by mounting the autonomy box described in HARDWARE on a USV system. The USV system was manually controlled for data acquisition and steered a branch of Claytor Lake in Virginia. Due to limited time for systems integration, an integration with the USV onboard state estimation system was not performed and only the LIDAR and IMU data was used. The system was able to track the pose of the USV with sufficient accuracy for autonomous control over a range of multiple hundreds of meters.

(a) (b)

Figure 6.8: (a): Mount of autonomy box on USV system (b): Map generated overlaid with Google Earth imagery



(a) (b)

Figure 6.9: (a): The robot used by team Rescube. The RPLidar is the small cylindrical device at the rear of the robot. (b): Map generated by the system at RoboCup German Open 2015. Photo and map courtesy of Peter Kopias.

6.6.3 Use with Low-Cost LIDAR Sensors

Cheap LIDAR sensors like the XV-11 sensor [124] or the RPLidar provide 360-degree sensor coverage at low update rates, but their price makes them an attractive option beyond research applications, also for hobbyist users. Multiple users of *hector_slam* demonstrated that the system works well with these sensors and can be used for navigation of low-cost robots within home environments. Examples are available online [4] and in literature [36].

Team Rescube used a RPLidar sensor for their participation in the RRL at the RoboCup German Open 2015 with their robotic system and could generate a consistent map of the rescue arena, despite significant motion of the platform and the use of only the low-cost sensor. Figure 6.9 shows the robotic system and the map generated using it.

6.6.4 Large Scale Mapping

As demonstrated in Figure 6.10, the system is capable of producing large scale maps, despite not providing explicit loop closure handling. The example map provided has been generated using a Forbot robot platform with a Hokuyo UTM-30LX LIDAR mounted. The map shows an area of 200m x 200m, clearly demonstrating the scalability of the system.

In multiple other works, the system is used as a front end for efficiently generating maps and trajectories. A back-end system then is used to optimize the resulting pose graph. One example of this is [119], performing mapping using a Turtlebot platform and using their own pose graph optimizing back-end. In [75], the system is used as a SLAM front end and combined with Karto as a back-end to achieve very accurate indoor mapping in large scale scenarios. In [91], the system is used as a front end for a backpack-mounted 3D mapping system.

[4] `https://www.youtube.com/watch?v=gy9cDuaNW1w`

(a) (b)

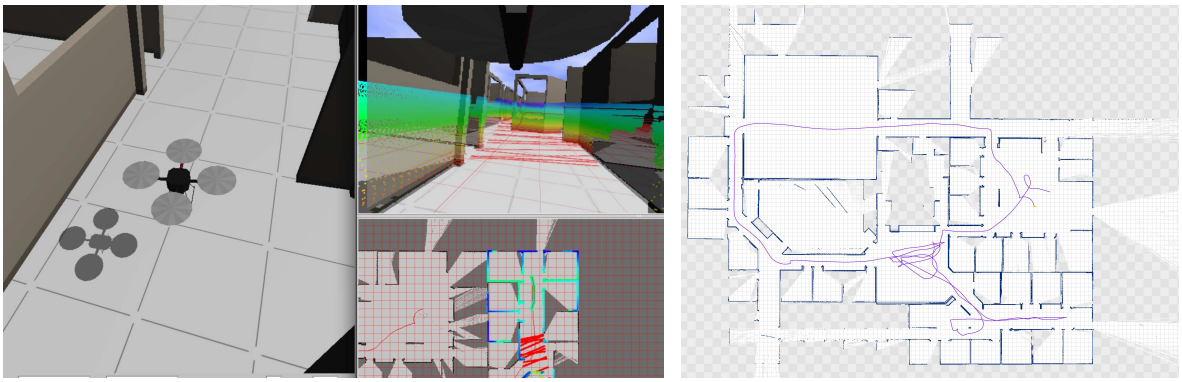
Figure 6.10: (a): The forbot robot with a Hokuyo UTM-30LX LIDAR mounted (b): Map generated by the system of the WTS museum in Koblenz, Germany. Note that the map is consistent, despite the large scale (200m x 200m) and no explicit loop closure handling. Photo and map courtesy of Johannes Pellenz.

6.6.5 Use on UAVs

The flexibility of the system is also demonstrated by using the datasets from [42]. These datasets are recorded onboard data from the City College of New York. A quadrotor equipped with a Hokuyo UTM-30LX LIDAR is used. The results obtained with our system show comparable quality compared to the results presented in the original work even though the system was not specifically developed or tuned for use with UAVs. Another example of the use of the system for real-time mapping on a quadrotor UAV is presented in [53].

A further example of using *hector_slam* on a UAV is available as a demo for the *hector_quadrotor* ROS packages. The indoor SLAM scenario is easily reproducible on any machine able to run Gazebo and provides a complete simulation of a quadrotor performing mapping of an office environment. Figure 6.11 shows an example screenshot and the resulting map after the UAV was teleoperated through the environment.

In [101], the approach is used as the basis for another UAV-based mapping and navigation system.



(a) (b)

Figure 6.11: UAV Indoor SLAM simulation: (a): Screenshot of the GUI. On the left the Gazebo simulation environment is visible. On the top right the view of the forward facing camera is shown, with LIDAR point cloud and map data projected into the image. A top down ortho view is visible on the bottom right (b): Final map generated after teleoperation of the UAV through the scenario.

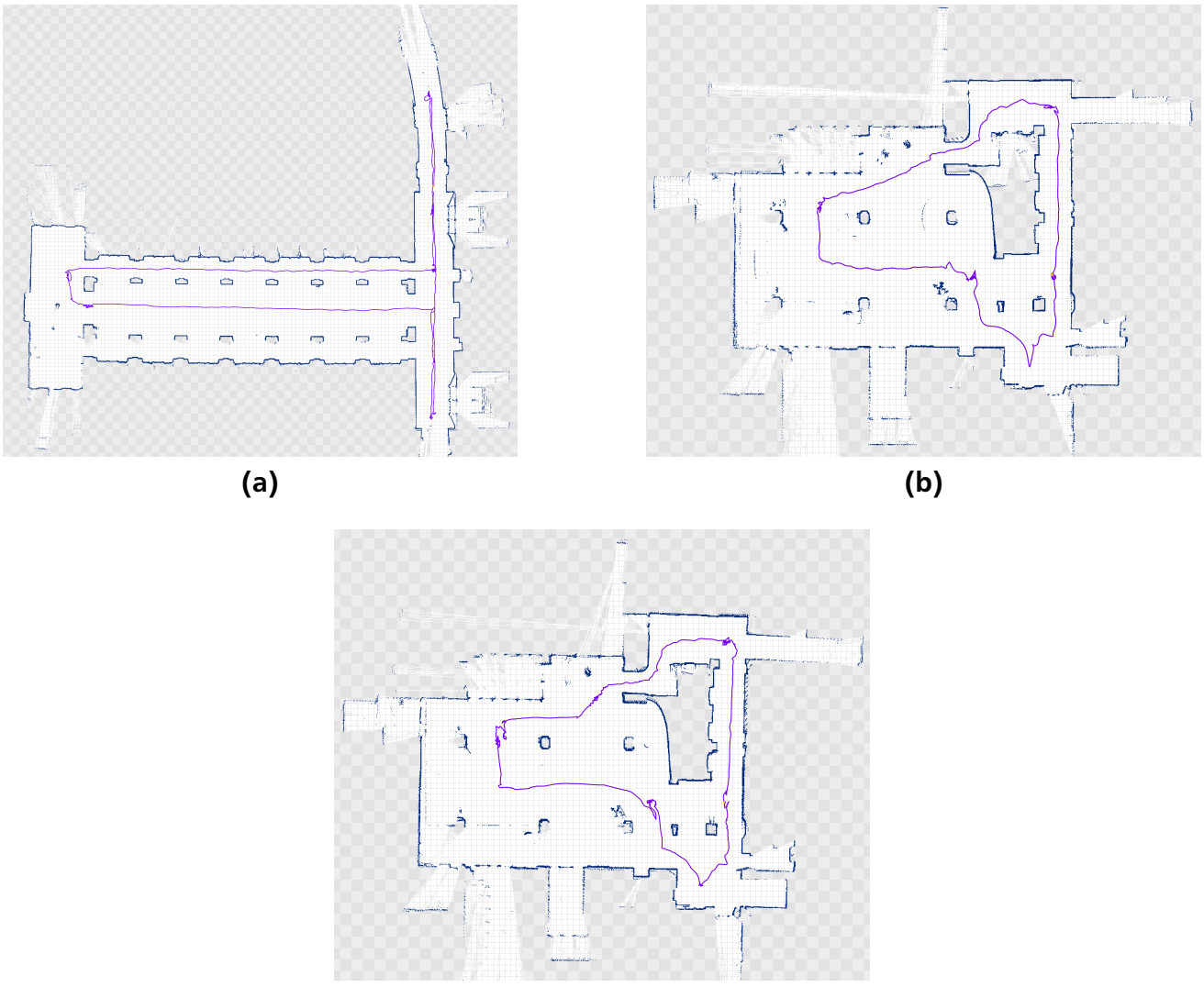


Figure 6.12: Examples of use of hector_slam on with quadrotor UAV datasets: (a): Hall dataset (b): Lobby dataset 1 (c): Lobby dataset 2

7 Navigation, Exploration, and Search for Victims

In USAR, the goal is to explore the environment, find trapped victim, extricate them and perform their medical stabilization. This chapter focuses on the first two of these tasks. An approach for exploration and victim search is detailed, enabling USAR robotic systems to perform both types of task with a high degree of autonomy. For evaluation, the scenarios of the RRL are used, as they are modeled by researchers at NIST to reflect many of the same challenges encountered during real USAR situations.

7.1 Related Work

Navigation for mobile robots is a frequently studied topic in recent years and existing solutions provide high performance in many different scenarios. The ROS navigation stack developed originally for the PR2 robot is widely used and has been demonstrated to provide consistent performance in office environments [96]. For Ackermann-steered vehicles such as the Hector UGV experiments showed that the use of the navigation stack is not feasible without significant modifications, however [90]. Another observed issue is high CPU consumption even when the robot is standing idle. This is at least in part caused by the online obstacle avoidance capability of the navigation stack, which is not as crucial in the USAR scenario.

The exploration of previously unknown environments has received a lot of attention in the research community. An approach that has found widespread adoption for robustness and simplicity is the exploration transform algorithm [161], which combines frontier-based exploration [162] with the path transform approach [165] for risk minimization. A benchmark of frontier based planning is provided in [65], demonstrating that re-evaluating exploration goals before reaching them can be beneficial for exploration time. This approach is also used in the exploration approach described below. An approach the exploration using multiple USAR robots leveraging dropped Radio-Frequency Identification (RFID) tags is described in [167] and evaluated using USARSIM both in simulation and the RoboCup Rescue Simulation League. This approach does not take the search for victims into account explicitly, however.

The focus of exploration approaches in the research is frequently focused on generating a complete map of the environment [23]. In [2] a comparison of exploration approaches for pure map exploration is provided. Once a part of the environment has been covered by the main sensor used for mapping, it is considered observed and the robot proceeds with exploration. As modern LIDAR sensors have ranges of tens of meters, a robot does not necessarily have to come close to a location that is considered observed with such approaches. This is crucial and problematic in cases when the range of some sensors is significantly lower than the range of others. If a victim detection sensor has only low range compared to the LIDAR sensor, a majority of victims will not be detected by a pure map-based exploration approach. In [13], the lower range of victim detection sensors is considered, but evaluation is based on map coverage.

The tradeoff between exploration and reporting the position of found victims is discussed in [138].

The Multi Autonomous Ground-robotic International Challenge (MAGIC) 2010 incorporated navigation and mapping challenges similar to some of those encountered in USAR scenarios.

While a larger scale scenario with multiple robots performing cooperative mapping was considered for the MAGIC challenge, the terrain was not as challenging as the RRL arena. A good overview is available in competitor publications like [26]. A ROS-based system by another competitor is described in [122]. The approach of the winner team is described in [112]. During the MAGIC challenge, competitors demonstrated highly capable multi-robot SLAM systems. The USAR scenario considered in this thesis differs from MAGIC tasks in these important aspects:

- Terrain in the MAGIC challenge exhibited less challenging elements than USAR scenarios used within this thesis
- Objects of interest to be found were large enough and color coded, so they can be found over large distances.

The research and achievements by MAGIC competitors are thus largely complementary to the contributions of this thesis.

7.2 Contribution

The contribution detailed in this chapter are multiple components for the exploration and search for victims in USAR situations that together allow for reliable autonomous navigation and detection of victims. In contrast to prior work, a holistic view is taken, incorporating aspects from path planning, exploration, recovery on failure as well as victim detection and approach strategies. The system has been demonstrated to be superior to competing approaches, as demonstrated during annual participation in the RoboCup RRL competition. Related publications are [98], [83], [82] and [84].

Multiple people have contributed to the system the contributions of this thesis are part of. These contributors are named here: Johannes Meyer(state estimation, motion control and world model), Thorsten Graber (elevation and cost mapping), Karen Kurowski (behavior control), Florian Kunz (base navigation), Mark Sollweck (basic exploration planner), Dorothea Koert (octomap integration), Paul Manns (differential drive control), Paul Schnitzspahn, Mykhaylo Andriluka and Konstantin Fuchs (victim detection).

7.3 Human-Robot Interaction and Supervision Aspects

The contributions in this chapter focus on improving the autonomous capabilities of USAR robots. With currently fielded robots using teleoperation almost exclusively as discussed in Chapter 2, the introduction of autonomous functionality promises to make operation more reliable and offload work from the operator.

While fully autonomous capability is demonstrated in simulated disaster scenarios, this serves for exploration of capabilities. Given the difficulty of USAR, tasks it is likely that autonomous capabilities will be introduced slowly over time. The interaction between the operator (taking on a supervisory role in the future) and the robot should thus always be considered for system design.

The system allows for the full range of control from teleoperation to full autonomy, as visualized in Figure 7.1.

For teleoperation, the operator can control the robot using a gamepad, providing base and sensor head velocity commands that are directly translated to robot action via onboard controllers.

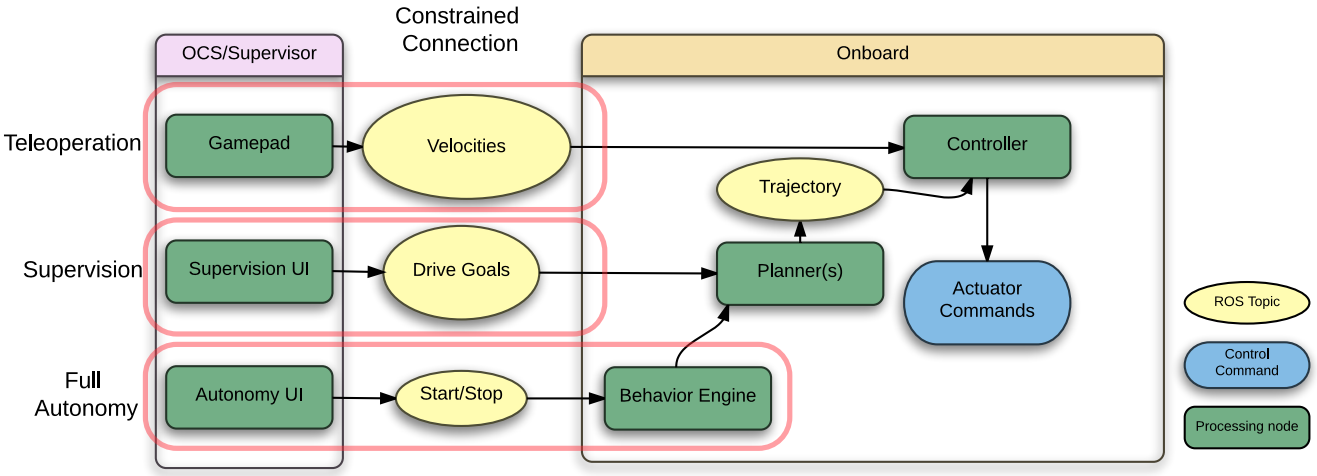


Figure 7.1: Supporting multiple levels of interaction for navigation of exploration and victim-search focussed UGV systems.

In assisted autonomy mode, the operator can provide goal points for the robot that the robot will try to reach autonomously. The robot avoids obstacles and leverages the planning system that is also used during autonomous operation.

In autonomous mode, the operator specifies the mission (for instance selecting victim search or pure exploration) and then sends a “Start” command. The robot then starts executing the mission autonomously, with the operator able to monitor and take over at any time.

7.4 Overview

Multiple researchers have contributed to USAR robotic system described in this Chapter. To facilitate comprehension of the following sections, a brief overview of the complete system and how the contributions in this thesis fit in is provided.

Figure 7.2 shows a schematic of most components relevant to this chapter. Components that are detailed in this chapter are shown in light blue, while others not focused in this thesis, but relevant for comprehension of the system are marked dark green. To reduce clutter, connections between components are not labeled but can be inferred from context. Dashed arrows denote ROS service calls, from caller to callee.

Using the *hector_slam* SLAM system described in Chapter 6 a 2D map of the environment is learned online. Fusing this estimate with IMU data, the pose of the robot is estimated.

Using the pose estimate, depth data from an RGB-D camera can be transformed into the global world frame. The resulting point clouds are used to generate a octomap-based 3D environment map in real-time. They are also used generate a 2.5D elevation map representation of the environment. This map is used for estimating traversability of the environment.

The cost mapping system fuses different map representations, generating a cost map that is used for navigation. Per default, the 2D map provided by the SLAM system and the elevation map are fused. Obstacles lower than the LIDAR sensor cannot be seen by the 2D SLAM system, so it is crucial to fuse information from the 2D focussed LIDAR and the 3D focussed RGB-D sensor.

The navigation system is based on a significantly modified variant of the ROS navigation stack. The motion planners are loaded as plugins into the corresponding process, hence the

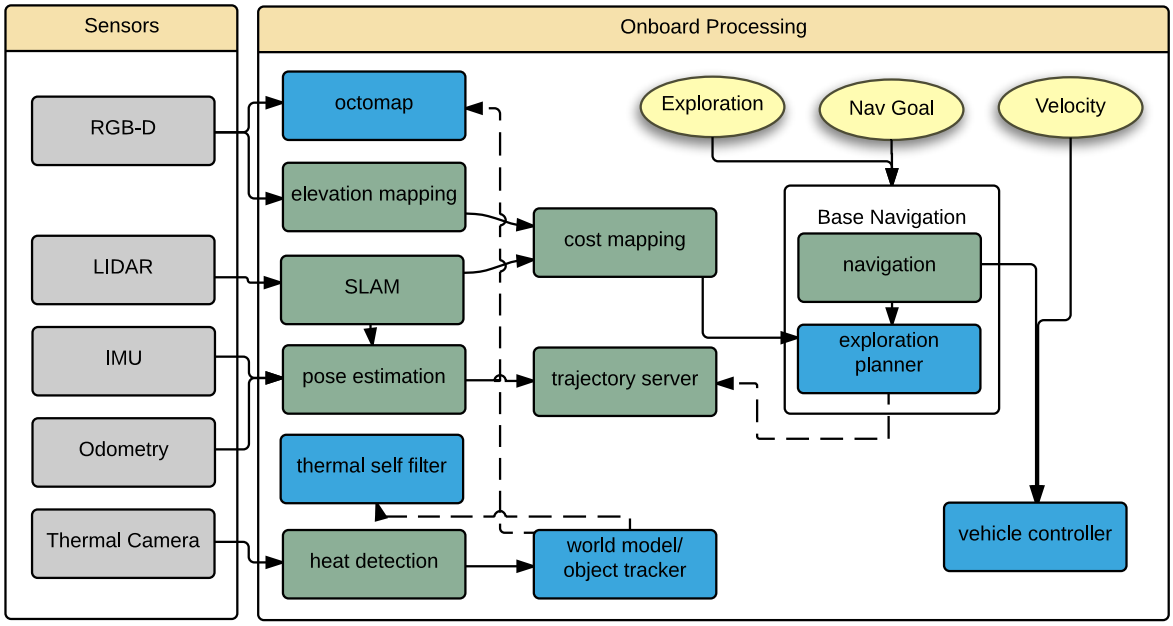


Figure 7.2: Navigation system overview schematic. Components that are detailed in this chapter are shown in light blue, while others not focused in this thesis, but relevant for comprehension of the system are marked dark green.

exploration planner and navigation system are shown within one container in 7.2. Depending on input by higher level systems or the supervisor as shown in Figure 7.1, the full range of control modes from teleoperation to fully autonomous exploration is supported.

A comprehensive overview beyond the contributions described in the following sections is available in [83] and [82].

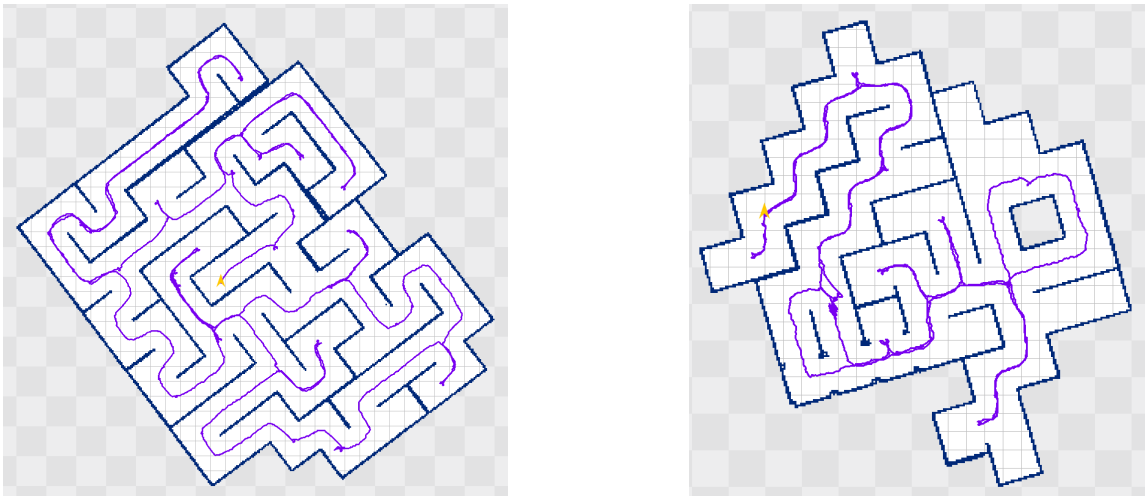
7.5 Exploration of Unknown Environments

The exploration of previously unknown environments is a crucial ability for accomplishing USAR tasks, as the primary goal is to search for and rescue trapped human victims. Exploration has to be thorough, as to not overlook victims in the environment and simultaneously the approach employed has to be highly reliable as to not block paths in the environment on failure or otherwise tie up response forces because it has to receive attention.

A crucial factor for reliability is hardware robustness. As this thesis focuses on software aspects, the reader is referred to the literature for a further discussion of UGV reliability [47], [27], [28]. In the following sections, contributions that allow reliable exploration are described. They contribute to robot capabilities for avoiding dangerous terrain, performing a thorough search for victims and objects of interest, and the ability to recover from failures.

7.5.1 Exploration Transform

The exploration approach used is based on the proven Exploration Transform [161] approach. This approach provides an elegant solution for the exploration of unknown environments, combining frontier-based exploration [162] with a discomfort/danger cost and a gradient following approach that allows finding a path without the risk of the planner getting stuck in local min-



(a) (b)

Figure 7.3: Examples for environment exploration: (a): Simulated artificial maze (b): Simulated Thailand Rescue Robot Competition 2012 arena

ima. Figure 7.3 shows two example maps of basic environment exploration using the exploration transform approach.

During evaluation during participation in the RRL competition, shortcomings of the original approach were observed, however. In the following, contributions that eliminate these shortcomings are described.

7.5.2 Inner Exploration

When no more frontiers between occupied and free space are available, the original exploration transform approach stops generating new goal points and paths for the robotic system to follow. This is undesirable if the robotic system is performing a USAR mission, as continuing exploration of the environment can contribute to creating a better environment model and finding victims that have been missed previously. To achieve this, the exploration transform approach is extended.

To provide goal points to the robot when no more frontiers are reachable anymore, the robot path traveled so far is retrieved. Positions on the path are used as a seed for the exploration transform algorithm. Then, all valid exploration transform positions are searched for the highest value. Afterwards a plan to the position with the highest value is planned and sent to the robot for execution. This approach results in the robot exploring positions that are farthest away from the path it has traveled so far, exploring closely areas that have not been visited before.

7.5.3 Recovery of Stuck Mobile Robots

While the risk of the robot getting stuck is minimized by avoiding non-traversable obstacles, this cannot completely prevent robots from getting into situations where a plan cannot be followed. It thus is important for robustness to provide recovery options that allow the robot to try and free itself from a stuck situation.

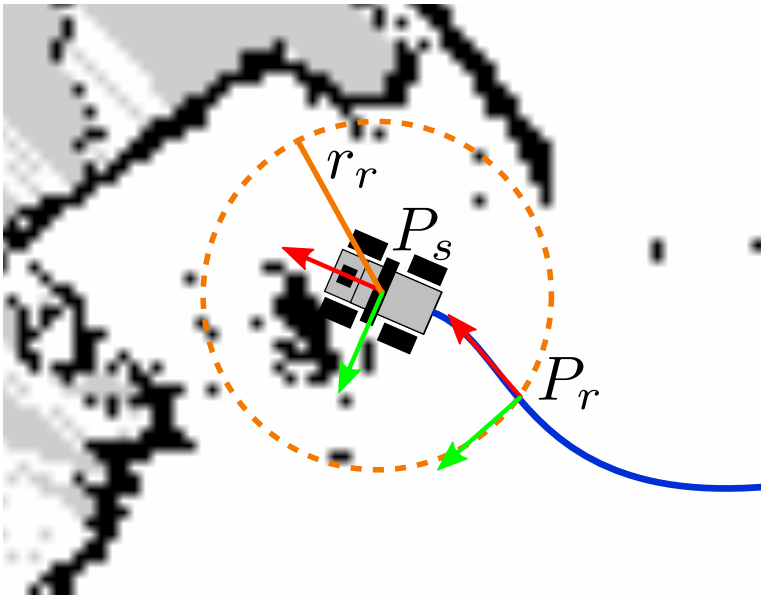


Figure 7.4: Inverse trajectory recovery: The robot is stuck at pose P_s and previously traveled on the blue path to get there. Based on a circle with radius r_r around this pose, a recovery pose P_r lying on the path toward the current robot pose is selected and sent to the vehicle controller as a target pose to attempt to recover the robot.

The ROS navigation stack provides recovery behaviors, but these are not applicable in USAR scenarios as they clear cost maps or try to rotate the robot in place. In the tight spaces and harsh terrain of a USAR scenario, such a strategy can lead to damage to the environment, robot or both.

Data: Robot pose P_s, traveled path T_r, threshold radius r_r
Result: Stuck recovery pose P_r
initialization;
for *Reverse iteration through path elements* t_i *of* T_r **do**
  distance $d_i = \|P_s - t_i\|$;
  if $d_i > r_r$ **then**
    return t_i;
  end
end
return invalid pose;

Algorithm 1: Recovery attempt through inverse trajectory search.

As an alternative, an inverse trajectory recovery approach that takes the path traveled by the robot into account is used instead as shown in Figure 7.4. Based on the current pose where the robot is stuck P_s, the path followed by the robot so far is traced back from end to start. A threshold radius r_r around P_s is used to determine the pose on a circle around the robot. This pose P_r is selected as the recovery target pose and sent to the robot controller. By following this approach, information about traversable terrain is implicitly taken into account and the robot attempts to get unstuck via a path that was already traversed before.

7.6 Improved Search for Victims

The goal in USAR is finding and extricating victims. Contributions detailing approaches for performing a more thorough search and for reducing the false position detection rate are presented in the next sections.

7.6.1 Victim Exploration

To detect objects of interest in disaster scenarios reliably, a thorough search has to be performed by (autonomous) robotic systems tasked with the exploration of the environment. As noted previously, exploration of unknown environments has been well studied in the past, but only few approaches consider different sensing modalities. Often, an environment that has been covered by a LIDAR sensor is considered explored. This approach is sufficient for tasks like learning a map of an unknown environment, but it does not take into account the limited effective detection range of sensors when searching for objects of interest, like victims in a disaster environment. The reliable detection of victims trapped under rubble using thermal imaging requires the USAR robotic system to come as close as possible to permit detection even when only very small portions of the victim are visible to sensors.

Approaches that attempt to estimate the information gain for exploration goals have been proposed [144]. With victims potentially covered by small holes in the environment, estimating potential information gain for a given pose in the environment is much more challenging than in many other scenarios, however.

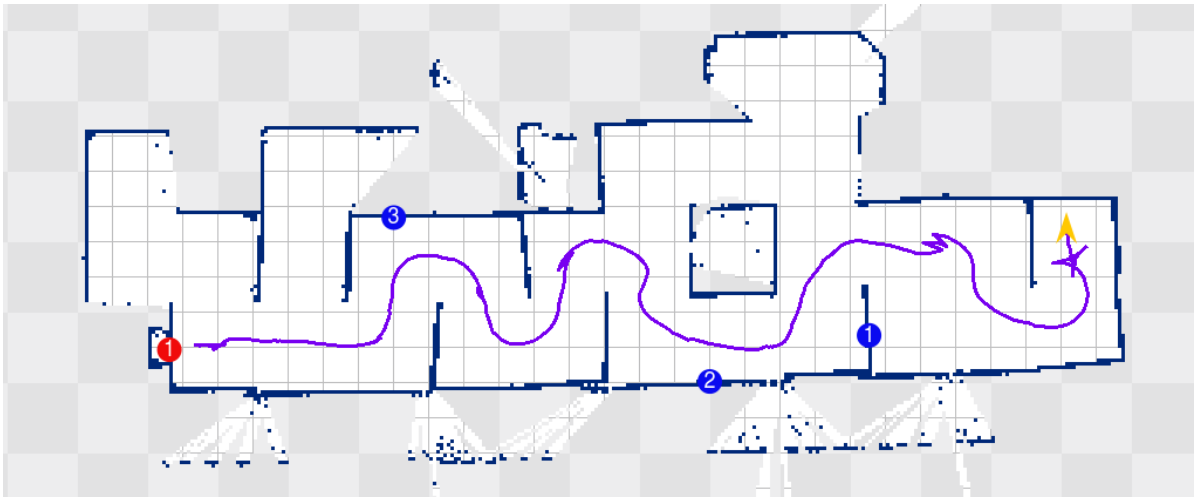
While approaches for an exhaustive search for voids in volumetric environment models have been demonstrated [39], computational complexity precludes their real-time application. As an alternative approach, instead of considering parts of the environment covered by sensors explicitly, the path planner is modified to generate paths that implicitly lead to improved coverage of the environment by relevant sensors.

This is achieved by not approaching frontier cells directly as in the original exploration transform approach but instead finding the closest free cell to the robot that is at least a threshold d_t away from the path traveled by the robot so far.

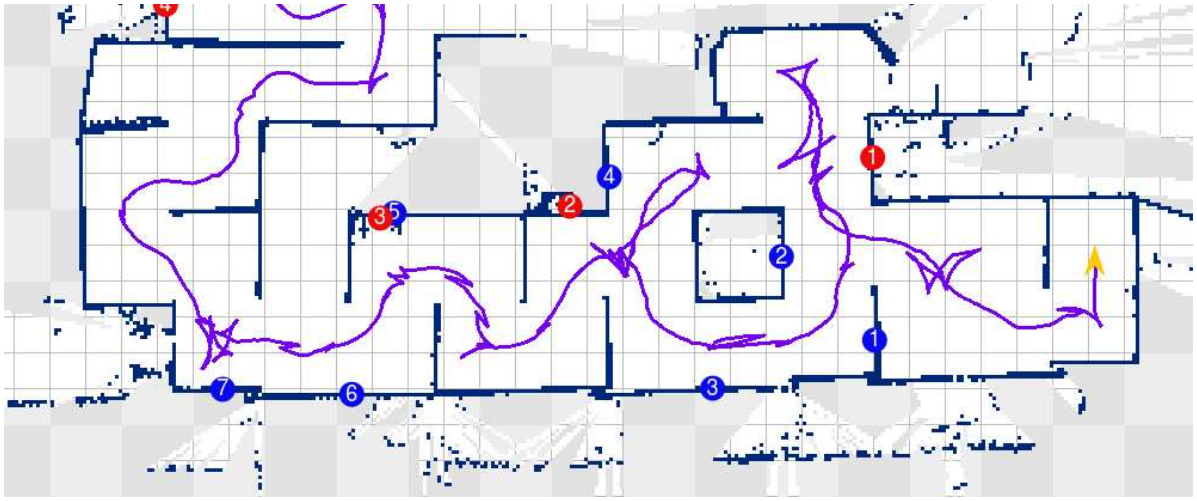
Effectively, this leads to the robot taking much shorter steps during exploration of the environment, resulting in better coverage. While a somewhat similar behavior could be achieved by artificially reducing LIDAR sensor range, an important advantage of the proposed approach is the fact that the distance per step can be varied dynamically at runtime. It is also possibly to switch between standard and victim exploration dynamically, which would not be possible when generating reduced range maps.

Figure 7.5 shows two examples from victim search missions at RoboCup 2014. For the upper map, the standard exploration approach was used. This resulted in the robot missing most victims in the map. In contrast to this, the victim exploration approach was used for the lower map. As is visible, it makes the robot visit also smaller cavities in the environment, resulting in more victims being detected.

In Section 7.9.1, a comprehensive comparison between both approaches is described.



(a)



(b)

Figure 7.5: Examples of victim search at RoboCup 2014: (a): Use of exploration transform. The UGV explores frontiers between known and unknown space in the grid map. (b): Exploration by using victim exploration, generating target poses close to robot path.

7.6.2 Robust Navigation towards Goal Poses

Both in autonomous and semi-autonomous operation, target poses for the robotic system might not be valid due to obstacles preventing the robot from reaching the target pose. Without careful consideration of this, the planner will report that planning failed as it is unable to plan to a given target pose. In many practical cases, however, a pose close by might be reachable, allowing continuation of the robot mission without interruption.

A common use case observed in the USAR scenario was USAR robots approaching victims for closer inspection. To reach those victims that are typically situated inside or close by to obstacles, just forwarding the victim pose to the planner as a goal pose would naturally fail. Instead, an *observation pose* has to be found that is sufficiently close to the victim as to provide sensors with a view that allows inspection and possible confirmation of the victim hypothesis.

To achieve this, an approach generating valid observation poses is added to the planner. Based on an obstacle transform map, an area around the target pose (2m x 2m per default) is sampled for valid poses the robot can reach. Only poses that are within free space (as opposed to unknown space) are considered valid here, as allowing unknown space can lead to wrong target locations in some degenerate cases.

The approach is shown in Algorithm 2. For all candidate poses, the vector from original target to candidate is checked against the original target orientation and the candidate only is used if the orientation difference is within a threshold. This is motivated by the fact that for observation of the target pose to be possible, it has to be in front of the robot. Simultaneously, this approach prevents the generation of target poses that lie behind walls. This could otherwise happen if thin walls are part of the environment. An example is shown in Fig 7.6.

The *observation pose* approach is used as a default setting when goal poses are sent to the robot. This notably includes cases where the robot tries to reach victim hypothesis for verification autonomously, but also cases where the supervisor provides a pose goal. Using the described approach, carefree planning is possible for both higher level behavior components and supervisor. Either the requested pose can be reached; in that case, a plan to it is created. If it is not reachable, the alternative *observation pose* is computed and sent to the planner.

Data: Robot pose P_V, goal pose P_g, map m, bounding box B
Result: Collision-free refined goal pose P_o
initialization;
Generate bounding box B around P_g;
Initialize closest distance d_{min} to max float;
for *all grid cells* g_i *of* m *inside bounding box* B **do**
    if g_i *not free cell* **then**
        continue;
    end
    Compute angle σ_i between P_g and g_i;
    if $\sigma_i >$ *threshold* **then**
        continue;
    end
    Compute distance d_i between P_g and g_i;
    if *distance* $< d_{min}$ **then**
        $d_{min} = d_i$;
        $g_{min} = g_i$;
    end
end
Compute angle from g_s to g_i;
Convert target pose from map to world frame;

Algorithm 2: Generation of collision free goal poses.

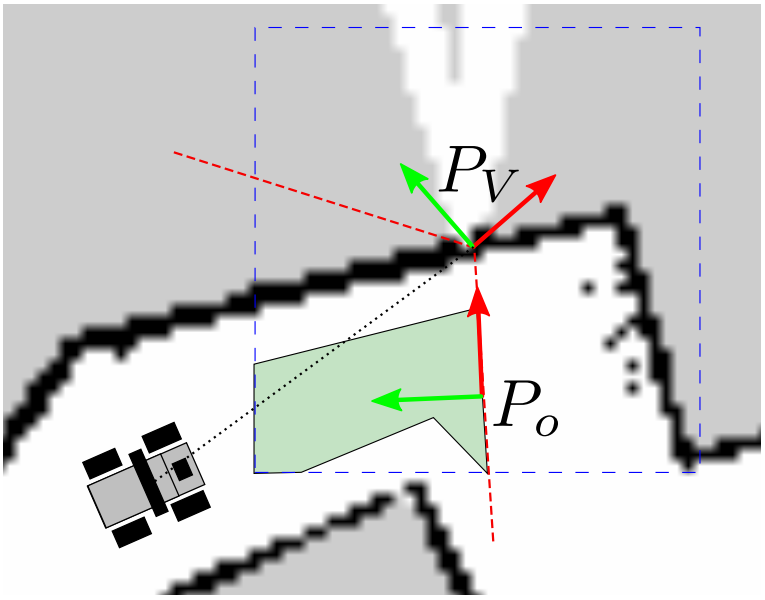


Figure 7.6: Generating observation poses: P_V is the original victim pose. Constraints based on victim orientation (dashed red lines), distance from original pose (dashed blue lines) and distance from walls result in the mint green polygon being searched and P_o getting selected.

7.6.3 Victim Verification

Victim detection using the UGV systems used within this thesis is mainly based on thermal sensors, possibly supported by other sensing modalities. A comprehensive description of the victim detection approach and semantic world model used for victim estimation is provided in [98] and a description of a method for detection of victims from standard camera images is provided in [4].

In [159], the distance to victim hypotheses is determined based on a Time of Flight (ToF) camera that is mounted parallel to the thermal camera used for victim detection. While the approach is demonstrated to work well, it requires a 3D sensor to be mounted alongside the thermal sensor. If the 3D sensor does not provide valid data at the time when a victim hypothesis is detected (for instance if it is dazzled by sun glare in outdoor light conditions), a distance measurement is not possible using the described approach.

In [60] a heat map based approach for victim detection based on low-cost sensors is proposed. Here, heat perception data is entered into a 2D occupancy grid map. The approach is using 2D data only and no provides no capability to perform 3D distance measurements, which is a disadvantage in complex environments.

The victim verification process is shown as a flowchart in Figure 7.6. Depth sensor data is not used directly for determining distances to victim hypotheses. Instead, the volumetric octomap generated onboard the robot is used and a raycast operation is performed into the octomap. This approach has two major advantages: There is no requirement to have a depth sensor mounted in parallel to the thermal sensor. This is advantageous as other sensors can be used for generating the 3D environment representation. Also, and perhaps more importantly, the distance lookup is performed on a aggregated 3D representation of the environment. It is sufficient if a 3D sensor

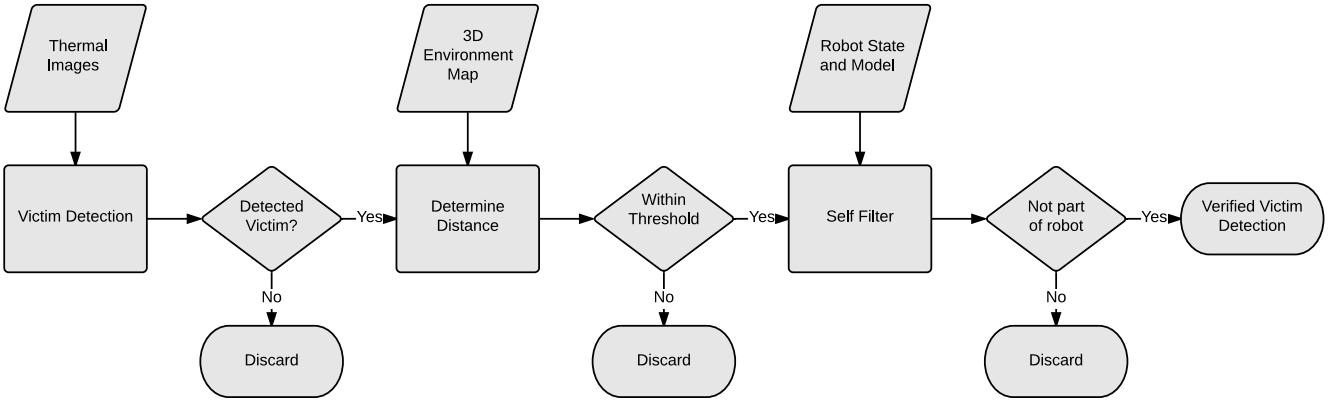


Figure 7.7: Flowchart of the victim verification process. After a victim detection candidate has been generated, a raycast-based distance query is first used to determine the distance of the potential victim. Afterwards, a self-filter verification check is used to determine if the detection is a part of the robot. Only if these tests pass, the victim percept is passed on to the semantic world model.

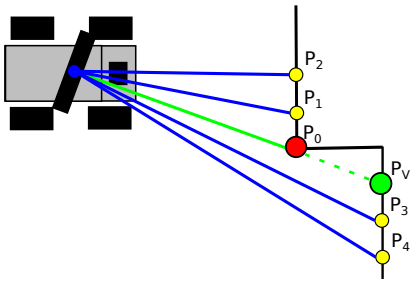
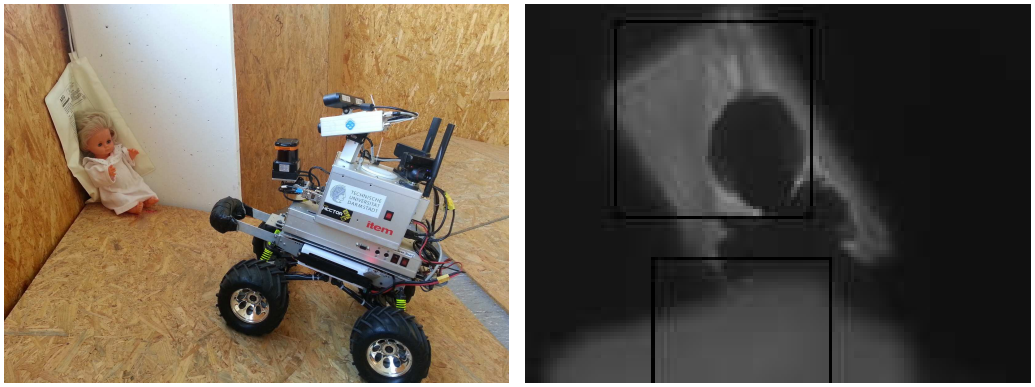


Figure 7.8: Example of a victim at P_V mapped to a corner P_0 due to sensor noise. The additional rays (blue) allow detection of the false mapping.

has observed the environment previously. There is thus no need for synchronized observation of the same area by both thermal and depth sensing at the same time.

In the distance verification step, using a raycast operation, an estimated distance for the victim hypothesis is determined. To avoid mapping victim hypotheses to wrong distances due to noise, a fan of rays is cast as shown in Figure 7.8. If a depth discontinuity is detected within the distance measurements, the victim hypothesis is discarded to avoid entering false victim positions into the world model. If no valid distance measurement is generated, the victim hypothesis is also dropped. For valid distance measurements, a robot relative cartesian pose for the hypothesis is computed and checked for being inside height thresholds. If the thresholds are exceeded, the hypothesis is discarded.

After the distance verification step, a self-filtering step is used to determine if the victim hypothesis has been generated due to robot parts being visible to the sensor. Depending on the hardware setup, the thermal sensor might be able to see parts of the robot during camera motion for victim search. During operation, parts of the robot, like drivetrain, sensors or computing hardware might heat up to temperatures that cause similar heat signatures to victims in the environment. An example situation demonstrating this issue is shown in Figure 7.9.



(a) (b)

Figure 7.9: Demonstrating the requirement for a thermal self filter step: (a): External view of robot after detecting a simulated victim (b): The thermal camera view. Both the heat blanket behind the victim and the heated up LIDAR sensor of the robot are detected as heat sources.

To filter these erroneous measurements, the hypotheses are converted into a point cloud representation based on the previously determined distance. In this representation, a single point represents a victim hypothesis. This point cloud is then used for a raycast against a simplified robot geometry model to determine if the victim hypothesis is a false positive caused by detection of a heated part of the robot robot. If this is the case, the hypothesis is discarded.

Only after these two verification steps, the *victim percept* is forwarded to the semantic world model system. There, it is used for a probabilistic update of victim hypotheses as described in detail in [98].

7.7 Path Following Control

As previously discussed, for USAR, is highly desirable that the robotic system follows planned paths closely, as the risk of getting stuck is greater than in many other environments. As the performance of available open source approaches was not deemed satisfactory during testing, considerations and implementation of path following controller are discussed here.

7.7.1 Common Base Controller

The common base controller uses a carrot-following approach similar to the one described in [64]. As described in [12], path shortcutting and oscillation can be drawbacks of carrot-following control when for highly dynamic vehicles. As the USAR scenarios considered here, these effects play a negligible role, as vehicles move relatively slow compared to many other applications.

The carrot following approach is used as a common back-end for both wheeled and tracked vehicle path following control.

7.7.2 Ackermann-Steered Vehicles

The four wheel steered Hector UGV has limited capability to perform holonomic motion. This capability is leveraged for path following control, with the system using sideways motion to align with the path without having to re-orient the whole vehicle. This controller has is mentioned here for completeness and has been implemented by Johannes Meyer.

7.7.3 Tracked Vehicles

The previously described controller for Ackermann-steered vehicles cannot be used for tracked platforms, as it uses a different kinematics model. A dedicated controller for tracked and differential drive vehicles is thus described here.

As most tracked robots can turn in place, a path following controller can leverage this capability. With a given maximum track velocity v_{max} and a wheel track d_w, the maximum angular rate is

$$\dot{\theta}_{max} = \frac{v_{max}}{\frac{d_w}{2}} \tag{7.1}$$

To follow the carrot moving along the planned path, the direction offset between the carrot and the robot is determined:

$$\theta_e = \theta_r - \theta_c \tag{7.2}$$

The desired angular rate is then computed using

$$\dot{\theta}_{desired} = \frac{\theta_e}{d_c * k_p} \tag{7.3}$$

where k_p is a proportional gain parameter for scaling the response. The maximum forward speed achievable with a given desired angular rate is limited by the speed the faster-moving track of the vehicle needs to achieve. The forward speed is thus computed to be

$$v_{desired} = v_{max} \frac{v_{max} - |\dot{\theta}_{desired}| \frac{d_w}{2}}{v_{max}} \tag{7.4}$$

Using this control approach, the tracked vehicles introduced in 3.1 can be used with the existing vehicle controller infrastructure to explore USAR scenarios.

7.8 Map Merging

In case of using multiple UGV systems in the same environment, it is highly desirable to establish a common reference frame between them. There are several benefits:

With robots capable of learning maps, merged maps can be generated. It is thus possible to let multiple robots explore different parts of the environment and provide a merged map to first responders. This is desirable as the manual alignment of multiple maps can be time-consuming and error prone, especially when maps are annotated with additional semantic information such as locations of victims or other objects of interest.

For coordinated autonomous operation in USAR environments, mutual localization of robots is a prerequisite for autonomous exploration, as environments are typically very narrow and the possibility of robots blocking each other have to explicitly be taken into account during exploration and motion planning.

In the literature, approaches using random walk and simulated annealing on grid maps [30] and later using a hough transform based approach [29] have been described. Another approach uses Voronoi partitioning and matches this partitioning between maps [131]. A disadvantage of this approach is the requirement to generate a Voronoi diagram from different maps with sufficient similarity. Another class of approaches based on graph optimization such as those described in Chapter 7.1 used in the MAGIC competition. These approaches require a complex back-end and tight integration between onboard SLAM and the map merging approach.

The map merging approach presented here uses feature-based matching that is based on methods used for image stitching [22]. By considering grid maps as images, standard feature detectors and descriptors can be used to extract salient features from them. Based on experiments with feature detector/descriptor combinations available within the OpenCV[1] library [20], Oriented FAST and Oriented BRIEF (ORB) [129] features were determined to be the best performing approach. Using fast binary features, this approach is known to not be scale invariant [63]. As the scale of both maps to be aligned is known, missing scale invariance is not a drawback for the map merging application.

The map merging approach ist described in Algorithm 3. First, maps are converted to images and the OpenCV standard ORB implementation is used to detect features on both. A feature matcher then computes matches between features from both maps. In a loop employing a Random Sample Consensus (RANSAC) [51] approach, three candidate feature matches are selected and an affine transform between them is estimated. This affine transform is then checked for plausibility based on scale. The quality of the fit is determined next and if not within a threshold, the hypothesis is discarded. Only if these early checks pass, the number of inliers is computed. The system per default computes a large number of RANSAC hypotheses and keeps the one with the highest number of inliers. The transformation with the highest number of inliers is returned at the end.

The described approach is advantageous in that it is independent of the onboard SLAM system used; as long as robots can provide a grid map representation of the environment, map merging can be performed. Experiments in Section 7.9.2 show that even noisy maps can be matched successfully.

7.9 Experimental Evaluation

A third party comparison of the open source exploration planning framework is available in [154], demonstrating that it provides advantageous obstacle avoidance capabilities compared to standard frontier-based exploration due to taking additional cost into account.

In the following two sections, specific aspects of the previously discussed contributions are evaluated. For a discussion of aspects relating to overall system performance in multiple challenging competition participations, the reader is referred to Chapter 9.4.

[1] http://opencv.org/

Data: Robot maps m_1, m_2, optionally known distance d_s between start points
Result: Merged map m_m, transform T_m
Generate ORB features f_1, f_2 for both m_1, m_2;
Compute matches c between f_1, f_2;
Initialize highest number of inliers $n_{max} = 0$;
for *fixed number of iterations* i **do**
    Randomly draw three matches c_i from c;
    Estimate rigid transform T_i between points in c_i;
    if *scale of* T_i *outside threshold* **then**
        continue;
    end
    for *points* p_1, p_2 *in* c_i **do**
        if $\|p_1 - p_2 T_i\| >$ *threshold* **then**
            continue;
        end
    end
    Compute number of inliers n_i by applying T_i to all matches;
    if $n_i > n_{max}$ **then**
        $n_{max} = n_i$;
        $c_max = c_i$;
    end
end
Compute rigid transform T_m using all inliers;
Compute merged map m_m by fusing m_1 and applying T_m to m_2;

Algorithm 3: Feature-based map merging.

7.9.1 Victim Search and Exploration

The primary goal of the contributions of this chapter is the efficient and reliable search for victims. In this section, the results of applying a conventional frontier-based exploration transform scheme is compared to the contributed victim exploration strategy.

Using simulation, the robotic system explores the environment shown in Figure 7.10. The scenario has a size of approximately 131 m^2 and contains 17 simulated victims. The difficulty of detecting victims varies between locations, as they are placed at different heights. To reduce the effects of noise in the experiment, it is repeated n times each for both approaches. The task of the robot is to explore the environment and find as many victims as possible. For this scenario, the robot is given 10 minutes time for each mission, with the counter starting once the "Start Mission" command is sent to the robot. As the simulation does not necessarily run in real-time, simulation time is used for all timing, ensuring fair measurements over all trials.

Using $n = 15$ trials each for both strategies, the robot explores the environment. As a metric, the number of grid map cells observed by LIDAR scan rays serves as an indicator for the area the robot explored. As the grid cell size is known, the covered area can easily be computed in more intuitive m^2. The evolution of the mean area explored using both approaches is shown in Figure 7.11.

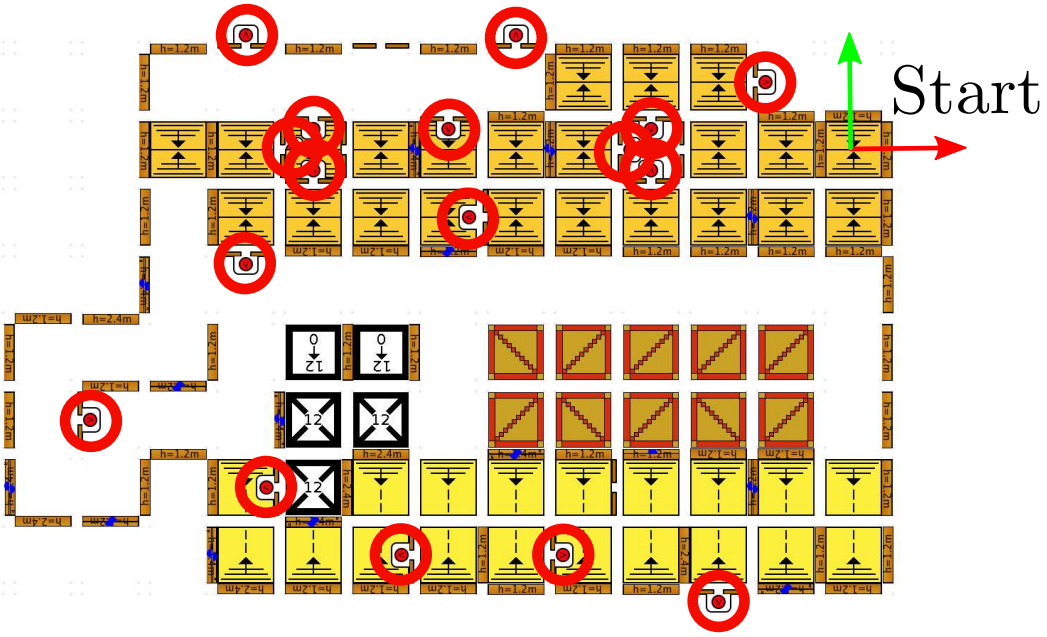


Figure 7.10: The scenario used for the exploration experiment. The start point for the robot is marked with the red/green axes. The victim locations are marked with red circles. Orange elements close to the start at the top of the map are half ramps and yellow elements at the bottom are full ramps.

The other metric used is the number of victims found by the robot. A time series evolution of the mean number victims found is shown in Figure 7.12.

Comparing both time series, it can readily be seen that the conventional exploration transform strategy covers more area in the same time. This faster exploration, however, comes at the cost of missing more victims than with the victim exploration strategy.

Figure 7.13 shows 4 example maps for each of the two exploration strategies. The maps have been randomly drawn from the 15 maps each generated for both exploration strategies. With the path of the robot shown in purple and victim locations in red, the differences between approaches are visible.

The map coverage for the exploration transform approach is larger, with the lowest part of the map explored in all instances. The path traveled also is smoother as plans directly guiding the robot to next frontiers is generated, as opposed to the multiple shorter distance goal poses generated by the victim exploration approach.

The number of victims discovered close to the start point in the left part of the arena is lower with the exploration transform approach due to the robot passing them by faster. It is notable that due to the inner exploration approach as discussed in Section 7.5.2, the robot revisits the left part of the arena in some cases, discovering more victims.

It is notable that although there is no inherent randomization happening in any of the onboard system components, there is variability in the paths taken and the number of victims found.

As discussed earlier, the tasks of exploration and victim search are contradictory in so far as exploration of a larger area of the environment invariably comes at the cost of a less thorough search for victims. The experiment confirms this and provides quantitative information about the relative differences for the given scenario.

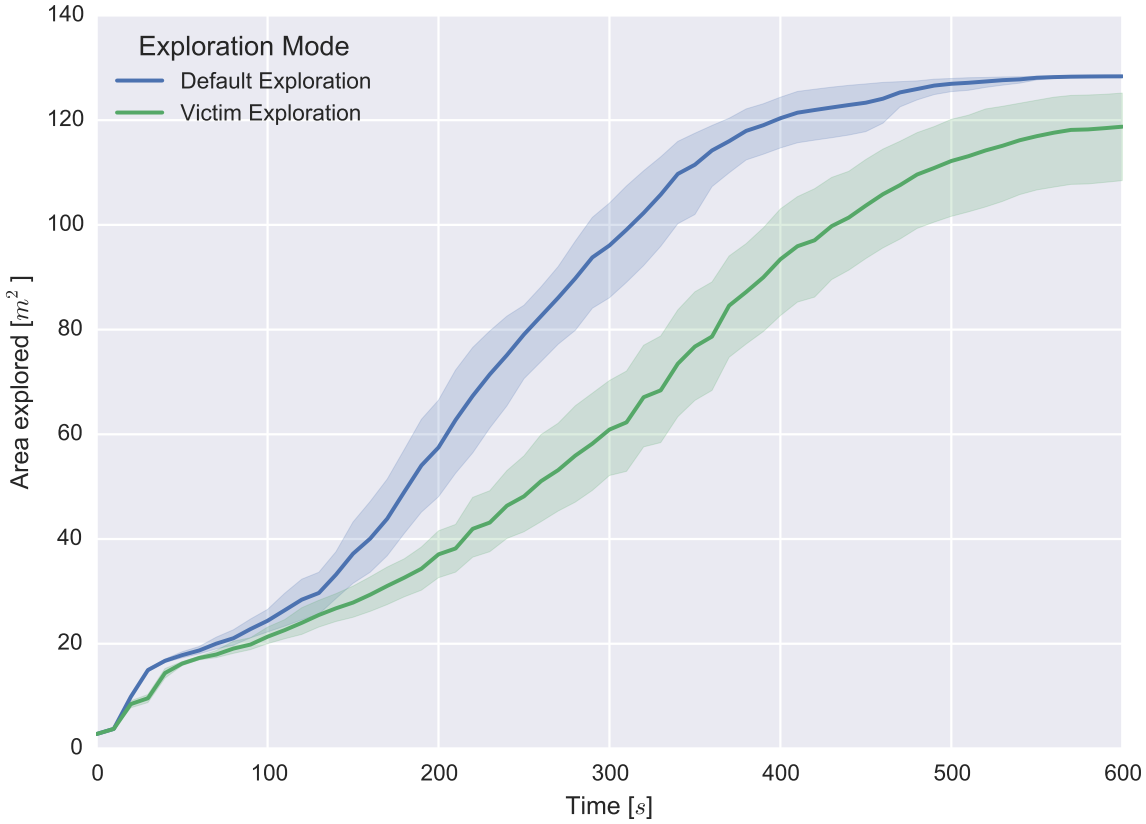


Figure 7.11: Time series visualization of the mean area explored using both methods. A 95% confidence interval is shown for both variants.

As the exploration approach computes the necessary data for planning online and is not using grid maps for tracking coverage, it is possible for the supervisor to change parameters and select an exploration strategy at runtime. It is thus possible to select the exploration strategy depending on the (possibly evolving) mission goals online.

7.9.2 Map Merging

In this section, experimental applications of the map merging approach are presented. For real world usage examples from the RoboCup 2015 competition, the reader might refer to Chapter 9.4 and specifically Figure 9.12.

As the approach is feature-based, there is no generally applicable rule how much overlap is required between maps; even small overlap can result in highly accurate merges if salient features can be detected and matched. Conversely, even large overlap can result in matching failures if no salient features can be matched.

Using exemplary test cases the applicability of the approach to real-world datasets and situations is shown. Figure 7.14 shows a typical application as it may be encountered in a USAR scenario where two robots capable of learning maps start close by each other and then explore the environment in different directions. This particular dataset and scenario were recorded dur-

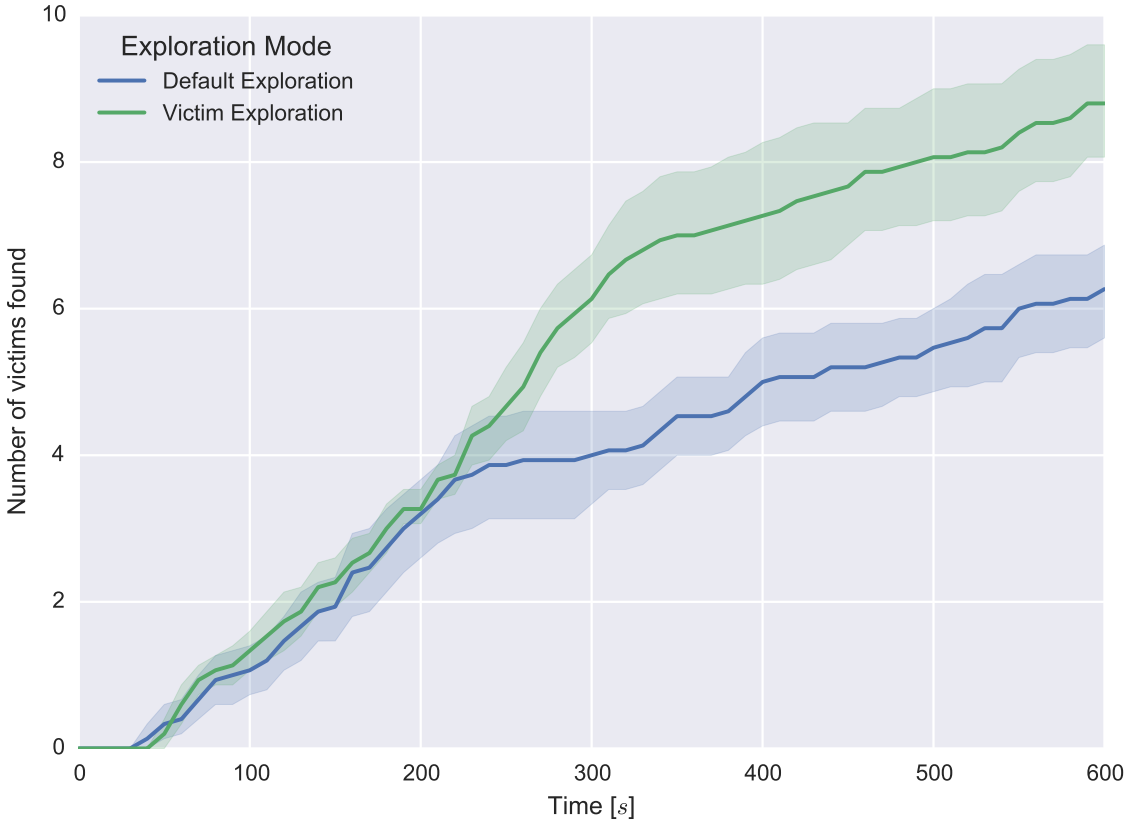


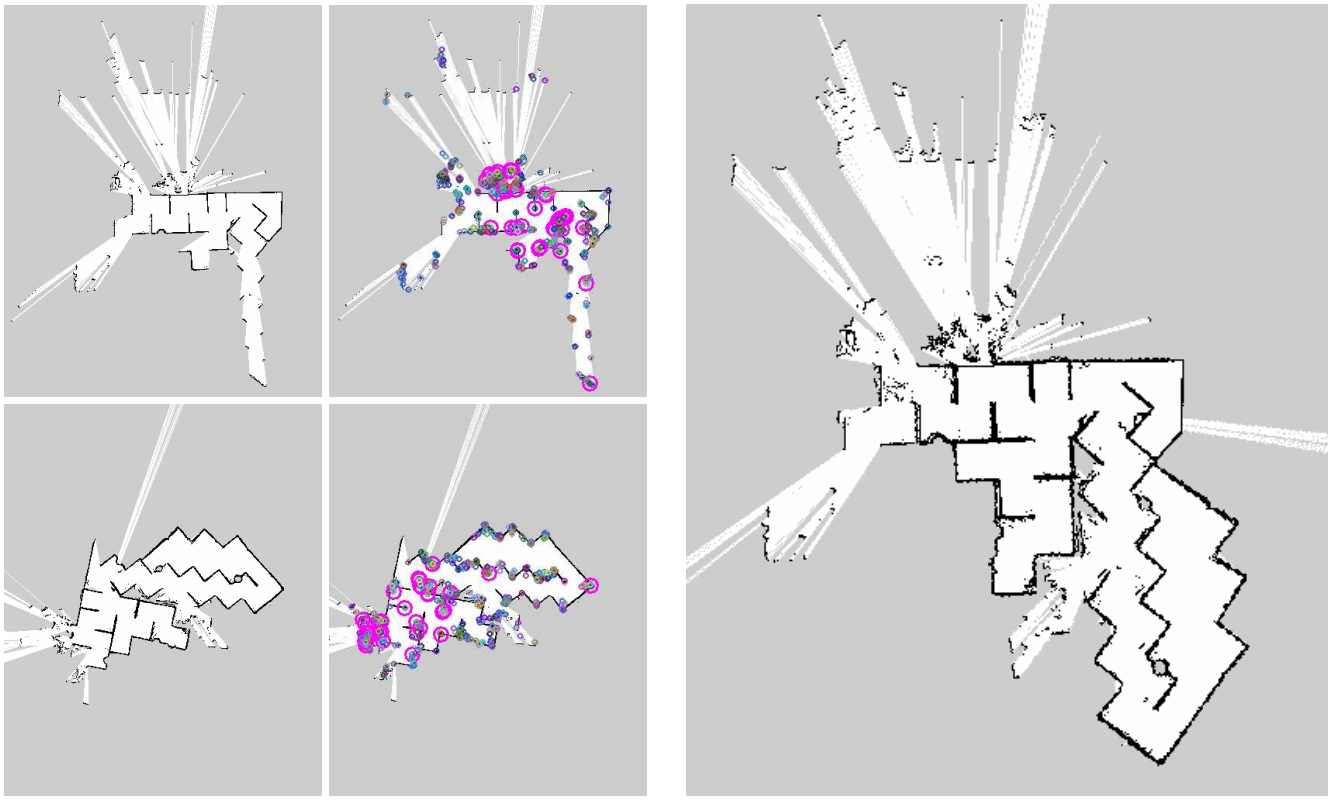
Figure 7.12: Time series visualization of the mean number of victims found using both methods. A 95% confidence interval is shown for both variants.

ing testing at RoboCup 2015. The initial overlap allows correct feature matching between both maps, despite the partly self-similar nature of the environment.

Figure 7.15 shows challenging scenario based on map data recorded at the RoboCup German Open 2011 competition. The upper map is partly erroneous as some walls of the environment were moved while the data was recorded. The lower map contains significant noise and erroneous occupancy information, as an IMU calibration issue resulted in erroneous filtering of LIDAR data, leading to scans of the floor being incorporated into the map. Despite these challenges, the resulting map is consistent and correctly registered. Per default, cells that are marked as occupied in one of the maps are also marked as occupied in the merged map; in cases like the shown one, this rule could be modified to trust one of the map providers more and let them overrule the other.

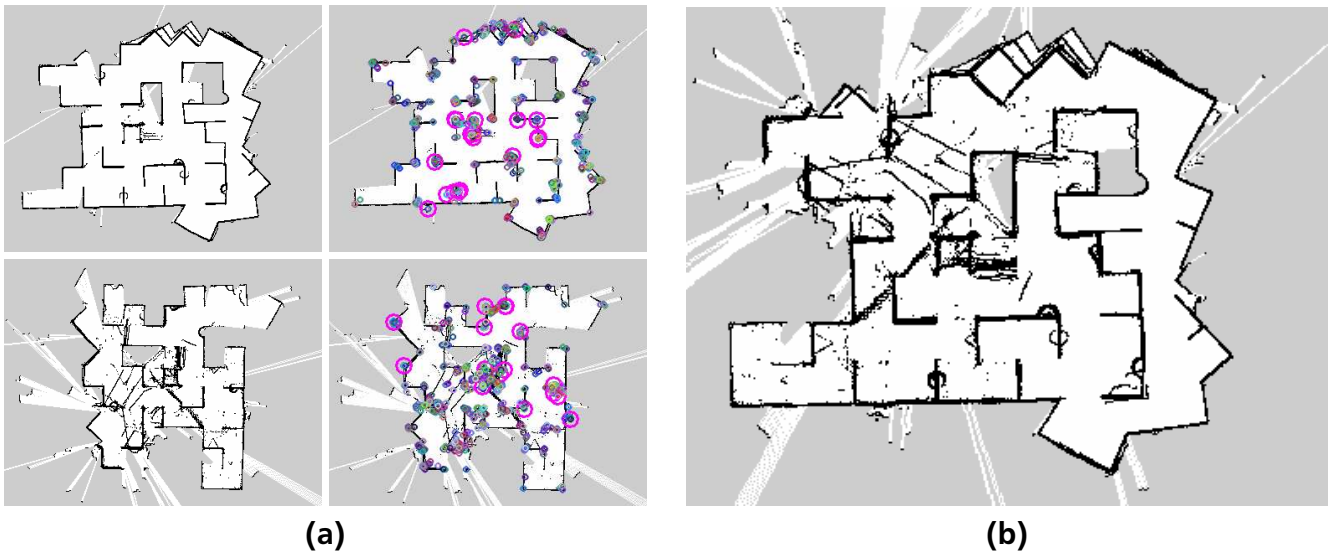
(a) (b)

Figure 7.13: Examples of maps generated during the exploration experiment: (a): Four out of 15 maps generated using the exploration transform approach. (b): Four out of 15 maps generated using the victim exploration approach.



(a) (b)

Figure 7.14: Map merging example: (a): Maps (left) and associated ORB features overlaid on them (right). Small circles are detected features and the larger magenta circles are features matched between maps. (b): The resulting correctly merged map.



(a) (b)

Figure 7.15: Map merging robustness to noise: (a): In the map on top, walls were moved during operation, leading to double walls in some portions of the map. For the lower map, erroneous IMU calibration lead to returns from the floor being added to the map. (b): Despite the significant differences, the map merging result is correct and consistent.

8 Manipulation for Complex Disaster Recovery Tasks

While the feasibility of exploration of unknown environments for autonomous robots has been demonstrated in Chapter 7, the ability to perform manipulation is often required in real-world environments. A ubiquitous example of a manipulation task is opening a door. There are however also many other scenarios in USAR that might necessitate manipulation, for instance performing shoring for the stabilization of a collapsed structure or the removal of debris that is preventing progress during exploration of the environment.

The nature of disaster environments makes it very challenging to perform autonomous manipulation. The main reason for the increased difficulty is the random nature of disasters, precluding the use of prior knowledge about the environment.

Instead, the operator has to make sense of the scene as he encounters it and then provide the robotic system with instructions on how to manipulate the environment. As detailed in Chapter 4, the nature of these instructions can range from pure teleoperation to task-level instructions. The cognitive task of making sense of the environment is thus adopted by the human supervisor in most cases. For this approach to work in practice, two important capabilities have to be available to the human-robot team:

The human supervisor has to be able to understand the environment. This means that sensor data collected by the robot has to be processed and transmitted to the operator in a way that allows reliable understanding of the scene and the achievement of SA. In a disaster response context, this is a challenging problem due to possible communication constraints between robot and operator station. These constraints can have different characteristics, such as low bandwidth, high latency or outages. A plain transmission of all sensor data to the OCS is thus often not possible.

In the opposite direction, the operator has to be able to reliably communicate intent to the robotic system. Ideally, this happens on a task level, with the operator providing high-level instructions. To provide robustness in case task-level capabilities fail, a teleoperation focused approach has to be available, however, to allow for proceeding with the task in a manually controlled fashion.

Research for the contributions presented in this chapter was performed while participating in the DRC with Team ViGIR. For this reason, research in this chapter concentrated on anthropomorphic humanoid robotic systems. However, as shown in Chapter 9.5, the contributions also generalize to other types of robotic systems.

8.1 Related Work

While humanoid robotics is an active research area already for a long time, the DRC program resulted in a wealth of research in areas such as controls, planning, and human-robot interaction. For the first time, humanoid robots and their human operators had to fulfill a variety of tasks in a common competition setup, shifting focus from specialized research topics such as walking or whole body control towards the realization of humanoid (and other) systems that provide comprehensive integrated perception, locomotion, and manipulation capabilities.

After the DRC Trials, publications by multiple competitors describe their approaches, but the majority of teams did not make them available as open source software that would allow for

reproduction of the presented results. The MIT DRC team heavily used optimization-based planning and control [48], using the Lightweight Communications and Marshalling (LCM) [102] as communication middleware and the Matlab-based Drake dynamics toolbox [148] as a planning and control back-end [1]. Team IHMC uses a proprietary middleware based on Java [72]. Both teams provide significant parts of their software as open source software, but so far do not provide a setup that allows running their full setup as used in the DRC in simulation.

In [43], a manipulation approach used with the BDI Atlas robot is described, focussing on some of the DRC tasks. In [11] another human-supervised manipulation control approach is described with a focus on the door DRC task.

For manipulation, bilateral teleoperation approaches allow teleoperation by the operator while the robot simultaneously provides force feedback. Albeit demonstrated to be a highly promising approach where applicable, potential stability issues when using bilateral approaches [160] make their use infeasible in condition with constrained and significantly varying communications conditions such as those considered in this thesis.

An overview of Team ViGIR's DRC related research is available in [85] and detail aspects in separate publications on footstep planning [146] and manipulation [127]. This chapter contains excerpts from these publications but provides further insights and details.

A highly informative general account on 'What happened at the DRC' by teams is available online [40].

8.2 Contribution

The contribution in this chapter is an approach for performing complex manipulation tasks under possibly time-varying communication constraints. Perception aspects and planning are considered in a holistic fashion, allowing for manipulation even under severe bandwidth constraints. As a back-end for other components, the system allows using the full range of control modalities from teleoperation to full autonomy. The applicability is demonstrated for challenging real-world scenarios. Related publications are [81], [85], [127], [126] and [146].

Multiple people have contributed to the system that the contributions of this thesis are part of. These contributors are named here: David C. Conner, Ben Waxler, Shawn Hanna (control, communication), Alberto Romay (Object template system), Alexander Stumpf (Footstep planning) Philipp Schillinger, Spyros Maniatopoulos (Behavior control), Felipe Bacim and Brian Wright (OCS).

8.3 Human-Robot Interaction and Supervision Aspects

For most manipulation tasks in disaster scenarios, the interaction with a human supervisor is indispensable, as human cognition is necessary to make sense of the environment state and decide on the tasks the robot has to perform.

For humanoid avatar robots as used within this chapter, safety of interaction is perhaps even more crucial than with more conventional UGV systems, as humanoids are always quite literally just one step away from falling. The exertion of unintended high forces on the environment thus has to be avoided.

[1] https://github.com/RobotLocomotion/drake

One of the goals of the contribution of this chapter is thus to permit manipulation of the environment state while keeping the robotic system safe. This is achieved by providing the supervisor with a rich environment model, by allowing her to use a predictive model for planning manipulation and by using a planning system that avoids collisions automatically.

8.4 World Modeling

The world model system has to provide state estimation and situational awareness to the supervisor-robot team. To effectively leverage the human supervisor's cognitive and decision-making capabilities, a state estimate of both the internal and external state of the system has to be made available via the often constrained communication link between robot and operator. With current state of the art sensors often providing sensor data at rates in excess of 100 MB/s this is both crucial and challenging.

The type of communication constraints under which the perception system has to work depends on used hardware and encountered scenario. Over the course of the DRC competition, multiple different communication constraints were emulated:

- In the VRC competition, a bandwidth budget for communication between robot and operator was allocated for each mission and communication was cut off after the budget was exceeded
- In the DRC Trials, communication was constrained by limiting bandwidth and introducing latency, alternating between a "good comms" and "bad comms" setting.
- In the DRC Finals, 3 communication channels were used, one 9600 baud line from robot to operator, one 9600 baud line in the opposite direction and one high bandwidth connection that is blocked for a period of 10-30 seconds

The world model system was designed and adjusted to provide situational awareness and state estimation for the operator under all of these conditions. To achieve reliable and efficient manipulation with a remote operator in the loop, obtaining 3D geometry data is crucial. In the following sections, the approach and components for providing SA to both human supervisors and the robot are described.

8.4.1 Sensors

To generate a comprehensive model of the environment, the available sensing modalities of the robotic system have to be fused and calibrated to be the most useful. For complex manipulation and locomotion with humanoids, two external sensing modalities have become a standard setup:

A LIDAR sensor used to generate a 3D point cloud model of the environment. With a range of tens of meters, the capability to read out intensity data per range measurements and largely range independent depth measurement accuracy and precision, LIDAR data is a crucial component of the perception system. As lightweight LIDAR systems commonly only scan a single scan plane of the environment, they have to be rotated or otherwise moved to generate full 3D data. This crucially also means that the creation and updating of a 3D representation is not instantaneous, instead taking time to get updated.

Property	Spinning LIDAR	RGB-D Type Sensor
Update rate [Hz]	40	5-30
Ranges/s	40,000	> 1,000,000
Range [m]	30	5
Range dependent noise	No	Yes
Outdoor capable	Yes	Partial

Table 8.1: Comparison of LIDAR and RGB-D type sensors. This table shows qualitative differences and detailed technical performance varies between actual hardware.

RGB-D sensing provided by stereo, time-of-flight or structured light cameras is the second modality used. As all three types of sensors produce an RGB image and additional depth information, they are viewed as belonging to one class for the purpose of discussion here. The sensors produce images with depth data at update rates of 5 to 30 Hz. Unlike the LIDAR that provides a planar scan, they provide a depth image with every update cycle. Comprehensive discussions of stereo and RGB-D sensor capabilities are available in [54], [77], [95] and [50].

Table 8.1 summarizes the qualitative differences between both sensor types. While LIDAR sensing provides a lower number of measurements, they are largely devoid of range dependent noise and more accurate. However, the need to aggregate data over time is a disadvantage when considering a moving platform or a dynamically changing scene.

8.4.2 World Model Server

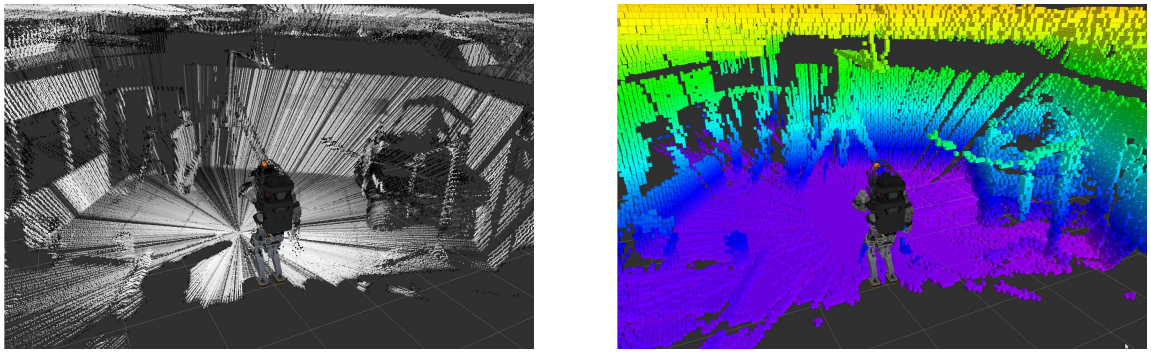
The world model server component preprocesses, collects and aggregates sensor data and makes it available to both onboard and OCS system components. Leveraging established open source libraries like the Point Cloud Library (PCL) [130] and octomap [68], the world model server allows queries of information about the environment with flexible level of detail and bandwidth consumption.

As described in Section 8.4.1, three-dimensional sensing is provided by onboard sensors, providing point cloud data. The setup used for this thesis is a Hokuyo UTM-30LX LIDAR and optionally an RGB-D type camera system. As RGB-D sensing generally has a smaller field of view, is sensitive to lighting conditions and has less consistent measurement accuracy, LIDAR data is used as the default main source for creating a 3D geometry model of the environment onboard the robot. To achieve this, the planar scans of the LIDAR have to be preprocessed and aggregated, so full 3D point clouds can be generated from them. The following preprocessing steps are employed:

First, scan data is filtered for spurious measurements commonly called "mixed pixels" that occur at depth discontinuities [151] [147] using the shadow point filter available as a ROS package.

The filtered scan is then converted to a point cloud representation. During this process, the rotational motion of the LIDAR on the slip ring is considered and high fidelity projection is employed, transforming every scan endpoint separately.

In the last step, parts belonging to the robot have to be filtered out of LIDAR data. To increase robustness against errors in kinematics calibration, a specialized robot geometry model uses simplified and enlarged collision geometries for self-filtering purposes.



(a) (b)

Figure 8.1: Visualization of sensor data: (a): LIDAR data with intensity information. (b) Resulting octomap representation.

LIDAR scans are saved to a ring buffer along with snapshots of coordinate frames used within the system. By employing this method, aggregate point clouds relative to different coordinate frames can be provided on request. A ROS Application Programming Interface (API) allows querying the world model via both ROS topics or services and retrieving region of interest point cloud or octomap data relative to different coordinate frames on demand. This capability can be employed by both onboard and OCS system components.

The primary onboard 3D geometry model is created using octomap, a volumetric, probabilistic approach using an octree as a back-end. Using this approach, the environment representation maintained onboard can be updated efficiently and in a probabilistically sound way. Even in case of changes in the environment or drift in state estimation, the environment model is updated accordingly and maintains a useful representation. Figure 8.1 shows both aggregated LIDAR data and a octomap representation of the same data.

The octomap environment model provides the main geometry representation and is used for multiple purposes. Using ray casting, distances to geometry can easily be determined. This feature can be used from within the OCS to perform ray cast distance queries against onboard geometry. In this case, only the ray cast information has to be transmitted to the robot and the distance information is transmitted back, utilizing only very low bandwidth.

The capability to request Region of Interest (ROI) data of the environment model allows to transfer small ROI geometry over the constrained connection on supervisor demand and also makes geometry available to other modules on request, like the footstep planning system. Similarly, it is possible to request 2D grid map slices of the octomap representation, aggregating 3D data into a 2D grid map. Using compression during transmission, this representation is very compact and often sufficient for supervisors to gain SA.

8.4.3 LIDAR Data Compression

In case of intermittent communication, the approach for querying the onboard world model for data from the OCS as described in the previous section can fail, as no data can be transmitted in periods of communication loss. Instead, it is desirable to transmit all geometry information available onboard to the OCS side as long as a communication window is available. A mirror of

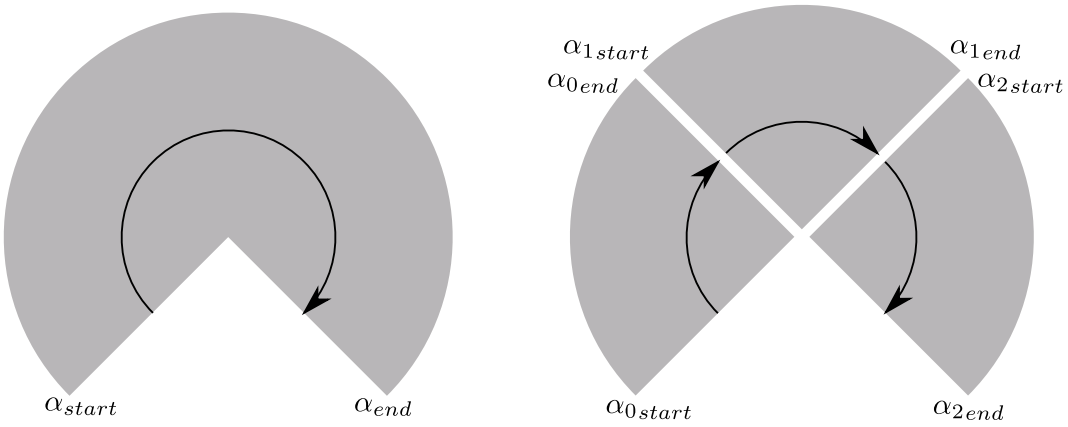


Figure 8.2: Splitting LIDAR scans for compression: A schematic view from the top is shown here and the rotation direction indicated by an arrow. A LIDAR scan can be described by the start angle α_{start} and end angle α_{end}. With a known angular resolution, scan points can be projected. To achieve a small packet size, the scan is split and intermediate start and end angles computed.

the world model can then be queried on the OCS side instead of relying on a connection to the remote onboard world model.

In case of intermittent communication between supervisors and robot, two instances of the world model server are used, one for the onboard/robot side and one for the OCS side. As direct transmission of point cloud data is error prone when experiencing packet loss, additional processing on LIDAR data is performed to make each packet compact enough to fit within a standard 1500 Byte User Datagram Protocol (UDP) packet and compress it as to be able to transmit a maximum of data during a communications burst.

For compression of LIDAR data, the Geographical Information System (GIS) research community developed solutions for large scale airborne LIDAR datasets [69], but these significantly differ in structure from those by small planar scanners. For this reason, an approach leveraging the special structure of data provided by planar scanners is presented here.

Direct transmission of point cloud data generated onboard the robot would cause prohibitive bandwidth cost as a point cloud representation with at least three floating point values for each cartesian point is not a compact one. For this reason, the natural and compact representation of a laser scan as an array of range values is leveraged and used instead. To fully reconstruct the 3D geometry captured by a single scan, a high fidelity projection of the scan has to be performed, however, taking into account motion of the LIDAR mirror during the data capture process. If this motion is not considered, scan data shows visible skew and ghosting (double walls) once it gets converted to a point cloud representation. The following approach is thus utilized:

- Perform a 3D high-fidelity projection onboard the robot and perform self-filtering. The onboard octomap and world model are updated simultaneously.

- Compress the scan data by writing the range values to a 2 Byte array representing millimeters and also encoding self-filtering information. Threshold and map intensity information to a single Byte.

Data	LaserScan [Bytes]	LocalizedLaserScan[Bytes]	Compressed [Bytes]
Header	≥ 16	-	-
Metadata	7×4	-	-
Ranges	4×1080	2×1080	$< \frac{1}{3} \times 2 \times 1080$
Intensities	4×1080	1080	$< \frac{1}{3} \times 2 \times 1080$
Total	8684	3240	< 1080

Table 8.2: Different LIDAR scan representations and the associated data size. As shown, the compressed size results in a packet size below the 1500 Bytes of a standard size UDP packet.

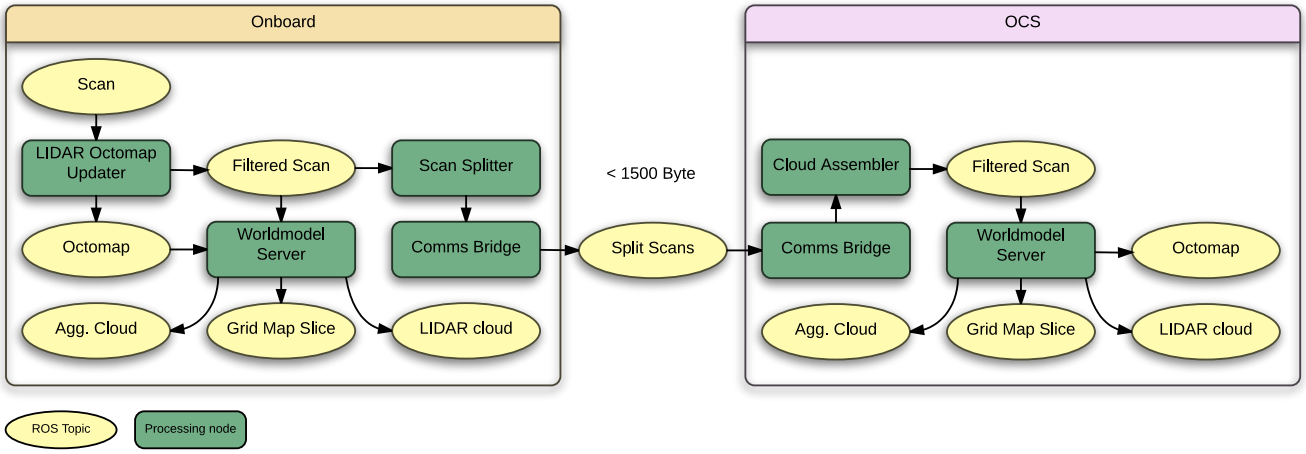


Figure 8.3: Overview of the world model server setup. The world model data is synchronized via the compressing LIDAR data and one instance of the world model server is running on the onboard and on the OCS side each.

- Add information about the scanner transform in world frame, one transform for the start of the scan and one for the end. This information allows performing a high fidelity projection of the scan after unpacking on the OCS side.

- Split the compressed scan into chunks that are small enough to be compressible to less than 1500 Bytes. A schematic of this approach is available in Figure 8.2. By using this approach, each compressed scan packet is a self-contained unit and can be unpacked and used on the receiver side without the need for packet reassembly.

On the OCS side, the compression process is reversed and resulting scan data is used to update the OCS world model. The size of a LaserScan message is dominated by the range and intensity fields. A Hokuyo UTM30LX-EW LIDAR, for instance, provides 1080 measurements per scan. For compression, floating point range values in meters are converted to millimeters and stored in an unsigned 16-bit number. Self-filtering of robot parts from LIDAR data requires knowledge of the whole transform tree of the robot and thus has to be performed on the onboard side if transmission of high bandwidth transform data to the OCS side is to be avoided. Per default, self-filtering is thus performed onboard and compressed laser scan data is annotated with a single bit per scan point indicating if the self-filter determined it belongs to the robot or objects attached to the robot.

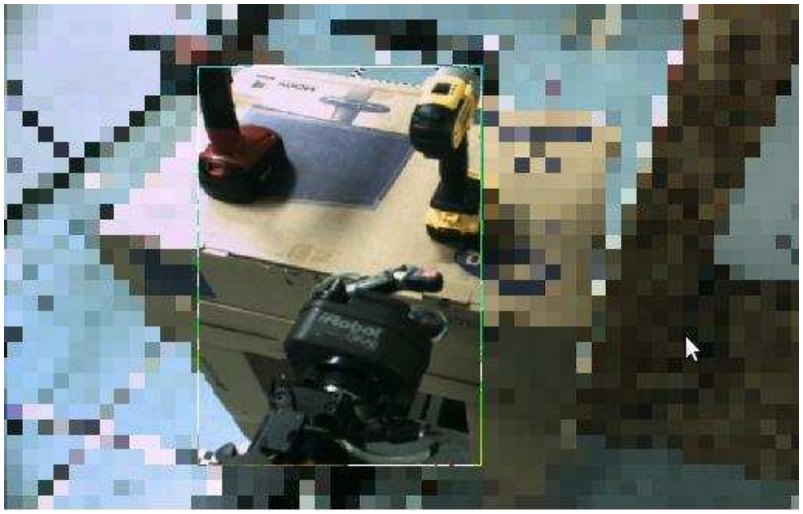


Figure 8.4: Variable resolutions for image data request. While the full camera image is transmitted and shown only at very coarse resolution, a region of interest specified by the operator is shown in high resolution.

Intensity data is converted from a floating point intensity to an unsigned 8-bit number. Here, a loss in fidelity is acceptable as intensity is mainly used for visualization and a range of 2^8 values is sufficient for presentation to the human supervisors.

Table 8.2 shows the different scan representation and their relative size. In Figure 8.3, the setup using one world model instance each on the onboard and OCS sides is visualized. The synchronization is performed using the previously described compressed scan transmission mechanism.

8.4.4 Sensor Data Processing for Situation Awareness

To provide the supervisor(s) with the necessary SA for complex manipulation tasks, not only geometry, but also image and texture data are crucial. In this section, components allowing for the processing of sensor data to achieve suitable representations and visualizations for obtaining supervisor SA are discussed.

Region of Interest Image Data

As images are readily compressible using standard compression methods, providing such data to the operator is often possible and can be feasible even when bandwidth is constrained. Often, only a limited region of interest in the full image is required. Examples are visually inspecting the quality of a grasp or the accuracy of end effector positioning. To provide this capability, the operator can request full image and region of interest independently, making it possible to show coarse resolution full images, but high-resolution regions of interest. To minimize communication requirements, an optional video frame-rate is part of the request and images can be sent at a fixed rate without the need for bi-directional communication. Figure 8.4 shows an example of a coarse resolution full image and a high-resolution ROI being used.

(a) (b) (c)

Figure 8.5: Different sensing modalities: (a): RGB camera image (b): Stereo depth image (c): Depth image generated from aggregated LIDAR data from stereo camera view point. It provides depth information also in parts of the image where no stereo-based depth data is available.

Mesh Generation

To provide a high-fidelity format for this 3D geometry data, an infrastructure for generating meshes out from both LIDAR point clouds and camera or LIDAR depth images was developed. Compared to plain point cloud visualization, this approach allows for a clear view of geometry and texturing of mesh surfaces, which allows for easier scene understanding by human supervisors.

Figure 8.6 shows a schematic of the mesh generation data flow. As indicated by the light blue OR gates, the mesh generation process can be based on different kinds of input data. Based on depth images, a mesh can be generated using a FastMesh [67] approach. The depth image can either be provided by an RGB-D type camera or it can be generated from LIDAR data. In the latter case, data has to be aggregated over time, however. Instead of depth images, LIDAR-based point clouds can also be used for mesh generation; in this case, the mesh is generated from LIDAR point cloud data directly. This approach does not have the restricted field of view of the depth image based one.

Figure 8.5 shows a comparison between stereo and LIDAR based depth images. Due to missing texture, large parts of the stereo depth image contain no data. Given the complementary advantages and drawbacks of both sensing modalities, as also described in Section 8.4.1, the supervisor can choose which option to use for mesh generation.

An example of generating meshes based on stereo camera RGB and depth data is shown in Figure 8.7. Three novel rendered viewpoints are shown, demonstrating how the approach combines the fidelity of image data with 3D geometry.

Fisheye Camera

The Atlas robot cannot perform rotation of the Multisense sensor head around the yaw axis, greatly limiting the field of view of the main sensor system. Prior to the Atlas v5 arm upgrade, this issue was much more severe, as the volume of good manipulability for the arms was outside

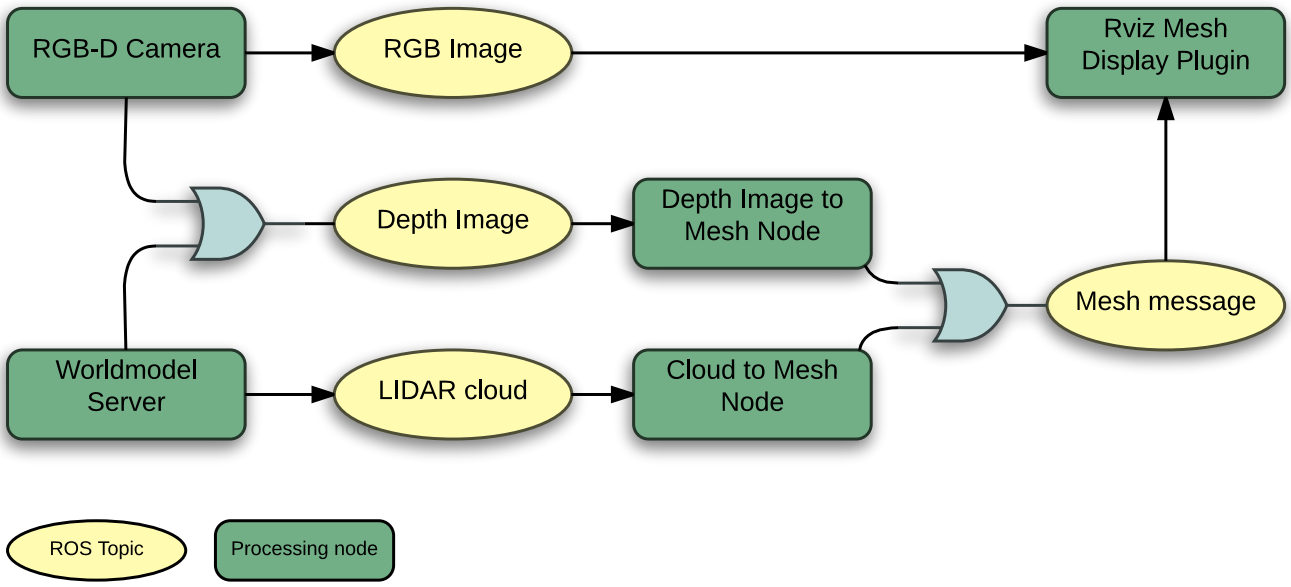


Figure 8.6: Options for generating a mesh representation of the environment. The RGB camera image gets texture mapped on a mesh generated from LIDAR or depth image data. The depth image either is either provided by a camera directly or can be generated from aggregated point cloud data.

the Multisense sensor field of view. To remedy this issue, a system for rectification the Fisheye lenses of the fisheye cameras was developed. Using a ROS integrated version of the OCamLib library [132], the fisheye distortion is calibrated. This allows generating novel rectified views from fisheye images not exhibiting severe distortion that otherwise makes judging of spatial relations difficult for operators.

Recomputing the rectification online, the system can track arbitrary frames on the robot or in the environment. It is thus possible to create a virtual pinhole camera that for instance tracks an end effector of the robot. Figure 8.8 shows both the fisheye image and a rectified view of the left hand of Atlas.

8.5 Manipulation

For manipulation, motions to move manipulators into desired configurations for grasping or other tasks need to be generated. As it can reduce operator workload considerably, a crucial capability is automated collision avoidance, both considering self-collisions of the robot (e.g. arm coming in contact with torso) and collision of robot parts with the environment. When performing manipulation in contact with the environment, motion must not lead to unplanned high internal forces acting on the robot, as these can quickly lead to damage to the robot, especially if it loses balance as a result. While force or admittance control approaches can reduce this risk, they are often difficult to implement due to limited force sensing and control performance on real systems. Preventing unintended contact in the first place thus serves as a risk reduction measure.

As high latency limits the usefulness of otherwise promising approaches for teleoperation of end effectors that rely on real-time feedback [92], direct control is not feasible. Instead,

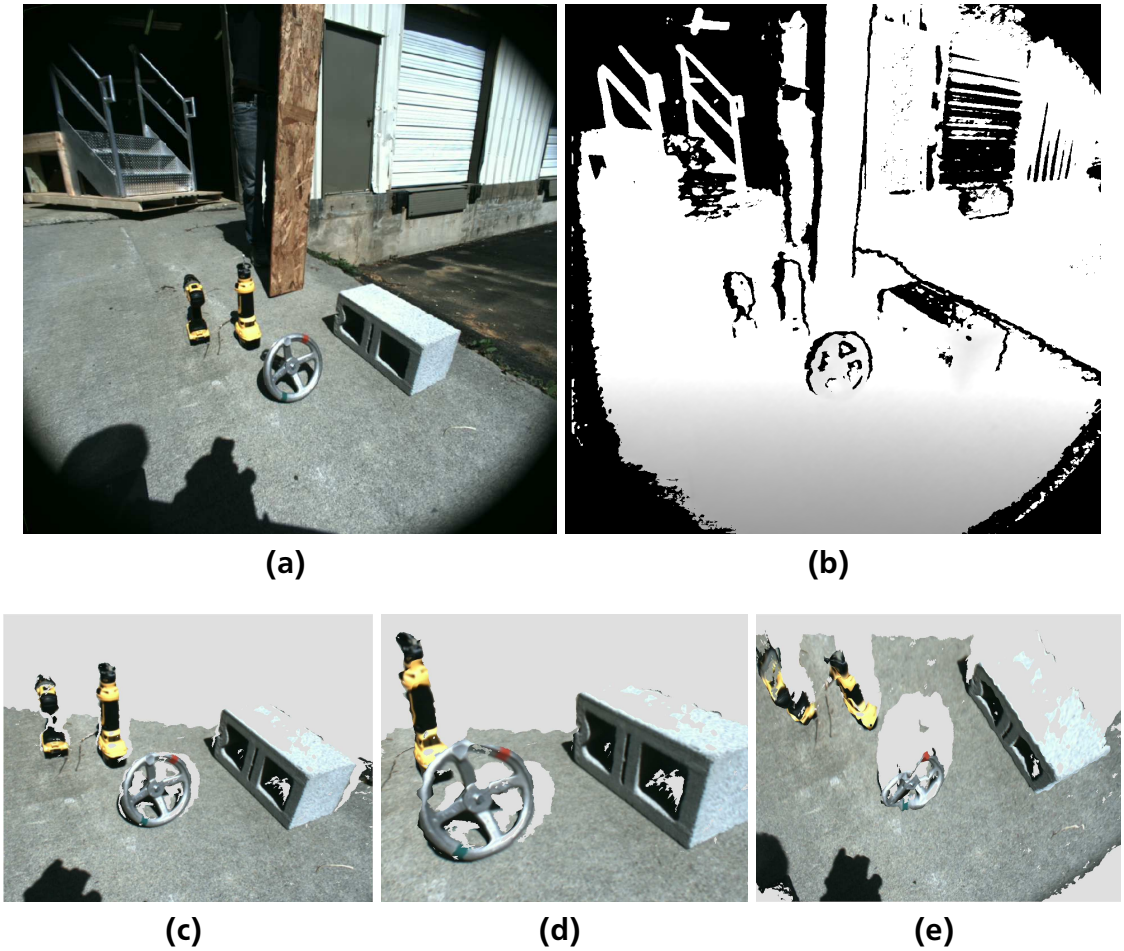


Figure 8.7: Rendering novel views based on textured mesh data: (a): RGB Image (b): Depth Image (c), (d), (e): Novel view points rendered based on applying texture to a mesh generated from the depth image

the supervisor(s) specify goal joint configurations or cartesian goal poses and requests robot onboard systems to reach them.

8.5.1 Kinematic Calibration

Joint angle sensors frequently exhibit a joint angle bias offset as well as a scale factor offset with respect to the true joint state. Without calibration of these effects, the joint angle configuration as measured by the joint sensors and the true configuration differ, making accurate manipulation difficult or impossible.

For kinematic calibration, a proven bundle adjustment approach previously used on the PR2 robot [117] available via the *calibration*[2] ROS package is adapted for use with the Atlas robot. To perform calibration, a calibration checkerboard is attached to the arm end effectors. A dataset comprising multiple camera views of the checkerboard using different arm joint configurations is then recorded. Due to the bundle adjustment approach employed by the calibration system, the transform between checkerboard and end effector does not have to manually specified, but

[2] `http://wiki.ros.org/calibration`

(a) (b)

Figure 8.8: Rectifying fisheye images for SA: (a): Raw fisheye image. (b): Simulated pinhole image of robot end effector.

can be estimated in a first calibration step, keeping the other arm kinematics parameters to be estimated afterward fixed. A second calibration step is then employed to jointly optimize for the checkerboard transform as well as the kinematics parameters. Figure 8.9 shows calibration results for the right arm. The reprojection error is significantly reduced after calibration.

The cause for the remaining calibration error has been determined to be caused by the Linear Variable Differential Transformer (LVDT) encoders used within the Atlas arms. As shown in [49], these encoders exhibit significant hysteresis and backlash effects, explaining the remaining calibration error. For this reason, the calibration approach was adjusted to move the arms into joint limits and record static offsets then.

8.5.2 Previewing Manipulation

As described in Chapter 8.4, the world model server provides the supervisor(s) with the necessary tools to achieve situational awareness of the environment state in a variety of different bandwidth conditions. To be able to reliably perform manipulation, an approach for predictive visualization of how the robot interacts and likely will interact with the environment in the future is required.

With the high number of DOF of humanoid systems and the challenges of balance control, judging the reachability and manipulability of the robot for a given task can be much more difficult than for more conventional robots. While inverse reachability approaches show promising results in the literature [152], [24], they do not consider constraints beyond kinematics and self-collisions. Such additional constraints are for instance sensor visibility constraints or control-related constraints due to appendage control performing better in some configurations than others. It would be possible to incorporate those into inverse reachability analysis, but this remains a largely unsolved topic for research at this time.

To provide an intuitive interface to human operators, the so-called g̈host robotïs used. This is an interactive puppet robot that can be used to predictively simulate the kinematics of manip-

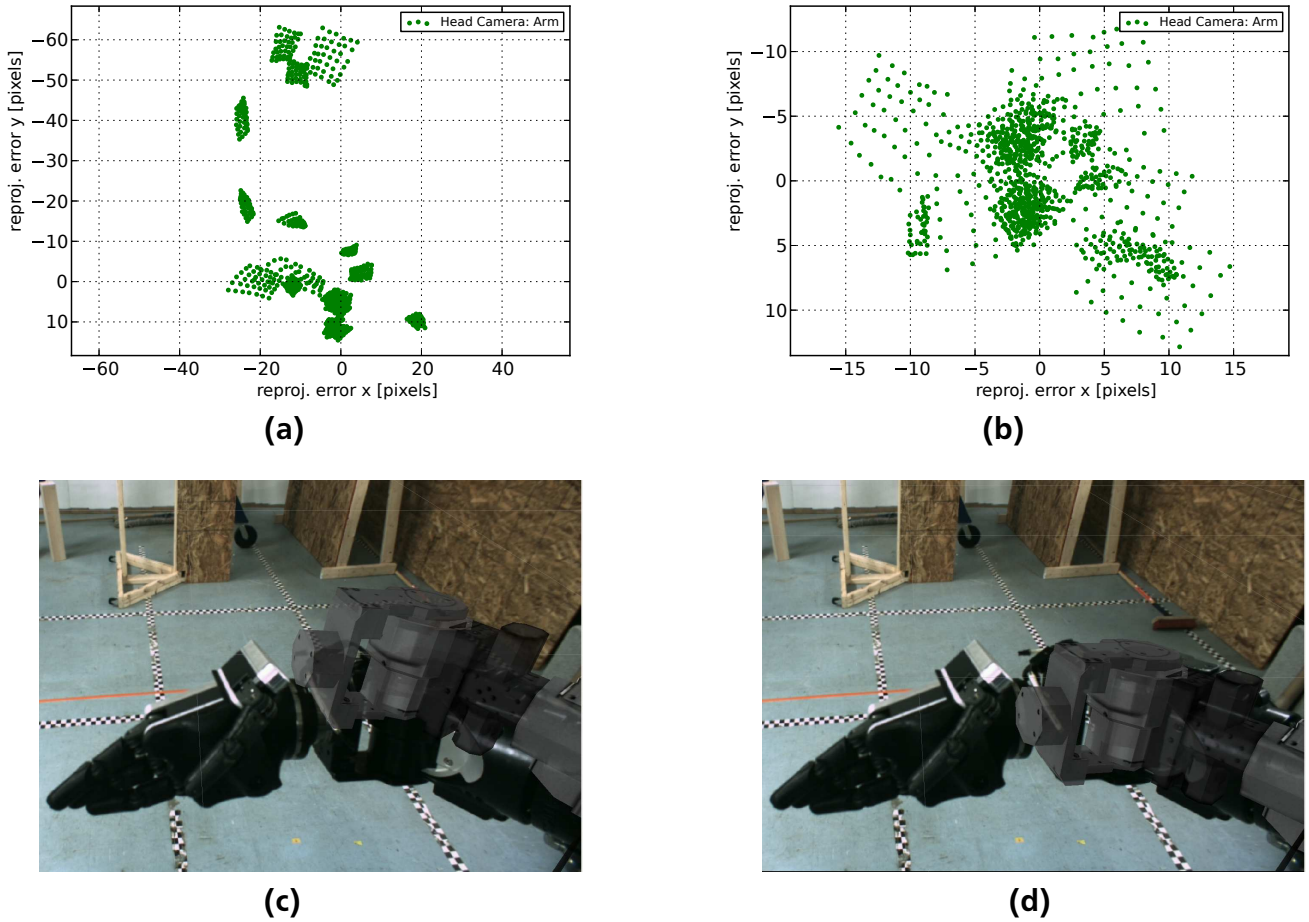


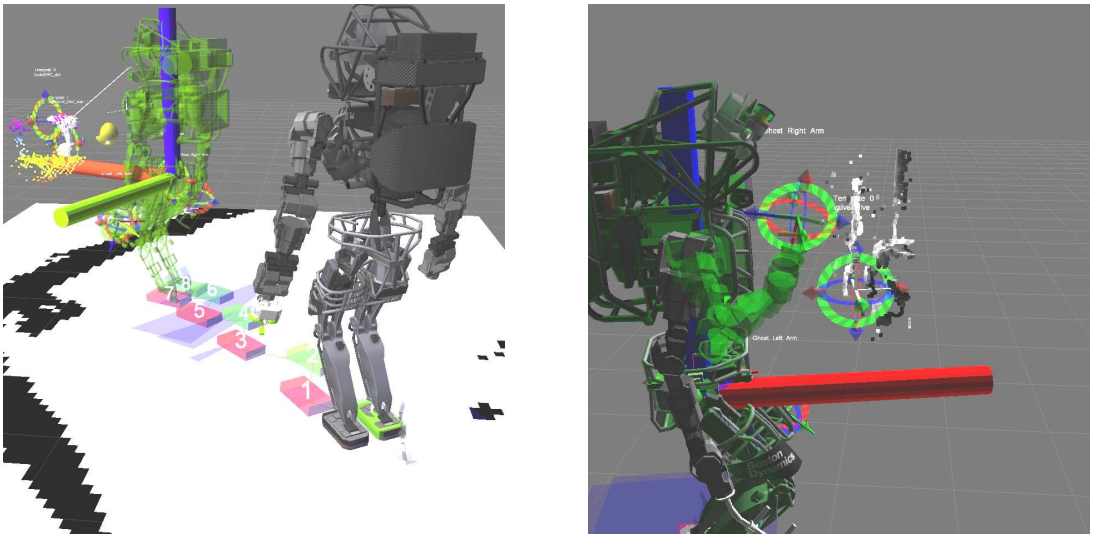
Figure 8.9: Kinematics calibration: Scatter plot of checkerboard corner reprojection error between forward kinematics based calculation of checkerboard corner positions and detections in camera images: (a): Before kinematics calibration (b): After calibration of joint angle offset biases and scale factors. Note the scale change in both plots. (c): Example of reprojection of forward kinematics based arm configuration before calibration (d): after calibration

ulation tasks. The state of the ghost robot can be modified in the user interface without effects on the real robotic system. Once the supervisor is satisfied with ghost robot based pre-planning, planning and motion requests can be generated based on the ghost robot state using a variety of different options detailed below.

The ghost robot is an essential tool for teleoperation and supervised autonomy and is used for the full range of manipulation and locomotion control. While it remains possible to move the robot by sending joint angles directly, this is discouraged due to the high risk involved in such actions.

As shown in Figure 8.11 the ghost robot state can be modified based via a ROS API that allows for the following options:

- Joint angles. The ghost robot can externally be set to be in a desired joint angle configuration. Importantly, a subset of joints can be used here.



(a) (b)

Figure 8.10: Two examples of using the ghost robot for previewing manipulation: (a): The ghost robot is used to preview the stand pose before performing manipulation (b): Previewing arm motion during the valve task at the DRC Trials. The solid robot is the current true state, while the translucent green one is the ghost robot.

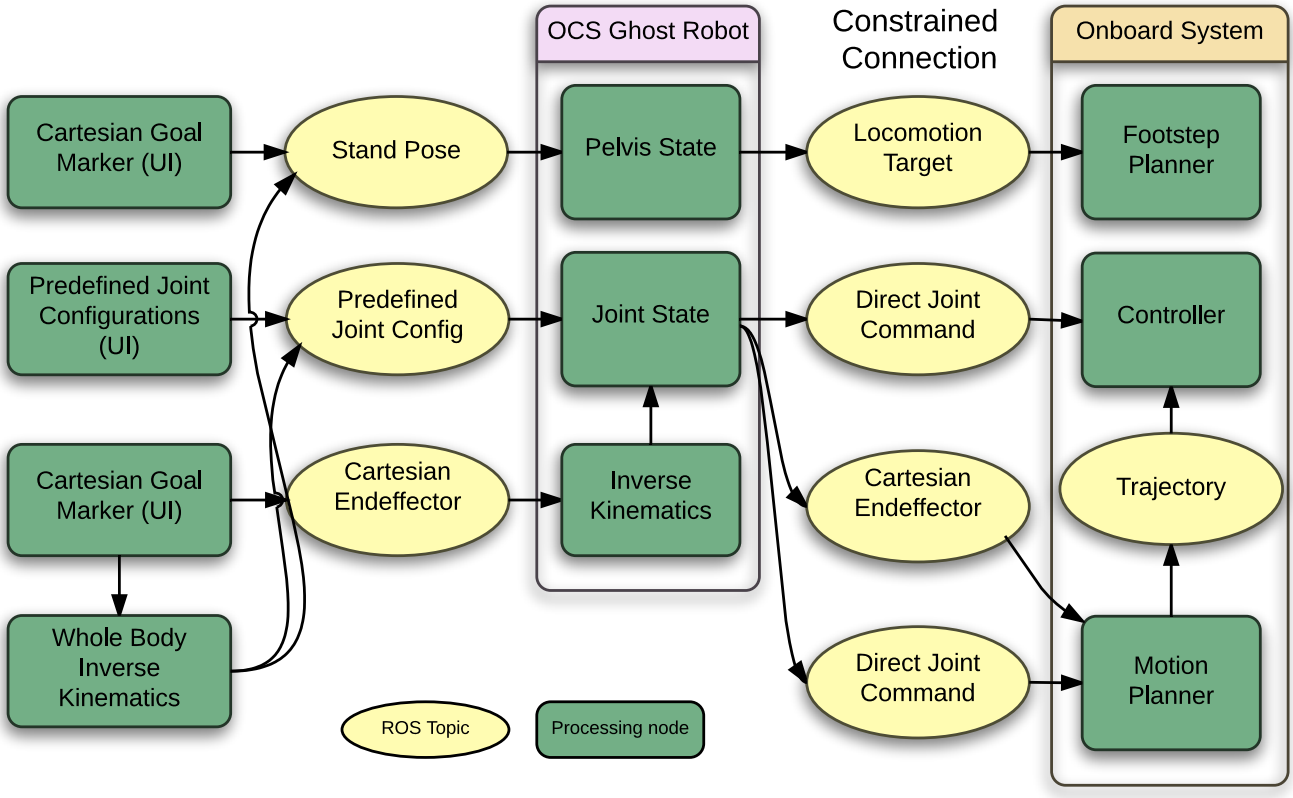


Figure 8.11: Schematic showing inputs and outputs for the ghost robot that is used for pre-planning manipulation tasks.

- Cartesian goals for end effectors. The ghost robot end effectors can be moved to cartesian goals. In this case, an Inverse Kinematics (IK) solver is used internally to solve for the joint positions.
- Cartesian goals for the robot pose. The ghost robot root frame (frequently the pelvis in case of a humanoid) can be moved to a desired cartesian goal pose

If a whole body IK solver is used externally, the ghost can also be set to a desired state by jointly using the joint angle and cartesian robot pose interfaces simultaneously.

Based on the ghost robot state, the following types of commands can be generated to be executed on the real robot:

- A goal pose for the footstep planner based on the ghost robot pelvis position in the global frame
- The joint configuration of one of the ghost's appendage groups can be sent to the onboard controller as a motion target
- The same joint configuration can be sent to the onboard motion planner, which then generates a collision-free trajectory for it.
- The cartesian end effector pose can be sent to the onboard motion planner, which then generates a collision-free trajectory to reach it.

It should be noted that the last two options are not equivalent on most humanoid robots, as balance control generally will shift the pelvis pose when the arm configuration of the robot changes, resulting in an offset for the first option.

Figure 8.10 shows the use of the ghost robot during the DRC Trials. It is used for determining a standing pose for the robot on the left and for planning manipulation of a valve on the right.

8.5.3 Planning System Details

Manipulation in disaster response situations often incorporates prolonged contact situations, for instance when opening a door or turning a valve. Especially in USAR scenarios, cluttered environments present a challenge, as obstacles have to be avoided during motion planning.

The manipulation planning system is based on the MoveIt! [3] [34] motion planning framework available for ROS. This framework provides a powerful API for planning and different planning components.

The system enables planning to goal joint configurations and to goal end effector poses and thus is directly compatible with the ghost robot approach described in the previous section. Two planning modes are available: The default mode is unconstrained planning, with joints free to move between the start and goal joint configurations. The other mode is a constrained motion mode. Here, motion is constrained to follow a Cartesian path between the start and goal end effector pose. In this case, waypoints are generated based on linear interpolation between start and goal position and orientations for waypoints are generated using Spherical Linear Interpolation (slerp) [142] between start and goal end effector quaternions. More complex constrained motions such as a circular motion for turning a valve are generated by concatenating multiple short linearly interpolated Cartesian paths as shown in Figure 8.12.

[3] `http://moveit.ros.org/`

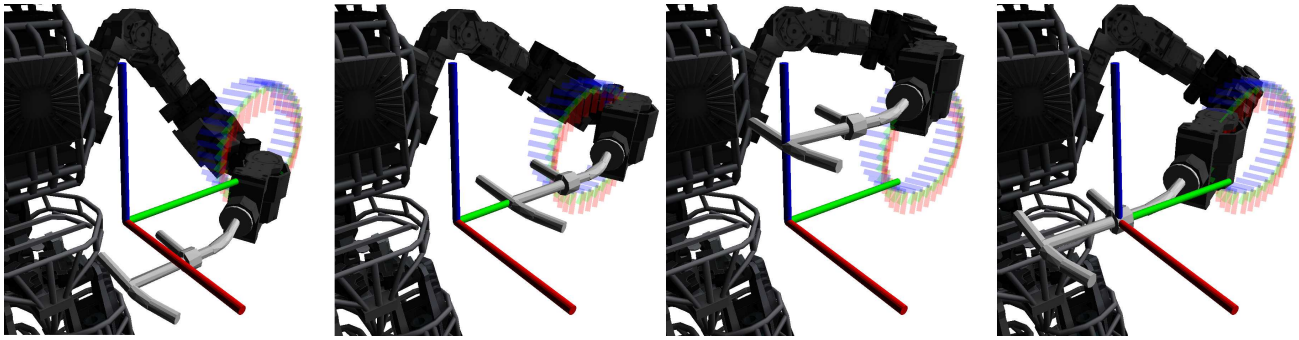


Figure 8.12: Clockwise circular path. The Hook rotates around the X axis(red) while the interpolated poses are shown for the last joint of the arm.

For obstacle avoidance, the volumetric octomap representation as described in Chapter 8.4 is used. As contact with the environment is required in many manipulation tasks, collision checking between end effectors and the environment can optionally be disabled by the supervisor(s). For instance, collision avoidance is needed to safely bring the robot hand into a position to pick up a drill. In order to grasp the drill, collisions between the palm and fingers of the hand and the drill handle must be allowed, however.

In challenging conditions, noise in sensor data that leads to geometric artifacts, preventing successful planning due to spurious collisions cannot be ruled out completely. To cope with such situations, collision checking against the octomap environment model can also be disabled for the complete robot geometry; in this case, the ghost robot changes color to warn the operator.

For motion planning, the number of joints (DOF) to use can selected by the supervisor(s). For instance on Atlas, planning can be performed using either 7 DOF with the arms only, or by including the torso joints and using up to 10 DOF. As the 10 DOF planning mode tends to result in higher control error or oscillation in some joint configurations, the operator can lock a selection of torso joints to restrict the planning space. The same approach can be used on other robotic systems transparently.

To allow for safety and robustness, the ability to select the desired trajectory execution speed with every planning request was introduced. Using standard MoveIt! functionality, trajectories were previously time parametrized according to the velocity limits supplied in the Universal Robotic Description Format (URDF) robot model. This approach turned out to be not flexible enough for challenging manipulation in contact that might require moving appendages slow for safety.

8.5.4 Planning Interface

To implement the described manipulation back-end, the MoveIt! API was used and DRC-specific capabilities were implemented in a separate *move_group* capability plugin. This offers the advantage of retaining standard MoveIt! library planning features, while simultaneously allowing the development of extended capabilities specific for disaster response manipulation tasks.

As shown in Figure 8.13, the planning system is exposed via a ROS Action server interface and thus provides feedback about the planning and plan execution process. The Action interface is the sole entry point for requesting and executing motion plans and (in order of increasing autonomy) used for teleoperation, affordance-based manipulation planning and for motion plan

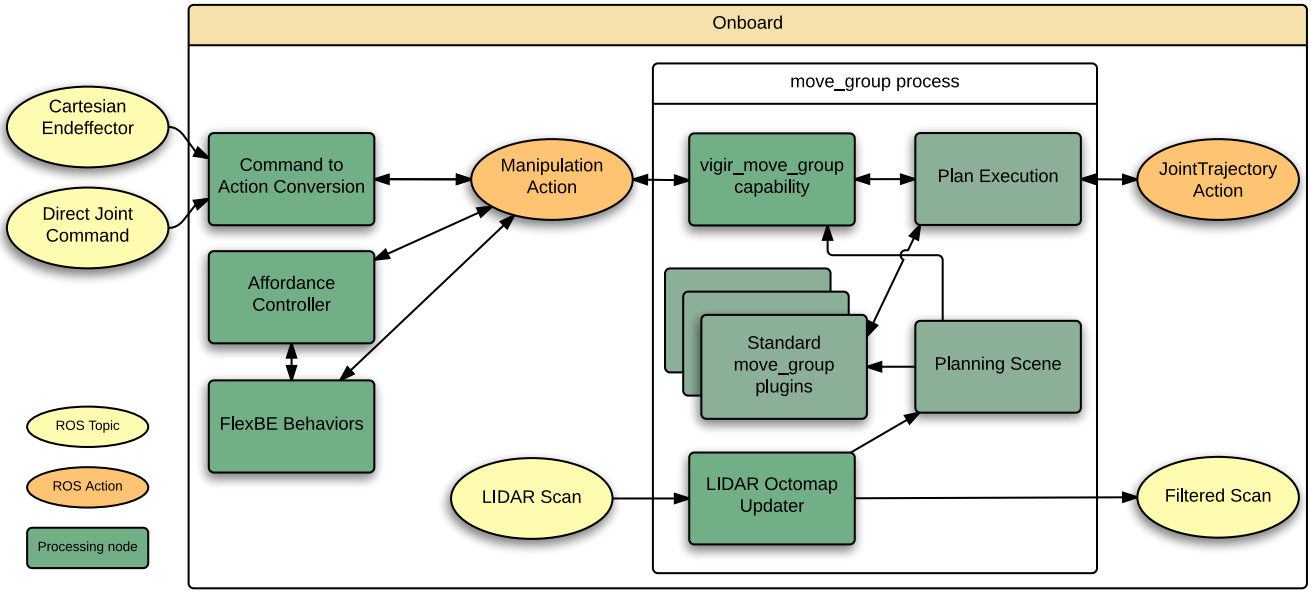


Figure 8.13: Overview of the planning back-end. Both the planning interface and the LIDAR octomap updater are loaded into the standard MoveIt! move_group process as plugins. Using this approach, existing functionality provided by MoveIt! is kept, but extended.

requests generated by the behavior executive. For teleoperation, an onboard node translates compressed and compact motion requests by the operator into an Action request that then gets forwarded to the planning system.

8.5.5 Supervised and Autonomous Control

The described planning system offers a powerful API that can be used to plan for complex manipulation tasks. In the preceding sections, both the teleoperation interface and the planning back-end are described.

To achieve both task-level supervised operation and autonomous control, two additional software components for manipulation use the described planning system as a back-end for performing manipulation: An object template framework and the FlexBE behavior engine. They are contributed by other researchers, but for the purpose of discussing their integration, a brief description of both follows.

Figure 8.14 shows an overview of how the different system components interact to achieve the full range of capability from teleoperation to full autonomy in interaction with one or more human supervisors.

Object Templates

Instead of directly controlling appendages, the object template-based approach for manipulation [127],[126] uses models of objects to be manipulated, so-called object templates. These are placed in the virtual environment model and serve as references for to achieve manipulation at a higher level of abstraction.

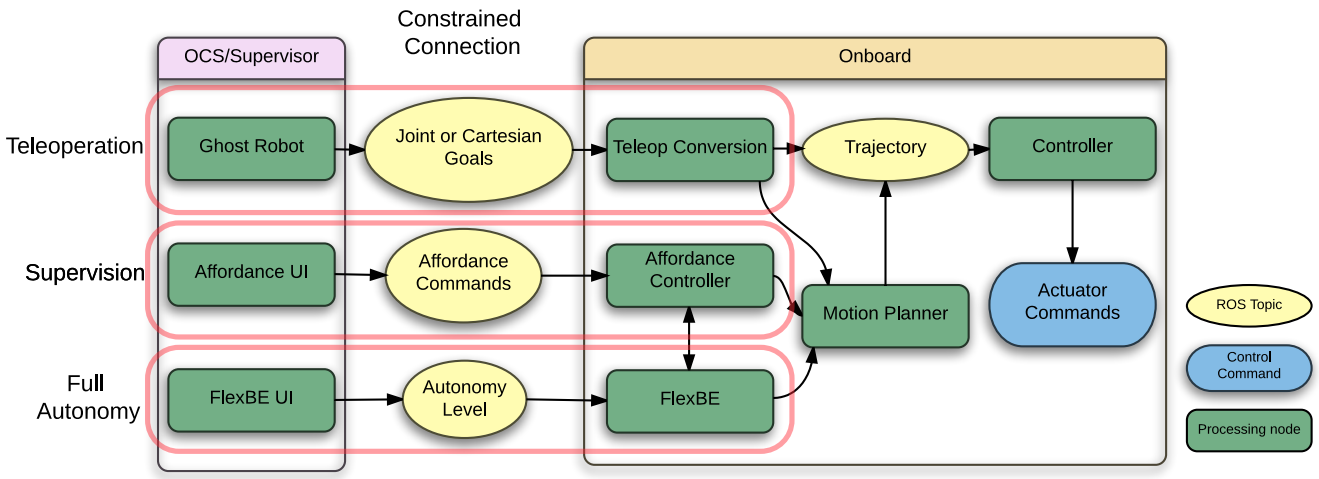


Figure 8.14: Supporting multiple levels of interaction for manipulation capable avatar robots.

With each template offering a set of affordances, motion can be specified by the operator on the affordance level. A door opening motion can for instance be commanded by using an "open" affordance defined for the door handle.

Automatic Behavior Control

For the autonomous execution of complex manipulation and locomotion tasks, the Flexible Behavior Engine (FlexBE) has been developed during the DRC. A hierarchical state machine based behavior control framework specifically designed for supervised autonomy, it allows for specifying online behavior editing and setting an autonomy level, indicating which state transitions are allowed autonomously and which of them require supervisor feedback.

The object template system is also used within FlexBE to represent manipulatable objects. Hierarchical state machines can thus take over responsibility for coordinating complex tasks from remote human supervisors where applicable.

8.5.6 Whole Body Planning

While the developed motion planning system performs well for many manipulation tasks requiring only upper body motion, sampling-based planning falls short for planning whole body motions that require the consideration of balance constraints. To also support this, the optimization-based planning approach available as part of the Drake framework developed at MIT has been integrated with the Team ViGIR planning system. Planning using Drake can transparently be used by specifying the plan request. Drake has also been integrated with the ghost robot on the OCS side and the operator can use Drake-based whole body inverse kinematics to pre-plan tasks like reaching towards the ground for picking up objects.

8.6 Experimental Evaluation

The applicability of the contributions in this chapter to a supervisor-robot team that has to solve complex manipulation and locomotion tasks under communication constraints is demonstrated as part in the next chapter as part of the experimental evaluation within the DRC.

9 Experimental Evaluation

In this chapter, systems-oriented evaluation and benchmarking are performed through a review of participation in renowned international competitions for disaster response (DARPA Robotics Challenge) and USAR (RoboCup Rescue Robot League).

Furthermore, based on a simulated case study with a new centaur-type robot, the applicability and combination of highly autonomous exploration for victim search and highly versatile manipulation capabilities using a single robot with a supervisor in the loop is demonstrated.

9.1 Component-Focussed Experiments

This chapter focusses on the experimental evaluations of contributions as part of comprehensive robotic systems. For an evaluation of isolated functional abilities for simultaneous localization and mapping as well as navigation, exploration, and search for victims, the reader is referred to Chapters 6.5 and 7.9.

9.2 Aspects of System-Oriented Evaluation and Benchmarking

Evaluation of a holistic system approach is a challenge in itself, especially when the holistic systems of interest are highly complex robotic systems, with caveats applicable in terms of availability and reliability.

In addition, the re-creation of comprehensive and realistic USAR scenarios is difficult. Competitions have proven to be a highly valuable tool for benchmarking robotics research [15], making complex and elaborate experimental scenarios available, while at the same time allowing to compare solutions for them in an open and reproducible way.

Recently, there has been increased interest in competitions as experiments. In [3] an overview of the use of competitions for benchmarking is given. A classification of benchmarks into task benchmarks and functional benchmarks is described, with the former testing tasks and thus system integration aspects and the latter testing specific capabilities (i.e. subsystems).

As already noted early [21], competitions and participation in them have to be embedded in larger research as to disseminate results and not make participants lapse into a mode where winning becomes the primary objective, regardless of advancing research. As a measure to avoid this trap, the vast majority of software described in this thesis is available as open source, allowing dissemination, use and improvement upon it by others.

The applicability of the contributions of this thesis to complex problems is demonstrated based on describing the participation in two most challenging international robotics competitions for disaster response and USAR: The DRC and the RoboCup Rescue Robot League competition.

The results achieved during participation in both competitions are discussed with a focus on the contributions of this thesis. It should be noted that the two DRC competition events incorporating real robots only allowed for a low number of attempts at the given tasks (one attempt per task at the Trials, two attempts at the whole scenario at the Finals). Additionally, a higher reliance on human supervisors and their training added additional uncertainty.

Figure 9.1 shows which control modalities and capabilities of the supervisor-robot team were used at the DRC and RoboCup Rescue competition.

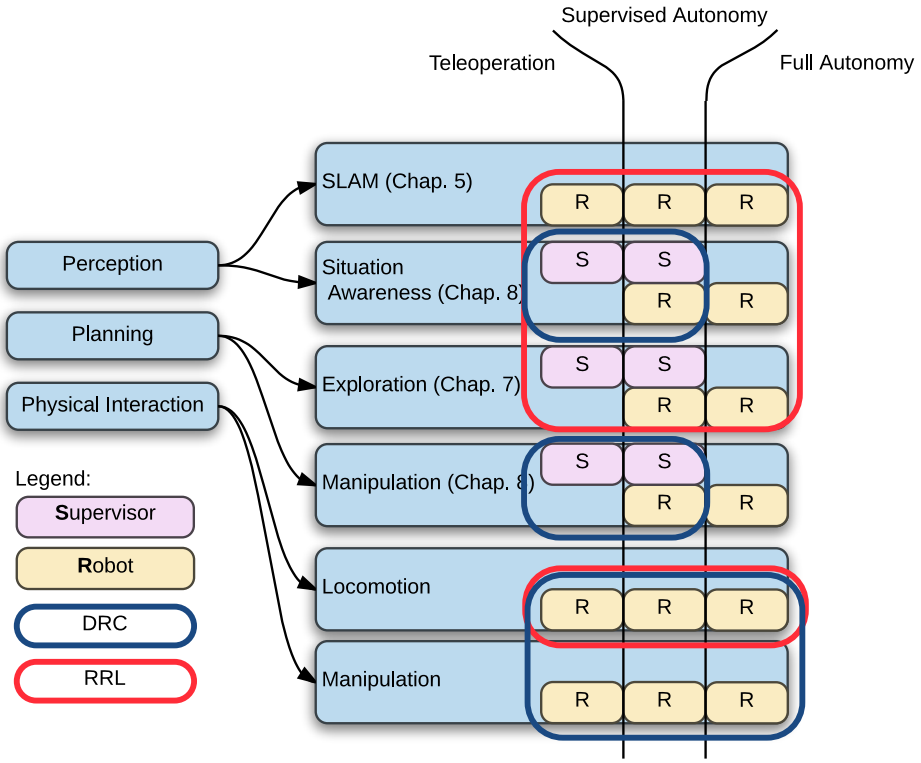


Figure 9.1: Control and interaction modes used within the evaluation scenarios.

9.3 DARPA Robotics Challenge

Over the course of the DRC, Team ViGIR participated in three competition events described below. Results for each of the three competitions are discussed with a focus on the contributions of this thesis. While each event was consistent and fitting into the overarching disaster response theme, featuring complex manipulation tasks, there also were significant differences that required adaption and further research between competitions. As an example of this, Table 9.1 shows the approaches used for the simulation of constrained communication in the different competition events.

9.3.1 VRC

The VRC competition took place in July 2013 and was used by DARPA to select the teams that would receive Atlas robots as GFE. This section describes the approach taken in the three VRC tasks and resulting performance. A comprehensive overview is available in [81].

For each task, 5 instances with varying conditions were used and for each of those, up to 4 points could be scored. The total number of achievable points was thus 60. The allowed time per mission were 30 minutes of simulation time. As simulation might run slower than real-time in some cases,

As visible in Table 9.1, constrained communications were simulated by providing an up- and downlink data budget and adding latency. In the worst case scenario, only a very small amount of data could be transmitted between supervisor and robot, making selective data transmission and management crucial.

	Uplink (from robot)	Downlink (to robot)	Remarks
VRC	Total ~115 kB for 30 minutes. 500 ms latency	Total, 7 MB for 30 minutes. 500 ms latency	Worst case (20% of scenarios)
Trials	1 MB/s, 50ms latency	1 MB/s, 50 ms latency	"good comms"
	100 kB/s, 500 ms latency	100 kB/s, 500 ms latency	"bad comms"
Finals	1.2 kB/s	1.2 kB/s	
	300 Mbit/s		Outages of 1-30 seconds after robot traverses door

Table 9.1: DARPA Robotics Challenge communication constraints overview. The method for modeling communication constraints differed significantly between all three competition events. For the VRC competition, only a fixed bandwidth budget was available. At the Trials, two communication states ("Good Comms" and "Bad Comms") were alternated every minute. At the Finals, two low bandwidth links were permanently available, and a high bandwidth link was available only intermittently after the robot passed through the door.

Tasks Overview

Task 1 (Driving)

After walking up to the vehicle, the driving task presented two challenges. Getting into the car was achieved by using a carefully designed motion that makes use of mechanical effects to reduce uncertainty in positioning the robot into a sitting posture in the car as visible in Figure 9.2a).

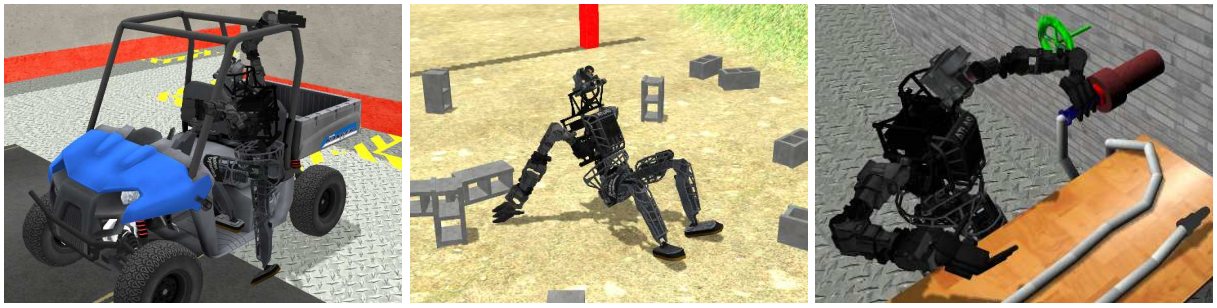
Actuation of the car controls was difficult, as the seat was modeled as a sticky surface. When performing the required manipulation in contact with the environment, the robot would slip in the seat unpredictably. This resulted in the loss of sitting posture in all runs.

Task 2 (Rough Terrain)

Due to a bug in the optionally provided walk controller which only showed itself during the competition, it was not possible to get into walking mode again when the robot has stood up after a fall. Therefore, after a fall during a task in the VRC, crawling was the only option left for locomotion. An open-loop key frame-based quadrupedal locomotion approach had been developed which was used in the VRC during the rough terrain task and whenever the robot had fallen. After walking up to the first gate, the robot encountered a simulated mud pit, which could not be traversed using the provided walking controller. Therefore, the robot was commanded into a sitting posture and backward crawling was employed as the mode of locomotion for the remainder of Task 2 as visible in Figure 9.2b).

Task 3 (Hose Manipulation)

The most challenging part of the hose manipulation task was aligning the hose with the standpipe after picking it up as visible in Figure 9.2c). While picking the hose up worked reliably using



(a) (b) (c)

Figure 9.2: VRC Tasks: (a): Driving (b): Rough terrain (c): Hose manipulation

our approach, moving it towards the location of the standpipe was more challenging, as movement of the hose was unpredictable when dropped on the table for re-grasping with the other arm. For this reason, the strategy was switched to walking with the hose in hand during VRC, which allowed alignment with the standpipe in one run.

Discussion

With the highly constrained bandwidth budget, the selective transmission of robot state and perception data was crucial. Due to the budget model used, it was also crucial to not accidentally send high rate data as that would automatically deduce from the budget, in a worst case scenario using up the bandwidth budget within seconds.

For the supervisor to achieve SA, the use of remote queries to onboard world model data was crucial. By using grid map slices generated from 3D octomap data, bandwidth requirements were low, while the provided maps still conveyed the necessary information for the supervisor to perform the given tasks.

Navigation was generally performed on grid maps. For manipulation, selective point cloud data and ROI image data was transmitted. The bandwidth limits were exceeded in none of the 15 task attempts, demonstrating the effectiveness of the contributed remote world model system.

From out of 126 Track B and C teams registered for the VRC, 26 passed qualification and 22 scored in the VRC. With a total of 27 points (Table 9.2) Team ViGIR was ranked 6th behind 5th and 4th with 29 and 30 points and before 7th and 8th with 25 and 24 points (Table 9.3). The distribution of points scored in the three tasks was quite similar for teams ranked 3rd to 8th. A major challenge in the VRC was the very short time span of only 8 months from the start of the project to the competition event. The approaches developed for providing SA to the operator under bandwidth constraints proved invaluable for achieving good results.

9.3.2 DRC Trials

In the DRC Trials, the described contributions were used with the real Atlas robot for the first time. A comprehensive description of the performance in the DRC Trials is available as part of [85]. Here, an evaluation with a focus on the contributions of this thesis is provided.

	Run 1	Run 2	Run 3	Run 4	Run 5	Total (Max. 20)
Task 1	0	1	1	1	0	3
Task 2	4	2	4	4	4	18
Task 3	1	1	2	1	1	6

Table 9.2: Detailed Team ViGIR VRC scores.

Team	Task 1	Task 2	Task 3	Total
IHMC	12	20	20	52
WPI	15	20	4	39
MIT	5	20	9	34
TRACLabs	4	20	6	30
JPL	5	20	4	29
ViGIR	3	18	6	27

Table 9.3: Top teams VRC results. Note that for brevity, this table does not show the complete field of competitors.

Figure 9.4 shows examples of the 7 tasks performed by Team ViGIR. The team opted to not perform the driving task at the Trials as it was determined to require significant development effort that would not be re-usable for much of the other tasks.

The time allowed for each task was 30 minutes. Figure 9.5 shows a timeline overview of all the tasks attempted. While a single operator was used for the VRC, two operators interacted directly with the robot, with the auxiliary operator being responsible for managing perception data and the primary operator responsible for commanding robot motion via teleoperation or task-level commands.

Kinematics calibration proved very challenging for the Atlas robot as the characteristics of the LVDT encoder-based joint sensing as described in Chapter 8.5.1 only were correctly identified after the competition. The primary operator thus had to compensate for kinematics calibration offsets manually and manipulation was mainly performed in teleoperation mode, at times visual servoing end effectors. Using this teleoperation-focused mode, the supervisor team relied heavily on the contributions of this thesis for perception and communication under bandwidth constraints.

Figure 9.3 shows the network setup at the Trials. A traffic shaper was used to inject communication constraints between operators and field computers, which ran the robot onboard software. The communication conditions injected are visible in Table 9.1. The “good comms” and “bad comms” conditions were alternated every minute, resulting in significant and varying latency between robot and supervisors.

Individual Task Results

A timeline of the performance of the 7 attempted tasks in the Trials is available in Figure 9.5 and can serve as a reference when reading the following discussion.

Door

The door task required the robot to open and walk through multiple doors. The first door was successfully opened using teleoperation within 5 minutes. As the door was very narrow, the automated footstep planning system failed to compute a valid footstep plan, so the main operator attempted manual footstep planning. The robot touched the door frame during this manual approach, leading to a fall. A second attempt lead to another fall. After this, time ran out and no points were scored.

Debris

The debris task required 10 pieces of debris to be moved out of a rectangular area. This proved to be very challenging due to the aforementioned limitations of kinematic calibration. After the first attempt at grasping a piece of debris was unsuccessful, pulling out the whole truss out of the debris field was attempted as seen in Figure 9.4b. While 5 pieces of debris were removed, they were not moved completely out of the required rectangle. For this reason, not points were scored.

Hose

The hose task required picking up a hose, carrying it and attaching it to a wye. The hose was picked up successfully using a combination of object template manipulation and teleoperation, scoring one point. The robot was then commanded to walk to the wye. The grasped hose was brought into contact with the wye, scoring another point. Team ViGIR was the fastest Atlas team to score this second point. The attempt at attaching the hose to the way was not successful, so two points were scored. A detailed analysis is available in [126]. A video summarizing task performance is available online[1].

Valve

The valve task required opening three different valves located on a wall. The valve task was successfully accomplished using a combination of template-based positioning of the robot and cartesian motion capabilities. Four points were scored on this task, as the completion point was also awarded.

Wall

The wall task required picking up a drill and cutting a triangular hole into a wall made of drywall. The drill was successfully picked up, but the grasp slipped during the cutting operation. The combination of hand and drill chosen by the team was determined to be the main contributing factor for the grasping issues experienced in later analysis. No points were scored, as only cutting at least one complete side of the triangle would have resulted in a point getting awarded.

Terrain

The terrain task required traversal of irregular terrain, featuring a chevron shaped hurdle and cinderblocks the robot had to ascend and descend. Using automated footstep planning, the robot proceeded successfully up to the middle of the second section of the task, scoring a point for traversing the first section. Upon stepping down from a cinderblock, one knee actuator

[1] https://www.youtube.com/watch?v=I8PB6GpvLeo

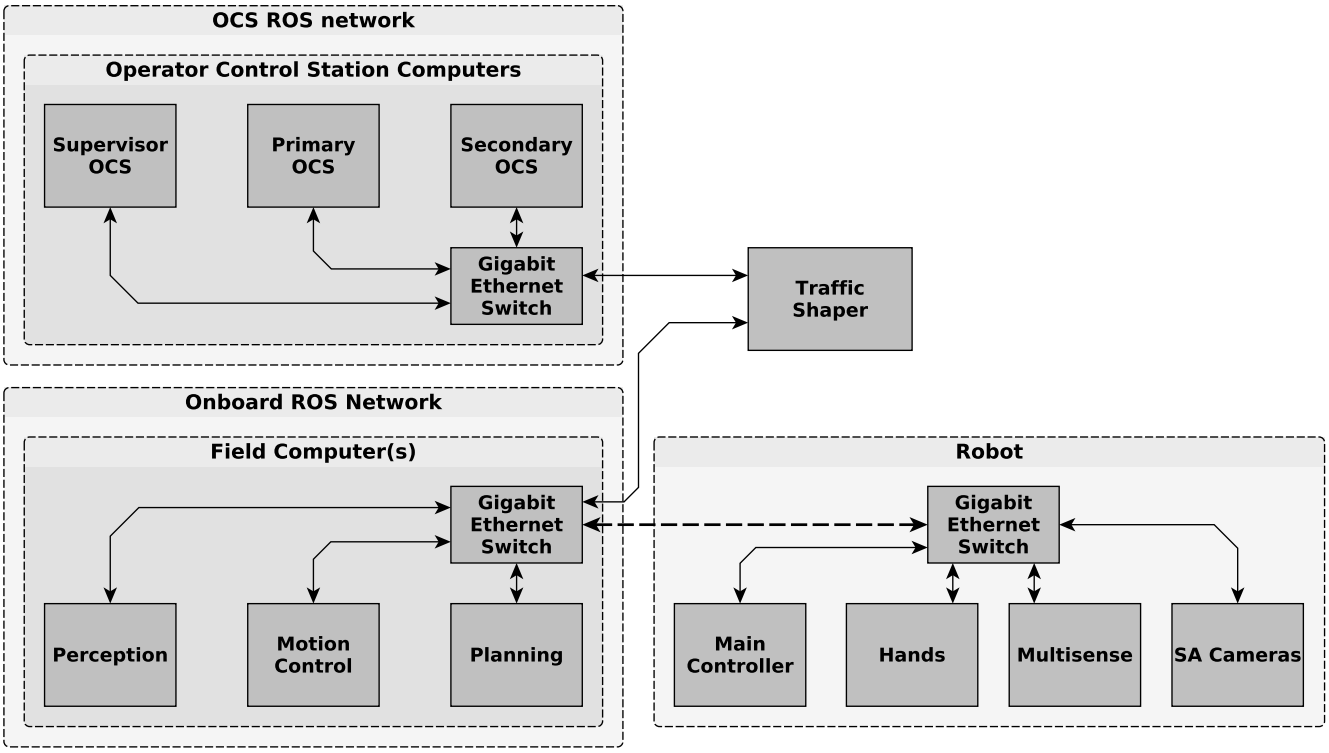


Figure 9.3: Network setup at the DRC Trials.

gave in, resulting in a fall. Another (rushed) attempt at crossing the second section resulted in another fall. One point was scored in this task.

Ladder

The ladder task required the robot to ascend a ladder. A first point was awarded for the robot having all contact points above the first step. To achieve this, the robot walked in front of the ladder. Using the hook hands it then supported itself from the fourth step, and the knees were bent backward. Using this method, a point was scored. After this attempt, another attempt at performing a step up the ladder was attempted but resulted in a fall.

Discussion

The aforementioned requirement for teleoperation due to limited kinematics calibration accuracy impeded the speed and reliability with which manipulation could be performed. It, however, has to be noted that using teleoperation, the supervisors were in full control of the robot at all times and the remote world model and manipulation system as contributed in this thesis worked highly reliably. This allowed performing all planned tasks, albeit slowly and with different degrees of success.

9.3.3 DRC Finals

The DRC Finals took place at Pomona, California on June 5th and 6th 2015. In the DRC Finals, three teams used Team ViGIR's software and thus contributions of this thesis, demonstrating the claimed flexibility and modularity.

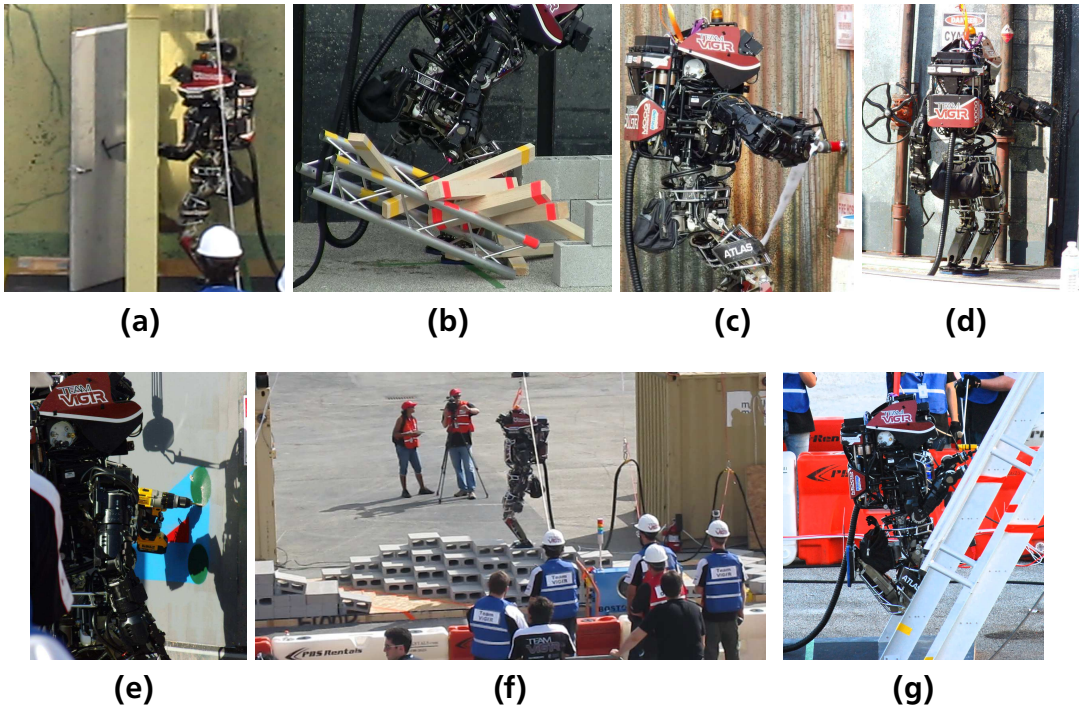


Figure 9.4: Participation in the DRC Trials: (a): Opening the first door (b): Pulling the truss out in the debris task (c): Attempting to connect the hose to the wye (d) Rotating the first valve (d): Using the drill (e): Stepping over cinderblocks (g): Robot on ladder with all support points above the floor.

Unlike in the Trials, tasks were not attempted separately. Instead, teams had 60 minutes time to score as many of the 8 tasks as they could. Each team was allowed two runs in the competition, one on the first and one on the second competition day. The first objective was reaching the door, behind which the other tasks were situated. This could be done either by starting the robot in a Polaris Ranger vehicle and letting the robot drive up to the goal line close to the door, or by starting outside the vehicle and letting the robot walk the whole distance. Scoring awarded 0 points for walking, 1 point for driving and 1 point for egress from the vehicle. Teams could opt out from performing egress. In this case, a reset had to be called and the robot manually extracted from the vehicle, resulting in a 10-minute reset penalty and no point for egress. Traversing the door was the next task, with one point for full traversal through the door frame.

After traversing the door, communication constraints went into effect as shown in Table 9.1, meaning that the high bandwidth connection for perception data had pseudo-random dropouts of up to 30 seconds length, with 1-second windows of communication in-between. 15 minutes before the run end, the drop outs stopped, allowing for full communication again.

Team Hector

Team Hector used the THOR-MANG robot as described in Chapter 3.3. While the system showed promising capabilities during the qualification for the DRC Finals and prior to them during testing, the slope of the ground at the Trials and hardware problems resulted in the robot falling in both Final runs.

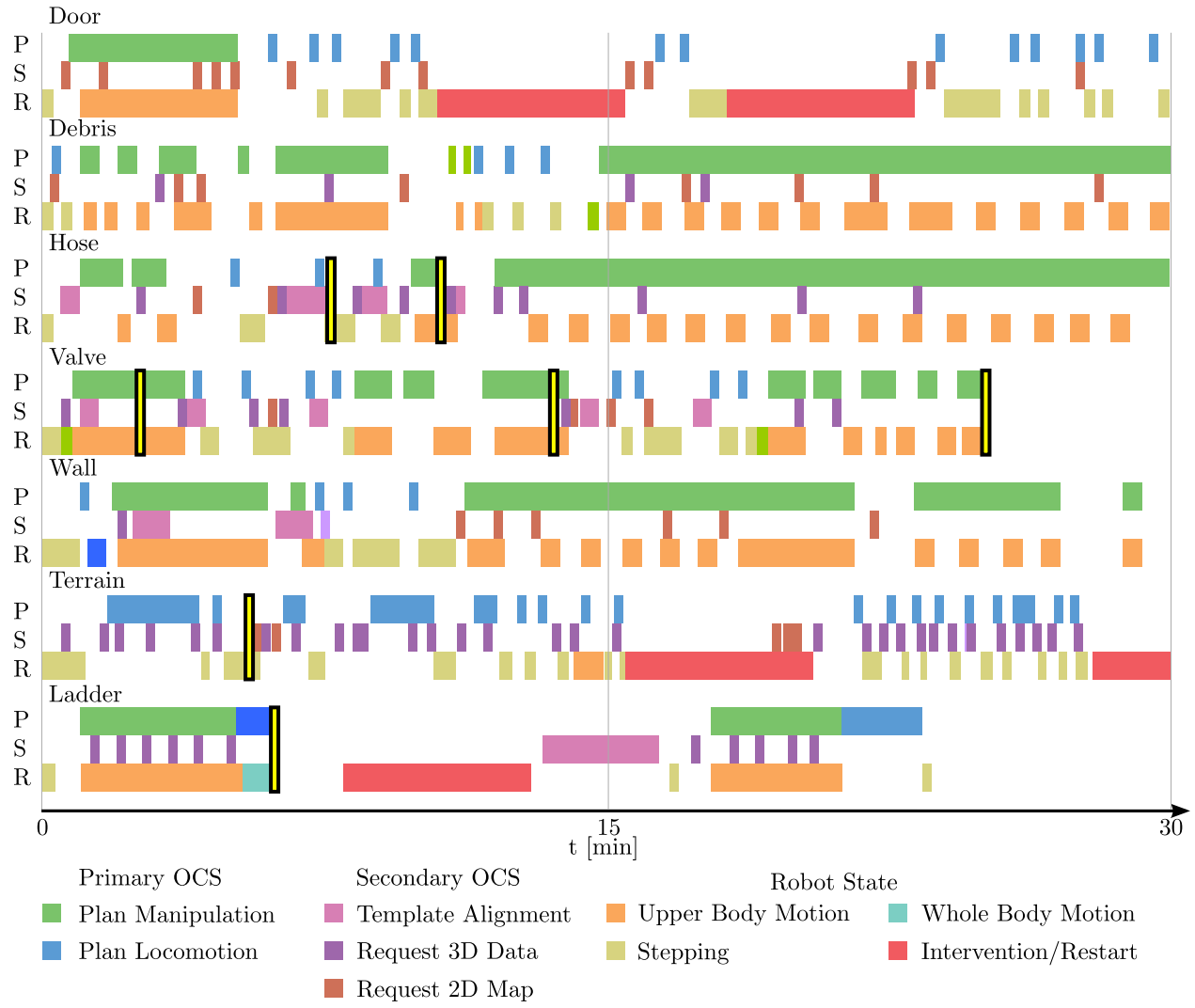
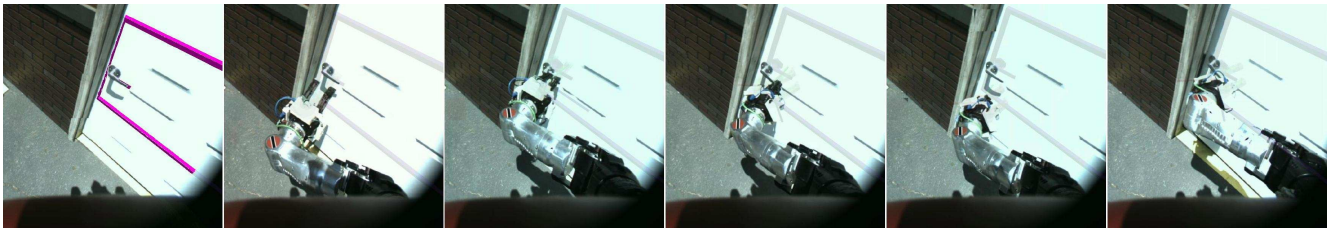
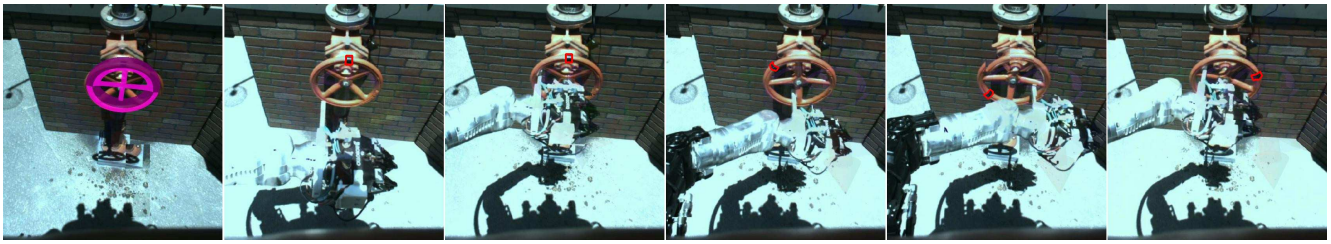


Figure 9.5: DRC Trials overview schematic showing the time and task distribution among Primary OCS (P) , Secondary OCS (S) and robot (R). The different considered modes shown are shown in the legend. Scoring events are marked using yellow bars with black border.

Figure 9.6: The THOR-MANG robot of Team Hector opening the door at DRC Finals Day 1. Images courtesy of Alberto Romay.



(a)



(b)

Figure 9.7: Team ViGIR Atlas robot view of performing tasks at the DRC Finals: (a): Door task. From left to right: Door Template aligned, Pre-grasp, Grasp, Turn Clockwise affordance, Push affordance (fails to open), door opened after Cartesian teleoperation. (b): Valve task. From left to right: Valve Template aligned, Pre-grasp, Grasp, Open affordance 45, 135, and 270 degrees. Images courtesy of Alberto Romay.

The driving task was performed reliably, but on both days the robot fell when attempting to perform the door task. The door was successfully opened on the first day as seen in Figure 9.6.

Team Valor

Team VALOR used the Electric Series Compliant Humanoid for Emergency Response (ESCHER) in the DRC Finals. The team decided to not attempt the driving task. ESCHER was the only robot that successfully walked the complete distance from the start point up to the door. The attempt at opening the door was not successful due to encountered hardware issues.

Team ViGIR

Team ViGIR used the untethered version of Atlas as described in Chapter 3.2. Originally, the team intended to skip the driving task. When it became clear that it would be allowed to not perform egress, but instead call for a reset, a decision was made to attempt the driving task. The performance for both competition days is briefly described the next two paragraphs.

Finals Day One

Starting in the Polaris Ranger vehicle, teleoperation was used to drive the robot down the vehicle course. An intervention with an associated 10-minute pause as specified in the DRC rules was then used to manually extract the robot from the car.

After the 10 minute penalty time, the door task was attempted. During the attempt to perform the door task, the supervisor team noticed that high-level behavior execution did not work as intended. This was later traced back to a faulty setup of the communications bridge system

and increased saturation of the wireless links used in the competition. The supervisor team thus switched from using assisted autonomy via FlexBE behaviors and use of object templates to using object templates and teleoperation. Using this approach, the door was successfully opened as visible in Figure 9.7a). The valve task was solved using mainly object affordance level control 9.7b). Before being able to actuate the switch in the surprise task, time ran out, ending the run. A video is available online [2].

Finals Day Two

The second-day mission again started by the supervisor team using teleoperation for driving the Polaris vehicle. Due to erratic network connectivity and possible operator error, a barrier was touched and a reset had to be called. In the second attempt, the driving task was performed successfully. The door opening task was performed using object template and automated behavior control. After the door was opened, the pump of the robot shut down for unknown reasons and the robot fell. After this forced reset another attempt at traversing the door was made, resulting in another fall. A video is available online [3].

Discussion

As the second competition day was characterized by hardware failures, a closer look at Day One performance is provided. While the communications bridge issues encountered prevented the use successful use of using autonomous behavior capabilities, they forced the supervisor team to switch to a more teleoperation-centered approach, thus demonstrating the flexible level of interaction laid out in the requirements in Chapter 4.

The perception system worked flawlessly, providing full LIDAR based environment geometry despite communication constraints after traversing the door only allowing intermittent communication over the 300MBit high data rate connection from the robot.

All three teams using Team ViGIR's software were able to leverage the contributions of this thesis, performing manipulation for opening the door at the Finals. Due to hardware issues, the full potential, however, could not be demonstrated at the DRC Finals.

9.4 RoboCup Rescue

The RoboCup Rescue Robot League competition serves as a benchmark that allows testing complex USAR robotic systems in comprehensive scenarios and direct performance comparison to approaches employed by other researchers. Focussing on robot autonomy, Team Hector performed very well using the contributions described in this thesis as visible in Table 9.4, consistently winning the Best in Class Autonomy (BICA) award in every RoboCup competition the team participated in since 2012.

The competition scenarios get successively refined and become more challenging to keep up with evolving capabilities of robotic systems and evolving requirements of responders. Scores in the competition can thus be considered as being a useful metric relative to competitors taking part in the same event, but fluctuations in difficulty and scoring metrics applied between com-

[2] https://youtu.be/VEsUICAa4rg
[3] https://youtu.be/Whw-tG0Wh9U

petitions means that the absolute value of scores cannot be meaningfully compared between competition events.

The RoboCup German Open traditionally has a higher focus on robot autonomy. The scores of the three top teams for the competitions from 2010 to 2015 are visualized in Figure 9.8. As can be seen, Team Hector always was among the top three teams. Videos showing missions at the RoboCup German Open 2014 competition from the perspective of the operator station are available online [4].

Figure 9.9 shows the evolution of performance as measured by counting the number of victims and QR codes found during the RoboCup 2014 competition. Over the course of the competition, performance steadily increased. The final mission is an exception to this trend as operator error lead to the vehicle rolling over and thus an early abort of the mission.

RoboCup 2014 Final Missions

Figure 9.10 provides a detailed timeline for the two highly successful final missions at the RoboCup 2014 competition. Over the course of both missions, the full range of capabilities of the robotic system was leveraged.

For both missions, the robot started in the yellow part of the RRL arena. In the yellow arena, victims can only be scored by fully autonomous robots. When the robot onboard system decides a victim is found, an alert in the user interface pops up and the supervisor can decide to let the robot either *confirm*, *ignore* or get closer to the victim.

On alert, the judge will score the victim depending on how many signs of life are visible via robot sensors. There thus is an *inspection* phase when a victim is found. If a false positive victim is detected, the robot might be forced to stop for a certain amount of time at the discretion of the judge.

Final Mission 1

The robot immediately started exploring the arena. Positioning after finding the first victim was not close enough for a good observation position, so the "get closer" option was selected twice by the supervisor.

A false positive victim was detected and ignored shortly afterward. The supervisor then decided to perform a soft reset, moving the robot back to the start position via teleoperation. After finding the second victim, two more soft resets were performed.

Moving towards the orange arena, the robot autonomously detected two more victims, meaning all of the yellow arena victims have been found. At this point in time, the supervisor switched to teleoperation mode to quickly move the robot into the orange arena. Under teleoperation, the fifth victim is found and manually added to the world model.

Arriving at the radio drop out zone, the robot was commanded to the victim located there using supervised autonomy mode, with the supervisor providing the goal pose for the UGV. The victim was then manually added and the robot returned to the start of the radio drop out zone on supervisor command, which doubles the points scored for the victim in the radio drop out zone.

The supervisor proceeded to explore the orange arena, scoring the seventh victim before the mission time elapsed.

[4] https://www.youtube.com/playlist?list=PLqdOEBv9QGrEiqlUklq0BI1QPU55IhTO2

Year	Event	Result
2011	RoboCup German Open	Overall Winner Best in Class Autonomy
2012	RoboCup German Open	Overall Winner Best in Class Autonomy
	RoboCup	Overall 2nd Place Best in Class Autonomy
2013	RoboCup German Open	Overall Winner Best in Class Autonomy
	RoboCup	Overall 7th Place Best in Class Autonomy
2014	RoboCup German Open	Overall Winner Best in Class Autonomy
	RoboCup	Overall Winner Best in Class Autonomy
2015	RoboCup German Open	Overall 3rd Place Best in Class Autonomy
	RoboCup	Overall 5th Place Best in Class Autonomy

Table 9.4: Team Hector RoboCup Rescue participation results.

Final Mission 2

In the second final mission, the first victim was scored quickly, but erratic teleoperation by the supervisor lead to a map inconsistency, prompting the operator to call for a full reset.

After the full reset, the robot was again started in fully autonomous mode, scoring the four victims in the yellow arena. A single false positive victim was detected in between.

Having scored all possible victims in the yellow arena, the supervisor again switched to teleoperation mode and scored the fifth victim manually.

While attempting to traverse a cross ramp, supervisor error resulted in the vehicle rolling over, damaging the RGB-D camera mount. The mission thus ended prematurely.

9.5 Simulated Case Study Combining Exploration and Manipulation

The contributions towards supervised autonomous exploration with UGVs and towards supervised complex manipulation using humanoid avatar robots were demonstrated and discussed based on highly complex real-world applications in the preceding section.

It remains to be shown that both contributions can be used to provide a single robotic system with the capability to both explore environments using high autonomy and perform manipulation under human supervision. For this, the Hector Centaur robot previously described in 3.4 is used in a simulation setting.

The robot is used to explore the environment given in Figure 9.13. This is the same environment as used for the experiment in Chapter 7.9, with a added door at the end of the initial maze. Without manipulation capability, a USAR robot is blocked by the door and cannot bypass

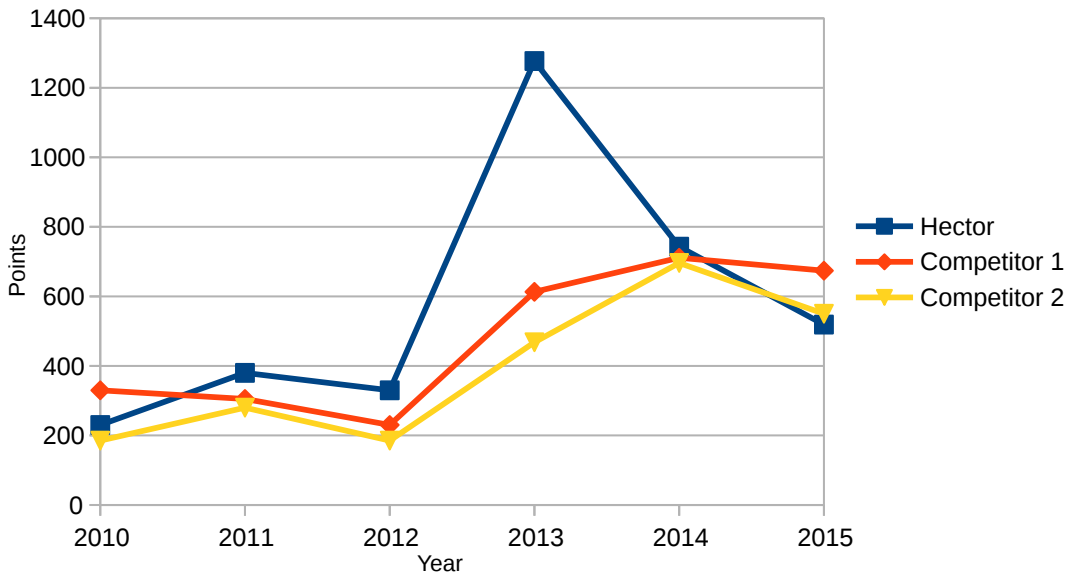


Figure 9.8: History of participation at the RoboCup German Open competition. The top three teams are plotted, with Team Hector always plotted in Blue. The other competitors varied by year and are anonymized for simplicity.

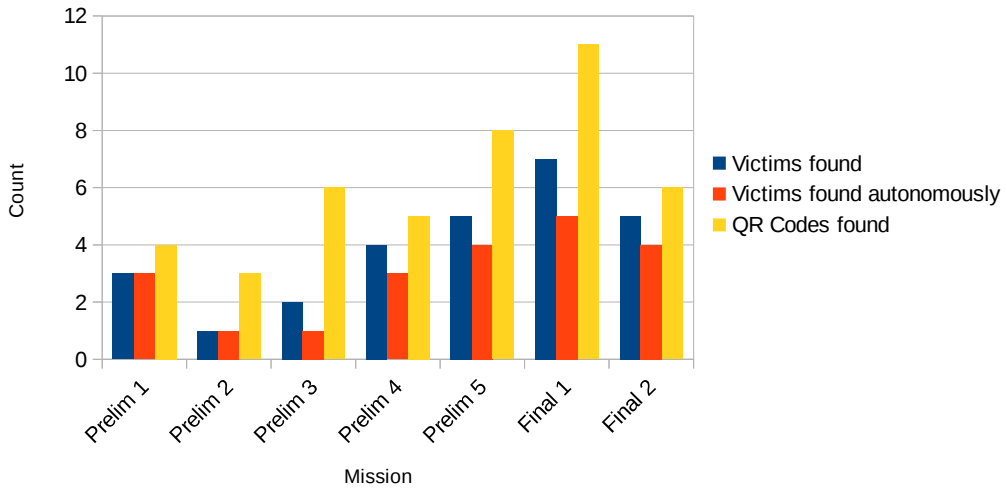


Figure 9.9: Progression of performance of Team Hector at RoboCup 2014 over the course of the competition.

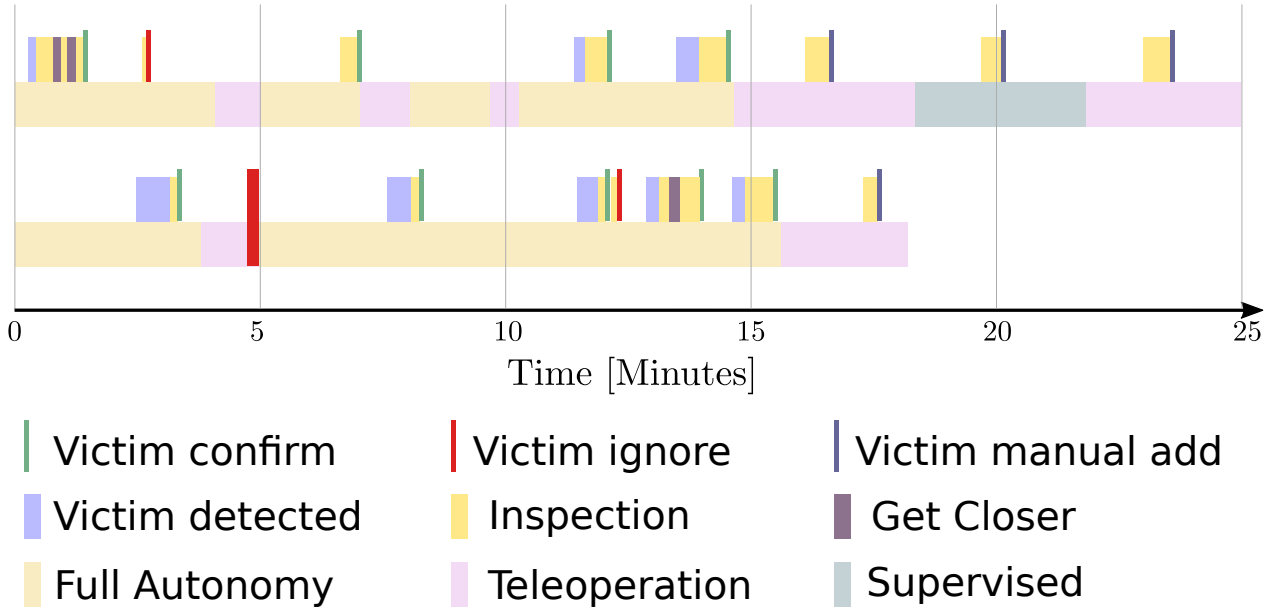


Figure 9.10: Detailed timeline for both RoboCup 2014 Final Missions. A detailed discussion is provided in Section 9.4. The upper bar corresponds to the first final mission. The corresponding map is shown in Figure 9.11f. The lower bar corresponds to the second final mission. The corresponding map is shown in Figure 9.11g.

it, ending the mission early. It can be expected that this is a common situation to be encountered by a USAR robotic system.

With the Hector Centaur robot using both the contributions for improved exploration and manipulation, it first explores part of the environment fully autonomously, with the supervisor observing progress. When the supervisor notices that the closed door prevents the robot from continuing exploration, she uses the manipulation capabilities of the robot to open the door using teleoperation or affordance-level control. Afterward, the supervisor can command the robot to keep exploring the environment autonomously or continue operating in a lower autonomy mode.

Figure 9.14 shows three screenshots of the robot opening the door with assistance from the supervisor.

It should be noted that while not implemented for this example, the robot could notify the supervisor once it requires help by performing semantic mapping and detecting the closed door.

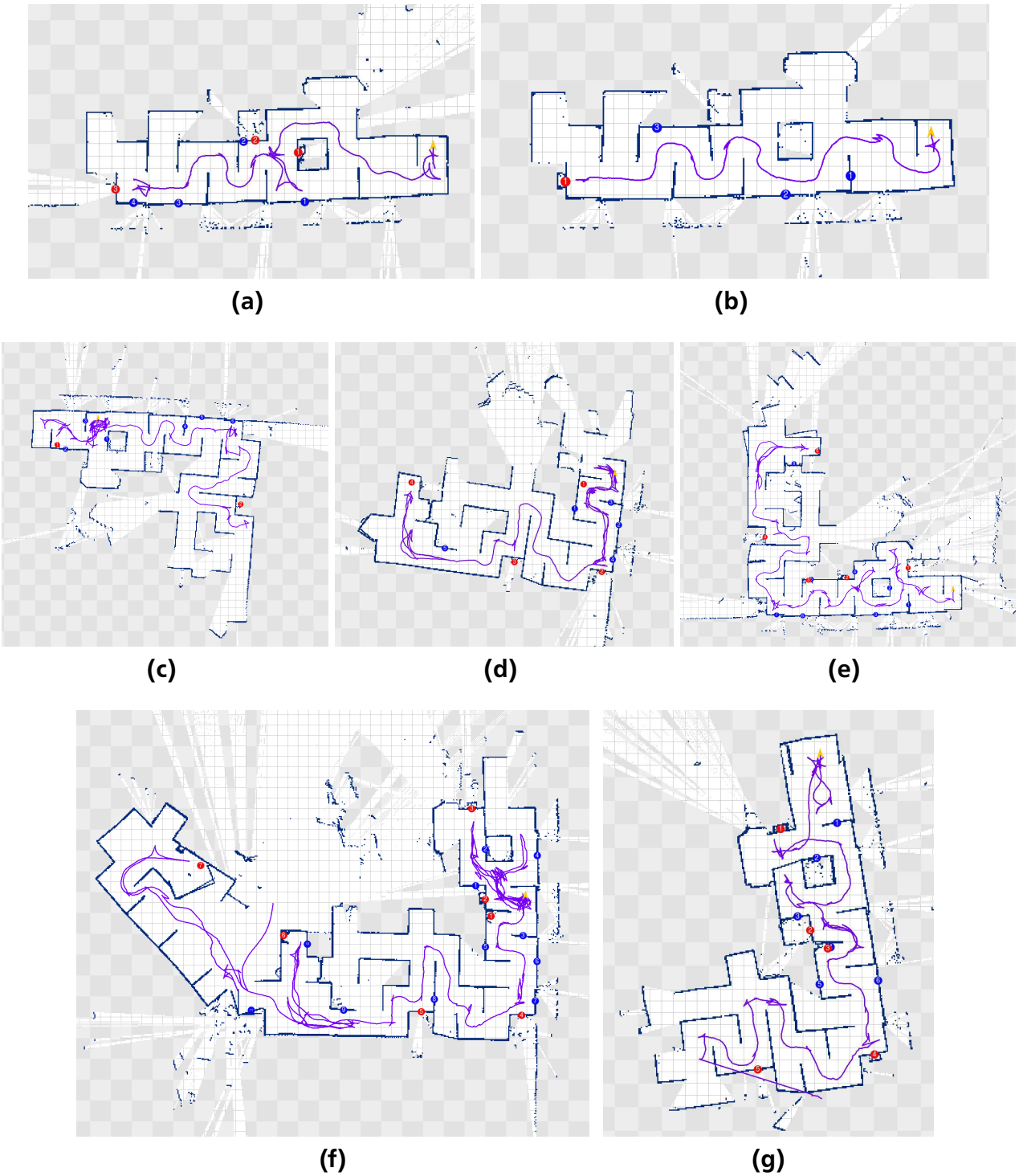


Figure 9.11: Results at the RoboCup 2014 competition. (a)-(e): Preliminary mission maps (f) and (g): Final mission maps. Red markers denote found victims, blue markers denote detected and mapped QR codes.

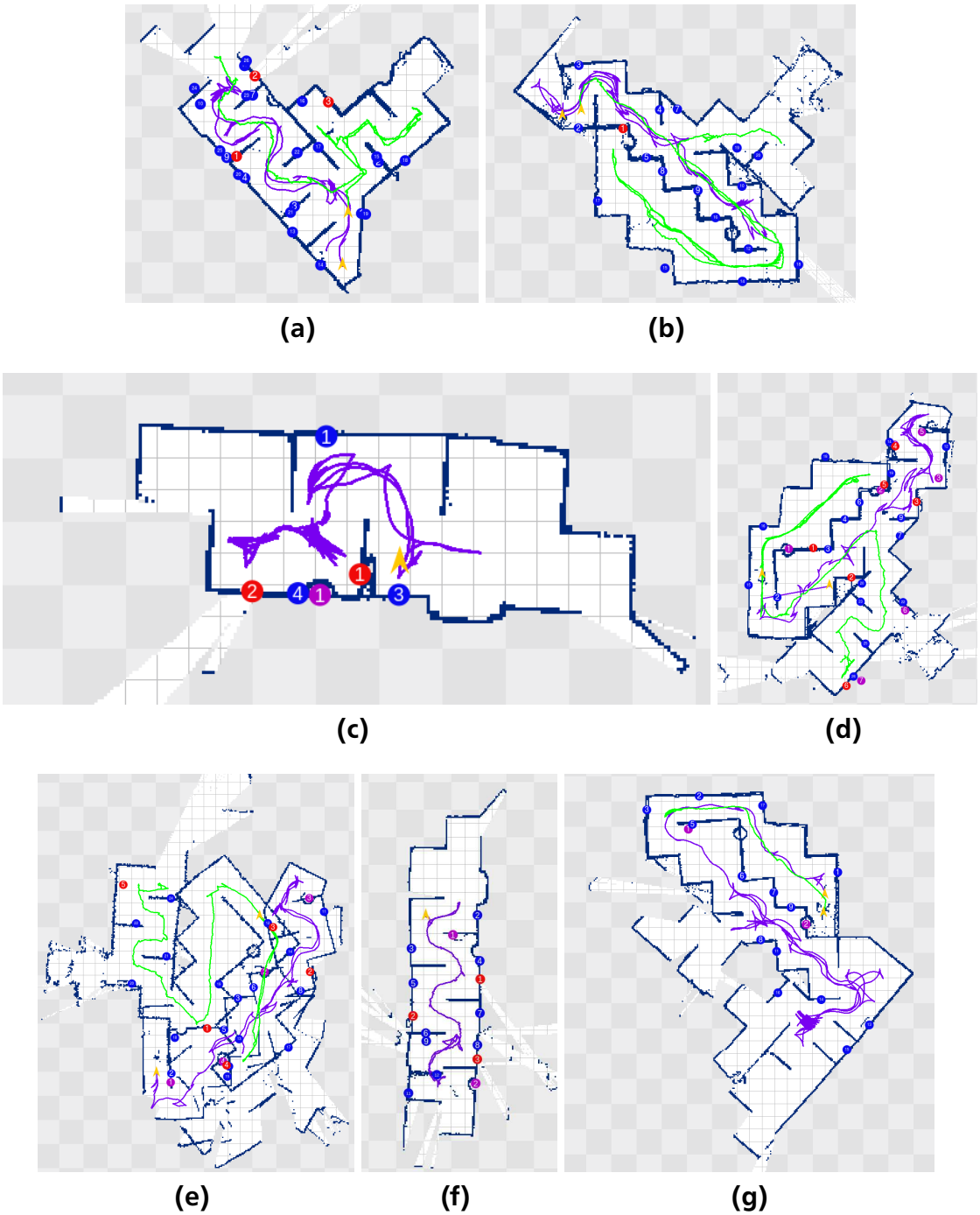


Figure 9.12: Results at the RoboCup 2015 competition. (a)-(d): Preliminary mission maps (e) and (f): Semi-final mission maps. (g): Best in Class Autonomy mission map. Red markers denote found victims, blue markers denote detected and mapped QR codes. Violet markers denote detected barrels.

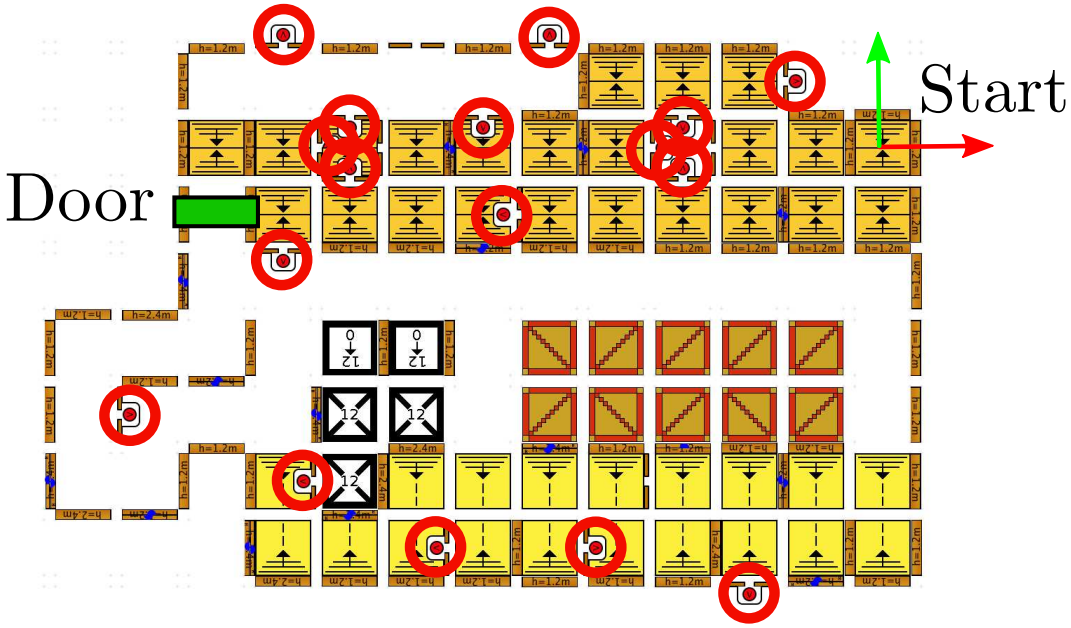
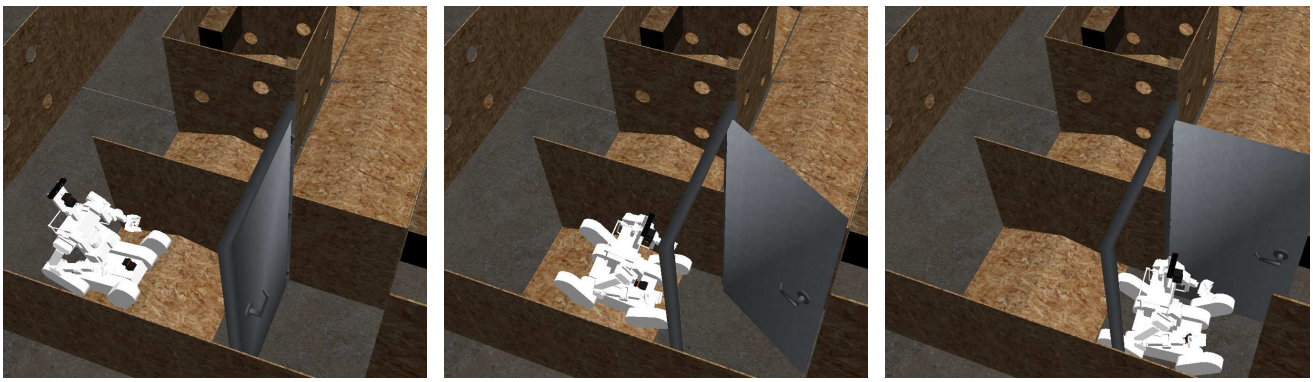


Figure 9.13: The scenario used for the exploration scenario requiring door opening.



(a) (b) (c)

Figure 9.14: Hector Centaur opening a door: (a): Robot approaching the door. (b): Supervisor assists in opening the door. (c): Supervisor switches control back to autonomy, robot proceeds with mission.

10 Summary

10.1 Contributions

This thesis makes several contributions to research and development of highly versatile supervised intelligent robotic systems for urban search and rescue and disaster response tasks which are summarized in the following.

Holistic Approach for Supervised Autonomous Robots for Disaster Response

In order to enable and support versatile robotic capabilities as well as flexible modes of control and interaction between human supervisor and robots, a holistic systems-oriented approach based on three pillars is presented in this thesis.

First, a broad range of heterogeneous robots with diverse capabilities ranging from exploration to manipulation is considered. These represent the diverse physical capabilities needed to address versatile tasks in urban search and rescue and disaster response.

Second, the full range of (remote) human-robot control and interaction modalities from teleoperation over supervised and assisted to full autonomy is considered as well as flexible changes between them during a mission. This is a necessary prerequisite to making the best use of human and robot capabilities to accomplish tasks in the degraded environment of a disaster scene, as semi-autonomous, much less fully autonomous robots will for a long time still not be sufficiently reliable and robust to be used for deployment in real disaster situations.

Third, an accordingly derived systems architecture is presented, which supports these requirements as well as modularity, adaptability, extensibility and openness.

Large parts of the system architecture have been used by two and functional components by three teams in the DARPA Robotics Challenge as well as by multiple other researchers. Related own publications are [81], [85], [83] and [82].

In addition to the holistic approach, a number of contributions are being made in this thesis to research and development of key functional components for the considered robotic systems.

Robust Simultaneous Localization and Mapping for USAR Environments

With *hector_slam* (Chapter 6) a flexible and robust SLAM approach is presented that is significantly more robust than other state of the art approaches. It allows for learning of a map of the environment even for the case of challenging motion behavior of the platform carrying the used sensors. It is available as open source and widely used by other researchers for diverse applications such as simulated USAR tasks, for mapping with quadrotor UAVs or with new low-cost LIDAR sensors. Related own publications are [86] and [136].

Navigation and Search for Victims

A comprehensive approach for the exploration and search for victims in USAR situations is described (Chapter 7). It allows for reliable autonomous navigation and detection of victims. Unlike most prior work, a holistic view is also taken here, incorporating exploration, recovery on failure as well as victim detection and approach strategies. Applied to ground robotic systems the developed approach demonstrates to be superior to competing approaches, as evaluated

during annual participation in the RoboCup Rescue Robot League competition. Related own publications are [98], [83], [82] and [84].

Manipulation System for Complex Disaster Response Tasks

An approach for performing complex manipulation tasks under varying communication constraints is described (Chapter 8). It also considers related perception and planning aspects in a holistic fashion, allowing for manipulation even under severe bandwidth constraints. The system allows using the full range of control and interaction modalities between supervisor and robot from teleoperation to full autonomy. The applicability is demonstrated for challenging real-world scenarios. Whereas related approaches have been developed in parallel by other teams participating in the DARPA Robotics Challenge, this is the only one which has already been applied and demonstrated on three different types of humanoid robots. Related own publications are [81], [85], [127], [126] and [146].

Experimental Evaluation

Evaluation through experiments is mandatory for performance investigations (Chapter 9). However, evaluation of components only provides limited insight into overall system performance, which is to a large extent governed by the interaction between components. This becomes especially relevant when considering different levels of autonomy and allowing the capability to seamlessly switch between them. This is in contrast to testing standalone components under laboratory conditions which, while providing insight into isolated components performance, does not allow evaluating overall system performance for complex tasks. Therefore, systems-oriented evaluation and benchmarking, which are usually not in focus of academic research but of high relevance for practical applications, play an important role in this thesis. These are performed through participation in renowned international competitions for urban search and rescue (RoboCup Rescue Robot League) and disaster response (DARPA Robotics Challenge). Based on a simulated case study with a new centaur-type robot, the applicability and combination of highly autonomous exploration for victim search and highly versatile manipulation capabilities using a single robot with a supervisor in the loop is demonstrated.

10.2 Outlook

While the contributions in this thesis provide a holistic design for the design of USAR robots and components that enable a human-robot team to execute complex tasks, further research is required until semi-autonomous versatile USAR robotic systems will be capable, reliable and robust enough to move out of laboratories and competitions into real world applications, to be used in disasters and save human lives.

Increasing capabilities and reliability have to be achieved in multiple related areas, from hardware to software. As prominent examples, perception (including semantic scene understanding) and locomotion are to be mentioned. Perception abilities are critical for proper assessment of situations of both, human and robot.

Full 3D perception is a prerequisite for many real-world applications as well as perception in degraded environments. With advanced and multi-modal sensors and corresponding approaches becoming available, significant advances can be expected. Once mature, the ability of reliable 3D perception and mapping will open up new opportunities for the use of robots in USAR tasks.

Also, planning and control of locomotion in arbitrary 3D terrain remain highly challenging tasks that are yet only partially solved. Especially different ground characteristics and non-rigid or potentially collapsible environments pose significant challenges that need to be addressed.

Finally, considering robots as avatars for human response forces, such versatile locomotion needs to be combined with equally versatile manipulation capabilities.

[1] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, H. B. Brown Jr, D. McMordie, U. Saranli, R. Full, and D. E. Koditschek. Rhex: a biologically inspired hexapod runner. *Autonomous Robots*, 11(3):207–213, 2001.

[2] F. Amigoni. Experimental evaluation of some exploration strategies for mobile robots. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2818–2823. IEEE, 2008.

[3] F. Amigoni, E. Bastianelli, J. Berghofer, A. Bonarini, G. Fontana, N. Hochgeschwender, L. Iocchi, G. Kraetzschmar, P. Lima, M. Matteucci, P. Miraldo, D. Nardi, and V. Schiaffonati. Competitions for benchmarking: Task and functionality scoring complete performance assessment. *Robotics Automation Magazine, IEEE*, 22(3):53–61, Sept 2015.

[4] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth, and B. Schiele. Vision based victim detection from unmanned aerial vehicles. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1740–1747, Oct 2010.

[5] R. C. Arkin. *Behavior-based robotics*. MIT press, 1998.

[6] A. Bachrach, R. He, and N. Roy. Autonomous flight in unknown indoor environments. *International Journal of Micro Air Vehicles*, 1(4):217–228, 2009.

[7] D. Bakken. Middleware, 2001.

[8] B. Balaguer, S. Balakirsky, S. Carpin, and A. Visser. Evaluating maps produced by urban search and rescue robots: lessons learned from robocup. *Autonomous Robots*, 27(4):449–464, 2009.

[9] S. Balakirsky and Z. Kootbally. Usarsim/ros: A combined framework for robotic control and simulation. In *ASME/ISCIE 2012 International Symposium on Flexible Automation*, pages 101–108. American Society of Mechanical Engineers, 2012.

[10] S. Balakirsky, C. Scrapper, S. Carpin, and M. Lewis. Usarsim: providing a framework for multi-robot performance evaluation. In *Proceedings of PerMIS*, volume 2006, 2006.

[11] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. G. Atkeson, M. Gennert, and T. Padir. Human-supervised control of the atlas humanoid robot for traversing doors. In *Humanoid Robots (Humanoids), 15th IEEE-RAS International Conference on*, 2015.

[12] M. J. Barton. *Controller development and implementation for path planning and following in an autonomous urban vehicle*. PhD thesis, The University of Sydney, 2001.

[13] N. Basilico and F. Amigoni. Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. *Autonomous Robots*, 31(4):401–417, 2011.

[14] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth. Robotic roommates making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 529–536. IEEE, 2011.

[15] S. Behnke. Robot competitions-ideal benchmarks for robotics research. In *Proc. of IROS-2006 Workshop on Benchmarks in Robotics Research.*, 2006.

[16] P. Biber and W. Straßer. The Normal Distributions Transform: A New Approach to Laser Scan Matching. In *IEEE/RJS Intern. Conf. on Intelligent Robots and Systems*, 2003.

[17] M. Biegl, R. Hasenauer, L. Silberbauer, P. Filo, J. Orgonas, B. Paholkova, and C. Weber. Marketing testbeds for high tech innovation: The case of taurob robotics. In *Management of Engineering Technology (PICMET), 2014 Portland International Conference on*, pages 1145–1168, July 2014.

[18] R. Bischoff, T. Guhl, E. Prassler, W. Nowak, G. Kraetzschmar, H. Bruyninckx, P. Soetens, M. Haegele, A. Pott, P. Breedveld, et al. Brics-best practice in robotics. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–8. VDE, 2010.

[19] A. Bonchis, P. I. Corke, D. C. Rye, and Q. P. Ha. Variable structure methods in hydraulic servo systems control. *Automatica*, 37(4):589–595, 2001.

[20] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[21] T. Braunl. Research relevance of mobile robot competitions. *IEEE Robotics & Automation Magazine*, 6(4):32–37, 1999.

[22] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[23] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 476–481. IEEE, 2000.

[24] F. Burget and M. Bennewitz. Stance selection for humanoid grasping tasks by inverse reachability maps. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5669–5674, May 2015.

[25] A. Burguera, Y. González, and G. Oliver. On the use of likelihood fields to perform sonar scan matching localization. *Autonomous Robots*, 26:203–222, May 2009.

[26] J. Butzke, K. Daniilidis, A. Kushleyev, D. D. Lee, M. Likhachev, C. Phillips, and M. Phillips. The university of pennsylvania magic 2010 multi-robot unmanned vehicle system. *Journal of Field Robotics*, 29(5):745–761, 2012.

[27] J. Carlson and R. R. Murphy. Reliability analysis of mobile robots. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 274–281. IEEE, 2003.

[28] J. Carlson and R. R. Murphy. How ugvs physically fail in the field. *Robotics, IEEE Transactions on*, 21(3):423–437, 2005.

[29] S. Carpin. Merging maps via hough transform. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1878–1883. IEEE, 2008.

[30] S. Carpin, A. Birk, and V. Jucikas. On map merging. *Robotics and autonomous systems*, 53(1):1–14, 2005.

[31] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. Usarsim: a robot simulator for research and education. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1400–1405. IEEE, 2007.

[32] S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis, and J. Wang. Quantitative assessments of usarsim accuracy. In *Proceedings of PerMIS*, volume 2006, 2006.

[33] J. Casper and R. R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 33(3):367–385, 2003.

[34] S. Chitta, I. Sucan, and S. Cousins. MoveIt! *IEEE Robotics Automation Magazine*, 19(1):18–19, 2012.

[35] P. Clements, D. Garlan, L. Bass, J. Stafford, R. Nord, J. Ivers, and R. Little. *Documenting software architectures: views and beyond*. Pearson Education, 2002.

[36] J. R. de Souza, D. Guimaraes, F. Pimentel, F. Sapucaia, C. Laranjeira, L. Jesus, M. A. Simoes, and D. Frias. Bahiart@ home 2015 team description paper. Technical report, Universidade do Estado da Bahia, 2015.

[37] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[38] A. Diosi and L. Kleeman. Fast Laser Scan Matching using Polar Coordinates. *The International Journal of Robotics Research*, 26(10):1125, 2007.

[39] C. Dornhege and A. Kleiner. A frontier-void-based approach for autonomous exploration in 3d. *Advanced Robotics*, 27(6):459–468, 2013.

[40] DRC-Teams. What happened at the DARPA Robotics Challenge? `www.cs.cmu.edu/~cga/drc/events`, 2015.

[41] I. Dryanovski, W. Morris, and J. Xiao. An open-source pose estimation system for micro-air vehicles. In *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, pages 4449–4454, 2011.

[42] I. Dryanovski, W. Morris, and J. Xiao. An open-source pose estimation system for micro-air vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4449–4454, 2011.

[43] C. Du, K.-H. Lee, and W. Newman. Manipulation planning for the atlas humanoid robot. In *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, pages 1118–1123, Dec 2014.

[44] G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, and S. Stinckwich. Simulating complex robotic scenarios with morse. In *SIMPAR*, pages 197–208, 2012.

[45] A. H. Eden and R. Kazman. Architecture, design, implementation. In *proceedings of the 25th International Conference on Software Engineering*, pages 149–159. IEEE Computer Society, 2003.

[46] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[47] H.-s. Eom, J. W. Cho, and K. Jeong. Some reliability considerations of ugv for remote-response in nuclear emergency situation. 2013.

[48] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. D'Arpino, R. Deits, M. DiCicco, D. Fourie, et al. An architecture for online affordance-based perception and whole-body planning. *Journal of Field Robotics*, 32(2):229–254, 2015.

[49] M. F. Fallon, M. Antone, N. Roy, and S. Teller. Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 112–119. IEEE, 2014.

[50] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart. Kinect v2 for mobile robot navigation: Evaluation and modeling. In *Advanced Robotics (ICAR), 2015 International Conference on*, pages 388–394, July 2015.

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

[52] T. Fong, C. Thorpe, and C. Baur. *Collaborative control: A robot-centric model for vehicle teleoperation*. Carnegie Mellon University, The Robotics Institute, 2001.

[53] M. Gäbel, T. Krüger, and U. Bestmann. Design of a quadrotor system for an autonomous indoor exploration. In *IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014*. Delft University of Technology, 2014.

[54] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[55] C. S. Goodman and D. E. Hogan. Urban search and rescue. *Disaster medicine*, 2007.

[56] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[57] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi. Fast and accurate slam with rao–blackwellized particle filters. *Robotics and Autonomous Systems*, 55(1):30–38, 2007.

[58] S. Grzonka, G. Grisetti, and W. Burgard. Towards a Navigation System for Autonomous Indoor Flying. In *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2009.

[59] M. Habbecke and L. Kobbelt. Iterative Multi-View Plane Fitting. In *Vision, modeling, and visualization*, pages 73–80, 2006.

[60] R. Hahn, D. Lang, M. Häselich, and D. Paulus. Heat mapping for improved victim detection. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 116–121. IEEE, 2011.

[61] B. Hailpern and P. Tarr. Model-driven development: The good, the bad, and the ugly. *IBM systems journal*, 45(3):451–461, 2006.

[62] R. Halterman and M. Bruch. Velodyne HDL-64E lidar for unmanned surface vehicle obstacle detection. In *SPIE Conference*, Apr 2010.

[63] J. Heinly, E. Dunn, and J.-M. Frahm. Comparative evaluation of binary features. In *Computer Vision–ECCV 2012*, pages 759–773. Springer, 2012.

[64] R. W. Hogg, A. L. Rankin, S. I. Roumeliotis, M. C. McHenry, D. M. Helmick, C. F. Bergh, and L. Matthies. Algorithms and sensors for small robot path following. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3850–3857. IEEE, 2002.

[65] D. Holz, N. Basilico, F. Amigoni, and S. Behnke. Evaluating the efficiency of frontier-based exploration strategies. *ISR/ROBOTIK 2010*, 2010.

[66] D. Holz and S. Behnke. Sancta simplicitas - on the efficiency and achievable results of SLAM using ICP-based incremental registration. In *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, pages 1380–1387, 2010.

[67] D. Holz and S. Behnke. Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In S. Lee, H. Cho, K.-J. Yoon, and J. Lee, editors, *Intelligent Autonomous Systems 12*, volume 194 of *Advances in Intelligent Systems and Computing*, pages 61–73. Springer Berlin Heidelberg, 2013.

[68] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[69] M. Isenburg. Laszip. *Photogrammetric Engineering & Remote Sensing*, 79(2):209–217, 2013.

[70] A. Jacoff, R. Sheh, A.-M. Virts, T. Kimura, J. Pellenz, S. Schwertfeger, and J. Suthakorn. Using competitions to advance the development of standard test methods for response robots. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pages 182–189. ACM, 2012.

[71] M. Johnson, J. M. Bradshaw, P. J. Feltovich, C. M. Jonker, M. B. Van Riemsdijk, and M. Sierhuis. Coactive design: Designing support for interdependence in joint activity. *Journal of Human-Robot Interaction, 3 (1), 2014*, 2014.

[72] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, et al. Team IHMC's lessons learned from the DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 32(2):192–208, 2015.

[73] S. Joyeux, J. Schwendner, T. M. Roehr, and R. I. Center. Modular software for an autonomous space rover. In *Proceedings of the 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2014), Montreal, Québec, Canada*, 2014.

[74] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. on Robotics, TRO*, 24(6):1365–1378, Dec 2008.

[75] R. Kaijaluoto et al. Precise indoor localization for mobile laser scanner. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 1–6, May 2015.

[76] R. Katz and T. J. Allen. Investigating the not invented here (nih) syndrome: A look at the performance, tenure, and communication patterns of 50 r & d project groups. *R&D Management*, 12(1):7–20, 1982.

[77] K. Khoshelham. Accuracy analysis of kinect depth data. In *ISPRS workshop laser scanning*, volume 38, page W12, 2011.

[78] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 739–743. IEEE, 1999.

[79] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[80] A. Kleiner and C. Dornhege. Operator-assistive mapping in harsh environments. In *Safety, Security & Rescue Robotics (SSRR), 2009 IEEE International Workshop on*, pages 1–6. IEEE, 2009.

[81] S. Kohlbrecher, D. C. Conner, A. Romay, F. Bacim, D. A. Bowman, and O. von Stryk. Overview of Team ViGIR's Approach to the Virtual Robotics Challenge. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–2. IEEE, 2013.

[82] S. Kohlbrecher, F. Kunz, D. Koert, C. Rose, P. Manns, K. Daun, J. Schubert, A. Stumpf, and O. von Stryk. Towards highly reliable autonomy for urban search and rescue robots. In *Proc. RoboCup Symposium 2014*, Lecture Notes in Artificial Intelligence (LNAI). Springer, 2014.

[83] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. von Stryk. Hector open source modules for autonomous mapping and navigation with rescue robots. In S. Behnke, M. Veloso, A. Visser, and R. Xiong, editors, *RoboCup 2013: Robot World Cup XVII*, volume 8371 of *Lecture Notes in Computer Science*, pages 624–631. Springer Berlin Heidelberg, 2014.

[84] S. Kohlbrecher, K. Petersen, G. Steinbauer, J. Maurer, P. Lepej, S. Uran, R. Ventura, C. Dornhege, A. Hertle, R. Sheh, et al. Community-driven development of standard

software modules for search and rescue robots. In *Proceedings of the 10th IEEE International Symposium on Safety Security and Rescue Robotics (SSRR 2012)*, 2012.

[85] S. Kohlbrecher, A. Romay, A. Stumpf, A. Gupta, O. von Stryk, F. Bacim, D. A. Bowman, A. Goins, R. Balasubramanian, and D. C. Conner. Human-robot Teaming for Rescue Missions: Team ViGIR's Approach to the 2013 DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 32(3):352–377, 2015.

[86] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160. IEEE, 2011.

[87] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent. Sparse Pose Adjustment for 2D Mapping. In *IEEE/RSJ Intern. Conf. on Intelligent robots and systems (IROS)*, Oct 2010.

[88] G.-J. M. Kruijff, V. Tretyakov, T. Linder, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, E. Pianese, S. Corrao, et al. Rescue robots at earthquake-hit mirandola, italy: a field report. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–8. IEEE, 2012.

[89] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. On measuring the accuracy of slam algorithms. *Autonomous Robots*, 27(4):387–407, 2009.

[90] F. Kunz. Evaluation von Navigationssoftware für mobile Such- und Rettungsroboter. Technical report, TU Darmstadt, 2011.

[91] H. A. Lauterbach, D. Borrmann, R. Heß, D. Eck, K. Schilling, and A. Nüchter. Evaluation of a backpack-mounted 3d mobile scanning system. *Remote Sensing*, 7(10):13753–13781, 2015.

[92] A. Leeper, K. Hsiao, M. Ciocarlie, I. Sucan, and K. Salisbury. Methods for collision-free arm teleoperation in clutter using constraints from 3D sensor data. In *IEEE Intl. Conf. on Humanoid Robots*, Atlanta, GA, 10/2013 2013.

[93] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *DARPA Image Understanding Workshop*, pages 121–130, Apr 1981.

[94] J. Machado Santos, D. Portugal, and R. P. Rocha. An evaluation of 2D SLAM techniques available in Robot Operating System. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–6. IEEE, 2013.

[95] K. D. Mankoff and T. A. Russo. The kinect: A low-cost, high-resolution, short-range 3d camera. *Earth Surface Processes and Landforms*, 38(9):926–936, 2013.

[96] E. Marder-Eppstein, E. Berger, T. Foote, B. P. Gerkey, and K. Konolige. The office marathon: Robust navigation in an indoor office environment. In *International Conference on Robotics and Automation*, 05/2010 2010.

[97] E. R. Messina and A. S. Jacoff. Measuring the performance of urban search and rescue robots. In *Technologies for Homeland Security, 2007 IEEE Conference on*, pages 28–33. IEEE, 2007.

[98] J. Meyer, P. Schnitzspan, S. Kohlbrecher, K. Petersen, O. Schwahn, M. Andriluka, U. Klingauf, S. Roth, B. Schiele, and O. von Stryk. A Semantic World Model for Urban Search and Rescue based on Heterogeneous Sensors. In *RoboCup 2010: Robot Soccer World Cup XIV*, pages 180–193, 2011.

[99] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk. Comprehensive simulation of quadrotor uavs using ros and gazebo. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411. Springer, 2012.

[100] N. Mohamed, J. Al-Jaroodi, and I. Jawhar. Middleware for robotics: A survey. In *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, pages 736–742. IEEE, 2008.

[101] S. Moon, W. Eom, and H. Gong. Development of large-scale 3d map generation system for indoor autonomous navigation flight–work in progress. *Procedia Engineering*, 99:1132–1136, 2015.

[102] D. Moore, E. Olson, and A. Huang. Lightweight communications and marshalling for low-latency interprocess communication. 2009.

[103] R. Murphy, J. Casper, M. Micire, and J. Hyams. Assessment of the nist standard test bed for urban search and rescue. *NIST SPECIAL PUBLICATION SP*, pages 260–266, 2001.

[104] R. R. Murphy. Trial by fire [rescue robots]. *Robotics & Automation Magazine, IEEE*, 11(3):50–61, 2004.

[105] R. R. Murphy. Findings from nsf-jst-nist workshop on rescue robotics. In *Safety Security and Rescue Robotics (SSRR), 2010 IEEE International Workshop on*, pages 1–4. IEEE, 2010.

[106] R. R. Murphy. *Disaster robotics*. MIT press, 2014.

[107] R. R. Murphy and J. L. Burke. Up from the rubble: Lessons learned about hri from search and rescue. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 437–441. SAGE Publications, 2005.

[108] R. R. Murphy, J. Casper, and M. Micire. Potential tasks and research issues for mobile robots in robocup rescue. In *RoboCup 2000: Robot Soccer World Cup IV*, pages 339–344. Springer, 2001.

[109] R. R. Murphy and S. Stover. Rescue robots for mudslides: A descriptive study of the 2005 la conchita mudslide response. *Journal of Field Robotics*, 25(1-2):3–16, 2008.

[110] S. Nikitaki, P. M. Scholl, K. Van Laerhoven, and P. Tsakalides. Localization in wireless networks via laser scanning and bayesian compressed sensing. In *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on*, pages 739–743. IEEE, 2013.

[111] E. Olson. Real-Time Correlative Scan Matching. In *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, pages 4387–4393, Jun 2009.

[112] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goeddel, M. Bulic, J. Crossman, and B. Marinier. Progress toward multi-robot reconnaissance and the magic 2010 competition. *Journal of Field Robotics*, 29(5):762–792, 2012.

[113] J. Pellenz. Mapping and map scoring at the robocuprescue competition. 2008.

[114] J. Pellenz, A. Jacoff, T. Kimura, E. Mihankhah, R. Sheh, and J. Suthakorn. Robocup rescue robot league. In *RoboCup 2014: Robot World Cup XVIII*, pages 673–685. Springer, 2015.

[115] K. Petersen. *General Concepts for Human Supervision of Autonomous Robot Teams*. PhD thesis, TU Darmstadt, Department of Computer Science, July 9 2013.

[116] K. Petersen and O. von Stryk. An event-based communication concept for human supervision of autonomous robot teams. *International Journal on Advances in Intelligent Systems*, 4(3 and 4):357 – 369, 2011.

[117] V. Pradeep, K. Konolige, and E. Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *Experimental Robotics*, pages 211–225. Springer, 2014.

[118] G. Pratt and J. Manzo. The darpa robotics challenge [competitions]. *Robotics & Automation Magazine, IEEE*, 20(2):10–12, 2013.

[119] A. Ratter and C. Sammut. Local map based graph slam with hierarchical loop closure and optimisation. *Australasian Conference on Robotics and Automation 2015*, 2015.

[120] E. S. Raymond. *The Cathedral & the Bazaar: Musings on linux and open source by an accidental revolutionary*. " O'Reilly Media, Inc.", 2001.

[121] S. Rebel, F. Hüning, I. Scholl, and A. Ferrein. Mqone: Low-cost design for a rugged-terrain robot platform. In *Intelligent Robotics and Applications*, pages 209–221. Springer, 2015.

[122] R. Reid, A. Cann, C. Meiklejohn, L. Poli, A. Boeing, and T. Braunl. Cooperative multi-robot navigation, exploration, mapping and object detection with ros. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1083–1088. IEEE, 2013.

[123] N. Ritter and M. Ruth. The geotiff data interchange standard for raster geographic images. *International Journal of Remote Sensing*, 18(7):1637–1647, 1997.

[124] C. A. Rockey. *Low-Cost Sensor Package for Smart Wheelchair Obstacle Avoidance*. PhD thesis, Case Western Reserve University, 2012.

[125] E. Rohmer, S. P. Singh, and M. Freese. V-rep: A versatile and scalable robot simulation framework. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1321–1326. IEEE, 2013.

[126] A. Romay, S. Kohlbrecher, D. Conner, and O. von Stryk. Achieving versatile manipulation tasks with unknown objects by supervised humanoid robots based on object templates. In *IEEE-RAS Intl. Conf. on Humanoid Robots*, page to appear, Nov. 3-5 2015.

[127] A. Romay, S. Kohlbrecher, D. C. Conner, A. Stumpf, and O. von Stryk. Template-based manipulation in unstructured environments for supervised semi-autonomous humanoid robots. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 979–986. IEEE, 2014.

[128] D. Rubinstein and R. Hitron. A detailed multi-body model for dynamic simulation of off-road tracked vehicles. *Journal of Terramechanics*, 41(2):163–173, 2004.

[129] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[130] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011.

[131] S. Saeedi, L. Paull, M. Trentini, M. Seto, and H. Li. Efficient map merging using a probabilistic generalized voronoi diagram. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4419–4424. IEEE, 2012.

[132] D. Scaramuzza and R. Siegwart. A practical toolbox for calibrating omnidirectional cameras. 2007.

[133] E. Schmidlin. *Can this robot pass through that aperture? Learning to make judgments for a tele-operated robot.* PhD thesis, Texas Tech University, 2014.

[134] F. E. Schneider, B. Gaspers, K. Peräjärvi, and M. Gårdestig. Current state of the art of unmanned systems with potential to be usedfor radiation measurements and sampling: Erncip thematic group radiological and nuclear threats to critical infrastructure task 3 deliverable 1. 2015.

[135] F. E. Schneider, D. Wildermuth, and H.-L. Wolf. Elrob and eurathlon: Improving search & rescue robotics through real-world robot competitions. In *Robot Motion and Control (RoMoCo), 2015 10th International Workshop on*, pages 118–123. IEEE, 2015.

[136] P. M. Scholl, S. Kohlbrecher, V. Sachidananda, and K. Van Laerhoven. Fast indoor radiomap building for rssi-based localization systems. In *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*, pages 1–2. IEEE, 2012.

[137] S. Schwertfeger. *Robotic Mapping in the Real World: Performance Evaluation and System Integration*. PhD thesis, IRC-Library, Information Resource Center der Jacobs University Bremen, 2012.

[138] S. Scone and I. Phillips. Trade-off between exploration and reporting victim locations in usar. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, pages 1–6. IEEE, 2010.

[139] F. Seco, A. Jimenez, C. Prieto, J. Roa, and K. Koutsou. A survey of mathematical methods for indoor localization. *Intelligent Signal Processing*, pages 9–14, 2009.

[140] R. Sheh, A. Jacoff, A.-M. Virts, T. Kimura, J. Pellenz, S. Schwertfeger, and J. Suthakorn. Advancing the state of urban search and rescue robotics through the robocuprescue robot league competition. In *Field and Service Robotics*, pages 127–142. Springer, 2014.

[141] M. Shimizu, N. Koenig, A. Visser, and T. Takahashi. A realistic robocup rescue simulation based on gazebo.

[142] K. Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH computer graphics*, 19(3):245–254, 1985.

[143] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2008.

[144] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *AAAI/IAAI*, pages 852–858, 2000.

[145] E. Strickland. 24 hours at fukushima. *Spectrum, IEEE*, 48(11):35–42, 2011.

[146] A. Stumpf, S. Kohlbrecher, D. C. Conner, and O. von Stryk. Supervised footstep planning for humanoid robots in rough terrain tasks using a black box walking controller. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 287–294. IEEE, 2014.

[147] P. Tang, D. Huber, and B. Akinci. A comparative analysis of depth-discontinuity and mixed-pixel detection algorithms. In *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*, pages 29–38. IEEE, 2007.

[148] R. Tedrake. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems, 2014.

[149] D. Thomas. *Middleware for Efficient Programming of Autonomous Mobile Robots*. PhD thesis, TU Darmstadt, Department of Computer Science, Nov. 19 2010.

[150] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, 2008.

[151] J. Tuley, N. Vandapel, and M. Hebert. Analysis and removal of artifacts in 3-d ladar data. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2203–2210. IEEE, 2005.

[152] N. Vahrenkamp, T. Asfour, and R. Dillmann. Robot placement based on reachability inversion. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1970–1975. IEEE, 2013.

[153] P. Velagapudi, P. Scerri, K. Sycara, H. Wang, M. Lewis, and J. Wang. Scaling effects in multi-robot control. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2121–2126. IEEE, 2008.

[154] K. Verbiest, S. Berrabah, and E. Colon. Autonomous frontier based exploration for mobile robots. In H. Liu, N. Kubota, X. Zhu, R. Dillmann, and D. Zhou, editors, *Intelligent Robotics and Applications*, volume 9246 of *Lecture Notes in Computer Science*, pages 3–13. Springer International Publishing, 2015.

[155] E. A. Wan and R. V. D. Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158. IEEE, 2000.

[156] J. Wang, M. Lewis, S. Hughes, M. Koes, and S. Carpin. Validating usarsim for use in hri research. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 457–461. SAGE Publications, 2005.

[157] S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.

[158] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal. Industrial adoption of model-driven engineering: are the tools really the problem? In *Model-Driven Engineering Languages and Systems*, pages 1–17. Springer, 2013.

[159] T. Wiley, M. McGill, A. Milstein, R. Salleh, and C. Sammut. Spatial correlation of multisensor features for autonomous victim identification. In *RoboCup 2011: Robot Soccer World Cup XV*, pages 538–549. Springer, 2012.

[160] B. Willaert, H. Van Brussel, and G. Niemeyer. Stability of model-mediated teleoperation: discussion and experiments. In *Haptics: Perception, Devices, Mobility, and Communication*, pages 625–636. Springer, 2012.

[161] S. Wirth and J. Pellenz. Exploration transform: A stable exploring algorithm for robots in rescue environments. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–5. IEEE, 2007.

[162] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.

[163] H. A. Yanco and J. L. Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, pages 111–119, 2002.

[164] T. Yoshida, K. Nagatani, S. Tadokoro, T. Nishimura, and E. Koyanagi. Improvements to the rescue robot quince toward future indoor surveillance missions in the fukushima daiichi nuclear power plant. In *Field and Service Robotics*, pages 19–32. Springer, 2014.

[165] A. Zelinsky. Using path transforms to guide the search for findpath in 2d. *The International Journal of Robotics Research*, 13(4):315–325, 1994.

[166] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, 13:119–152, Oct 1994.

[167] V. A. Ziparo, A. Kleiner, L. Marchetti, A. Farinelli, and D. Nardi. Cooperative exploration for usar robots with indirect communication. In *Proc. of 6th IFAC Symposium on Intelligent Autonomous Vehicles, IAV*, 2007.

Own Publications

Journal Papers

Stefan Kohlbrecher, Alberto Romay, Alexander Stumpf, Anant Gupta, Oskar von Stryk, Felipe Bacim, Doug A Bowman, Alex Goins, Ravi Balasubramanian, and David C Conner. Human-robot Teaming for Rescue Missions: Team ViGIR's Approach to the 2013 DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 32(3):352–377, 2015.

Conference Papers

Mykhaylo Andriluka, Paul Schnitzspan, Johannes Meyer, Stefan Kohlbrecher, Karen Petersen, Oskar Von Stryk, Stefan Roth, and Bernt Schiele. Vision based victim detection from unmanned aerial vehicles. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1740–1747. IEEE, 2010.

Johannes Meyer, Paul Schnitzspan, Stefan Kohlbrecher, Karen Petersen, Mykhaylo Andriluka, Oliver Schwahn, Uwe Klingauf, Stefan Roth, Bernt Schiele, and Oskar von Stryk. A semantic world model for urban search and rescue based on heterogeneous sensors. In *RoboCup 2010: Robot Soccer World Cup XIV*, pages 180–193. Springer, 2011.

Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160. IEEE, 2011.

Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar von Stryk. Comprehensive simulation of quadrotor uavs using ros and gazebo. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411. Springer, 2012.

Stefan Kohlbrecher, Karen Petersen, Gerald Steinbauer, Johannes Maurer, Peter Lepej, Suzana Uran, Rodrigo Ventura, Christian Dornhege, Andreas Hertle, Raymond Sheh, et al. Community-driven development of standard software modules for search and rescue robots. In *Proceedings of the 10th IEEE International Symposium on Safety Security and Rescue Robotics (SSRR 2012)*, 2012.

Philipp M Scholl, Stefan Kohlbrecher, Vinay Sachidananda, and Kristof Van Laerhoven. Fast indoor radio-map building for rssi-based localization systems. In *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*, pages 1–2. IEEE, 2012.

Stefan Kohlbrecher, David C Conner, Alberto Romay, Felipe Bacim, Doug Bowman, Oskar von Stryk, et al. Overview of team vigir's approach to the virtual robotics challenge. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–2. IEEE, 2013.

Stefan Kohlbrecher, Johannes Meyer, Thorsten Graber, Karen Petersen, Uwe Klingauf, and Oskar von Stryk. Hector open source modules for autonomous mapping and navigation with rescue robots. In *RoboCup 2013: Robot World Cup XVII*, pages 624–631. Springer, 2014.

Alexander Stumpf, Stefan Kohlbrecher, David C Conner, and Oskar von Stryk. Supervised footstep planning for humanoid robots in rough terrain tasks using a black box walking controller. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 287–294. IEEE, 2014.

Alberto Romay, Stefan Kohlbrecher, David C Conner, Alexander Stumpf, and Oskar von Stryk. Template-based manipulation in unstructured environments for supervised semi-autonomous humanoid robots. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 979–986. IEEE, 2014.

Stefan Kohlbrecher, Florian Kunz, Dorothea Koert, Christian Rose, Paul Manns, Kevin Daun, Johannes Schubert, Alexander Stumpf, and Oskar von Stryk. Towards highly reliable autonomy for urban search and rescue robots. In *RoboCup 2014: Robot World Cup XVIII*, pages 118–129. Springer, 2015.

Alberto Romay, Stefan Kohlbrecher, David C. Conner, and Oskar von Stryk. Achieving versatile manipulation tasks with unknown objects by supervised humanoid robots based on object templates. In *IEEE-RAS Intl. Conf. on Humanoid Robots*, page to appear, Nov. 3-5 2015.

Workshop Papers

Stefan Kohlbrecher and Oskar Von Stryk. Modeling observation uncertainty for soccer playing humanoid robots. In *Proc. 5th Workshop on Humanoid Soccer Robots at the*, pages 6–8. Citeseer, 2010.

Stefan Kohlbrecher, Alexander Stumpf, and Oskar Von Stryk. Grid-based occupancy mapping and automatic gaze control for soccer playing humanoid robots. In *Proc. 6th Workshop on Humanoid Soccer Robots at the*, 2011.

Wissenschaftlicher Werdegang[1]

05/1999	Allgemeine Hochschulreife
10/2000 — 10/2009	Studium der Informatik an der Technischen Universität Darmstadt
10/2009	Diplom in Informatik
seit 11/2009	Wissenschaftlicher Mitarbeiter, Fachbereich Informatik, Technische Universität Darmstadt

Erklärung[2]

Hiermit erkläre ich, dass ich die vorliegende Arbeit, mit Ausnahme der ausdrücklich genannten Hilfsmittel, selbständig verfasst habe.

[1] gemäß § 20 Abs. 3 der Promotionsordnung der TU Darmstadt
[2] gemäß § 9 Abs. 1 der Promotionsordnung der TU Darmstadt