Prototyping and Benchmarking Autonomous Mobile Service Robot Applications

Thomas Wisspeintner

Dissertation zur Erlangung der Doktorwürde
der Naturwissenschaften

Eingereicht am Fachbereich Mathematik und Informatik
der Freien Universität Berlin, Juni 2009

Erstgutachter: Prof. Dr. Raúl Rojas

Zweitgutachter: Prof. Dr. Daniele Nardi

Tag der mündlichen Prüfung: 26. Oktober 2009

Abstract

This thesis describes methods and approaches which aim to accelerate scientific research and development in mobile service robotics through standardization in hardware and software and through application-oriented system benchmarking. It contains three major contributions. The first contribution is the design and implementation of a mechatronic construction kit for efficient and multipurpose mobile service robot prototyping called *VolksBot*. The relevance and multipurpose character of this development is demonstrated through various robot components, platforms and applications that were implemented on the basis of this kit and which are used by an international robotic community. Second, is the sensor development for mobile robotics and autonomous urban driving. This includes an adaptive catadioptric vision system for varying light conditions *IAISVision* and a continuously rotating 3D laser range finder *3DLS-K* which was used as a sensor component for autonomous driving in the *DARPA Urban Challenge 2007*. The third major contribution is the design and implementation of the *RoboCup*@Home initiative, an international scientific competition for benchmarking and developing personal domestic service robots. Here, system benchmarking is applied to test the robots' performance in specific abilities in a realistic and uncertain setting which includes human-robot interaction. In summary, the combination of standardization, modularization, community building and benchmarking in mobile robotic research and development is proposed as a suitable approach to cope with the high level of related complexity and unsolved problems in the domain. Furthermore, this approach aims to foster and accelerate the development and the distribution of useful, affordable and broadly accepted robotic applications and products in the near future.

Acknowledgements

This thesis would not have been possible without the personal and practical support of numerous people. My sincere gratitude goes to my girlfriend, my father, my family and all my friends, and to my colleagues and superiors for their support and patience over the last few years.

Many people at the Autonomous Robot department at Fraunhofer Institute IAIS encouraged me in various ways during my work and research. I wish to thank Professors Thomas Christaller, Paul Plöger and Frank Pasemann, and Dr. Ansgar Bredenfeld, for their continuous guidance and support of my professional and scientific career in the field of autonomous mobile robotics at Fraunhofer IAIS. I am truly grateful for the time and the freedom they granted me to finish my research besides the persistent requirements of daily work.

My thanks go out to my former colleagues and friends Peter Schöll, Keyan Zahedi, Reiner Frings, Hartmut Surmann, Stefan May, Kai Pervölz, Yasutake Takahashi and Walter Nowak, with whom I enjoyed working on several projects, and who supported, encouraged and inspired me in my daily work. My former students also deserve my true appreciation; I was privileged to supervise them, and I also learned a lot from them.

I am especially grateful to the Board of Trustees and the Presidents of the RoboCup Federation, in particular, Professors Manuela Veloso, Minoru Asada, Hiroaki Kitano, Daniele Nardi, Hans-Dieter Burkhard, Gerhard Kraetzschmar, and my friend Adam Jacoff from the Board of Executives. Without their trust, support and advice, the RoboCup@Home league would have remained just a concept that never came to fruition. In this context, I would like to warmly thank my friend and colleague Tijn van der Zant for hours of inspiring discussions, for creating new ideas and concepts and for putting them into practice together. The success of this initiative also highly depended and still depends on the dedication, the hard work and the valuable feedback of the growing community, the teams, the Organizing and Technical Committee of *RoboCup*@Home. In particular, I wish to thank Luca Iocci, Stefan Schiffer, Jesus Savage, Komei Sugiura and Javier Ruiz-del-Solar for all their efforts.

Professor Daniele Nardi receives my hearty thanks for his support, and for being an experienced mentor and guide during the conceptual design and the successful introduction of the RoboCup@Home initiative and beyond. I truly appreciate the time he spent and the productive feedback which he gave me as reviewer of this thesis.

Finally, this dissertation would not have been possible without the expert guidance of my esteemed advisor, Prof. Raúl Rojas. Many times he acted as a role model to me, showing me how to set, pursue and attain goals in a team of motivated and capable people in a short amount of time and under challenging conditions. I am especially grateful that he gave me the opportunity to be part of *Team Berlin* which reached the semi-finals at the *DARPA Urban Challenge*. Being part of this groundbreaking project has been one of the most enriching experiences in my career. As advisor of my thesis, I would like to give him my sincere thanks for his direct and valuable feedback, as his comments were always extremely perceptive, helpful and appropriate.

Contents

Preface

This thesis describes an integral set of methods and approaches which aim to accelerate scientific research and development in mobile service robotics through standardization in hardware and software, and through application-oriented system benchmarking. In particular, it covers the design of a construction kit for prototyping mobile service robots called *VolksBot*. This modular kit consists of mechanical, hardware, software and sensor components and offers a high level of reconfigurability. The development of two specific sensor systems is further discussed in detail, namely the catadioptric vision system *IAISVision* and a 3D-LIDAR system *3DLS-K*. Furthermore, the definition and implementation of a set of standard benchmark tests for personal domestic service robots is presented, which has become the largest international domestic service robotic competition called *RoboCup*@Home.

VolksBot is a mechatronic construction kit for prototyping mobile service robot applications. The modular and multi-purpose robot construction kit consists of reusable components in hardware, software and mechanics. To allow for a broad field of applications, the initial kit was further mechanically enhanced to fit the high physical demands of rough terrain, outdoor conditions and high payload (*VolksBot RT*). Further, an evolutionary design approach was used to optimize the morphology of a *VolksBot* variant to obtain high mobility performance. This approach has led to the implementation of a robot capable of climbing stair-cases (*VolksBot XT*). As a result, the multipurpose character and the feasibility of this construction kit approach are demonstrated by showing the range of applications, the distribution of the *VolksBot* kit and the meeting of the initially set design goals. Applications include an underwater rover for maritime research, robot rescue, robot soccer, autonomous transportation, domestic service robotics, as well as research and education. Up to now, *VolksBot* and its components have been used by over 60 institutions from research, education and industry world-wide.

IAISVision is an adaptive catadioptric color vision system. The physical design of the camera system can be adapted to various environmental conditions and applications through an iterative design approach in simulation. A 3-step method which consists of a PI camera parameter control using reference colors, image segmentation and color classification is presented as a method to obtain a high grade of color constancy. Thus, it allows for adaptive color recognition under various light conditions indoors and outdoors which is shown through analysis of the experimental results.

3DLS-K is a continuously rotating 3D-LIDAR system, which I developed and integrated as a component for the VolksBot construction kit and which I later applied to the domain of autonomous urban driving. Insight into the sensor hardware development and the acquisition of 3D data is given. The relevance and the universal applicability of this development is demonstrated by the successful implementation and integration of the system as a sensor component for the autonomous car *Spitit of Berlin* of *Team Berlin*, which participated in the *DARPA Urban Challenge* in 2007 and reached the semi-finals. In particular, a method for scene analysis of the 3D data at intersections and an according behavior system on the basis of a finite state machine is presented, which was evaluated during the competition.

The *RoboCup*@Home initiative targets the development and deployment of autonomous service and assistive robot technology being essential for future personal domestic applications. The *@Home* framework consists of independent benchmarks testing relevant robot key abilities in a dynamic and realistic environment which contains a high level of uncertainty. Further focus is put on the exchange of knowledge in a multidisciplinary community, the iterative adaptation of the benchmarks, the human-robot interaction and on high-level system integration. Performance metrics measurements indicate a significant increase of the robots' performance during the past several years. Since 2006, 25 teams from Asia, Europe, the US and Australia participated, and national competitions in China, Mexico, Iran, Japan and Germany were established. In 2009, 26 teams from 14 countries pre-registered. Novel and relevant personal service robotic applications and technology, as well as scientific contributions, are beginning to emerge from within the growing community, which currently includes 250 members.

The combination of standardization, modularization, community building and benchmarking in mobile robotic research and development is proposed as a suitable approach to cope with the high level of related complexity and unsolved problems in the domain. I strongly believe that such an integral approach can help to accelerate and direct progress toward relevant and useful application and product development in the field of mobile service robotics in the near future.

1

Introduction

Scientific research and development of autonomous mobile robots has been a continuous effort over the course of the last four decades. Obviously great progress has been made in this period in relevant research fields such as artificial intelligence, knowledge representation, online-learning, real-time processing, control, actuator and sensor development or high-level system integration. Still, looking at early autonomous mobile robot development [1], one can see that for the majority of today's mobile robots, the applied principles and the general system architecture have not changed significantly over time. Similarities, especially in hardware and used control methodologies, are still clearly observable. Since the beginning of these developments in the late 1960s, predictions about the near breakthrough of this technology for personal domestic use and service robotic applications have been made, with a recent one made by Bill Gates [2]. Although this breakthrough has not yet happened on a large scale, there are some promising indications that it will occur in the near future. There has been a significant increase in related research activities, and several service robotic applications are beginning to appear on the market, which will be further discussed this chapter.

This thesis aims to accelerate scientific research and development in mobile service robotics by identifying, designing and implementing methods and approaches for standardization in robotic hardware and software development, and for application-oriented system benchmarking. The proposed combination of standardization, rapid prototyping community-building and benchmarking aims foster the development and the distribution of real-life applications and products in the domain of mobile service robotics in the near future.

The following sections describe major challenges in mobile robotic research and development, and they push for standardization in hardware and software to manage the high complexity of the related problems. Further, an overview on prominent service robotic applications, projects and products already available on the market is given. Finally, the state of the art on standardization efforts in mobile robotic hardware and software development, and on mobile robotic benchmarking and scientific competitions, is summarized.

1.1 Challenges in autonomous robotic research and development

Autonomous mobile robotics research and development is a highly multidisciplinary effort, requiring contributions from many different fields such as computer science, electrical and mechanical engineering, artificial intelligence, mechatronics, biology, or brain-, cognitive-, social- and material science. The demand on collaboration, knowledge exchange and resources is comparably high in robotic-related projects. As the speed of innovation is quite high, the previously explored technology and the robotic systems themselves become outdated quickly, resulting in new system development from scratch on a frequent basis. Combined with high maintenance efforts for these often highly specialized, monolithic and non-standardized systems, this pulled a lot of resources away from the actual research to be done in the past. The lack of standardization often goes hand in hand with a lack of robustness of these prototype systems, a factor which leads to even more maintenance effort. From own experience, I believe that the high demand on resources to conduct robotic-related projects combined with the high complexity and uncertainty of the real-life environment and problems appears to be a primary reason for the still minor role of robotics in today's everyday life. As the market for robotic hardware and components is still very small, the prices of these often new and hardly tested prototypes are very high. For new components, such as e.g. sensors, robot developers often play the role of alpha testers and have to deal with unfinished development or undocumented functionality. Still, there are some promising indications that this situation will change as more relevant base technology like bus systems, embedded controllers, sensors and actuators is being developed and used in the automotive domain. With advanced driver assistance systems like adaptive cruise control, collision warning, pedestrian protection, lane change assistance, night vision, Car2car communication or automatic parking, modern cars have never been closer to the autonomous robotic domain. Both domains are beginning to merge and form a common basis. With large quantities and high pressure on price combined with a high demand on quality and robustness, these technologies and related components will surely play a more and more important role in product development of mobile service robotics, too.

Most of the robotic research is still being conducted in university environments. There, the systems are frequently used for education purposes; a wide range of relevant technical knowledge can be conveyed, students involved are usually highly motivated, and secondary skills like group work and project-oriented work are implicitly trained. Under these circumstances, usually a high level of fluctuation of people (students and researchers) occurs. As a consequence, the danger of loss of knowledge in these projects is high and needs to be addressed. Standardization, process orientation, documentation, use of frameworks, modularization and usage of agile software development methods [3] can help to prevent re-development due to loss of knowledge and to increase the overall quality in the development process.

The potential application domains for mobile service robots are already quite diverse. They include safety, security and surveillance, transportation, intelligent vehicles, autonomous underwater vehicles, space exploration or edutainment. Furthermore, socially relevant domains like urban search and rescue, personal assistance, caregiving and domestic service robotics exist and focus especially on Human-Robot-Interaction(HRI). Although the high potential of these application domains is commonly recognized, most of the projects and initiatives addressing them are research-oriented and do not focus on product development, due to the many unsolved problems still present in unconfined real world environments. For personal use, some consumer products already exist. Such applications include floor cleaning[1], lawn mowing[2] and surveillance[3]. Still, these service robots are specialized in a certain task and lack some important properties of a multipurpose, autonomous and intelligent domestic service robot, such as HRI. Prominent examples of domestic and personal assistant robot research projects are ReadyBot[4], PR2[5], Wakamaru[6] and PaPeRo[7]. Recent research projects in the domain of rehabilitation and caregiving[8] also show promising results.

[1] iRobot (http://irobot.com)
[2] Robomow (http://www.friendlyrobotics.com)
[3] Robowatch (http://robowatch.com)
[4] ReadyBot (http://www.readybot.com)
[5] PR2 (http://www.willowgarage.com)
[6] Wakamaru (http://www.mhi.co.jp/kobe/wakamaru/english)
[7] PaPeRo (http://www.nec.co.jp/robot/english/robotcenter_e.html)
[8] Care-O-Bot(http://www.care-o-bot.de/english)

1.2 Standardization in robotic development

Over the last several years, the push for standardization in robotic development processes has been growing. Achieving synergies and exchange of knowledge is crucial due to the large and diverse problem space in robotics research and development. Furthermore, it allows for high-level system integration and the maintenance of an operable robot platform while still being able to focus on individual research aspects. Standardization efforts include standard robot platforms, software development frameworks, middle-ware and interface definitions, reference architectures, simulation, benchmarking and robot construction kits.

Software architectures, frameworks and middleware The RoSta project[1] focuses on standardization and reference architectures for mobile service robots. Software frameworks for robot control include Carmen [4], Player/Stage [5], MRPT[2], MRS[3], or Orocos [6]. OpenCV[4] is an open source software library containing algorithms for computer vision with diverse applications, as shown in [7].

Simulation With higher computational capabilities of modern computers and steadily improving physical simulation tools like, for example, ODE [8], and robot simulators, such as USARSim[5] [9], robot simulation plays a relevant role in the robotic development process because it helps avoid some of the difficulties related to physical robot design and operation. Still, it cannot (yet) fully compensate for the experiences gained with a real physical system interacting in the real world. Depending on the actual field of application, this is mainly due to a still high level of abstraction of the world model in simulation and a limited amount of contained uncertainty. This is especially true when the scenario contains Human-Robot-Interaction that is still hard to simulate.

Robot construction kits and development platforms With respect to hardware, robot construction kits and development platforms offer an efficient way to reduce hardware-related design and maintenance efforts. Still, most construction kit approaches focus on education and edutainment, and, therefore, their use for application-oriented prototyping of mobile service robots is very limited. Prominent examples of

[1] http://www.robot-standards.eu
[2] The Mobile Robot Programming Toolkit (http://babel.isa.uma.es/mrpt/index.php/Main_Page)
[3] Microsoft Robotics Studio (http://msdn.microsoft.com/en-us/robotics/default.aspx)
[4] The Open Computer Vision Library (http://sourceforge.net/projects/opencv)
[5] Unified System for Automation and Robot Simulation (http://sourceforge.net/projects/usarsim)

commercially available construction kits used for education purposes are Lego Mindstorms [10], Fischertechnik Mobile Robots[1] or the Vex Robotics Design System[2]. The MoRob Project [11] aimed to develop a construction kit for education as well as research. The approach was similar to that of the Tetrixx kit [12] but provided a larger scope due to its ability to include additional hardware such as sensors [13]. On the other hand, robot development platforms offer a specific mobile robot base which usually can be equipped with additional hardware components. Prominent examples are ER1[3], Koala[4] or Pioneer and PeopleBot[5].

1.3 Mobile robot benchmarking and competitions

Benchmarking has been recognized as a fundamental activity to advance robotic technology [14, 15], and many activities are in progress. Some projects and special groups are working on defining standard benchmarking methodologies and data sets for many robotic problems, like HRI, SLAM or navigation. Examples for such initiatives are the the EURON Benchmarking Initiative[6], the international workshops on Benchmarks in Robotics Research and on Performance Evaluation and Benchmarking for Intelligent Robots and Systems, held since 2006[7]. The Rawseeds project[8] aims at creating standard benchmarks especially for localization and mapping.

Benchmarking can be distinguished in two classes: *system benchmarking*, where the robotic system is evaluated as a whole, and *component benchmarking*, where a specific functionality is evaluated. Component benchmarking is very important to compare different solutions to a specific problem and to identify the best algorithms and approaches. Among the many examples, much effort has been put on mapping and SLAM (e.g. [16, 17]), and on navigation (e.g. [18, 19, 20]). Conversely, system benchmarking offers an effective way to measure the performance of an entire robotic system in the accomplishment of complex tasks, as such tasks require the interplay of various sub-systems or approaches. In this kind of benchmarking, a standard reference environment, reference tasks and related performance metrics are to be defined. Examples of system benchmarking are given in the fields of interactive robots [21] and socially assistive robots [22].

[1] Fischertechnik RoboPro (http://www.fischertechnik.de/en)
[2] VEX Robotics (http://www.vexrobotics.com)
[3] ER1 (http://www.evolution.com)
[4] Koala II (http://www.k-team.com)
[5] Pioneer and PeopleBot (http://www.mobilerobots.com)
[6] http://www.euron.org/activities/benchmarks/index
[7] These workshops are summarized at http://www.robot.uji.es/EURON/en/index.htm
[8] http://www.rawseeds.org

Moreover, competitions provide an efficient means of interacting and communicating among research groups because they are often associated with scientific conferences or workshops and provide visibility on their research to a large audience. Finally, annual competitions enable feedback on a yearly basis about the increase in performance and allow research teams to set up medium-term projects. Among the many robotic competitions, AAAI Mobile Robot Competitions was one of the first, being established in 1992 [23]. RoboCup (founded in 1997) [24] currently has the largest number of participants (e.g. 440 teams with over 2600 participants from 35 countries in 2006), and the DARPA Grand Challenge is likely the most recognized competition in terms of public and media attention, and it is the one that is most directly application-oriented. Initiatives directly related to Domestic Service Robotics mainly aim for a single specific task. For example, the AHRC Vacuum Contest[1] and the 2002 IROS Cleaning Contest[2] [25] are focused only on floor cleaning, while ROBOEXOTICA[3] focuses on the single task of robots preparing and serving drinks. A more general initiative is given by the ICRA HRI Challenge[4]. However, it is still at a preliminary stage since evaluation criteria for benchmarking the performance have not been defined. Furthermore, educational contests, such as EUROBOT[5] or RoboCup Junior[6], are organized with the main goal of presenting robotics to young students, and thus they deal with simpler tasks and robotic platforms. Many of these competitions have obtained very relevant results which will be analyzed further in section 4.2.
This thesis proposes methods and approaches for accelerating research and application-oriented development in the domain of autonomous mobile service robotics. It structures as follows:

Chapter 2 details the conceptual design and implementation of a mechatronic construction kit called *VolksBot* for physical rapid prototyping of mobile service robot applications. It demonstrates how the concept for a modular and multipurpose robot construction kit consisting of reusable hardware, software and mechanics components was specified and implemented. It also shows how the kit was extended to fit the high physical demands of rough terrain and high payload, which are essential to many service robotic applications. This work further discusses an evolutionary design approach used to optimize robot morphology and obtain high mobility performance. As will be

[1] http://www.botlanta.org
[2] http://robotika.cz/competitions/cleaning2002/en
[3] http://www.roboexotica.org/en/mainentry.htm
[4] http://lasa.epfl.ch/icra08/hric.php
[5] http://www.eurobot.org
[6] http://rcj.sci.brooklyn.cuny.edu

explained, this resulted in a wheeled robot capable of climbing staircases (*VolksBot XT*) on the basis of the *VolksBot* kit. Various examples of implemented service robotic applications on the basis of the construction kit will demonstrate the multipurpose character and the feasibility of the kit. Briefly, these applications include an underwater rover for maritime research and sea-bed analysis, robot rescue, soccer and domestic service robotics. Further results include measures on construction times and on the distribution and use of *VolksBot* components in robotic research and education.

In chapter 3, two sensor systems of the *VolksBot* construction kit are discussed in more detail. This includes the development of an adaptive catadioptric color vision system *IAISVision* and the development of a continuously rotating 3D laser scanner *3DLS-K*. First, the hardware design of the vision system is discussed, showing how the physical design can be easily adapted to various environment conditions. Then, a 3-step method, which consists of a PID camera parameter control using reference colors, image segmentation and color classification, is presented. This method provides a high grade of color constancy and therefore allows for adaptive color recognition under various light conditions, which is shown via experimental results. Then, the design of the new, continuously rotating laser scanner *3DLS-K*, and its application to autonomous urban driving, is presented. The system was used as a sensor component for the autonomous car "'Spirit of Berlin"' during the DARPA Urban Challenge 2007.

Chapter 4 addresses the problem of benchmarking and developing personal service robotics applications through scientific competitions. As will be detailed, this resulted in the conceptual design and implementation of the *RoboCup*@Home initiative in 2006, currently the largest international competition for benchmarking domestic service robots. Here, the main challenges are the large degree of uncertainty of the dynamic and realistic environments, and the related human interaction which the robots have to cope with. Furthermore, the application-orientation requires a large effort towards high level integration combined with a demand for general robustness of the systems. First, the need for an interdisciplinary community effort to iteratively identify problems, to define benchmarks, to test and finally to solve the problems is advocated. Then, the concepts and the implementation of the *RoboCup*@Home initiative as a combination of scientific exchange and competition is presented as an efficient method to accelerate and focus technological and scientific progress in the domain of personal service robotics. Finally, the progress in terms of performance increase in the benchmarks and technological advancements is evaluated and discussed.

Chapter 5 provides the final conclusions and summarizes the future challenges in the fields. Finally, chapter 6 lists the contributions of this thesis.

2

VolksBot - A construction kit for multipurpose robot prototyping

2.1 Component-based robot development and prototyping

Autonomous mobile robotics research and development is a highly multidisciplinary effort, requiring contributions from many different fields such as computer science, electrical and mechanical engineering, artificial intelligence, mechatronics, signal processing, sensor technology or control theory. For application-oriented development, additional robustness and long lifetimes are required. The demand on collaboration, knowledge exchange and resources are comparably high in robotic-related projects. Looking at robotic hardware development in the past, these systems were often monolithic, highly integrated prototypes that took a long time to develop, were costly and hard to maintain. As the innovation rate is quite high, the contemporary technology and robotic systems became outdated quickly, resulting in new system development from scratch on a frequent basis. Combined with the elevated maintenance needs for these often highly specialized and non-standardized systems, in the past this drained a lot of resources away from the actual research to be done. The lack of standardization often goes in hand with a lack of robustness of these prototype systems, which requires even more upkeep. Further difficulties experienced in robotic-related research projects in university environments include the fluctuation of people involved in the project, the long training times for new project members and the loss of knowledge. In general, the high demand on resources to conduct robotic-related projects combined with the high complexity and uncertainty of real life environments appears to be a major reason for the still minor role of robotics in modern daily life, as it decelerates scientific progress significantly.

With the aim of addressing and reducing the impact of these difficulties related to mobile robot design, I initiated and conducted the conceptual design and implementation of a multipurpose, robust and cost-effective construction kit for mobile robot prototyping, the *VolksBot* project from 2003 until 2008 at Fraunhofer IAIS. The approach should enable developers to maintain a focus on a specific domain while still being able to have a clear understanding of the entire system. This is achieved via different levels of abstraction and well-defined interfaces to hardware and software modules. Furthermore, (re-)usability should be maximized by having well-documented system components of manageable size.

In this chapter, the methodology and the implementation of this modular, component-oriented design approach is presented, resulting in the prototyping of various robot platforms and components for different domains, including research, education and application prototyping. Originally applied to indoor scenarios like RoboCup Middle Size, the concept was extended to fulfill the demands of real-life applications like outdoor use, higher payload, velocity or scalability in morphology and hardware configuration of the platform.

State of the art in modular robot design Component-based prototyping concepts have been applied successfully in developing robots mainly for indoor applications or in the field of education where some approaches use a construction kit. The advantages of using a kit are quite obvious as this usually reduces development time and costs by fostering reuse of existing components. On the other hand, universal modules are not specialized, thus one loses in performance. There is always a trade-off between the general applicability and the performance in modular approaches.

Significant work has been done in the field of rapid prototyping of robots in the past. Won et al. [26] have shown that rapid prototyping is a viable method of creating articulated structures of robotic systems. Reshko et al. [27] have illustrated methods to quickly produce prototypes of desired quality in considerably little time by using ready-made components such as servo motors, sensors and standard plastic parts such as Lego blocks. Examples for robot construction kits mainly used for education and edutainment are Lego Mindstorms [10], Fischertechnik Mobile Robots[1], Tetrixx [12] or the Cubesystem [28]. Though aspects of modularity are addressed well by these systems, they are limited in on-board computational power and focus on miniaturization and low-cost hardware. As a consequence the aspect of application-oriented rapid

[1] Fischertechnik RoboPro (http://www.fischertechnik.de/en)

prototyping of fully autonomous robots is hardly provided in these approaches and on-board perception is limited. On the other side, several robot platforms with higher complexity in sensors, actuators and higher processing power are usually specialized for a certain field of application or a certain scenario[1][2][3]. Besides, many of these types of systems are specific in their morphology; their mechanics and hardware do not fit with a construction kit approach. The MoRob Project [11] aimed to develop a construction kit to build robots which could be used for education as well as research. The approach was similar to that of the Tetrixx kit, but it provided a larger scope due to its ability to include sensory equipment [13] in the development kit.

This chapter is structured as follows. First, design goals for the robot construction kit are derived from the analysis of the problems and challenges mentioned in section 1.1. Then, concrete design criteria for implementation of the kit are derived from these goals, and the interrelation between goals and criteria are shown. In section 2.3 the application of these criteria to form the *VolksBot* robot construction kit, consisting of reusable modules and components in mechanics, hardware and software, is presented. Section 2.4 shows how this kit was physically enhanced to build robots suitable for outdoor use and high payloads. This extension allows the design of a set of new robot variants(*VolksBot RT*). Furthermore, the design of a high mobility rover (*VolksBot XT*) by application of evolutionary design optimization is presented in section 2.5.3. To demonstrate the feasibility of the design approach for application-oriented prototyping, section 2.6 lists various applications ranging from robot soccer to autonomous transportation, robot rescue and service robotics which have been efficiently implemented on the basis of the extended construction kit. Section 2.7, demonstrates how the defined design goals and criteria have been met on a large scale by showing the relationship of these goals and criteria to the implementation of the construction kit and the robot platforms. Measures of the time required to assemble *VolksBot* platforms further support the claim of rapid prototyping. The results conclude with an overview on the distribution and use of *VolksBot* variants and components in the international robotic community. The chapter concludes with an outlook on future work.

[1] ER1 (http://www.evolution.com)
[2] Koala II (http://www.k-team.com)
[3] Pioneer and PeopleBot (http://www.mobilerobots.com)

2.2 Design goals and criteria

With the aim of developing an approach for multipurpose robot prototyping, I derived several design goals in hardware and software in order to address many of the challenges and problems mentioned in section 1.1. These design goals are listed and explained in the following:

G1 Reduce costs, time and resources in mobile robotics projects

G2 Managing system complexity

G3 Allow for exchange and reuse of existing components

G4 Allow for efficient reconfiguration and extension of the systems

G5 Low maintenance efforts, simple assembly procedures

G6 Allow for efficient integration of existing technology

G7 Foster exchange of knowledge

G8 Robust and scalable mechanical design

G9 Allow for a wide range of robot variants and applications

G10 Allow for short training periods for new users

G11 Achieve synergies through standardization

The goals are labeled in brackets(G1-G11) for later reference. One of the major goals is to reduce costs, time and resources needed to conduct mobile robotic projects(G1). This should motivate more groups from various backgrounds to start or continue activities related to mobile robotics in education and research. It should further help to generate interest and open the market for new service robot applications with more companies being willing to invest in mobile robot technology and prototyping projects. The complexity of robotic systems has grown consistently with the complexity of the applications they have been designed for. Modern mobile robots usually require a variety of sensors, actuators and controllers, but they also need methods for signal processing, sensor data fusion, planning, localization, navigation and control of the robot, especially when being used in real world environments. The approach, therefore, should enable the developers to manage this constantly growing system complexity(G2). The system should allow the exchange and reuse of existing components in hardware and software(G3). With this, reinvention and re-implementation of existing technology, as well as new system development from scratch, should be avoided. Also, an existing robot platform should be easily reconfigurable and extendable by use of common hardware and software components(G4) allowing for quick adaptation to new applications and tasks. Assembly and maintenance of the platform should be simple and efficient, and it should require a minimal number of special tools or machinery(G5). This way, users are independent from having access to a large mechanical workshop, and experts have more time to spend on research and development. Often, groups have previously worked in the domain of mobile robotics. Because of their experience, they should be able to efficiently integrate pre-existing technology into the kit(G6) or extend and upgrade their existing system with construction kit components. The approach should help to foster the exchange and distribution of knowledge(G7). The design of robotic systems usually requires the interplay of many different individual skills which are distributed over a group or multiple groups of people. Component-oriented design and modularization can help to facilitate exchange and counteract loss of knowledge. The mechanics of the kit should be robust and scalable and allow for high payloads and high dynamics(G8). To offer a wide range of possible applications, the kit should allow the design and configuration of diverse mobile robot variants for different tasks and environments(G9). The training periods for new users should be short(G10), addressing the problem of fluctuation of people and allowing new people to be productive in a shorter amount of time. Synergies should be achieved through standardization(G11). Setting standards in mobile robotics projects to foster synergy between different research groups is an active research topic[1][2]. Recently, Microsoft has introduced Robotics Studio[3] to foster exchange and synergy on the software level. The RoboCup four-legged league has provided excellent examples of how using a standardized platform in combination with consequent code sharing can accelerate research and development in this domain.

From these design goals, I derived a set of design criteria for the robot construction kit in hardware, software and mechanics. While the design goals are of more general means, the design criteria define a set of rules for the concrete implementation of the construction kit. The following list summarizes these design criteria:

[1]The Object Management Group (http://www.omg.org)
[2]The Rosta project (http://www.robot-standards.org)
[3]Microsoft Robotic Studio (http://msdn.microsoft.com/en-us/robotics

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
G1	x	x	x	x	x	x				x
G2		x						x	x	x
G3	x	x	x	x	x	x			x	x
G4	x	x	x	x	x			x	x	x
G5	x	x		x			x			x
G6	x					x			x	x
G7					x	x	x	x	x	x
G8	x		x							
G9			x							x
G10		x			x	x	x	x	x	x
G11	x				x	x				

Table 2.1: Relationships between design goals and design criteria

C1 Extensive use of standardized, industrial components

C2 Small number of different components with high reconfigurability

C3 Fine granularity of modules

C4 Build up mechanical component library in CAD

C5 Build up software library with documentation and coding standards

C6 Use and integrate existing software and frameworks

C7 Apply documentation standards for components

C8 Introduce multiple abstraction layers in robot hardware and software design

C9 Clear interface definitions for hardware and software components

C10 Avoid dependencies between components

The relationships between the design goals(G1-11) and design criteria(C1-10) are shown in table 2.1, and example relationships for each design criterion are given in the following:

To reduce the costs and efforts for manufacturing and hardware design(G1), standardized and available industrial components should be used if applicable(C1). To keep the system complexity low(G2) and to be able to maintain the construction kit(G5), the number of components should be kept minimal(C2), but offer a high grade of reconfigurability(G4). Components should posses a fine granularity(C3) and should be universal to reduce costs(G1) and ensure reuse(G3,G9). A comprehensive mechanical component library should be built up using standard CAD software tools(C4). Before actually building the robot, a complete assembly should be done in CAD to avoid major design errors and allow fast iterations during the design phase through component reuse(G1,G3,G4). The same holds true for software development, where a software library should be assembled using state-of-the-art software development standards for architecture, documentation and coding conventions(C5) to achieve synergies(G11). In addition to developing individual software, existing software and frameworks should be used and integrated into the approach(C6,G6). When developing a component in hardware or software, documentation standards for developers and users should be applied(C7). This, in turn, reduces training periods for new users(G10) and fosters an exchange of knowledge(G7). Different layers of abstraction should be provided during system integration and development in hardware and software(C8). This should help to reduce training times(G10) and allow a wide range of people from different technical backgrounds to work with the system(G2). Clear interface definitions for hardware and software components have to be defined and maintained(C9) to be able to manage the system complexity(G2). Furthermore to keep the number of possible variants high(G9) and the system complexity low(G2), dependencies among components should be avoided(C10).

2.3 Modular composition of the VolksBot construction kit

On the basis of the design goals and criteria listed in the previous section, I developed and implemented a concept for a mobile robot construction kit composed of reusable modules and components in mechanics, hardware and software called *VolksBot* [29] [30]. Fig. 2.1 shows the first version of a *VolksBot Indoor* robot built with this kit. Here, a differential drive unit, a catadioptric vision system, batteries, a control notebook and a motor controller are mounted on the central chassis frame consisting of X-beams. A modular software framework is used for the robot control (see section 2.3.4). This initial development established the basis for further continuous enhancements of the kit towards prototyping mobile service robots for various applications (see section 2.6). Details on the concept and the modular composition of the *VolksBot* construction kit are given in the following:

2.3.1 Layers for modular hardware composition

First, an overview of the construction kit's hardware composition is given by referring to the design criteria in brackets(C1-C10).

Figure 2.1: The first VolksBot Indoor variant

To account for a high grade of reconfigurability(C2) and to avoid dependencies between components(C10), the hardware assemblies of all *VolksBot* robots follow a common hierarchal structure (see Fig. 2.2). This structure consists of four layers(C8). On the top layer, the *Assembly layer*, a complete *VolksBot* assembly is implemented as a combination of the *Drive assembly*, the *Chassis assembly* and the *Electric/Electronic assembly*. These assemblies are composed of modules from the *Module layer* which again are composed of a combination of components from the *Component layer*. The lowest layer is the *Connection layer*, where standardized mechanical connections of the module or component to the chassis frame(C1), or electrical connections like power supply or interfaces to batteries or the Control PC, are implemented(C9).

All robot assemblies are constructed as a combination of the same hardware modules, such as the *Universal Drive Unit* (UDU, see section 2.4.2), the *Motor Unit* or the *Motor Control Unit*. These modules consist of components such as bearing blocks, X-Beams(C1), chain wheels and shafts. Structuring the assemblies in such manner allows the build-up of a hardware component pool and therefore ensures the reusability(C3) of modules and components in the real robot assembly, as well as in the CAD model. In addition, this method provides compatibility between the modules while allowing the

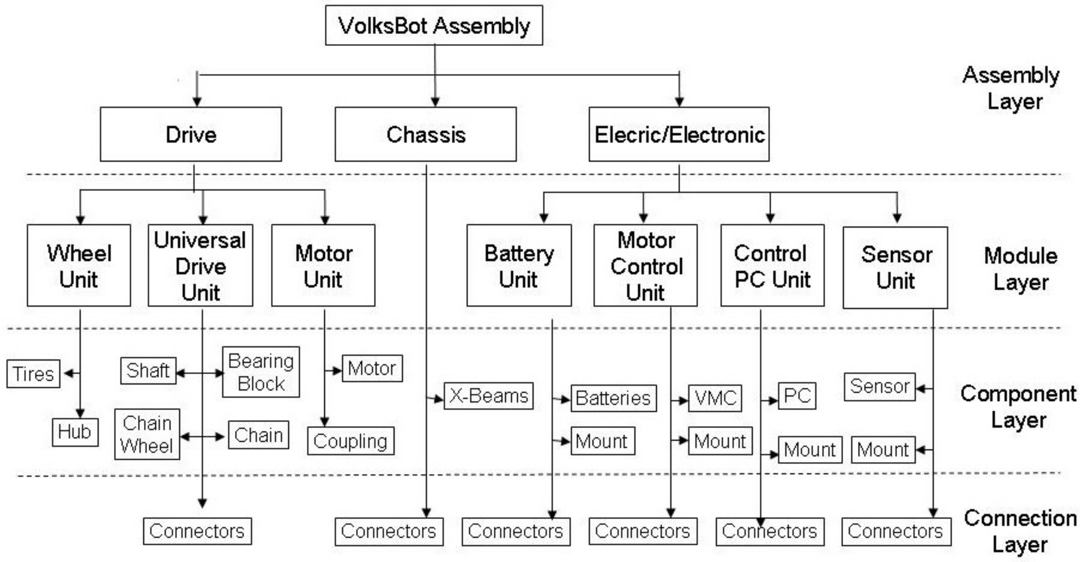


Figure 2.2: Hierarchal structure of the robot hardware composition

implementation of a wide range of robot variants as dependencies between the modules are minimized(C10) and interfaces are clearly defined(C9).

2.3.2 CAD component library

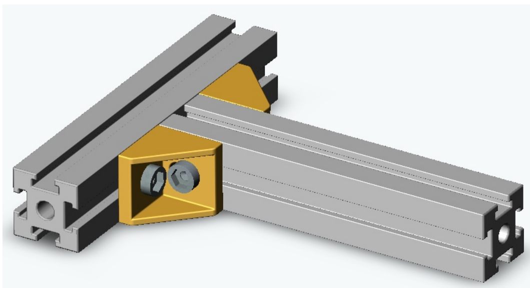


Figure 2.3: A basic assembly in SolidWorks

According to this hierarchal structure, all robot assemblies and their underlying hardware modules and components are modeled in CAD[1], building up a common CAD component library(C4), before physical assembly of a robot platform. This library helps to reduce the mechatronic design efforts enormously(G1) because new robot variants are designed mostly by recombination and adaptation of existing modules and components. Fig. 2.3 shows the 3D-view of a chassis frame element assembly in SolidWorks.

[1]Solidworks (http://www.solidworks.com)

In this example, the frame elements (X-beams) from the *Component layer* and the angular connectors and the screws from the *Connection layer* were modeled in CAD once, put in the CAD component library and are reused constantly throughout various robot designs. The library contains models of all required mechanical and hardware modules and their components like frame elements, batteries, battery mounts, sensors, motor controllers, actuators and mechanical connectors. A new robot variant is first completely modeled by use of these components in CAD before physical implementation of the robot design. An example of the hardware composition of an advanced robot assembly in CAD, i.e. a *VolksBot XT* (see section 2.5.3) is shown in figure 2.4. In this example new modules like the *Leg-Lever Unit* were designed in CAD by use of existing components such as chassis frame elements and the *Universal Drive Unit* (see section 2.4.2). These were made available in the CAD component library for later reuse. The CAD component library further facilitated the implementation of an as-

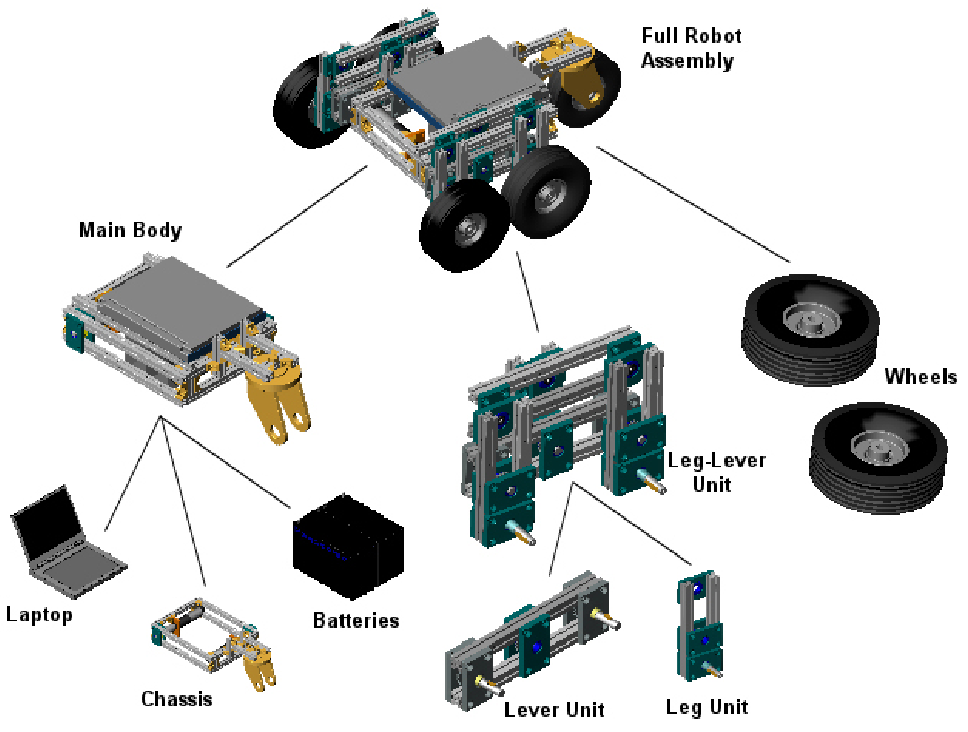


Figure 2.4: Hardware composition of a VolksBot XT variant in CAD

sembly manual(C7) for each of the standard *VolksBot* variants, i.e. *VolksBot Indoor v.2*, *VolksBot RT3, RT4* and *RT6* (see section 2.4.3). These detailed assembly manuals provide step-by-step instructions for robot assembly, which allows even inexperienced users to assemble a robot. An excerpt of an assembly manual is given in the Appendix.

Chassis frame In accordance to the design criteria(C1), standard aluminum machine construction extrusions (X-beams)[1] of 20mm width and compatible connectors are used to build up the robot's chassis frame (see Fig. 2.5). These beams provide high rigidity, are light-weight and offer a variety of assembly and connection options(C2). Size and shape of the robot's chassis frame can be adapted individually by basic mechanical processing (i.e. cutting and screwing)(G5). By using compatible pluggable t-nut-connectors, it is possible to establish new physical connections without having to decompose the frame. All sides of the X-beams can be used to connect to additional modules and components. All hardware modules are directly connected to the chassis frame. Therefore, only physical dependencies between the module and the chassis frame occur, not between the modules themselves. This way, a high grade of flexibility in the actual robot design is obtained: Direct physical dependencies between the modules would limit the number of possible robot variants(C10).

For the first VolksBot indoor variant (see Fig. 2.5), all modules, i.e. the battery unit, the motor control unit, drive units and sensors, are connected to a rectangular single-layered chassis frame. This allows for simple reconfiguration like repositioning of modules and scaling of the platforms. With the later aim of developing rough-terrain robots, and to account for higher robot payloads, a double-layered frame was introduced in combination with a new *Universal Drive Unit* (see section 2.4.2). This has led to the design of myriad *VolksBot* variants (*VolksBot RT*), which will be presented in section 2.4.3.

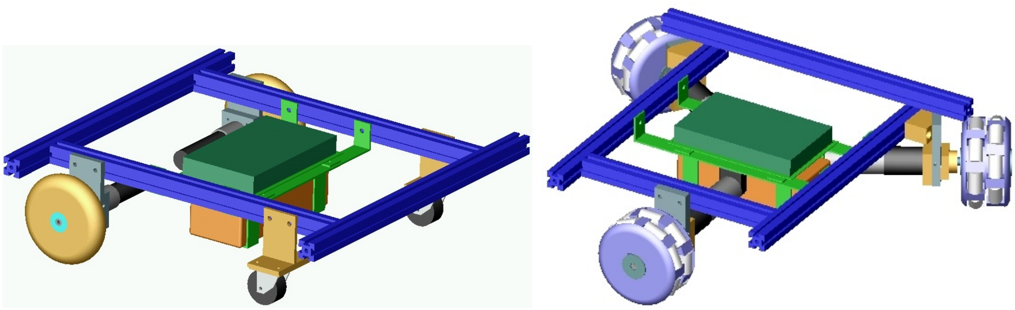


Figure 2.5: Chassis frames for differential and holonomic drive of VolksBot Indoor v.1

Differential and holonomic drive system Fig. 2.5 illustrates the frame construction for the first *VolksBot Indoor* versions equipped with a differential and holonomic drive system. The direct-drive units used for the indoor versions of *VolksBot Indoor v.1*

[1]X-beams by ITEM (http://www.item.info/en)

consist of a scalable DC motor (20-40W), a scalable planetary gear and a tick-encoder attached to the supported wheel shaft via a damped claw coupling. Due to the low weight of a *VolksBot Indoor v.1* robot of only 4.5kg, the robot can accelerate to a speed of up to 3.6m/s depending on the chosen gear ratio.

Instead of implementing a differential drive, holonomic drive units can be attached to the frame. The holonomic drive consists of three drive units which can directly replace the differential drive without any further modification of the robot. The unit itself is built up the same way as the differential drive, except for using 90W DC motors for higher speed and acceleration and *Cat-Trak Transwheels*[1], allowing a movement in x, y and ϕ direction. A triangular aluminum adapter block is used to attach the two front drive-units to the chassis frame via screw connection, providing an angle of 120° between the wheel axes. The third wheel is directly attached to the front chassis frame.

At this early stage of development of the construction kit, these two initial *VolksBot* assemblies demonstrate well the implementation of the design goals(G3, G4, G5). The holonomic drive system was used in the RoboCup MidSize League in 2004 and 2005. The first two *VolksBot Indoor v.1* variants were mainly used as educational platforms and acted as a starting point for further enhancement of the construction kit towards application-oriented prototyping. This enhancement includes the development of a new drive system, the UDU, which will be explained in detail in section 2.4.2.

2.3.3 Electric and electronic hardware components

Also, for electric and electronic hardware, e.g. sensors, motors or motor controllers, I specified and implemented a component library with is continuously being enhanced and maintained. The library holds commercially available products as well as in-house developments of Fraunhofer IAIS. Commercial components that are already integrated include (D)GPS, 2D laser range finders, inertia sensors, industrial and embedded PCs, compasses, stereo cameras and manipulators. A selection of integrated commercial components and their hardware specification is given in table 2.2. After the selection of a new hardware component, it is tested, and software interfaces are defined and implemented. Then, demo applications and documentation for each component are implemented. This allows efficient reuse and integration of the components when a new platform is built and facilitates access and exchange by diverse developers. A standard notebook or PC is used as the robot's main control unit. This solution offers scalability in performance and cost. Furthermore, a wider range of method and

[1] Cat-Trak Transwheels: http://www.kornylak.com

Component	Company	Hardware Specification	Interface
Compass	Honeywell HMR3300	3DOF,Resolution 0.1° Update rate 8Hz	RS232 5-15VDC
DGPS	Afusoft Raven 6	Pos. accuracy 0.5-3m Correction data SAPOS-EPS Update rate 1Hz	RS232 8-35VDC
GPS	Holux GR213	Pos. accuracy 5-25m Chipset SIRF3,Update rate 1Hz	PS2,USB
IMU	Xsens MTi	6DOF, Resolution 0.05° Update rate 120Hz,Accuracy <1°	RS232,USB 4.5-30VDC
2D-LIDAR	SICK LMS200	Field of view 180° Ang. Res. 1-0.25° Dist. Res. 1cm, range <80m Update rate 13-53ms	RS232/422 24VDC
2D-LIDAR	Hokuyo URG-04LX	Field of view 240° Ang. resolution 0.36° Update rate 100ms	USB 5VDC
Pan-Tilt Unit	Directed Perception PTU-D46-17	2DOF,Ang. Resolution 0.01° Speed 300°/sec,Payload <6lbs	RS232/422 9-30VDC
Stereo Camera	Videre Design STH-DCSG	CMOS Sensor 640x480,1/3" Frame rate 30Hz	IEEE1394
Motors	Maxon RE35 and RE40	90W(RE35),150W(RE40),24VDC Gear ratio 1:15-1:150 HEDL Encoder 500 ticks/turn	HEDL 0-24VDC
Motor Controller	Maxon EPOS 24/5	Single channel EC/DC control I_{max} 10A,I_{cont} 5A	CAN,RS232 11-24VDC
Manipulator	Neuronics Katana 450 6M	6DOF, Payload <400g Manipulation range 517mm Precision 0.1mm,Weight 4.8kg	CAN,USB Ethernet 24VDC
Industrial PC	Sontheim IPC-3	1.8GHz Pentium M,VGA PC104+,shock res. HDD Compact Flash optional RS422 and WLAN	USB,LAN IEEE1394 RS232,CAN 12-24VDC
Embedded PC	VIA EPIA-N Nano ITX	1GHz VIA C3,Mini IDE,VGA Power consumption <20W	USB,RS232 LAN, PS/2 12-24VDC

Table 2.2: Listing of integrated commercial hardware components

algorithms compared to micro-controllers can be used. By not having to transfer them to an individual processor type, pre-existing methods may be more quickly integrated. Furthermore, with PCs, stable drivers are available for a large number of sensors and actuators, and system compatibility is ensured for a longer amount of time.

The following overview details the in-house hardware components which I specified and integrated into the *VolksBot* construction kit. The hardware development of these components was conducted in the *VolksBot* project at Fraunhofer IAIS in the period from 2003 to 2008 by our project team under my direction. The hardware design and development of the *IAISVision* system was done solely by myself.

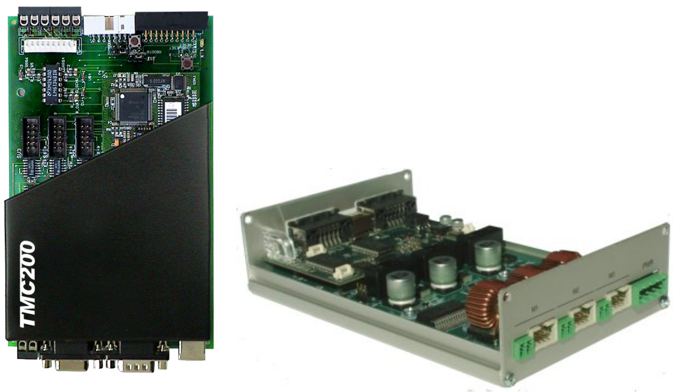


Figure 2.6: TMC200 and VMC motor controllers

TMC200 and VMC Motor controllers The motor controller *TMC200* is connected via serial interface to the control PC. The controller offers odometric data analysis, thermal motor protection, battery voltage monitoring, velocity and current PID control for three DC-motors up to 150W power. In 2006 a revised version called *VMC*(VolksBot Motor Controller) was developed with improved properties in hardware I/O and thermal dissipation. The new *VMC* operates at 12-24VDC input and hosts two AD-inputs and three digital I/Os. PID control can be set individually for up to three DC motors with 6A continuous load per channel. A Windows/Linux API is provided for communication via CAN and RS232. Firmware and API are available as open source. Besides the use as standard motor controller in the *VolksBot*, the component integrates well in other robot platforms. For example, three of the four most successful teams in the RoboCup Middlesize World Championships in 2006 used it in their robots. About 300 units were produced and used for *VolksBot* and by various research groups, robotic projects and companies worldwide. The motor controllers are depicted in Fig. 2.6.

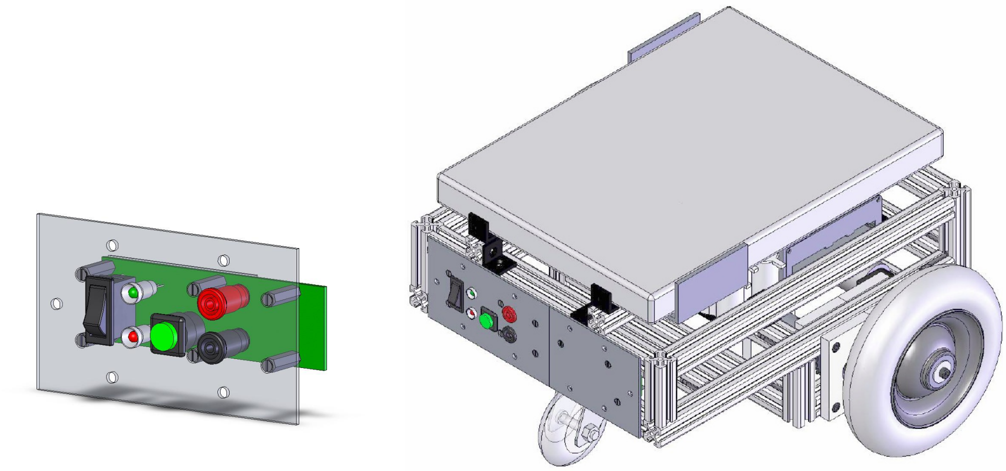


Figure 2.7: Power panel for power management on VolksBot Indoor v.2

Power panel The *VolksBot Power panel* (see Fig. 2.7) combines various features for on-board power management and safe operation of the robots. The *Power panel* includes an *Emergency stop* function, protection against deep battery discharge, usage of automotive circuit breakers against over current and reverse voltage protection. The panel hosts a main power switch, plugs for external battery chargers and LEDs for indication of battery and robot state. Furthermore, it allows for standardized power wiring for all *VolksBot* variants via defined and fused power circuits. Optionally, the panel can be equipped with DC/DC converters to provide a fused and stabilized 5VDC, 12VDC and 24VDC power supply for additional components. It was designed to fit between the double-layered chassis frame of the *VolksBot RT* and *VolksBot Indoor v.2* robot variants (see section 2.4.3).

MBoard The *MBoard* is a universally applicable I/O board, which can be used to control up to 32 servo motors synchronously. Originally developed as controller for the modular manipulator construction kit *Rapero* [31], I integrated it as *VolksBot* component to access analog sensor data and to control servo motors. Thirty-two analog input channels can be sampled and read out simultaneously at an update rate of 50Hz and 10bit resolution. Together with the *Rapero* servo modules, it can be used to directly implement pan, tilt or pan-tilt units equipped with infra-red sensors, ultra-sonic sensors or cameras, as shown in figure 2.8.

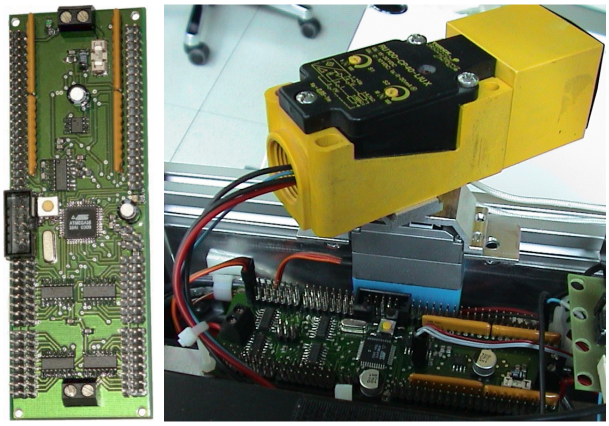


Figure 2.8: MBoard and pan sensor unit mounted on VolksBot

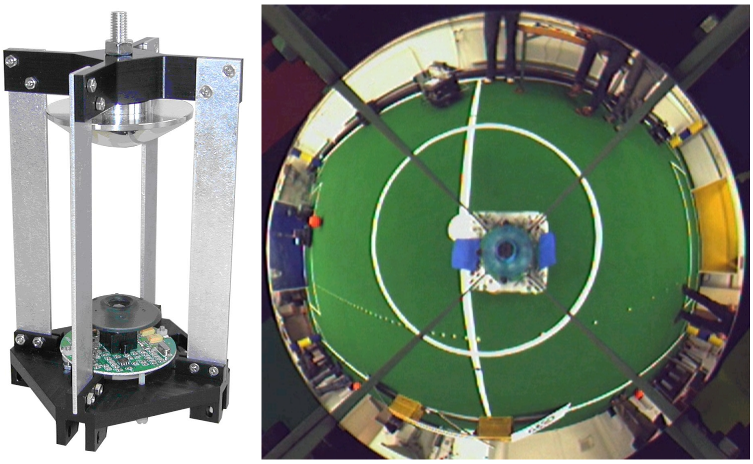


Figure 2.9: IAISVision system and camera image

IAISVision In addition to the integration of regular cameras, as an original requirement for the soccer robots used in RoboCup Middle Size League, I developed the catadioptric vision system *IAISVision*. The vision system includes an IEEE1394 CCD camera and a hyperbolic mirror as shown in figure 2.9. Further details on the hardware development of the system and its use to build up an adaptive color vision system are given in section 3.1.

3DLS-K a continuously rotating 3D-LIDAR In 2007, I conducted the development of the Fraunhofer 3D-LIDAR system (3DLS-K). Two industrial laser range finders rotate around the vertical axis of the system, acquiring depth and remission information for a 360° field of view (see Fig. 2.10). The system consists of two SICK LMS 291-S05 laser range finders mounted on an angular adjuster plate. The scanners have an apex

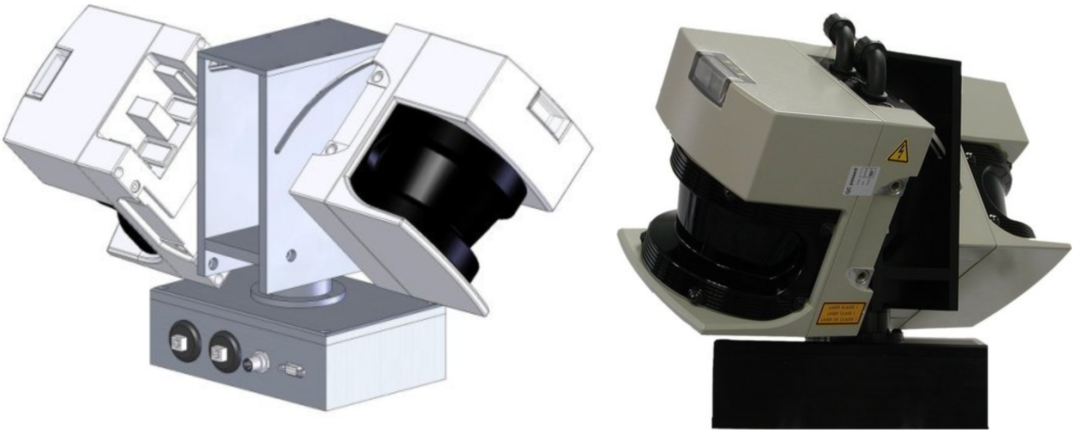


Figure 2.10: CAD assembly and prototype of the Fraunhofer 3DLS-K LIDAR

angle of 180° and a resolution 1 to 0.25°. Depending on the resolution, the response time to acquire one two-dimensional scan is from 13 up to 53ms. The maximum scan range is 80m. Originally I integrated the LIDAR system as a sensor component for *VolksBot RT* robots. Later, I adapted it as a sensor system for the autonomous car *Spirit of Berlin* [32] which participated in the DARPA Urban Challenge 2007[1]. Further details on the hardware development of the system as well as its application to service robotics and autonomous driving are given in section 3.2.

2.3.4 Component based software design

As valid for mechanics and hardware components, also in software, a framework concept with well-defined components was established(C1). First, an interface-based architecture on an implementation-independent level is formally specified. From this interface specification, module templates are generated that can be filled with an individual implementation. The software system of the robot is composed of software components. Each component is specified by its interface and its execution model. The interface of the component consists of pins which may be connected to pins of other components by data edges. Typically a data edge carries control signals or a data flow of a specific data type. The architecture of the robot's software system is described by these interfaces and their connections. The architecture does not prescribe anything about the realization of the component, i.e. its later implementation as a module. The Interface Description Language, IDL, is used to specify these interfaces formally.

[1] DARPA Grand Challenge (http://www.darpa.mil/grandchallenge)

IDL is standardized by the Object Management Group OMG[1] and is widely used as description language for middle-ware in the CORBA context. IDL neither enforces a particular programming language for the components nor a specific operating system like Windows or Linux. It concentrates on the formal description of the external view of a component and on the connectivity between components.

2.3.4.1 Visual programming with ICONNECT

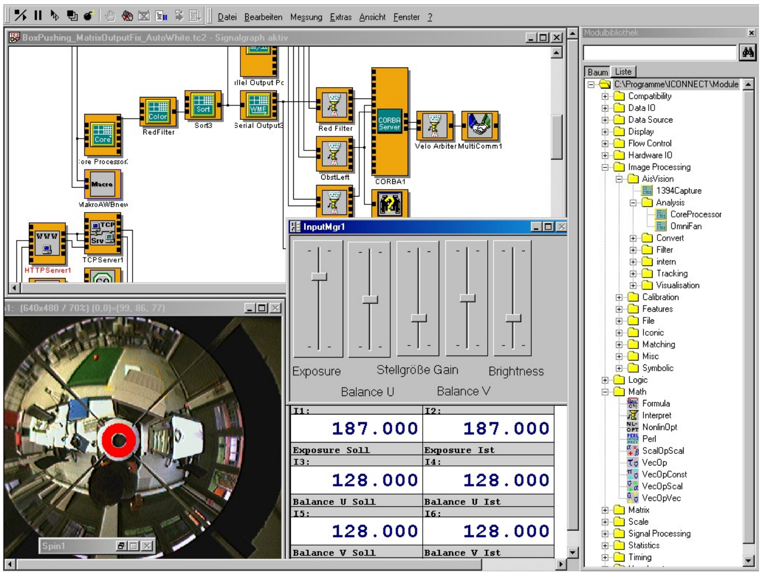


Figure 2.11: *ICONNECT* programming environment with graph, GUI and module library

ICONNECT [33] [34] is used as a framework for the visual composition of signal graphs. Signal graphs are composed of interconnected modules. A module in *ICONNECT* consists of a compiled DLL and has a visual black-box representation with input and output pins in the graph editor. For each module, relevant parameters can be entered in a parameter dialog or can be changed during run-time via optional input pins. In Fig. 2.11 the *ICONNECT* programming environment is depicted, including an example of an easy to build graphical user interface. A main advantage of *ICONNECT* compared to similar approaches [35] [36] is a unique feature that allows one to execute signal graphs on a PC in real-time without recompiling the whole graph. This makes *ICONNECT* very suitable for an iterative development process and for rapid prototyping of robotic applications.

[1] OMG (http://www.omg.org)

The existing module library of *ICONNECT* already contains a lot of functionality relevant for mobile robot control including signal processing of sensor data, image processing, control, hardware IO, logic, neural networks, network communication, data visualization and GUI design. Very important for system integration of a robot system is the fact that all interfaces of the PC are accessible in *ICONNECT*. This important feature abstracts from the interfaces of the control PC and eases integration of hardware components connected to the PC. In addition, direct access to the memory of the PC is possible.

The VolksBot software concept extends the framework as it adds robot-specific software modules to *ICONNECT*. The technical mechanism to insert these user-defined modules to *ICONNECT* exploits module templates which are automatically generated from the IDL interface specification of the new component. These module templates do not yet contain an implementation of the module's intended functionality, but they provide a complete set of the specified interfaces. Fig. 2.12 shows an example of the generated module DDBehavior1 that is instantiated in an example module graph. For each

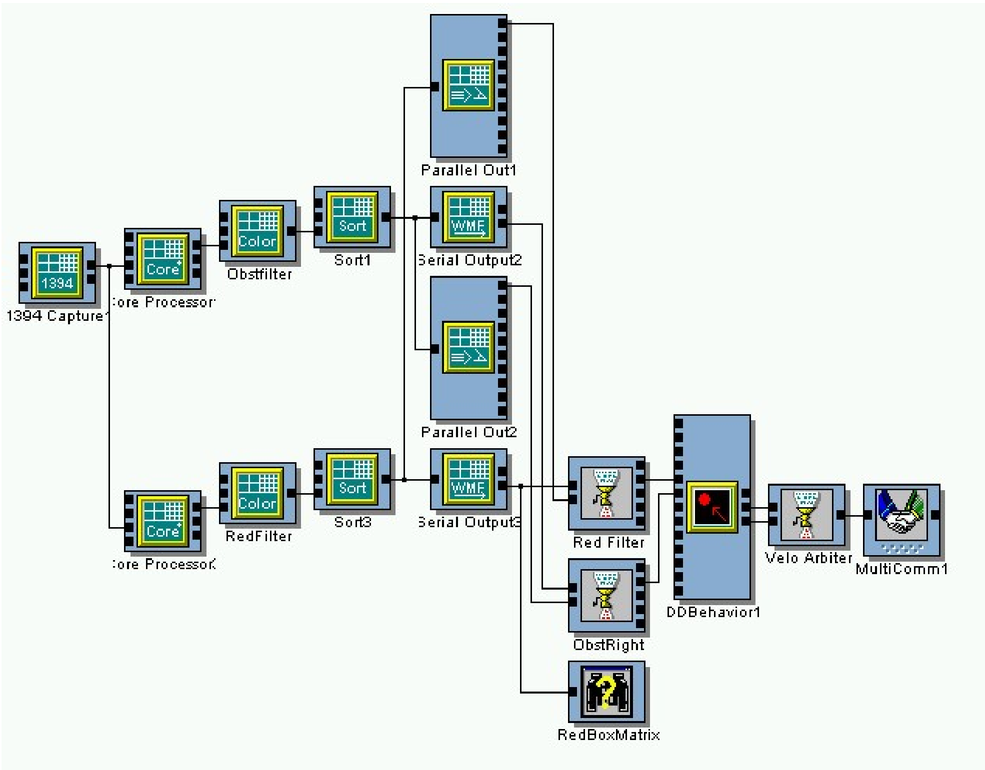


Figure 2.12: Example *ICONNECT* signal graph including data acquisition, signal processing , behavior control, and actuator output (from left to right)

VolksBot hardware component, an interface module is implemented for communication with the component. With this, the low-level protocols to access sensors or actuators via serial port, CAN bus, or a FireWire link are usually hidden to the user. The user simply connects pins of instantiated modules via an easy to use GUI providing drag-and-drop functionality. The activation of a module in a module graph is determined by its execution model. By choosing different execution models, e.g. event-driven or synchronous data flow, different control architectures may be implemented. This includes feedback loops and concurrent execution of parallel module graphs. The main advantage of using *ICONNECT* is its support for direct real time execution of module graphs on a PC and its specific support to construct control cockpits, data monitoring tools and image displays with the GUI construction kit that come with *ICONNECT*. An example of a control and data monitoring cockpit is shown in Fig. 2.11.

The module library of *ICONNECT* allows one to add personal modules. According to the framework concept, these user modules are specified on an abstract level by their IDL interfaces first. The second step is to generate an empty template module which already can be used and connected in a module graph. The third step it to program the functionality of the module.

The system offers many options to implement the functionality of a module. The first option is to script the code in Perl or Visual Basic. The read access to input data and the write access to output data is done using predefined functions in the script code. This allows the user to focus on the implementation of the functionality without taking care of interface handling. Another option is to program the module in C or C++, compile it to a DLL, add it to the library and use it by dynamically executing the DLL code. This provides a better performance than scripting. The third option allows one to insert functionality to a module graph via CORBA middleware [37]. Here, the implementation of the module consists of a CORBA server, which can be accessed from any external system, e.g. via WLAN. Thus, the functionality can be implemented using arbitrary programming languages. These simple extension mechanisms allow one to add existing libraries as well as his or her own development to the *VolksBot* software library(C5, C6). Table 2.3 summarizes the implemented *VolksBot* module extensions in *ICONNECT*. The code generator of the behavior design tool *DualDynamics-Designer* [38] [39] was modified. Now it is able to generate either a Java client which can control the robot via the CORBA mechanism, or it can generate C code for the implementation of the *DD-Behavior* module in *ICONNECT*. A special module template was created that allows one to integrate functionality of the *OpenCV* image processing library [40] via copying and pasting the *OpenCV* source

Module Name	Function
DD-Behavior	Interface to robot behaviors implemented in DD-Designer
OpenCV	Interface to OpenCV image processing library
CMU1394	Generic interface to IEEE1394 video cameras via CMU driver
CORBA	Execution of a CORBA server
ODE-SIM	ODE-based simulator with 3D visualization
MATLAB	Interface to MATLAB
IAISVision	Interface, calibration and signal processing for IAISVision
Joystick	Generic interface to USB Joysticks
TMC/VMC	Interface to motor controller
Odometry	Generic odometry calculation for all VolksBots
Kalman	Extended Kalman Filter for sensor data fusion

Table 2.3: Overview on ICONNECT software modules developed for VolksBot

code into the module's parameter box. This feature allows one to efficiently integrate methods like face recognition or stereo vision to the software framework. A module for generic FireWire camera access based on the *CMU IEEE1394 driver*[1] was developed. Another module encapsulates an ODE-based simulator [8] of the robot including a 3D visualization. The simulator module can be connected to the sensor inputs and actuator outputs of the *DD-Behavior* module. In this, it closes the senso-motoric feedback loop, which is usually closed by the real-world environment the robot is acting in, and therefore offers an efficient approach for iterative development or debugging of the behavior system. Furthermore, a module that interfaces with *MATLAB* [41] was written. This module sends *MATLAB* function code to *MATLAB* during initialization. During run-time the function is executed in *MATLAB* receiving data coming from the input pins of the *ICONNECT* module and sending results to the output pins of the module in *ICONNECT*. To integrate the *IAISVision* system, interface, calibration and signal processing modules for the *IAISVision* system including color segmentation, edge and color blob detection were implemented. These *IAISVision* modules were mainly used in the context of participation in the RoboCup Middle Size league. Further module development include an interface to USB Joysticks, an interface to the *TMC/VMC* motor controllers, an odometry module based on wheel encoder ticks and the implementation of an extended Kalman Filter. This *Kalman* module allows for sensor data fusion of robot position information from various sources including compass, GPS, IMU, LIDAR and odometry.

For each implemented software module, an HTML documentation is written, and a compact - preferably self-explaining - example graph in *ICONNECT* is built(C5,C7).

[1]CMU IEEE1394 driver: http://www.cs.cmu.edu/ iwan/1394

In conjunction with the goal of only building modules of fine granular functionality(C3), this fosters reuse and reduces initiation time when new team members enter the project(G10,G7). With this approach, loss of knowledge is reduced and even people with background other than computer science are able to program the robot on this abstraction level(C8). This approach is especially beneficial for projects with many people of different backgrounds working together. Compatibility between modules is ensured by a clear interface definition, and project members can have a common and quite intuitive understanding of the entire robot control software. Combined with the advantages mentioned above, the use of such a framework sets some restrictions to the developer as it limits the choice of possible methodologies when developing software. Another possibly negative aspect is the use of a particular operating system, such as Windows, which is required for the *ICONNECT* framework. One way to overcome these limitations is to implement a more general software library which does not provide certain advantages of a differentiated framework like *ICONNECT* but which may be appealing to a wider group of users.

2.3.4.2 FAIRLib - The Fraunhofer Autonomous Intelligent Robot Library

To form a broad basis and foster synergies in robotic-related software development at Fraunhofer IAIS, together with my colleague Stefan May, I designed the specification and conducted the development of the *FAIRlib*, the Fraunhofer Autonomous Intelligent Robot Library. Prior to this initiative, a wide range of software architectures, frameworks, programming languages, operating systems, coding and documentation standards were used in robotic-related in-house software development. With the development of the *FAIRlib*, in-house software development was standardized and made accessible for all robotic-related projects including *VolksBot* throughout the *Autonomous Robot department* of the institute and its cooperating partners(C5,C6,C9).

The *FAIRlib* is a platform independent C++ software library containing various algorithms and methods in the context of robot control, navigation, localization and mapping (SLaM), interfacing and processing of sensor data. The *FAIRLib* (see Fig. 2.13) consists of two layers(C8): an operating system-dependent layer on the bottom and an operating system-independent layer on top. The lower layer hosts basic hardware I/O functions, data structures and a math library. Here, dependencies on specific hardware and operating systems are tolerated as they are hard to avoid. The top layer contains high level device drivers, a graphics and an algorithm library. Here, the described dependencies can and should be avoided, which allows one to use the same set of methods

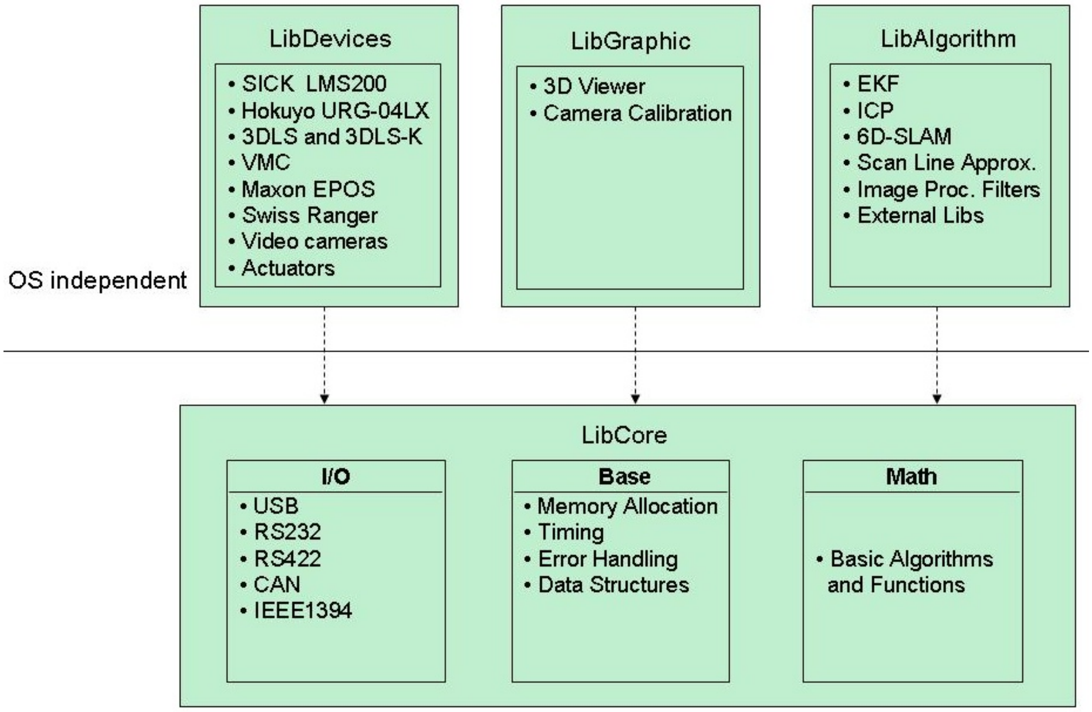


Figure 2.13: The FAIRLib architecture

for different hardware and operating systems. The *FAIRlib* currently consists of four libraries. On the lowest, operating system-dependent level, *LibCore* consists of three sub-libraries. The *I/O* library provides low-level access to hardware interfaces such as USB, RS232, RS422, IEEE1394 and CAN. The *Base* library provides direct access to and allows allocation of memory, timing functions, error handling and specification of data structures (e.g. 3D-LIDAR scans). Furthermore, a math library is implemented which is used by the upper layer of *FAIRlib*.

The top layer of the *FAIRlib* currently hosts three libraries:

- LibDevices hosts device drivers for various sensors such as the 2D-LIDAR systems *SICK LMS200* or *Hokuyo URG-04LX*, the Fraunhofer 3D-LIDAR systems *3DLS* and *3DLS-K* (see section 3.2), the *VolksBot* motor controllers *VMC* (see section 2.3) and *Maxon EPOS*, the 3D-camera *Swiss Ranger*[1], IEEE1394 and USB cameras or actuators like servo motors, grippers or pan-tilt units.

- LibGraphic provides access to common graphical user interfaces, for example a 3D-Viewer for point clouds from *3DLS-K* or a camera calibration utility for catadioptric and regular cameras.

[1]Swiss Ranger 3D camera: http://www.mesa-imaging.ch

- LibAlgorithm covers various algorithms for sensor data analysis and fusion, navigation, localization and mapping, such as an extended Kalman Filter for sensor data fusion, the ICP(Iterative Closest Point) algorithm [42] for scan matching, 6D-SLAM [43] and the *Scan Line Approximation* algorithm [44] for edge detection by use of range information from 3D scans and range images. Moreover, it contains a collection of frequently used image processing filters and integrates external libraries, e.g. *OpenCV*.

Doxygen[1] is used for code documentation which allows for generating documentation on a user and developer level. To ensure compatibility between the *FAIRLib* and the *ICONNECT* framework, now, before programming an *ICONNECT* module in C++, the functionality is first implemented into the *FAIRLib* which ensures even more universal use and reuse by a larger group of users and programmers.

2.4 Enhancements in physical performance and flexibility

The first *VolksBot Indoor* platforms presented in section 2.3 use a single layered chassis frame which allows a light-weight construction, but limits the kinds of possible applications, especially with regard to limited payload, size, clearance and mobility of the platforms. To overcome these limitations and to allow for prototyping of a wider range of service robotic applications, in a next step, I enhanced the mechanical hardware of the construction kit to allow for a broader field of applications. This enhancement accounts for robot operation in multiple environments such as rough terrain or outdoor, higher robot mobility, the possibility to construct larger platform and higher robot payloads.

2.4.1 Rough terrain robots and locomotion

A survey on rough terrain robot locomotion [45] has shown that wheeled robots are easiest to construct and maintain. They can obtain greater speeds, carry higher payloads and are usually more reliable compared to tracks or legs. Some of the most successful rough terrain wheeled platforms, such as Pioneer from ActivMedia[2], offer high payload and considerable ground clearance while the GOAT approach [46] from Carnegie Mellon University and Georgia Tech has an active adaption system. Powerbot[3] offers a payload of up to 100kg. Most of these rough terrain platforms, however, are fixed

[1]Doxygen: http://www.doxygen.org
[2]Pioneer robot: http://www.activrobots.com
[3]PowerBot: http://www.activrobots.com

in their structure and do not follow a construction kit approach. There has been a significant contribution to rough terrain robot design and analysis by researchers and developers worldwide. One of the most popular international competitions which demand rough terrain robots is Robocup Rescue [47]. Rescue scenarios as well as volcanic exploration [48] present extreme environmental challenges for robots with respect to robot mobility but also to cognition and autonomy. In the works of Kook, Jun and Krovi [49], passive articulated leg-wheeled subsystems have been examined and evaluated for rough terrain locomotion capabilities. The emphasis here was to create a design which adjusted to increasing terrain roughness while maintaining simple actuation requirements. The candidates evaluated were the *Single Degree-of-freedom Coupled Serial Chain (SDCSC)* [50] mechanism and the *Four-Bar link* mechanism. Jarvis [51] has presented a design of a large articulated six-wheeled robot for rough terrain navigation with dimensions of 2.5m by 1.7m. The passive articulation is implemented by a joint located at the body center. WorkPartner [52] is an articulated wheeled platform with active body and leg joints. Each leg has three active joints, and the wheels are individually powered. With a mass of 200kg, this robot can carry a payload of about 40kg. The maximum speed obtained by this robot on relatively flat ground is 7km/h. They introduce a locomotion method called *rolking* which is a hybrid of rolling and walking by selectively controlling either the articulated legs or wheels or by simultaneously controlling both. The *rolking* locomotion mode combines the advantages of both legs and wheels in difficult terrain conditions.

2.4.2 Development of the Universal Drive Unit

The physical enhancement of the initial *VolksBot* construction kit should account for the following attributes, which I defined to amend the general design criteria from section 2.2. The enhanced design criteria are labeled in brackets[E1-E10] for further reference:

- E1 Increased rigidity
- E2 Increased torque/payload
- E3 Increased mobility
- E4 Increased ground clearance
- E5 Increased grip on the terrain
- E6 Allowing a housing of the robot

E7 Damping of mechanical vibrations

E8 Scalable torque/speed ratio

E9 Compatibility with the original kit

E10 Minimal number of actuators and components

Furthermore, the physical extension of the construction kit should maintain the advantages of the existing construction kit specified by the design goals and criteria in section 2.2, in particular, allowing for a wide range of variants(G9) and maximal reuse of new and existing components(G3). Following these requirements, I enhanced the

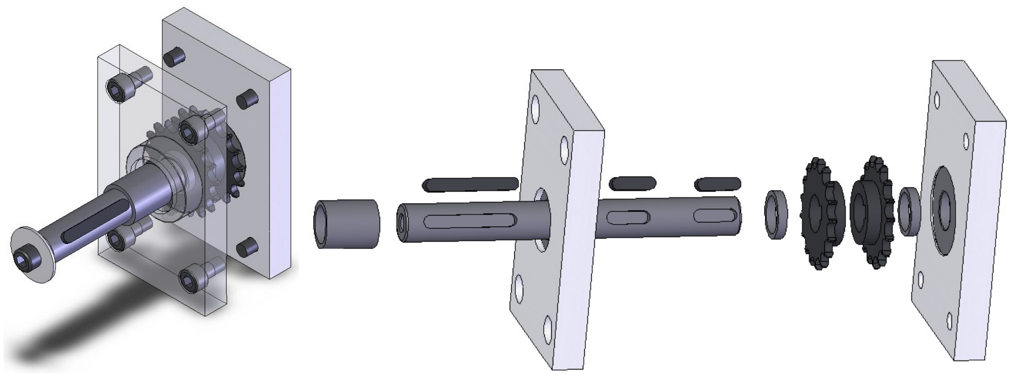


Figure 2.14: Regular and exploded assembly drawing of the UDU

concept for the robot's base frame construction and the drive system. The wheels on each side of the new base platform are driven by a single 90W or 150W DC-motor which provides a continuous torque of up to 15Nm(E2). A double-layered chassis frame is introduced which provides higher rigidity(E1). It also offers the possibility of a complete housing of the robot base(E6) as required for outdoor use. The force transmission to each wheel is achieved by use of the *Universal Drive Unit* (UDU). The UDU (Fig. 2.14) consists of two aluminum bearing blocks, a steel shaft and two chain sprockets(E10). It can be mounted by screw connection at any desired position on the sides of the chassis frame, which makes it possible to easily customize the wheel distances and wheel diameter. The steel shaft is supported by two bearing blocks which provide high rigidity(E1) to support payloads up to 80kg(E2). The shaft can be directly driven by a motor or driven indirectly via chain connection by another UDU. The chain sprocket is mounted in between the two bearing blocks allowing for an encapsulation of the entire drive unit.

In order to drive the shaft, it is connected to a motor block (Fig. 2.15 left) with a standard claw coupling. Once assembled, this can be used to drive other units via

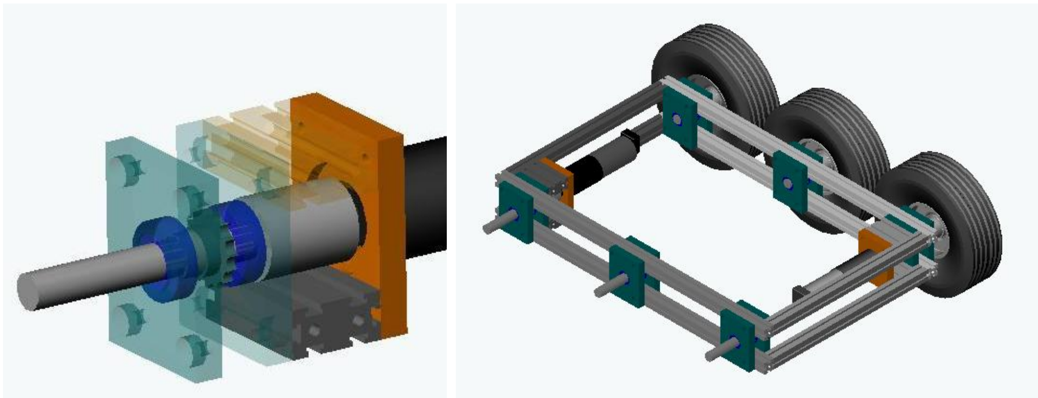


Figure 2.15: Drive unit with attached motor block (left) and full drive assembly (right)

chains or belts or drive the wheels directly. This way, a repetition of UDU assemblies allows for the actuation of all wheels on one side of the robot through a single motor(E10). An assembly (Fig. 2.15 right) consisting of three drive units and a single motor block illustrates this component reuse to construct a complete robot base. In order to change the motor type, only the motor block needs to be modified. With the connection between the wheel to the shaft made by standard hub connectors, air-filled tires(E7) of different diameter ranging from 18cm to 40cm and different profiles(E5) all having the same hub adapter can be connected to the shaft of UDU via a fitted key connection. With motor gear ratios ranging from 1:14 to 1:150 torques up to 15Nm can be transmitted, and maximum velocities range from 0.7m/s to 3.6m/s, depending on the motor-gear-wheel combination. Depending on the wheel diameter, a ground clearance of the robot between 6.5cm and 16cm can be obtained(E4). In conjunction with the increased grip(E5) and the damping of vibrations(E7) of the tires, this provides a significant increase in the robot's mobility(E3).

Only three UDU components need special machining(E10). Special parts include motor block, bearing block and shaft. Other components are industry standards. Table 2.4 gives an overview on the UDU components along with their properties. The following details how the physical extension of the existing *VolksBot* construction kit allows for the development of new *VolksBot* variants(G9), including the *VolksBot RT3, RT4, RT6* and *VolksBot Indoor v.2.*

2.4.3 Robot variants based on the extended VolksBot kit

The combination of the new UDU module and the double-layered chassis frame allows for the design of various *VolksBot RT* (Rough Terrain) variants in a short amount of

Component	**Dimensions**	**Properties**
Shaft	Diameter 12mm	Steel: CF53
Bearing	Inner Diameter 12mm Outer Diameter 28mm Strength 8mm	Mass: 20g Max. dynamic load: 5100N Max static load: 2360N
Bearing Block	Length 50m Width 10mm Height 80mm	Aluminum Alloy (AlCuMgPb)
Motor coupling	Inner Diameter 12mm Outer Diameter 32mm Length 57.15mm	Max. angular separation: 7° Max. torque: 25Nm
Chain & Chain sprocket	Number of Teeth: 17 Pitch 6mm	Max. Force: 3000N Max torque: 51Nm
Keyway & Key	Height 5mm Width 5mm Length 14&25mm	Steel: C45K
Shaft spacer	Inner Diameter 12mm Outer Diameter 15mm Length 3mm & 17mm	Steel: CF53 Mounted between sprockets and bearings

Table 2.4: Components of the Universal Drive Unit

time. This is due to the fact that mainly existing *VolksBot* modules and components together with industrial standard components are used to construct the robots(G3). This way, compatibility with the initial kit is maintained(E9).

Conversion and compatibility To demonstrate the compatibility of the new development with the existing kit(E9), the conversion of a *VolksBot Indoor v.1* robot to a *VolksBot RT* rover is illustrated in Fig. 2.16. It involves the removal of the drive unit from the chassis frame of the indoor robot and mounting it onto the chassis frame of an *RT* platform. Hardware components like the control PC, the motor controller or the camera system can directly be reused, and they are connected to the *Power Panel*, batteries and motors. *VolksBot* modules and components can also be used to upgrade pre-existing robot platforms. Figure 2.17 shows how an outdated Fraunhofer outdoor robot called *Pegasus* was equipped with *VolksBot* components such as the chassis frame, the *VMC* motor controller, the *MBoard* and a sonar sensor unit.

Reuse of modules and components from the extended construction kit also allows the efficient design of a set of new standard *VolksBot* variants which were used in in-house research projects (e.g the Outdoor project[1] and the ProfiBot project[2]) and were made available for distribution[3]. The new *VolksBot RT* robots are used by various research

[1]Outdoor project: http://www.iais.fraunhofer.de/602.html
[2]ProfiBot project: http://www.iais.fraunhofer.de/profibot.html
[3]VolksBot Website: http://www.volksbot.de

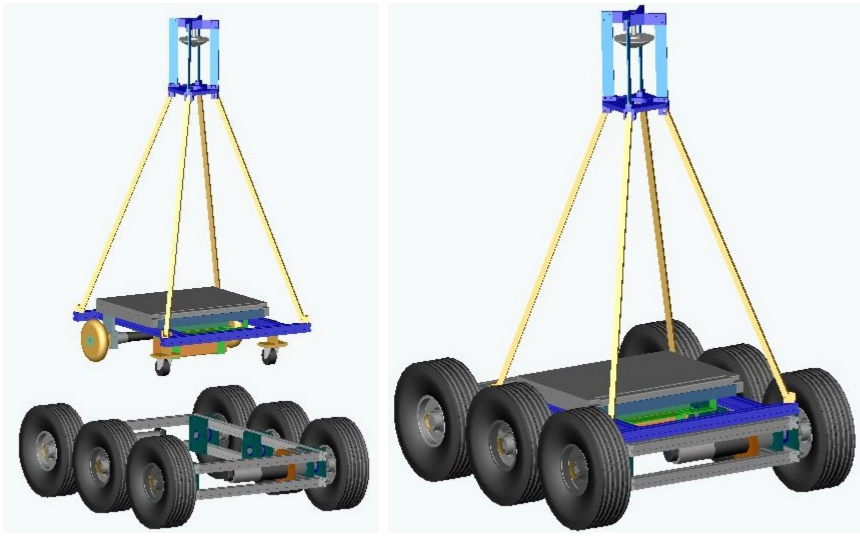


Figure 2.16: Conversion from VolksBot Indoor to RT

groups and companies worldwide. An assembly manual for each of these new variants was created. An excerpt of a *VolksBot RT4* assembly manual is given in the Appendix B. The variants are presented in the following.

VolksBot RT6 Equipped with two 150W DC-motors, the six-wheeled *VolksBot* rough terrain variant *RT6* (Fig. 2.18) is able to climb a slope of 43° at a maximum speed of 1.1m/s and a payload of 40kg. Because the motor gear is exchangeable, the torque/speed ratio can be adjusted according to the application; this ranges from 2.2Nm and a maximum velocity of 5.5m/s up to 15Nm at a maximum velocity of 0.56m/s, when using the 21cm diameter wheel for the standard *RT6*.

VolksBot RT4 The underlying design of *RT4* and *RT6* are nearly identical. The two variants only differ in the length of the chassis frame, in the amount of *UDUs* and in number and diameter of the wheels. Applying different hardware configurations, further RT variants can be efficiently built: one can vary the dimension of the chassis frame (between 30cmx30cm and 80cmx150cm), use four different wheel types with diameters between 18cm and 40cm, two different motors (90W and 150W) and four different motor gears with motor gear ratios between 1:15 and 1:150. The standard *RT4* (see Fig. 2.19) uses wheels of 26cm diameter, which allows for maximum velocities ranging between 0.7m/s and 6.8m/s at maximum continuous torques between 15Nm and 2.2Nm depending on the motor gear ratio.

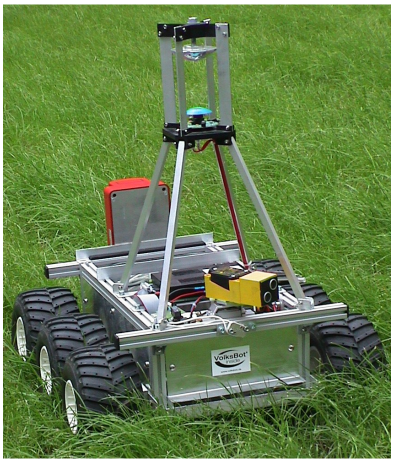


Figure 2.17: Pegasus - an outdoor rover equipped with VolksBot components

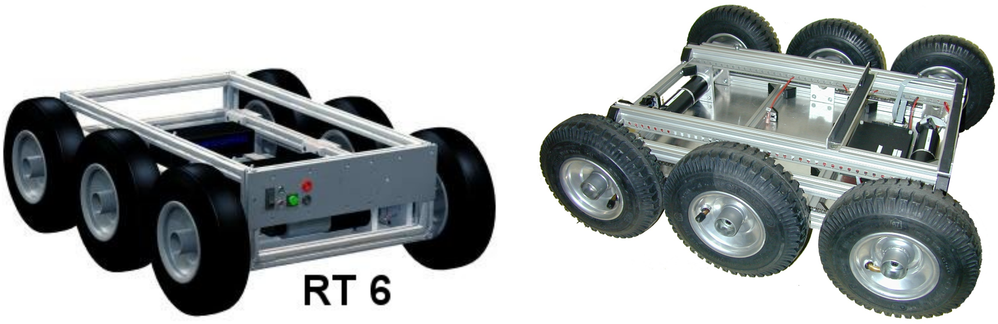


Figure 2.18: CAD model and image of VolksBot RT6

VolksBot RT3 Aside from arranging the UDUs horizontally along the double-layered chassis frame, it is also possible to arrange them vertically (see Fig. 2.20 left). In addition to maintaining the scalability in length and width, this minor change in the configuration offers two distinct advantages to the entire platform. First, the ground clearance of the platform can be easily adjusted by varying the length of the vertical chassis frame(E4). Second, the entire drive unit, including motors, can be completely covered shielding the controller and sensors from possible electromagnetic disturbances(E6). Fig. 2.20 (right) shows a *VolksBot RT3* variant using the new module. The robot is equipped with tires with a 26cm diameter. A standard passive caster is used as the third wheel.

VolksBot Indoor v.2 After the development of the three *RT* variants, I conducted a redesign of the original *VolksBot Indoor v.1* version to take advantage of the im-

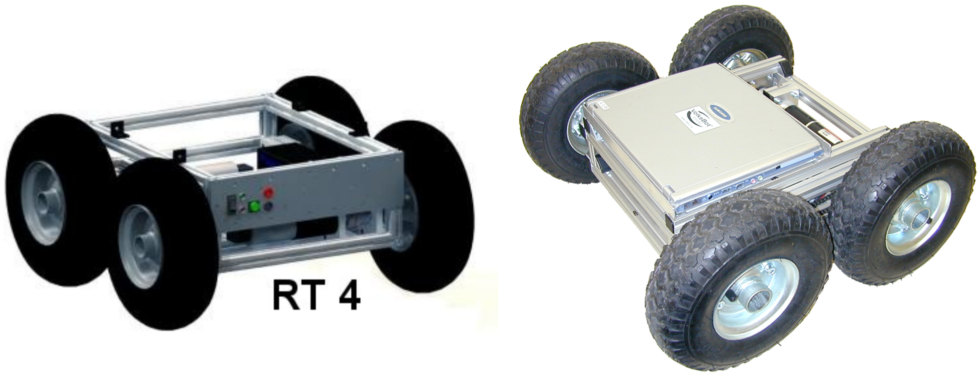


Figure 2.19: CAD model and image of VolksBot RT4

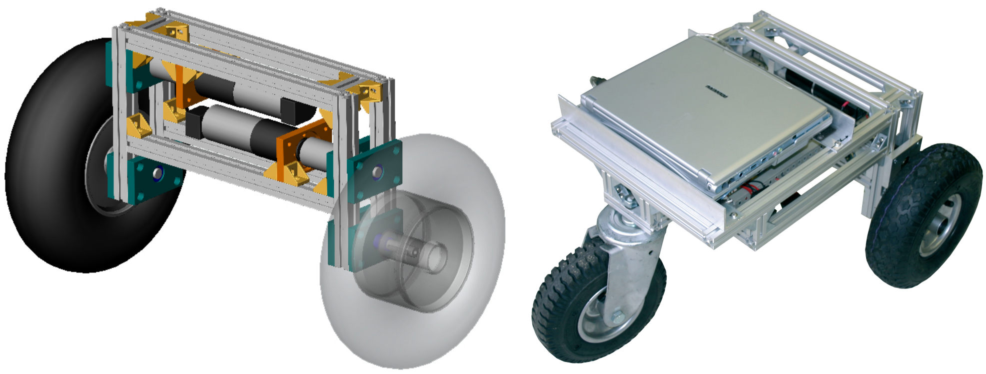


Figure 2.20: Vertical arrangement of UDUs to form a differential drive unit for VolksBot RT3

proved rigidity and payload of the *RT* development, as well as to maintain compatibility between the systems. The redesign included the use of the double-layered frame and the UDU. 90W DC motors and larger wheels with 180mm diameter were integrated (see Fig. 2.21). With its compact size and high payload, this platform allows for various indoor tasks, e.g. the participation in the RoboCup@Home competition (see section 2.6.1.3) or the use in the *ProfiBot* project[1] as an educational platform for mechatronics in vocational schools.

2.5 Evolutionary design optimization for enhanced mobility

A wide range of robot variants can be designed with the extended construction kit. But all platforms previously implemented still have limited mobility over rough terrain

[1]The ProfiBot Project (http://www.iais.fraunhofer.de/profibot.html)

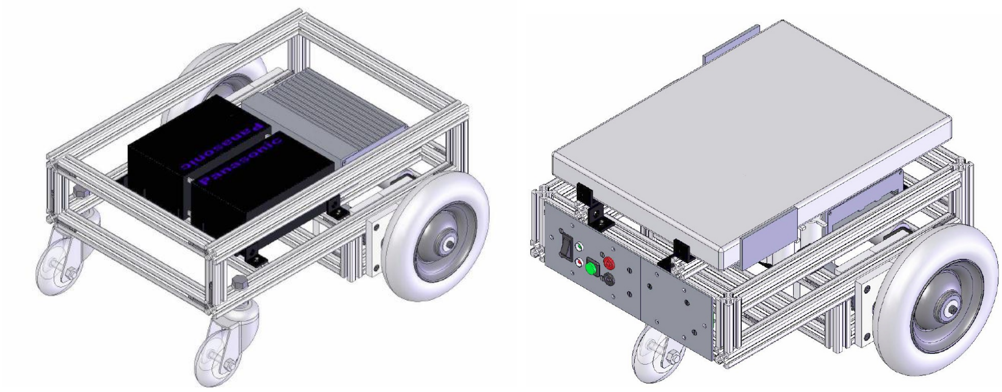


Figure 2.21: VolksBot Indoor v.2

due to structural limitations like fixed wheel configuration or the lack of a suspension system. Overcoming such limitations is vital; especially in certain application scenarios, including urban search and rescue, the demands on robot mobility are very high. Robots have to overcome difficult obstacles like staircases or random step fields. After a brief introduction of a high mobility platform by EPFL, the *Shrimp* rover, I will demonstrate how mobility of the given kit was further enhanced by the design of a new actuation module, the *Parallel Bogey Unit*, by recombining existing construction kit components. Further, performance optimization in mobility is achieved by applying parameter evolution to the physical design of the new robot. This approach resulted in the implementation a high mobility platform called the *VolksBot XT*, details of which will be presented in the next section.

2.5.1 Shrimp, a high mobility robot

There has been a significant contribution to high mobility and rough terrain robot design and analysis by researchers and developers worldwide which have already been discussed in section 2.4.1. A promising approach to obtain high mobility on rough terrain with a wheeled robot platform is demonstrated by the *Shrimp Rover* [53] [54]. It uses a parallel bogey mechanism to passively adapt the wheel positions to uneven terrain (See Fig. 2.22). The *Shrimp* is able to overcome obstacles of twice its wheel diameter and is able to climb regular stairs. The six wheels are independently actuated by motors installed in the wheel hubs. In spite of the excellent rough terrain mobility of *Shrimp*, one of its disadvantages is the low maximum payload of 3kg. Furthermore, the *Shrimp* has a direct drive which requires all six motors to be controlled simultaneously.

Since these motor units are close to the ground, they are susceptible to damage in rough terrain.

Figure 2.22: The Shrimp Rover (EPFL) adapting to rough terrain

2.5.2 The parallel bogey as VolksBot module

The parallel bogey is a four bar link mechanism with parallel links. Figure 2.23 il-

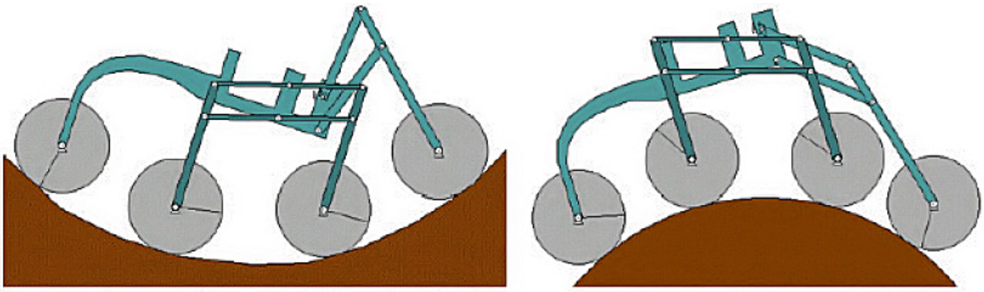


Figure 2.23: Parallel bogey on convex and concave ground

lustrates the function and the passive adaptation capabilities of the parallel bogey mechanism on convex and concave surfaces. Inspired by the *Shrimp* robot presented above, I conducted the design and implementation of a parallel bogey rover [55] by exclusive reuse of the components from the existing *VolksBot* construction kit. The entire robot structure along with the new parallel bogey unit is constructed by use of UDUs and X-Beams. The joints of the new *VolksBot* parallel bogey unit are realized by a modified version of the UDU, which is shown in figure 2.24. The only difference to the original UDU is a common shaft which connects two UDUs together. The UDU variant is equipped with chain sprockets so besides acting as parallel bogey joint, it is also capable of transmitting torque from the motors mounted inside the robot's body to the wheels.

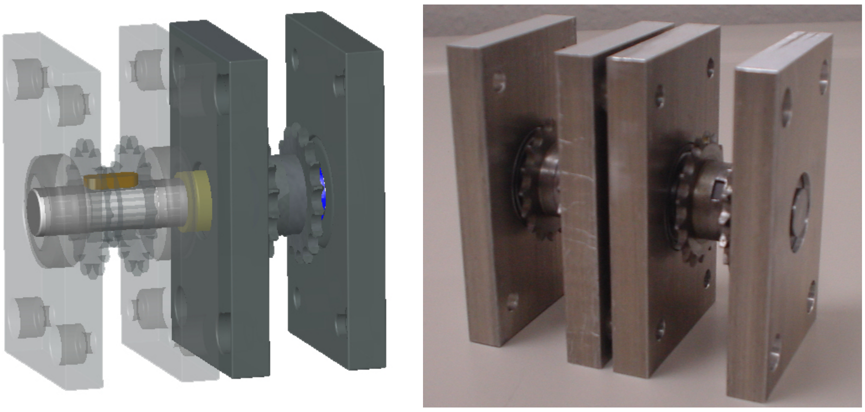


Figure 2.24: Modified UDU used as joint for the parallel bogey

The direct drive unit for the wheels was replaced by a strand of a central hinge unit mounted to an upper and a lower horizontal lever unit connecting two vertical leg units. Legs and levers build a parallelogram. Levers, hinges and legs are double barred such that they can accommodate an inlying chained transmission line. A total of four chain drives and eight UDUs (see Fig. 2.26) build up the parallel bogey unit. Using this kind of force transmission provides a lean construction, as only one 150W motor per side is used. The new module exclusively consists of *VolksBot* components and easily allows for later variation and expansion of the design. As illustrated, the chain drive is integrated in the lower link of the parallel bogey. In total, four variants of the UDU are used in the assembly of the parallel bogey unit. These variants are shown in figure 2.25.

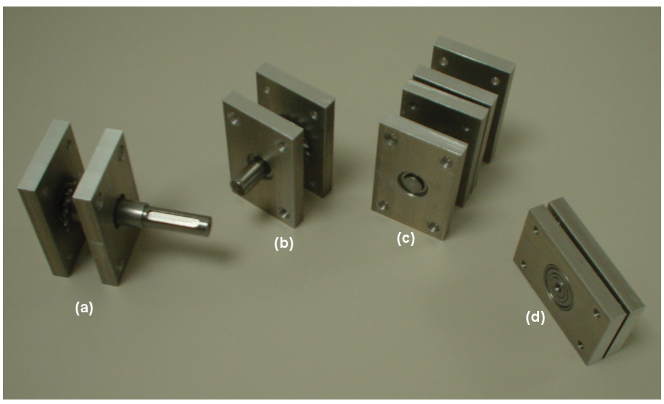


Figure 2.25: Variants of the UDU used in a parallel bogey unit

Variant (a) drives the wheel units, while variant (b) connects to the motor and drives the other UDUs via chain. Variant (c) forms the lower joints of the parallel bogey unit and transmits power to the wheels. Variant (d) forms the upper joints of the parallel bogey and is not used for power transmission.

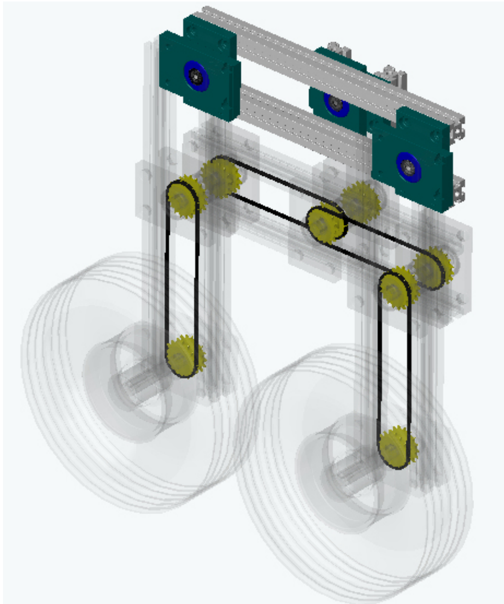


Figure 2.26: CAD drawing of the parallel bogey unit containing eight UDUs and four transmission chains

The chain transmission system requires only one motor per parallel bogey unit and hence can be easily scaled according to motor power and payload. Furthermore, the entire drive system including the motors can be encapsulated and protected inside the robot's frame.

2.5.3 The VolksBot XT variant

On the basis of the *VolksBot* construction kit, I conducted the design of an enhanced robot variant called *VolksBot XT*, using the new parallel bogey module and a passive caster wheel and tested its performance in ODE [8], a physical simulator (Fig. 2.27) on various terrains. The major drawback of the robot is that the passive caster wheel does not add to any driving power. It has been shown that the mobility of the rover on rough terrain highly depends on the robot shape and geometry (morphology), as well as on the constraints offered by the terrain. Additionally, these parameters do not influence separately, but rather as a combination. The optimization, therefore, needs a method which can deal with such complex dependencies. This motivated the use of evolutionary design optimization [56] as an approach for the optimization process. Also, this approach combines well with the use of a construction kit, as all variations could theoretically be built on the basis of one common kit with minor modifications.

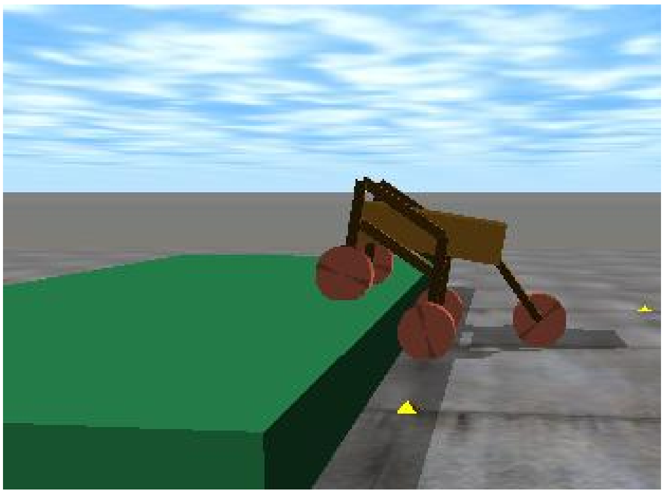


Figure 2.27: Simulation in ODE of the 5-wheeled XT

2.5.4 Evolution of morphology

Evolutionary algorithms, such as genetic algorithms, have proved to be extremely useful for optimization problems (e.g. The traveling salesman problem). Evolving neural networks [57] for the gait control of legged robots have shown very promising results. Robust and efficient control patterns could be created for various individuals. Bentley [56] classifies evolutionary design into four aspects: *Evolutionary Artificial Lifeforms, Evolutionary Art, Creative Evolutionary Design* and *Evolutionary Design Optimization.* In particular, for automated robot design, one can distinguish between robot configuration *synthesis* and *optimization.* Applied to robot design, configuration synthesis targets the generation of bodies and geometries in order to create a novel robot. Contrarily, configuration optimization aims to refine geometrical properties to improve the performance of a predefined robot structure. Various physical structures, including robot bodies and sensors, have evolved in the past [58] [59] [60]. Paul and Bongard [61] evolved the morphology and controller of a five-link bipedal walking robot simultaneously. Although the combination of evolution and rapid prototyping has been shown by Pollack [62], the component-based approach is clearly absent in this situation. Most importantly, the previously mentioned optimizations were done with a priori knowledge of the work environment. The environments developed for these robots are carefully designed and static. The success of these systems is questionable when the environments contains more uncertainty, as in a typical rough terrain environment.

2.5.4.1 System modeling in simulation

The modular assembly of the *VolksBot XT* allows for the application of robot configuration optimization on the robot's structure. The system, consisting of the rover and terrain, was modeled in an ODE simulation environment [8]. The model of the rover is shown in Fig. 2.27.

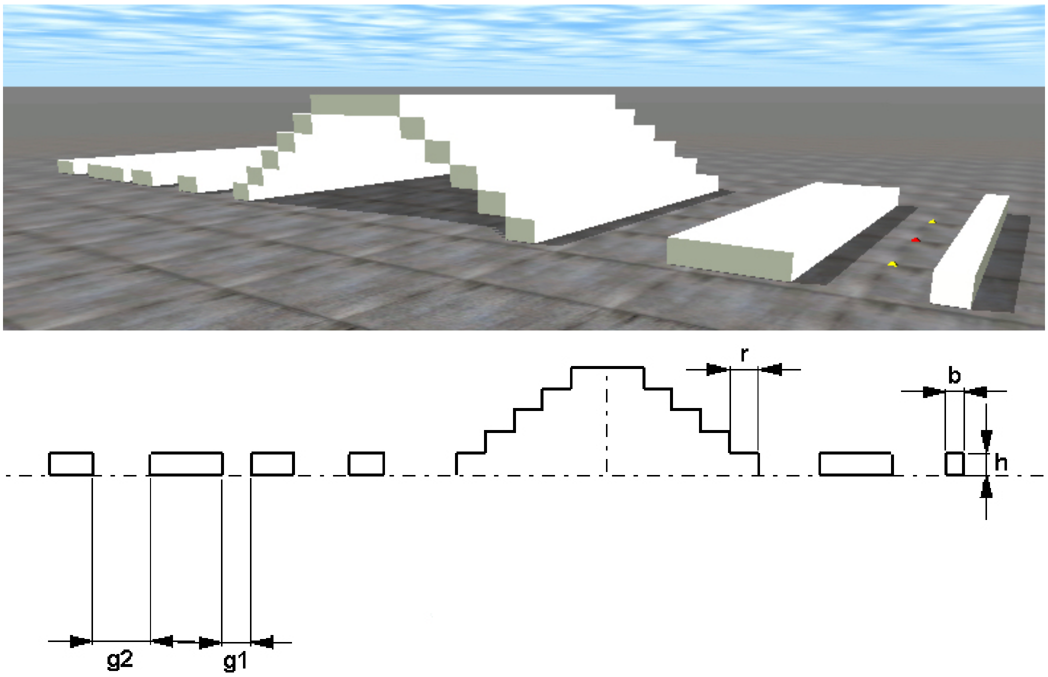


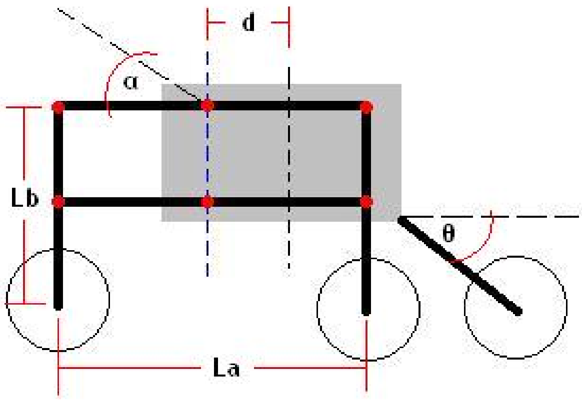
Figure 2.28: Evolution environment in simulation

The environment consists of cuboidal obstacles aligned in a straight path which the robot has to cross. To include aspects of robot rescue scenarios, a staircase has been added to the path (see Fig. 2.28). To avoid catering to a single environment, randomness is added by changing the following environment parameters during the evolution process:

- Surface friction coefficient μ
- Block height h
- Block width b
- Step width r
- Gap width $g1$ and $g2$

Initial experiments in simulation have shown that mobility is heavily influenced by the following geometrical parameters of the robot:

- Lever length (wheel distance) L_a
- Leg length(L_b)
- Lever hinge displacement (d)
- Lever angle limit (α_{max})
- Castor angle (θ)



Therefore, I applied an evolutionary optimization approach to these parameters which will be explained in the following.

2.5.4.2 The evolution process

The evolution tool employed, ISEE (Integrated Structure Evolution Environment) [57], is a software platform for the evolution of recurrent neural networks (RNN). It connects the ODE simulator to the evolutionary core program, EvoSun via the general interface, Hinton. The EvoSun program implements variation, evaluation and selection operators acting on populations of the neural networks. Hinton acts as a general interface to realize the interchange of evolved networks between EvoSun and simulation. For the optimization process, a fixed neural network structure (Fig. 2.29) of five input neurons for the parameters and one output neuron, which provides the fitness function, is used. Robots with higher mobility will travel faster and further over the obstacle track, hence fitness is the distance covered by the rover in a given time. Since each synapse has an effect on the output neuron, the evolution varies the synapse weights which represent the actual geometrical parameters. The synapse weights which are evolved have limits ranging from -1 to +1. Then these weights are scaled according to the given physical limits of the robot dimensions and fed into the simulation.
Two simulation parameters are controlled externally, the friction coefficient μ and the height of the block h. Both parameters are used to iteratively increase the difficulty of the terrain during the evolution process. The rest of the environment parameters are changed randomly within predefined limits during evolution to avoid over-fitting. The synapse weights make up a five-dimensional search space which contains the optimal solution. By setting a low initial difficulty level of the terrain, many individuals of

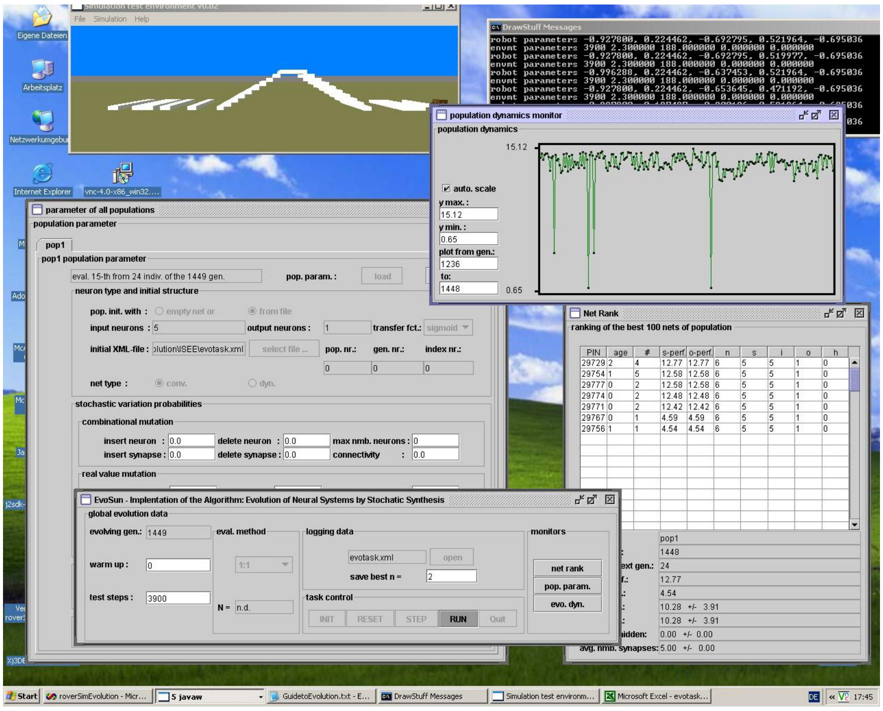


Figure 2.29: The ISEE evolution environment

different morphologies attain high fitness. Due to this, the population is spread into the search space so that many combinations of parameters can be tested by the evolution. In Evosun, the modified parameters during the evolution process were:

- Change Weight: The probability of varying the synapse weights
- Delta Weight: The mean delta value of a synapse weight change
- Birth Gamma: Selects the individuals which are allowed to produce offsprings. With a higher Birth Gamma, individuals with higher fitness are preferred.
- Average Population Size

Once a set of individuals with high fitness over generations is attained, the difficulty level is raised and Evosun parameters are modified, for example, by lowering the delta weight in order to converge to a smaller search space.

2.5.4.3 Evolution results

The initial run of the evolution took 1450 generations to converge into a stable set of individuals which could go over a set of randomly changing terrains. The fitness values (see Fig. 2.30) show that in the end, only four failures occurred in a span of over 200 generations despite permanently varying the environment after each generation.

Figure 2.30: Fitness values at the end of the evolution process showing a high rate of success

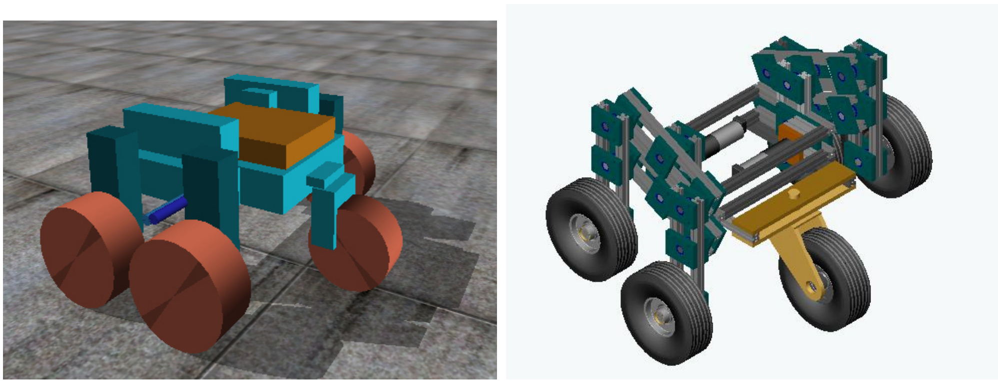


Figure 2.31: ODE simulation and CAD model of the robot as result of the evolution process

Comparing the morphology before (Fig. 2.27) and after evolution (Fig. 2.31), significant differences can be observed. After the evolution, the center of mass is low, the wheels are closely placed and the caster wheel is close to the body. Since the gaps between the wheels are small, the robot can easily overcome obstacles which otherwise tend to get stuck in the gaps. The lower center of gravity stabilizes the system in particular when moving down stairs. It was also noticeable that the low center of mass increases the climbing ability especially on grounds with low friction.

Although at first glance, the results obtained are obvious considering classical design techniques, a deeper look shows that the combination of these parameters obtained from the evolution is the key to the ability of the robot to move over the wide range of different environments offered.

Still, further performance analysis and testing of the solution has shown that the passive caster wheel very much limits the mobility in certain situations. This exhibits a fundamental drawback of the initial robot design which can not be compensated by the evolutionary design optimization approach. Therefore, I decided to modify the initial robot design before repeating the evolution process.

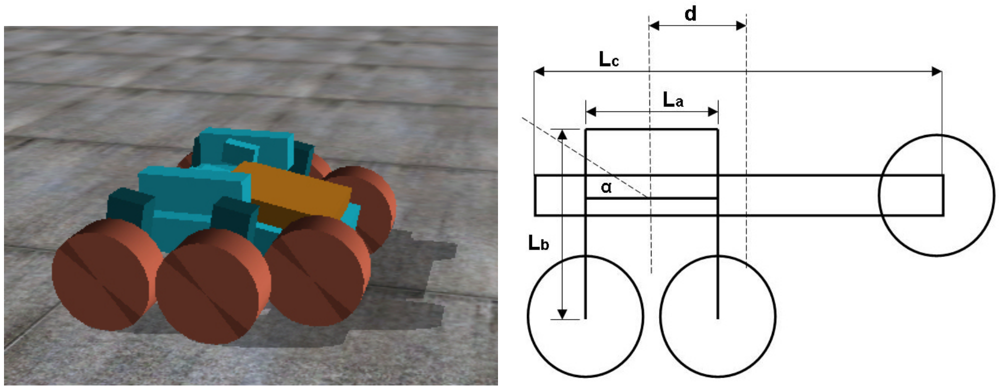


Figure 2.32: Modification of VolksBot XT structure and evolution parameters

In the new design, a six-wheeled version of the *VolksBot*XT (see Fig. 2.32), all six wheels of the robot are actuated. The modification from five wheels to six wheels involves the removal of the passive caster wheel and the integration of two wheel units positioned at the end of the elongated chassis frame. Due to the universal properties of the UDUs, the new drive units can be directly connected to the motor unit via a chain transmission inside the chassis frame. In addition to the geometrical parameters of the five-wheeled variant (i.e Lever length L_a, Leg length L_b, Lever hinge displacement d, Lever angle limit α_{max}) the Chassis length L_c was added as parameter for the optimization process. Evolving these geometric parameters of this new design, even with significant variation of the environment and constant increase in difficulty (increasing height of obstacles and lowering friction), the process converged after 1100 generations. The simulated robot can climb blocks of 240mm height and climb stairs with step lengths ranging from 250mm to 370mm and up to slopes as high as 40° on a smooth stone surface. The obtained parameters were then used assemble the new *VolksBot XT* platform. The CAD model and the real robot on the basis of the new parameter set are depicted in Fig. 2.33

The performance of the new *VolksBot XT* variant in various environments is shown in Fig. 2.34. The depicted stairs are outdoors with a slight incline (Fig. 2.34 top), yet the robot successfully manages and also climbs various indoor staircases with less friction at an incline of 40° (Fig. 2.34 center). Further performance evaluation was done on random step fields (Fig. 2.34 bottom). The robot was able to pass over this truly

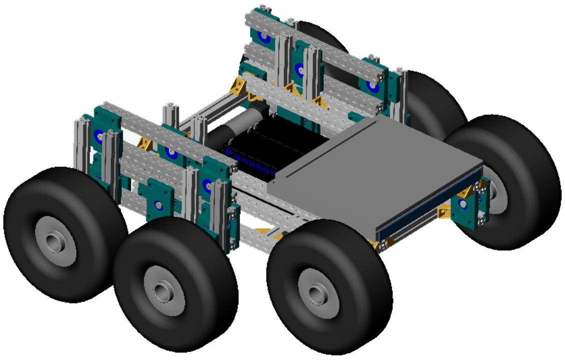



Figure 2.33: The new VolksBot XT

rough terrain even though it was not present in the environment during evolution. The robot's performance was further evaluated at the RoboCup Rescue Workshop 2005 (see section 2.6.1.2) where the high mobility performance of the platform could be further confirmed.

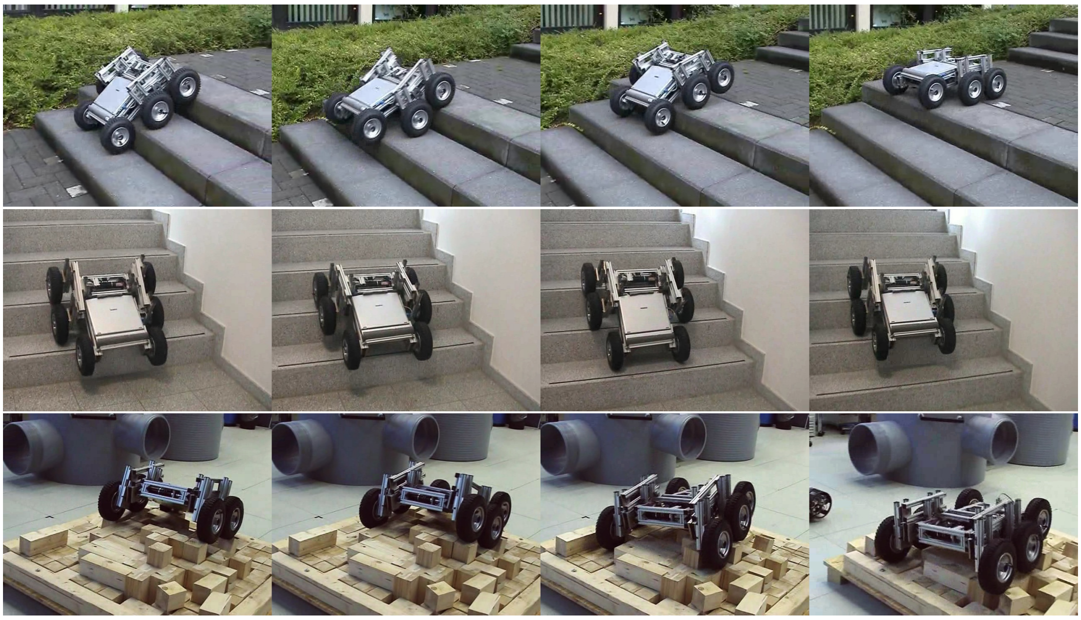


Figure 2.34: XT climbing an staircase outdoors, a steep staircase indoors and moving over a random step field

2.6 Application prototyping with VolksBot

The universal applicability of the *VolksBot* robot construction kit is demonstrated by various projects and application prototypes which are based on this development. A selection of these are presented in the following:

2.6.1 VolksBot in RoboCup

I conducted various development projects with *VolksBot* in the context of RoboCup. Our RoboCup team, as well as other teams, used and are still using different *VolksBot* platforms and components to participate in the RoboCup MiddleSize, Rescue and @Home league.

2.6.1.1 VolksBot in RoboCup soccer

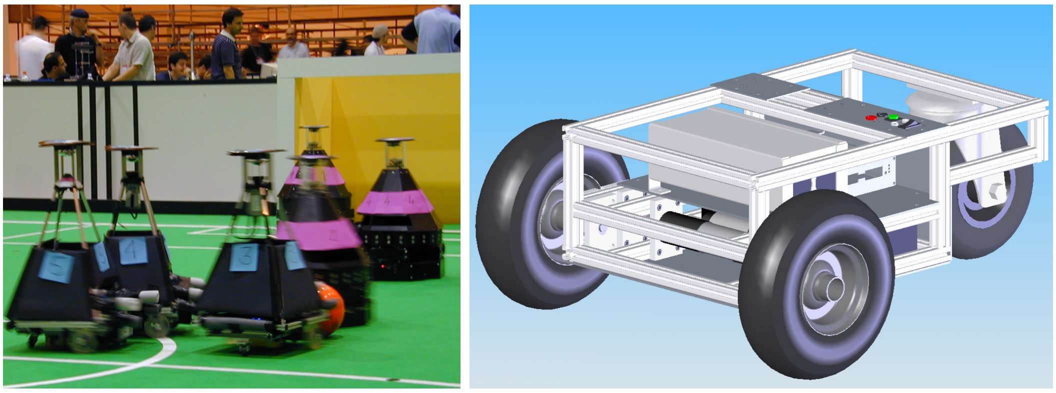


Figure 2.35: VolksBot in MSL and CAD model of an outdoor soccer platform

In the beginning of 2004 the international student-team (AIS/BIT) was established to participate in the RoboCup Middle Size League (MSL) using *VolksBot*. The main demands on MSL robots are quite different from other scenarios and require higher dynamics, superior motion control and real-time color vision. To meet these demands, the team had to introduce only a few additional plug-in components to the existing *VolksBot Indoor v.1* platform. *ICONNECT*, in combination with DD-Designer and the ODE simulator (see section 2.3.4.1), was used for the design of the robot control. Especially in highly dynamic situations, participation in MSL demanded high image quality. Therefore, the web-cams of the *IAISVision* system were replaced by *Sony DFW-500 IEEE1394* cameras. Furthermore, I developed an additional pneumatic kicker module for the *VolksBot MSL*. Later developments included the integration the holonomic drive unit, the new motor controller VMC, as well as a more compact chassis frame. Furthermore, on the basis of *VolksBot RT*, I designed and implemented a variant as a demonstrator for a potential new RoboCup outdoor soccer league. Using 150W DC motors and a gear ratio of 1:15, together with a lowered center of mass and increased width compared to the standard *RT3* variant, the robot has a maximum speed of 6.8m/s.

Fig. 2.35 shows *VolksBot MSL* robots during a game at the International RoboCup Competitions 2004 in Lisbon and the CAD model of the outdoor soccer prototype.

2.6.1.2 VolksBot in Urban Search and Rescue

To evaluate the grade of usability of the construction kit by people unfamiliar with the system(G10), the six-wheeled version of *VolksBot RT* was used as experimental base platform at the RoboCup Rescue Workshop 2004 in Rome (see. Fig. 2.36). There, within 15 hours of lab activities, together with my colleague Walter Nowak, I instructed two groups consisting of three and six people - with no prior experience with the system - in how to build a functional tele-operated rescue robot with semi-autonomous behavior. The task of one group was to build an interface for the robot operator including

Figure 2.36: VolksBot RT6 used as a Rescue Platform at the RoboCup Rescue Workshop 2004 in Rome

visualization of the robot's states, camera image and LIDAR data. Later on it was required to set the robot's state e.g. from manual to autonomous mode and build an interface to a joystick and throttle for proper tele-operation. The task of the other group was to design and program the entire robot control system on the robot PC including, signal processing of laser-scanner data, image processing, compression and wireless data transmission of the IAISVision image stream, interfaces for tele-operation and manual override, autonomous behavior and motor-control. An obstacle-avoidance method based on vector field histograms [63] was modified to achieve the desired behaviors like general obstacle avoidance, left and right-wall following or centering between the aisle.

The *ICONNECT* signal-graph in Fig. 2.37 gives a deeper insight into the software structure of the system. This graph describes the robot's main functionality. It communicates with the laser scanner and the motor-controller via two different sub-graphs, which are started in the initialization phase by triggering the modules "Start 1" and "Start 2". Using such sub-graphs offers the possibility to have different cycle times, which is very useful when reading data from sensors with different sampling rates. Furthermore, such sub-graphs offer a good way for encapsulating reusable software. The scanner data is received by "MultiComm1", which provides generic data transfer between the two signal graphs. "Joystream" contains a UDP server, which connects to the operator's remote computer, receiving Joystick data necessary for manual control of the robot. The two "TimeOut" modules act as signal buffers and ensure that the system stops in case of communication break-down. "PolarXY" converts the scanner-data to Cartesian coordinates. "VectorField" analyzes the scan data and returns free-space information about the robot's surroundings, which is fed into the "ActionSelect" module. Here, the desired direction from the operator is processed together with the free-space information and an optimal path is determined. "SlowDown" and "E-stop" analyze the near-region of the robot and reduce the translational speed with respect to the obstacle distance, or in case of a signal break-down. The finalized desired translational and rotational velocity are sent to the motor-controller sub-graph via "MultiComm2".

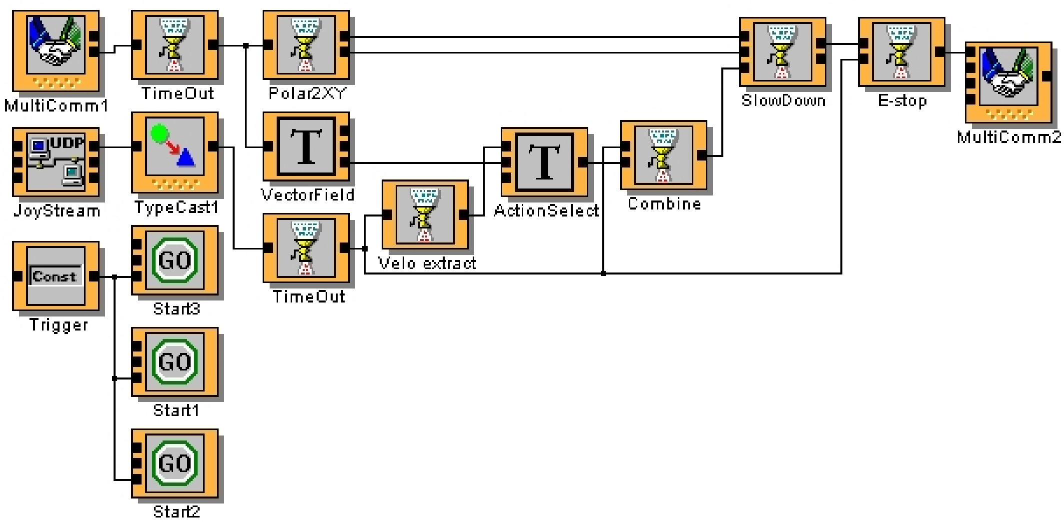


Figure 2.37: The main control graph developed an the RoboCup Rescue Workshop 2004

The two groups worked together well, first defining the interfaces, implementing sub-graphs, then combining and testing the results iteratively. On the whole, the *VolksBot* concept of rapid system prototyping proved to work successfully, resulting in a stable, functional system within very short amount of time, constructed by a heterogeneous group of people originally unfamiliar with the system. A successful demonstration of the system was done at the end of the workshop. Similar results were obtained at the RoboCup Rescue Workshop in 2005, where I used the *VolksBot XT* (see section 2.5) as a high mobility experimental platform. There, the task of the group was to design various behaviors like autonomous stair climbing and homing, and then to integrate the new functionality into the existing software framework from the previous year's workshop. Figure 2.38 shows the robot traversing a random step field during the final presentation of the workshop. These results strongly indicate that the following design

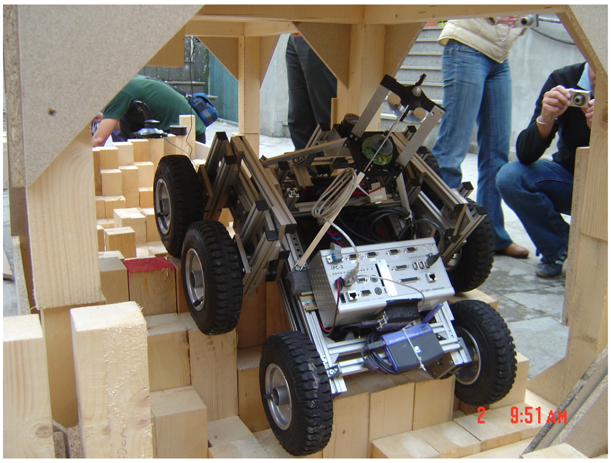


Figure 2.38: VolksBot XT traversing a random step field at the RoboCup Rescue Workshop 2005

goals from section 2.2 have been met:

- G1: Reduce costs, time and resources in mobile robotics projects
- G2: Be able to manage system complexity
- G3: Allow exchange and reuse of existing components
- G4: Allow simple reconfiguration and extension of the systems
- G7: Foster exchange of knowledge
- G10: Allow for short training periods for new users

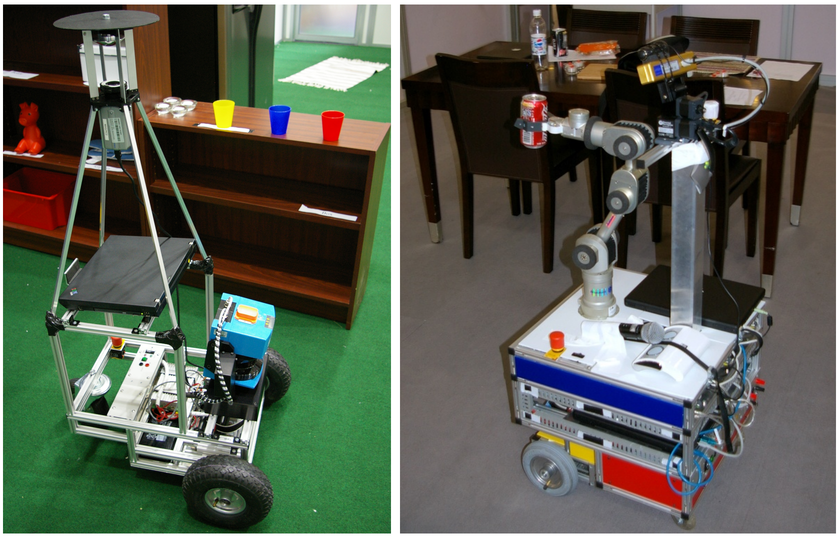


Figure 2.39: VolksBot participating at RoboCup@Home 2007 in Atlanta (left) and 2008 in Suzhou, China (right)

2.6.1.3 VolksBot in RoboCup@Home

For the RoboCup World Championships 2007 in Atlanta, our team started the development of a modified *VolksBot RT3* robot to take part in the RoboCup@Home league [64] three weeks before the tournament. Thanks to the modular design, it was possible to integrate two notebooks, the 3D LIDAR system *3DLS* and an *IAISVision* system on the robot within this limited mount of time by a small two-person team. By reusing existing MSL and FAIRlib software, it was possible to participate in four different benchmark tests. The robot in the @Home scenario of 2007 is depicted in Fig. 2.39(left). For the competitions in 2008, I conducted the design of a new platform, which was a modified version of the *VolksBot Indoor v.2* variant. The new platform uses a 2D LIDAR system, a stereo camera, a web-cam, a pan-tilt unit and a Katana arm from the *VolksBot* hardware component library described in section 2.3.3. With this development, our team *BIT-Bots* won second place in the RoboCup@Home championships in 2008. Figure 2.39(right) shows the new @Home platform delivering a can in the final of the competition. In chapter 4, further details on the RoboCup@Home initiative are given.

2.6.2 Prototyping of real world applications with VolksBot

Besides application in RoboCup, the VolksBot construction kit has been used in various other domains including autonomous transportation, surveillance and underwater robotics.

2.6.2.1 The VolksBot PeopleMover platform

Figure 2.40: The PeopleMover, a prototype of an intelligent vehicle on VolksBot basis

The *PeopleMover* (see Fig. 2.40) is an extended version of the three-wheeled *RT* variant introduced in section 2.4.3. It was used as a technology demonstrator for the *VolksBot* robot construction kit at the RoboCup German Open 2005. Its task was to autonomously transport people with a maximum weight of up to 80kg between predefined points while simultaneously avoiding obstacles. The reuse of software developed for the RoboCup Rescue workshop 2004 (see section 2.6.1.2) helped to build this prototype within only one week. The modifications which led to this prototype are briefly described in the following. To enable the robot to transport a person, the chassis frame of the original platform was elongated by 20cm, a seat, handles and foot rests were constructed based on the X-Beams and plastic housing for the robot was provided. In addition, a joystick and 2D LIDAR previously used for the rescue robot were attached.

In *ICONNECT*, modules from the rescue workshop such as the *TimeOut*, *SlowDown* *VectorField* or *ActionSelect*, as well as the entire laser-scanner sub-graph, were reused to create the signal-graph depicted in Fig. 2.41.

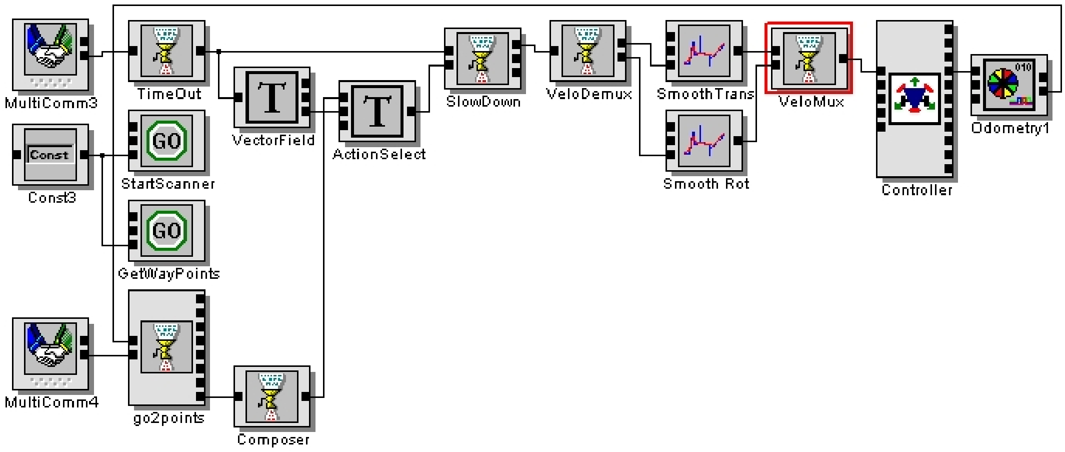


Figure 2.41: The main control graph of the People Mover

In addition to the functionality from the Rescue scenario previously described, an autonomous navigation system was implemented based on odometry information. The predefined way-points are provided by a separate sub-graph through *MultiComm4*. *Go2points* is a behavior which navigates the robot to the next way-point. Since the output interface of this module is identical to that of the joystick module used in the Rescue scenario, it can be directly connected to *ActionSelect*. To limit the dynamics of the robot for safety reasons, the velocity signal from *VeloDemux* is smoothed by using a moving-average algorithm for translational and rotational velocity in *Smooth Trans* and *Smooth Rot*. The *Controller* module, interfacing the motor-controller board via RS232 sends the desired velocities and receives motor encoder ticks. These ticks are sent to *Odometry1* where odometry-based pose (x, y, Θ) is computed and fed back to the behavior.

As a result, the efficient realization of this *VolksBot* variant strongly indicates that the following design goals from section 2.2 have been met:

- G3: Allow exchange and reuse of existing components
- G4: Allow simple reconfiguration and extension of the system
- G5: Robust and scalable mechanical design
- G6: Allow efficient integration of existing technology

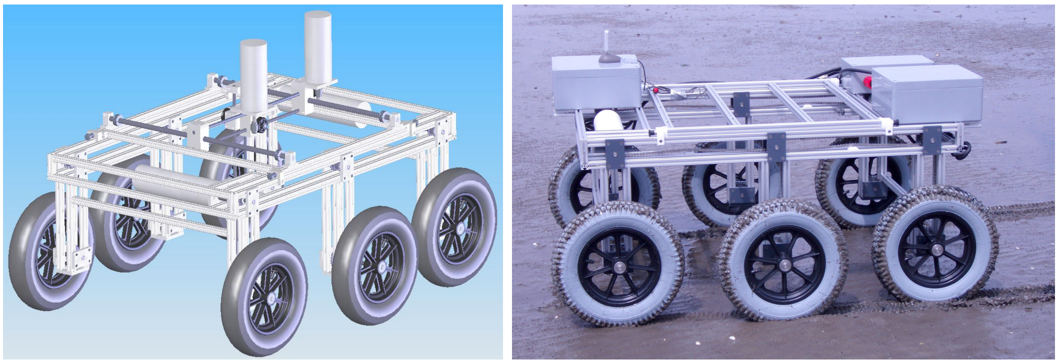


Figure 2.42: CAD assembly (left) and image of MarBot (right), an underwater VolksBot variant

2.6.2.2 MarBot - underwater rover based on VolksBot

In cooperation with the Alfred-Wegener-Institute for Polar and Marine Research (AWI)[1], I developed an autonomous underwater robot for marine seabed analysis on the basis of the *VolksBot* kit. Instead of providing a complete housing for the robot, only the robot's sensitive hardware parts, like the motors, the motor controller, the batteries or the control PC, were shielded from the surrounding salt water by housings provided by AWI. Besides the underwater environment the robot was designed for, various other demands had to be met regarding the design of the MarBot. Payload and size of the platform had to be increased to allow the installation of additional sensors and actuators. These included a mass spectrometer for advanced soil analysis which was mounted on a three-axis manipulator. Therefore, I designed an exchangeable center frame carrying the additional hardware, which allows fast reconfiguration of the robot during an expedition. Also, the ground clearance had to be increased to 400mm to minimize the dispersion of sediments while driving. The resulting platform is illustrated in Fig. 2.42. It has six actuated wheels of 400mm diameter, a total size of 1200x700x650mm, a maximum speed of 1m/s and it weighs 30kg. The construction followed the design principles of the *VolksBot RT* series using the UDU with chain transmission. Only a few drive unit parts like the bearing blocks and bearings had to be replaced by plastic parts to avoid corrosion. A Nano ITX barebone PC was used for the control of the robot. It communicated to a base station via WLAN and UDP connection in shallow water allowing remote control and monitoring of the sensor data. In software, both a cockpit for the operator and the robot control software have been implemented in *ICONNECT* thanks to the existing module library. Future development will include

[1] Alfred Wegener Institute: http://www.awi.de

autonomous operation by use of multiple sensors like GPS, IMU, compass and vision allowing the robot to go from shallow water to depths of up to 30m.

2.6.2.3 The VolksBot Fuel Cell platform

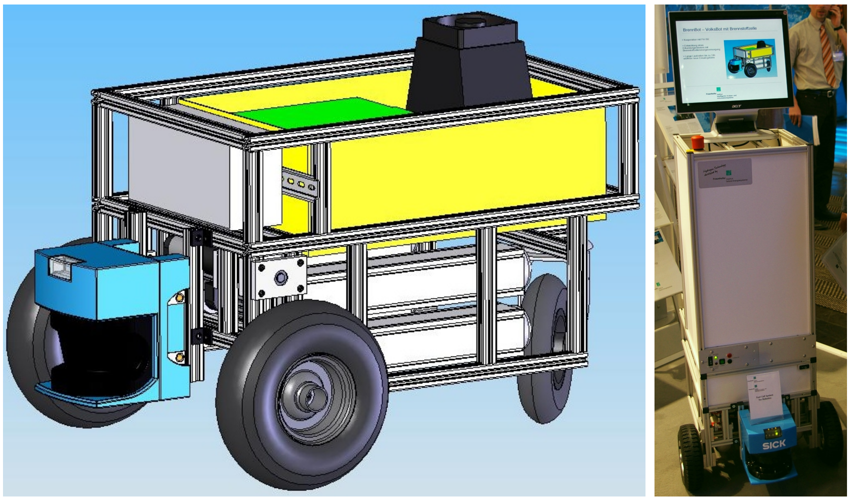


Figure 2.43: CAD assembly and image of VolksBot Fuel Cell

In cooperation with the Fraunhofer Institute ISE[1], I developed a *VolksBot* variant which uses a fuel cell as central power supply. The fuel cell provides up to 400W power at 24 VDC. Depending on the mode of operation and the size of the metal hydride tanks filled with hydrogen, it allows up to 24h of continuous operation. In this case, the application of the *VolksBot* prototyping concept allowed me to customize the chassis design of the robot according to the physical specification of the fuel cell and its underlying components. This fact again indicates the flexibility and general applicability of the *VolksBot* concept. A *VolksBot RT3* variant was used as the basis for this development. The robot was further equipped with a SICK LMS200 LIDAR, an industrial PC and a TFT display. It was exhibited at the Hannover fair 2007 in Germany.

2.7 Results

The variety of efficiently implemented *VolksBot* variants presented in the previous section strongly indicates the feasibility of the approach for application prototyping of mobile service robots. In this section, first the accomplishment of the defined design goals and criteria is verified. Then a measure of assembly times for different platforms

[1] Fraunhofer ISE: http://www.ise.fraunhofer.de

is analyzed. Finally, the distribution of *VolksBot* robots and components in the robotic community is presented.

2.7.1 Verifying design goals

In the following, it is analyzed if the design goals from section 2.2 have been met by the implementation of the *VolksBot* construction kit.

[G1] Reduce costs, time and resources in mobile robotics projects Reduction of costs, time and resources is achieved by efficient reuse and recombination of existing construction kit components. A standard *VolksBot RT* robot can be built at hardware costs of 1800 Euro within 7 hours assembly time (see section 2.7.2). An application prototype like e.g. the MarBot (see section 2.6.2.2) can be built at hardware costs of 6000 Euro. Adaptation of the hardware design took roughly 25h, mechanical assembly took 15h and programming took 10h. Comparing these numbers with previous robot development without the construction kit, I estimate a reduction of 50 to 70 percent in time and cost depending on the level of reusability of components, which can be obtained by use of the kit.

[G2] Be able to manage system complexity The component oriented design allows to manage and reduce system complexity. The hardware design of all robot platforms follows a common hierarchal structure (see section 2.3.1). The use of standardized hardware modules, components, interfaces and connectors in combination with a detailed assembly manual eases and standardizes the assembly process significantly. In software, the use of *ICONNECT* (see section 2.3.4.1) provides programming on three abstraction levels: parametrization of control graphs, composition of control graphs and module programming. The black-box representation of reusable functionality in parameterizable and documented software modules further reduces overall system complexity and fosters reuse.

[G3] Allow exchange and reuse of existing components The construction kit approach allows for an efficient exchange and reuse of components in hardware and software. The implementation of ten robot platforms for different applications on the basis of one common construction kit has been demonstrated in section 2.4.3 and 2.6.

[G4] Allow simple reconfiguration and extension of the systems In mechanics, simple reconfiguration and extension is ensured by use of the standard chassis frame consisting of X-beams in combination with the Universal Drive Unit (see section 2.4.2). Additional hardware can be attached via standard connectors and accessed via defined software interfaces. In software, efficient reconfigurability and extension is fostered by use of the *ICONNECT* framework which has been demonstrated in section 2.6.1.2 and 2.6.2.1.

[G5] Low maintenance efforts, simple assembly procedures Low maintenance efforts and simple assembly procedures are fostered by high accessibility of hardware components, use of standard connectors and limited physical dependencies between components. An excerpt of an assembly manual in the Appendix gives insight into the step-wise assembly process which requires minimal skills and tools. Furthermore, the extensive use of high quality industrial hardware components (e.g. motors, drive unit components and frame unit) helps to reduce maintenance efforts significantly.

[G6] Allow efficient integration of existing technology Efficient integration of existing hardware components into the construction kit has been shown in section 2.3.3. In software, existing software libraries have been integrated and used (see section 2.3.4).

[G7] Foster exchange of knowledge Knowledge exchange is fostered by a comprehensive documentation of the construction kit components including the assembly manual and documentation of software modules. Example implementation of reusable functionality is provided in software, and complexity can be scaled in multiple abstraction layers.

[G8] Robust and scalable mechanical design The enhancements of the original indoor construction kit towards the design of rough terrain platforms (see section 2.4) included the introduction of the double-layered chassis frame and the *Universal Drive Unit*. These enhancements allowed for a high scalability of the robot design which has been presented in section 2.4.3.

[G9] Allow for a wide range of robot variants and applications The implementation of ten robot platforms for different applications on the basis of one common construction kit has been demonstrated in section 2.4.3 and 2.6. Applications include:

soccer, urban search and rescue, domestic service robotics, autonomous transportation, outdoor exploration, underwater robotics, research and education.

[G10] Allow for short training periods for new users Training periods for new users are minimized by the help of the clear component oriented structure of the construction kit with limited dependencies between the components. Together with documentation standards for hardware and software components, interface definitions, assembly and example applications, this allows new users quickly develop a broad understanding of the entire system as well as to work into specific aspects of the system, which has been demonstrated in section 2.6.1.2.

[G11] Achieve synergies by standardization Consequent standardization in hardware and software includes the design of the CAD component library (see section 2.3.2), the electric and electronic component library (see section 2.3.3), the software module library in *ICONNECT* (see section 2.3.4.1) and the development of the *FAIRlib* (see section 2.3.4.2. Reuse, recombination, adaptation, integration and new development of standardized components and modules forms the basis of the *VolksBot* construction kit approach and has proved to increase synergies and efficiency in mobile robot design and application development significantly.

In summary it can be confirmed that all previously defined design goals for the implementation of the *VolksBot* robot construction kit have been met. Obviously the concrete application of such a construction kit to build an individual robot variant or application often requires an adaptation and enhancement of the kit. The open and component oriented structure of the kit fosters such enhancements which allows for a constant extension and adaptation of the kit itself, preventing it from being outdated after a certain time.

2.7.2 Measures of assembly times

For two selected *VolksBot* variants, assembly times were measured to verify the claim of fast building times(G1) and of good usability of the kit even by inexperienced users(G5,G10). First, the times used by an expert and a beginner to build two non-trivial variants, namely a *PeopleMover* (see section 2.6.2.1) and the six-wheeled version of *VolksBot XT* (see section 2.5), were measured. All build times in table 2.5 are in given in minutes. Using the kit, an expert can build a *PeopleMover* in roughly 7 hours, while the much more involved *XT* needs roughly double time. Even a beginner is able

Type	PeopleMover		XT		
Level		Expert		Beginner	Expert
Special body parts (min.)	upper	20	chassis frame	70	40
	middle	25	up. lever	100	50
	lower	25	low lever	150	70
	drive	50	legs	200	120
	seat	30	hinge block	30	20
	handles	45	Par. bogey	150	120
Common parts (min.)	Frame	75		400	240
	Chains	80		320	200
	Wheels	10		10	10
	Wires	70		50	30
Total(min.)		430		1480	900

Table 2.5: Assembly times for VolksBot variants

to assemble this elaborate design in about 21 hours. Also, for all other possible variants such as the *RT3, RT4* or *RT6* the assembly times are very much in the same range as for the *PeopleMover*.

2.7.3 Use and distribution of VolksBot robots and components

Initially developed to accelerate in-house robot design at Fraunhofer IAIS, the *VolksBot* construction kit and its underlying modules and components have been made available for distribution to research institutions, universities, schools and companies world-wide. Table 2.6 gives an overview over the quantities of *VolksBot* platforms and components used in-house and distributed externally in the period of 2003 until 2008. In total, over 200 VolksBot platforms and 200 components were used in internal projects, by cooperating partners and by over 60 universities, schools, research institutions and companies world-wide.

2.8 Summary and Conclusions

In this chapter, I presented a concept for application oriented prototyping of mobile robots, the *VolksBot* robot construction kit, consisting of reusable modules and components in hardware, software and mechanics. After defining design goals and deriving design criteria for the implementation of the kit, the modular composition of the kit is explained in detail. The hardware composition of the kit follows a multi-layer hierarchal structure which provides a high level of reconfigurability, minimizes dependencies between modules and provides clearly defined interfaces between the components. Following this structure, I designed a mechanical component library in CAD which allows to design all *VolksBot* variants in CAD prior to actual physical assembly. This CAD

Robot Platforms	In-house use	External use
VB Indoor v.1	11	32
VB Soccer	6	22
VB Indoor v.2	8	17
VB RT	13	33
VB XT	2	5
VB FuelCell	1	1
MarBot	0	1
ProfiBot	12	34
Concept studies	7	0
Robots Total	**60**	**145**
Components	**In-house use**	**External use**
Motor controller	16	86
3DLS	5	14
IAISVision	15	28
Hyperbolic mirror	9	43
Components Total	**45**	**171**

Table 2.6: In-house and external use of VolksBot variants and components

library reduces efforts for new platform development enormously due to extensive reuse of existing components and the possibility of fast iterations during a design cycle. Also in electric and electronic hardware, I composed a component library which holds commercial products as well as in-house developments such as sensors, actuators or control hardware with defined interfaces in hardware and software. In software, a commercial framework for the visual composition of modular signal graphs, *ICONNECT* was used and enhanced with software modules containing robot specific functionality. The design of an operating system independent software library called *FAIRlib* allowed reuse and exchange of robotic related software methods beyond the *VolksBot* project and the use of *ICONNECT*. In section 2.4 I demonstrated how the hardware of the original indoor kit was enhanced to obtain higher physical performance and flexibility in the robot design. The introduction of the double layered chassis frame in combination with the *Universal Drive Unit* has been presented as the key for the efficient design of a variety of rough terrain, outdoor and heavy duty robot platforms (*VolksBot Indoor v.2, VolksBot RT3 RT4 and RT6*). For applications demanding for high mobility such as stair climbing or urban search and rescue scenarios, the *VolksBot XT* was designed by use of the new parallel bogey drive unit and by applying evolutionary design optimization of the robot's morphology during the development process in simulation. An overview on all standard *VolksBot* variants is given in Fig. 2.44.

Various application prototypes have been realized on the basis of the *VolksBot* construction kit. Several *VolksBot* variants were designed for participation in the Robocup Middle Size league, the Robot Rescue league and the RoboCup@Home league. Finally,

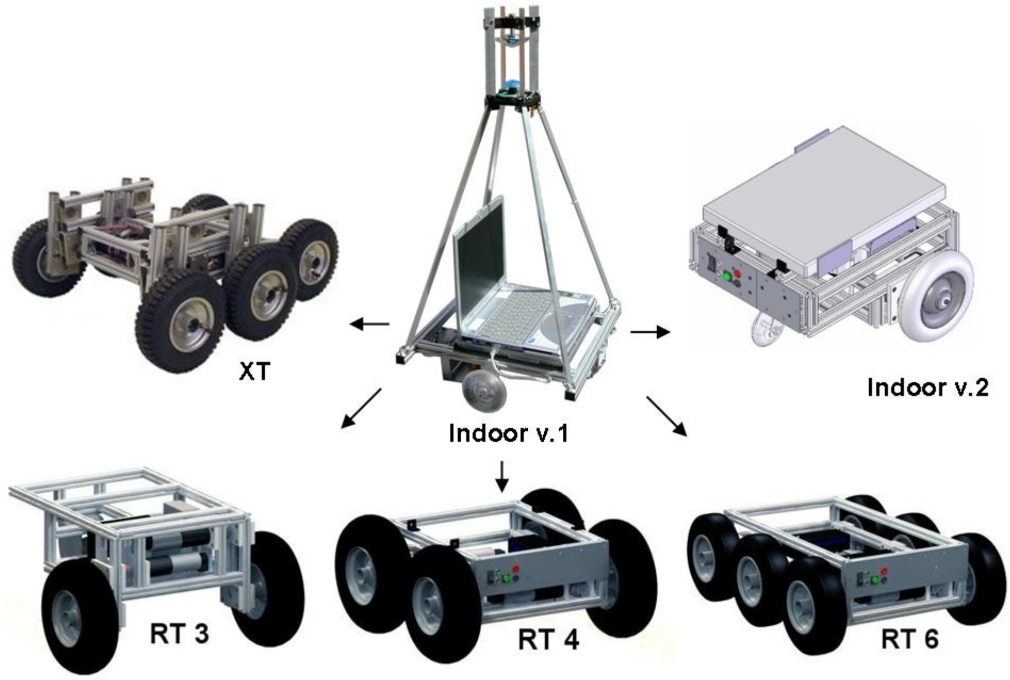


Figure 2.44: Overview on standard VolksBot variants

a demonstrator for autonomous transportation of persons (*PeopleMover*), an underwater variant (*MarBot*) and a fuel cell powered service robot (*VolksBot Fuel Cell*) were efficiently implemented with only minor modifications necessary and by extensive reuse of modules and components from the *VolksBot* construction kit. An overview on the conducted application development based on the construction kit is shown in Fig. 2.45. Successfully implementing this variety of robot platforms and applications on the basis of one common construction kit demonstrates the feasibility and effectiveness of the *VolksBot* design approach.

As a further result, it was shown that the defined design goals have been met on a large scale by showing the relation of these goals and criteria to the implementation of the construction kit and the robot platforms. Measures of the time required to assemble *VolksBot* platforms further support the claim of rapid prototyping. The results conclude with an overview on the distribution and use of *VolksBot* variants and components in the international robotic community.

Future work will include the consequent enhancement of the construction kit for new applications. This includes the development of a modular, steerable drive unit for autonomous transportation of high payloads and people. An initial CAD model of this robot is shown in Fig. 2.46 (left). Furthermore, the development and integration of a track drive unit and the additional installation of a flipper module (see Fig. 2.46 right)

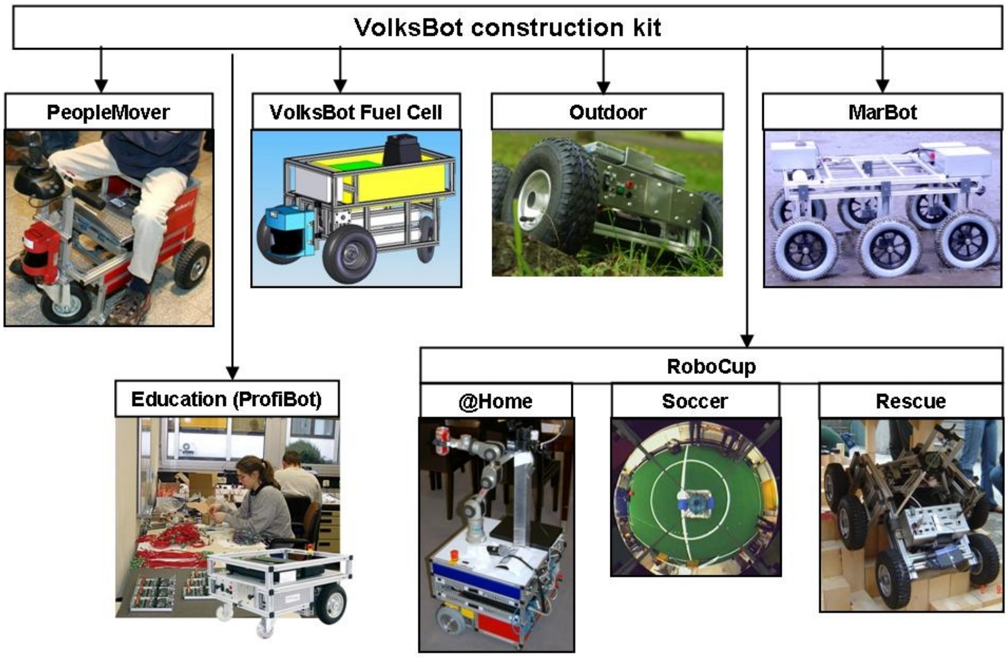


Figure 2.45: Overview on VolksBot application development

as a replacement for the wheels should enhance mobility in the future significantly. In software, the FAIRlib will be enhanced by reusable modules for indoor and outdoor localization, navigation and functionality in the domain of domestic service robotics and human robot interaction.

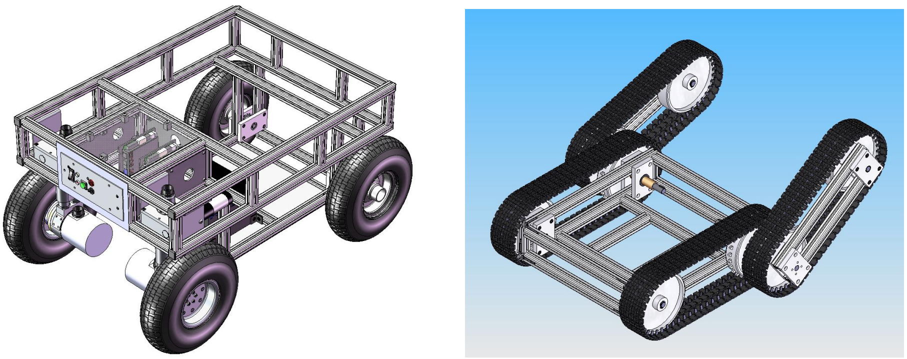


Figure 2.46: VolksBot with steerable drive unit(left) and with track drive and flippers(right).

3

Sensor development for service robots

In the previous chapter, the *VolksBot* construction kit was presented as an efficient approach for prototyping autonomous service robots. Operating such systems in the real world requires a focus on the robot's perception capabilities, as such environments are usually highly dynamic and uncertain. Therefore, in this chapter, the development and application of two sensor systems for the *VolksBot* construction kit are discussed in more detail, namely the *IAISVision* system and the 3D-LIDAR system *3DLS-K*.

First, the hardware development of the catadioptric vision system *IAISVision* is presented. Then, a three step method which consists of PI camera parameter control using reference colors, image segmentation and color classification was developed and applied in order to obtain robust color perception under changing light conditions for both indoor and outdoor situations. In particular, an intrinsic camera parameter controller is used to obtain a high grade of color stability in the YUV color space under varying light conditions. Then, an image segmentation method is applied in order to detect spatially coherent regions of uniform color belonging to objects in the image. Finally, a probabilistic classification method is applied to label the colors by use of a Gaussian color distribution model. Experiments on a *VolksBot* robot were done in a combination of artificial and natural light indoors and outdoors. The results show both the feasibility and the problems of this approach that occur under these highly diverse light situations. In particular, the application in a RoboCup soccer scenario and in possible future outdoor use is investigated. This work was carried out in cooperation with my former colleagues Yasutake Takahashi and Walter Nowak [65]. My contribution to this work includes the simulation, hardware design and development of the *IAISVision* system (section 3.1.1), the development and implementation of the adaptive camera parameter control method (section 3.1.3.3), as well as the conduction of the experiments and the analysis of the experimental results (section 3.1.4).

In the second half of this chapter, the system development and application of the 3D-LIDAR system Fraunhofer *3DLS-K* is discussed in detail. After giving insight into the sensor hardware development and the acquisition of 3D data, the sensor application to the domain of autonomous urban driving is presented, including an approach for road detection on the basis of the *Scan Line Approximation* algorithm. The relevance and the multipurpose character of the sensor development is shown by presenting the successful implementation and integration of the system as a sensor component for the autonomous car *Spirit of Berlin* which participated in the DARPA Urban Challenge in 2007. Here, a method for scene analysis of the 3D data at intersections and an according behavior system on the basis of a finite state machine were developed.

3.1 Adaptive color recognition with IAISVision

The use of video cameras and computer vision is well known to provide large amounts of information about the environment for mobile robots at comparably low cost. But besides object recognition and classification, one of the major challenges in this domain is to operate these systems in environments with changing light conditions. In many computer vision applications, one focus is to keep light conditions stable. Obviously, for mobile robots operating in real life environments, this is not possible and the systems have to cope with different types and combinations of illumination, like different types of artificial light sources, sunlight and shadow with varying intensity. In addition, color information, which humans can easily classify, may appear very differently in the camera images under these unstable conditions.

Also in RoboCup, a mid-term goal is to get away from highly defined artificial light conditions to have the robots cope with natural lights and objects. There, a possible next step is to install an outdoor soccer league to foster development in this direction in the future. Up to now, many teams require manually calibrated color look-up tables and fixed camera parameters. Obviously these systems only work under defined and constant illumination. Other approaches in this field avoid the use of color information at all. An overview on approaches addressing the problem of color constancy will be given in section 3.1.2.

3.1.1 Design of a catadioptric vision system IAISVision

As an original requirement for the *VolksBot* soccer robots used in the RoboCup Middle Size League (see section 2.6.1.1), I developed the catadioptric vision system *IAISVision* and integrated it as a component into the *VolksBot* construction kit. The vision system consists of an IEEE1394 CCD camera, a hyperbolic mirror and a sensor mount as shown in figure 3.1. Before construction, the system was modeled entirely in simulation using

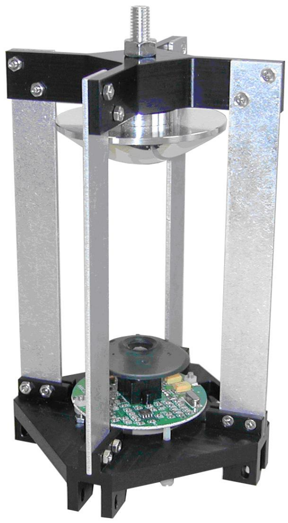


Figure 3.1: The *IAISVision* system

the ray-tracing software POV-Ray[1]. In an iterative process, all relevant geometrical parameters of the system were initially optimized for the use on a RoboCup Middle Size field. These parameters include height of the mirror with respect to the camera $h_m c$, height of the entire vision system above the ground h_g, diameter of the mirrord_m, focal distance $l_f d$ of the camera, and especially the two parameters a and b of the mirrors hyperbolic surface equation 3.1 with r being the radius and z the dimension along the optical axis.

$$\frac{z^2}{a} - \frac{r^2}{b} = 1 \tag{3.1}$$

The criteria for this iterative optimization process were full visibility of all landmarks from any position in the field, including goals and corner posts, and full visibility of the robot's close region. The simulated and the resulting real camera image are depicted in Fig. 3.2. Note that in the rendered image, the distance from the camera position to the right end of the field is doubled compared to the left end for calibration

[1]POV-Ray ray tracer(http://www.povray.org)

reasons. This optimization can be repeated for other scenarios and applications with the described method. Various cameras have been integrated, ranging from cost effective web cams to sophisticated industrial cameras. The mirror for IAISVision is custom built. An aluminum cylinder is used as raw material. The individual surface equation parameters r, a and b for the respective application are applied during the automated turning process. Then, the mirror is polished and a layer of raw aluminum is applied on the surface by vapor deposition. In a final step, the surface is protected against corrosion and scratches by application of a glass layer.

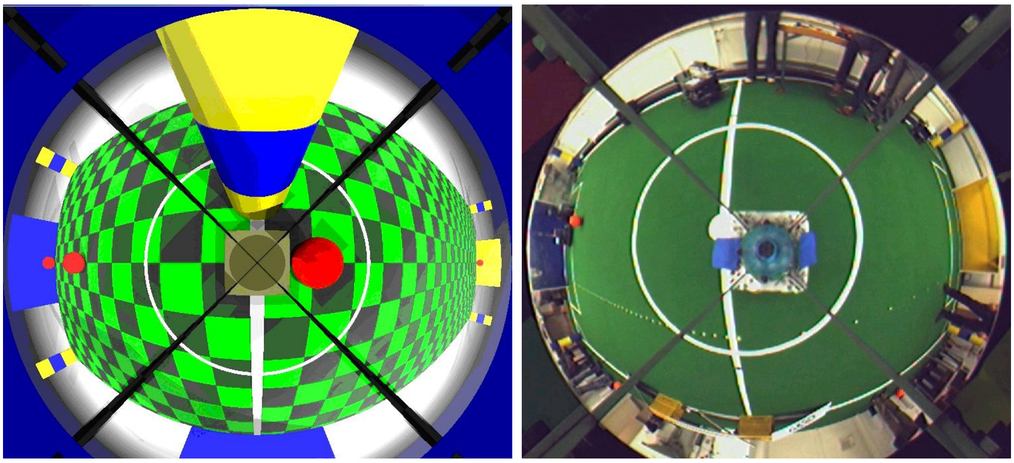


Figure 3.2: Rendered and real camera image of *IAISVision*

3.1.2 Approaches for color constancy

A vast amount of research has been done in the field of color constancy and adaptive vision systems for changing illumination. The focus mainly lies on the identification of illumination-independent descriptors for surfaces in a scene [66]. Usually this includes the two tasks of determining the type of illumination of a scene and mapping color values to a set of descriptors. A specific approach in this general procedure modifies the colors in an image in order to compare it with a reference image which was acquired under different illumination [67]. In general, the existing approaches can be categorized into physics-based methods which build up models containing the underlying physical processes, such as the dichromatic reflectance model, and statistics-based methods. The latter try to correlate distributions of colors under different illuminations, usually requiring more colors to be present in the image. Examples are max-RGB, the diagonal method, gray-world methods, [68], gamut mapping [69], as well as methods based on machine learning [70]. Other approaches use chromaticity (normalized) color spaces such as YUV or HSI, where the brightness value of each color is stored explicitly. [69] shows that methods using only chromaticity perform similarly to methods using the full RGB space, but they are more robust to shadows.

Due to camera characteristics and physical limitations, the intensity of light also has an influence on the color value in the image. This motivates the use of an online camera calibration method to keep the brightness in the image stable before doing the color segmentation and classification.

As several authors [71] [72] [73] point out, such approaches have to deal with enormous differences in the appearance of one color. In the color space, previously separated regions of uniform color may overlap, and the values of a set of colors change in various and highly nonlinear ways, especially when the type of illumination changes, e.g. from natural to artificial light.

The application in the domain of autonomous mobile robots enables the use of online methods, such as online adaptation of camera parameters. The problem here lies in the nonlinear control and calculation of the control error. The influence of adjusting a camera parameter can vary significantly among different cameras, as many parameters only influence the software preprocessing of the raw camera image and not physical values like aperture or shutter. Those parameters and underlying preprocessing models are often not specified, and the relation between parameter value and effect is usually very non-linear. As a possible solution, learning methods as in [74] can be applied.

One approach for controlling the camera parameters is to determine the control errors by the use of reference colors in the image. For example, white can be used to set the camera's white balance parameter and the brightness (white level). A defined reference color like red can be used to control saturation. With catadioptric camera systems like the IAISVision system (see section 3.1.1) a concentric multicolored ring can be mounted around the camera lens. With this, the reference color ring is always visible in the image and does not interfere with the actual image itself.

In structured and well defined environments, such as RoboCup soccer, other approaches make use of semantic knowledge about the environment (i.e the specification of the field and the objects in RoboCup soccer) [75] [76] [77].

A similar approach in a different scenario was taken by [78] to detect roads, assuming that a road is mostly flat, its color changes slowly and the car is driving on it. Alternatively, [79] use a three step method to identify pixels usable as white reference, but only white balance is controlled. Also, it requires the color white to be present in the environment during the calibration process.

In several approaches, first a segmentation or edge-detection step is done, and then colors of the detected segments are classified [80] [81]. Here, the focus lies on improved color recognition and fast processing time.[82] and [83] classify colors by modeling color distributions as Gaussians. [84] shows that predefining a discrete set of illumination conditions (bright, intermediate, dark) already improves the classification result significantly.

In the following approach, further evidence for the advantages of the differentiation between various light conditions is given, showing the benefits resulting from a continuous adaption of the camera parameters.

3.1.3 The processing steps for stable color perception

The method presented in the following consists of three major processing steps:

1. Segmentation of vertical lines into spatially uniform color regions and calculation of mean color values for each segment

2. Classification of each segment to a set of color representatives

3. PI-Control of camera parameters by use of reference colors

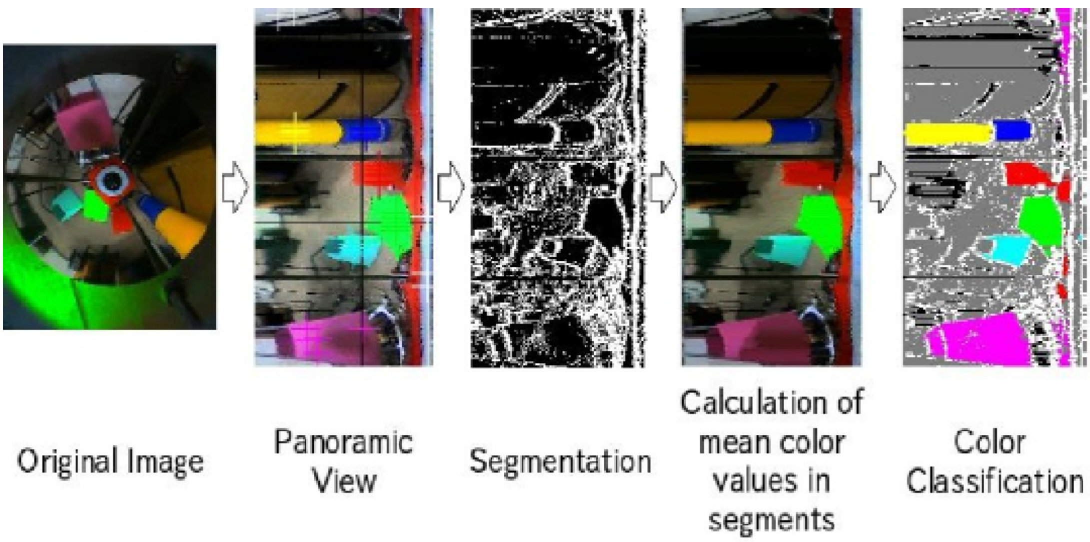


Figure 3.3: Overview of the involved image processing steps

Fig. 3.3 gives an overview of the steps involved in the image processing. These steps will be described in detail in the following.

3.1.3.1 Image segmentation

A boundary-based Markov Random Field method for line-based segmentation of an image is used. Markov Random Fields have been proposed as a model for the visual field in the human brain. Many variations of Markov Random Field have been developed, and some of them have been already applied to the task of image segmentation. This method provides a sophisticated way to segment an image into spatially uniform regions. In the following, the idea of boundary-based Markov Random Field is introduced briefly. First, we define an energy function $E(f, l|d)$ as follows:

$$E(f, l|d) = \frac{1}{2}\sum_i (f_i - d_i)^2 + \lambda \sum_i (1 - l_i)(f_{i+1} - f_i)^2 + \theta \sum_i l_i \tag{3.2}$$

where d is the intensity process vector representing the observed image line. Each intensity value d_i is supposed to include some noise. f is the estimated value vector. l is called line process. l_i represents the discontinuity (edge) between the ith pixel and pixel $i + 1$. It is 1 if it is a boundary, and 0 otherwise.

The first term of equation 3.2 is for data fitting and tries to minimize the error of estimation. The second term is for smoothness in space. While there is no boundary specified by the line process l_i, it tries to minimize the difference between conjunct pixels f_i and f_{i+1}. When the line process l_i is 1, i.e. there is a boundary; then no constraint between the conjunct pixels is introduced. The third term of equation 3.2 is a constraint on the number of boundaries. This means there should be fewer boundaries in the image than number of pixels. In order to minimize the energy function 3.2, a hill-climbing method is used and derivatives of f_i and l_i are introduced:

$$\frac{\partial f_i}{\partial t} = \lambda\{(1 - l_{i-1})(f_{i-1} - f_i) + (1 - l_i)(f_{i+1} - f_i)\} - (f_i - d_i) \tag{3.3}$$

$$\frac{\partial l_i}{\partial t} = -l_i + H(\frac{\lambda}{2}(f_{i+1} - f_i)^2 - \theta) \tag{3.4}$$

where $H(\cdot)$ is a step function. Each parameter is updated with the above derivative iteratively until it reaches convergence.

After obtaining the segmentation of the image, the mean value of each segment is calculated and used for color classification.

3.1.3.2 Color classification

A probabilistic method based on Mahalanobis distances is applied to classify colors. For each color, a Gaussian model of the color distribution must be provided beforehand. It consists of a mean vector and a covariance matrix in YUV space. The Mahalanobis distances are calculated by the mean color value of each segment and the color distribution models. To illustrate this, one reference color is assumed to be a distribution with mean $\mu = (\mu_y, \mu_u, \mu_v)$ and covariance matrix Σ. The Mahalanobis distance between a color value $x = (x_y, x_u, x_v)$ and this distribution is defined as:

$$D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{3.5}$$

Each segment is associated to the reference color with the minimal Mahalanobis distance to the segment's mean value, provided the distance is below a predefined threshold. Tuning this threshold value allows one to influence the ratio between unidentified pixels and false positives.

3.1.3.3 Camera auto calibration on the VolksBot

Changing light conditions cause problems for color vision, especially when segmenting certain predefined colors inside the color-space via thresholding, as the distribution inside the color space varies. In this approach, the variation of the distribution is stabilized by controlling the respective intrinsic camera parameters such as Brightness, Exposure, Saturation and White Balance. This can of course only be done within the physical limits of the respective camera sensor. As depicted in figure 3.4, the system consists of a cone made up of reference color rings including white, black and red. The cone is placed over the camera and partially covers the previously unused center area inside the camera image. The cone has a slope of 45°, to average the incident light on the reference rings to account for multiple light sources from different directions.

Fig. 3.5 shows the *ICONNECT* signal graph used for the calibration process. For each of the three reference areas in the camera image, the median of the respective YUV values is determined (Modules: White, Red and Black). This data is casted and forwarded to five PI-Controllers (Module: PI Control), working in parallel, responsible for setting the respective camera parameters: Exposure, Brightness, Whitebalance U, Whitebalance V, and Saturation. For each of these controllers, an optimal set of desired values (Y-Des, U-des, V-des, Bright-des, and Satur-des) has to be determined once. These values depend on the characteristics of the used camera.

In some experiments, the Gain parameter was also controlled using the Y-channel of the white reference area as control input. This results in a much wider working range regarding the intensity of illumination. But, as Gain uses the same actual value as exposure, a dependency between these control variables occurs. Furthermore, higher Gain values result in higher signal noise. The desired values for the camera parameter

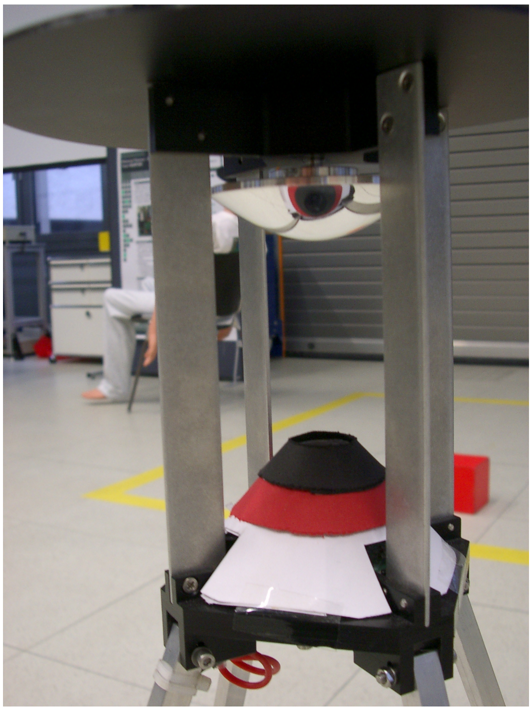


Figure 3.4: Catadioptric camera system IAISVision with reference color calibration cone

control have to be determined once, in the same way as is done when calibrating the camera manually, e.g. by analyzing at the distribution of the YUV-color space or the camera image. After that, the controller can be run in parallel to any other control graph. At the technical challenge of the RoboCup Middle Size League in 2004, I presented this approach on a *VolksBot* robot. The robot was constantly following a ball: inside the well illuminated Middle Size field, outside the field and outside the building in bright sunlight without any human interference. My team won second place in this challenge. In the following experiments, a different camera was used due to improved image quality, but the described method remains similar. Flat rings of white and red color were put around the lens of a Sony DSW-500 camera replacing the cone. Instead of Exposure, Iris is being controlled due to a different internal camera parameter model of the new camera. The control of White Balance and Saturation remain unchanged. The parameters of each PI controller can be tuned iteratively by observing and analyzing the step response switching from dark to bright illumination, from bright to dark and between different types of illumination. Fig. 3.6 shows a step response with optimized control parameters for the Brightness controller, switching illumination from dark to bright. In this example, the controller only operates in every fifth step. After PI parameter optimization, an appropriate desired value for each

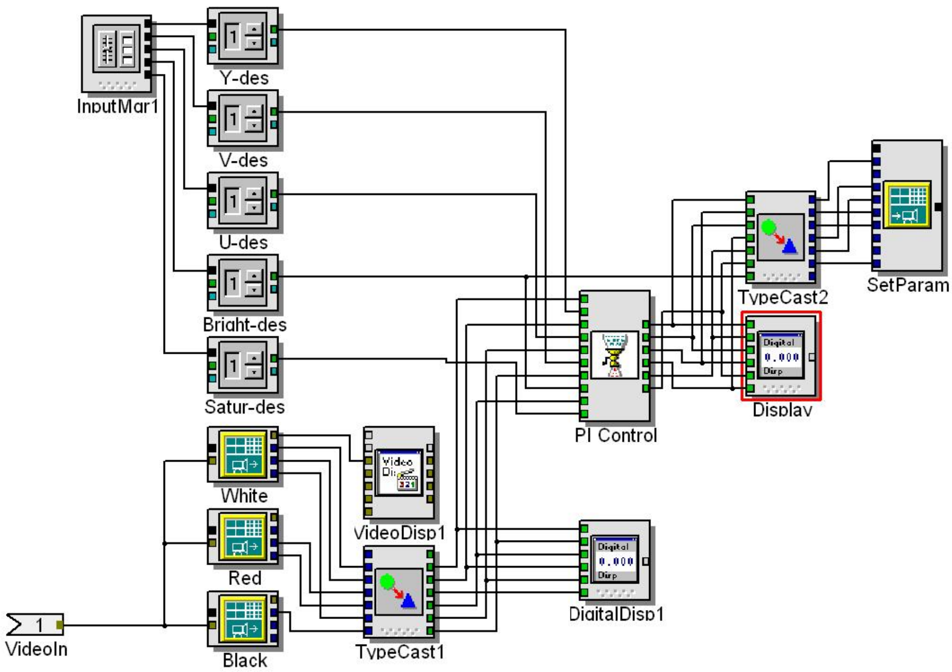


Figure 3.5: Iconnect signal graph for camera parameter control

controller has to be determined. This can be done by visual inspection of the YUV color distribution, the same way as doing a manual calibration of the camera. An example visualization of the camera color distribution is given in Fig. 3.7. In a distribution optimal for color classification, the colors should be widely spread in the color space. On the other hand, colors should not be over saturated, i.e. the distribution should not reach the borders of the YUV cube. Furthermore, the center of the distribution should lie in the center of the YUV cube.

3.1.4 Experiments with VolksBot

To evaluate the performance of this approach, several experiments in indoor and outdoor environments under different light conditions were conducted. The experiments were carried out on a *VolksBot* robot with a modified catadioptric *IAISVision* camera system. Processing was done on an on-board notebook with a Pentium M 1.8GHz processor. The complete vision processing takes less then 20ms for one image, depending on the number and sizes of recognized color regions. Thus, the algorithm works in real time.

The vision system consists of a Sony DSW-500 camera facing a hyperbolic mirror, thus producing 360° panoramic YUV images. A ring of white and red paper is fixed

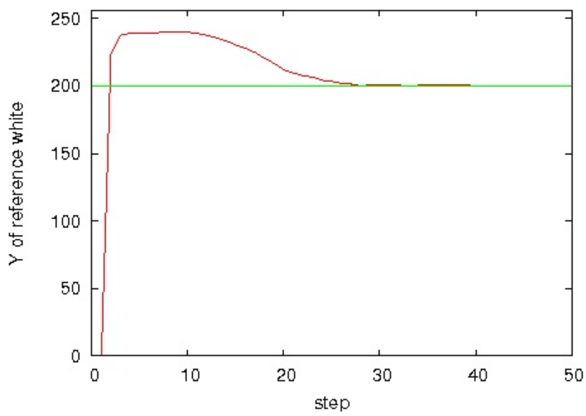


Figure 3.6: Step response with optimized control parameters for the brightness controller

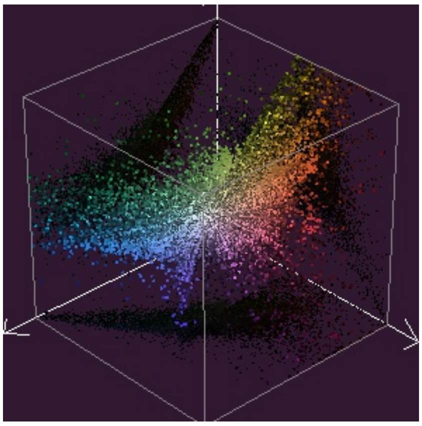


Figure 3.7: Example color distribution of a camera image in the YUV space

around the camera lens, see Fig. 3.3 left. This ring provides the reference colors used by the camera parameter controller without interfering with the view of the scene itself.

The colored objects used for color classification are mainly taken from the RoboCup scenario, in particular blue and yellow goals, a green field with white lines, cyan and magenta markers, a red ball and black robots. For outdoor tests, a subset was used.

To account for a broad range of light conditions, the following situations are considered:

1. Indoor: only artificial light of one light source (630 Lux)

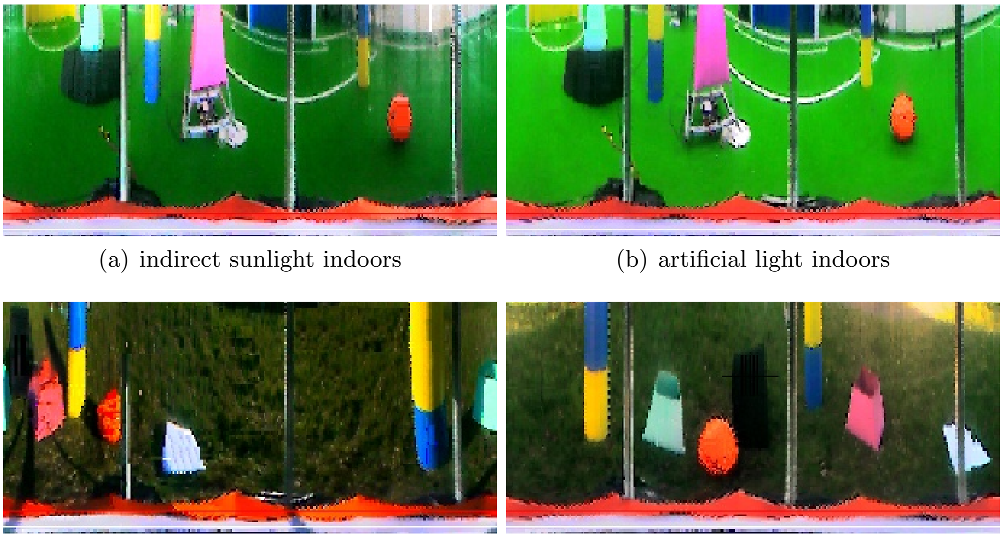
2. Indoor: mixed artificial and indirect sun light (1370 Lux)

3. Indoor: only indirect sun light (500 Lux)

4. Outdoor: camera and objects in direct sun light (97,000 Lux)

5. Outdoor: camera and objects in shadow (2,550 Lux)

Fig. 3.8 shows the camera images with active PI control under these different light conditions.



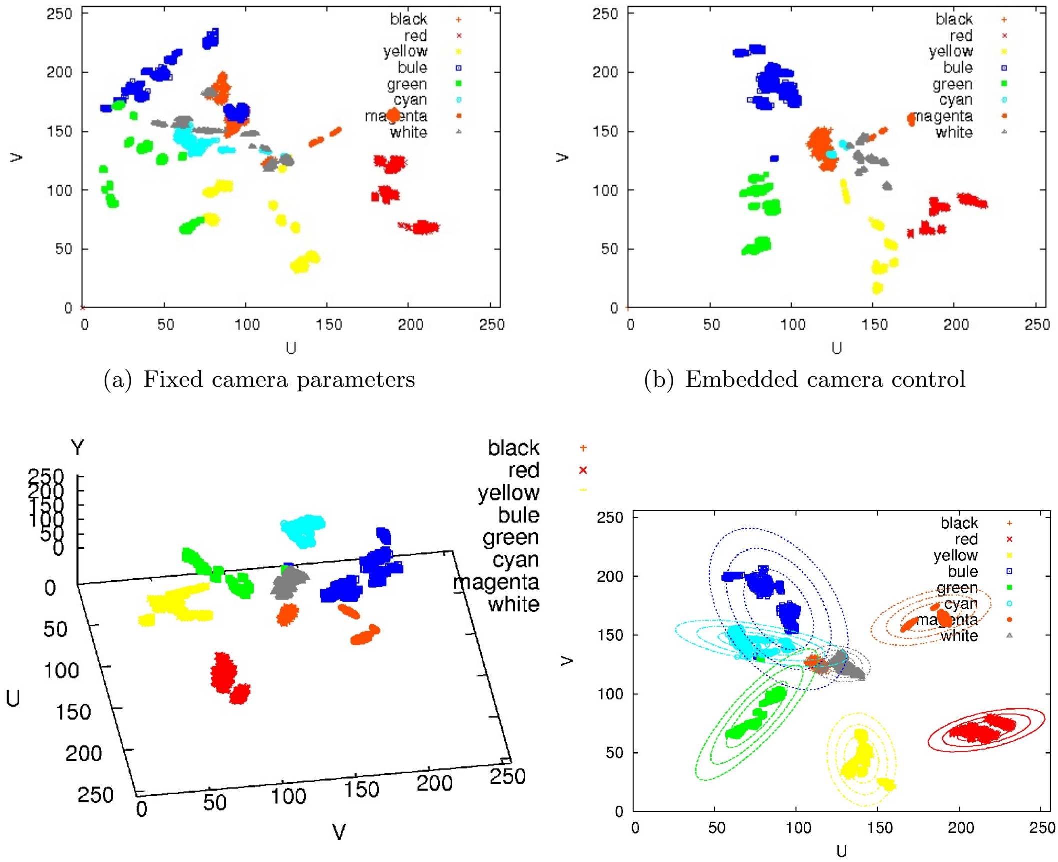
(a) indirect sunlight indoors
(b) artificial light indoors
(c) sunshine outdoors
(d) shadow outdoors

Figure 3.8: Captured panoramic images with active PI control under various light conditions

3.1.4.1 Color constancy

Fig. 3.9 shows the merged distributions of YUV values obtained from the color objects under the three different indoor light conditions. Fig. 3.9a and Fig. 3.9b show the YUV color distribution as 2D-projection on the UV-plane of the color space with fixed camera parameters and with the embedded auto control feature of the camera enabled. Fig. 3.9a shows the 3D-view and Fig. 3.9b the 2D-projection on the UV-plane of the color space using the PI control on reference colors of the camera parameters.

One can observe that the color drift is greatly reduced when applying the PI controller, while the colors drift heavily for the other two approaches. Not using PI control, the drift can be so big that previously separated color distributions overlap, making it impossible to deduce a unique color class from one YUV value. Also with the PI control the colors significantly drift depending on changes of direction of illumination, changes of intensity, changes of the ratio of different kinds of illumination or reflections. Still, the PI control provides better robustness and optimized spreading of the distributions



(a) Fixed camera parameters (b) Embedded camera control

(c) 3D-View with PI control (d) UV-plane with PI control

Figure 3.9: Plots of YUV color distribution indoors

compared to the other approaches evaluated. The system provides highest color constancy when both the object and the reference colors rings are exposed to equal light conditions.

The biggest change in color value occurs without any parameter control. It is interesting that not only the brightness Y, but also U and V change when illumination intensity decreases. This indicates that a simple brightness normalization is not enough to identify colors robustly, giving reason to also control the saturation value of the camera. Table 3.1 lists mean values μ and standard deviations σ in the direction of Y, U and V axis for three object colors under diverse light conditions with different control methods for the indoor and outdoor experiments. Comparing the standard deviations of the different approaches for a certain color, like e.g. red, the lower drift of the PI control method can be confirmed. It is apparent that the standard deviation with PI control is almost always smaller than for the others.

	Indoor									Outdoor		
	PI			No PI			Embedded			PI		
red μ	127.9	216.8	69.0	133.2	179.6	87.9	162.2	194.1	81.1	134.9	213.6	60.6
red σ	15.6	8.9	4.6	44.6	53.2	33.2	17.7	12.6	10.8	8.8	6.3	10.1
yellow μ	187.0	140.0	44.0	206.7	118.4	81.6	219.1	148.9	51.9	189.5	163.5	24.1
yellow σ	26.0	5.2	9.8	34.8	21.3	34.1	20.5	10.1	30.9	16.3	7.3	11.5
blue μ	63.1	89.2	172.1	98.6	64.3	191.6	101.4	89.8	192.0	83.2	84.1	186.1
blue σ	24.1	5.2	9.8	43.8	26.8	23.1	32.2	8.7	14.0	22.8	21.6	23.8

Table 3.1: Means $\mu_{y,u,v}$ and standard deviations $\sigma_{y,u,v}$ of typical colors in YUV space under various light condition in indoor/outdoor environment

The conditions change drastically when going from indoor to natural light conditions outdoors. The image in Fig. 3.10 shows the YUV distributions of the object colors and their projection into the UV-plane for the outdoor experiment in direct sunshine and shadow. The reason for the observable higher color drift lies in having a huge intensity

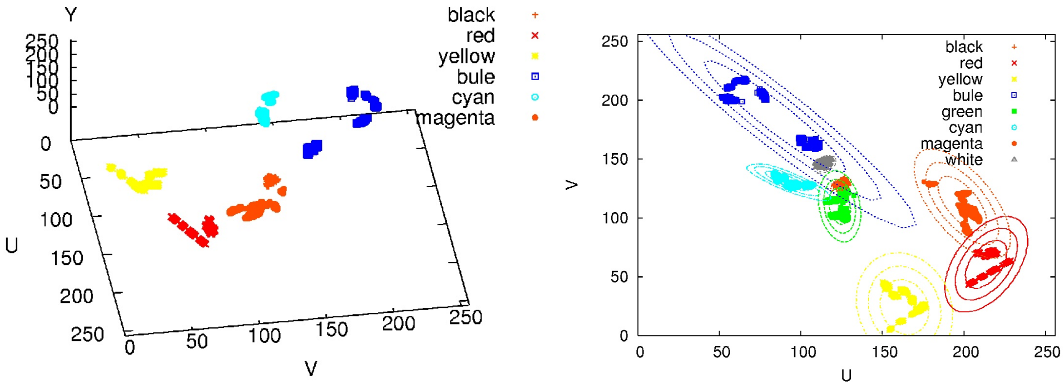


Figure 3.10: YUV color distributions in outdoor environment

range from 2550 to 97,000 Lux between shadow and direct sunlight.

Especially in the experiment undertaken in direct sunlight, these extreme illumination ranges occur in a single scene, having the same objects partly exposed to direct sunlight and partly lying in its own shadow. Furthermore, the drift in color space is highly dependent on the pose of the objects relative to the light source and to the camera. Also surface properties of the objects have a bigger influence here. Related to this huge illumination range, one can also see the need for the saturation control, as saturation of an object color decreases for dark and bright situations significantly.

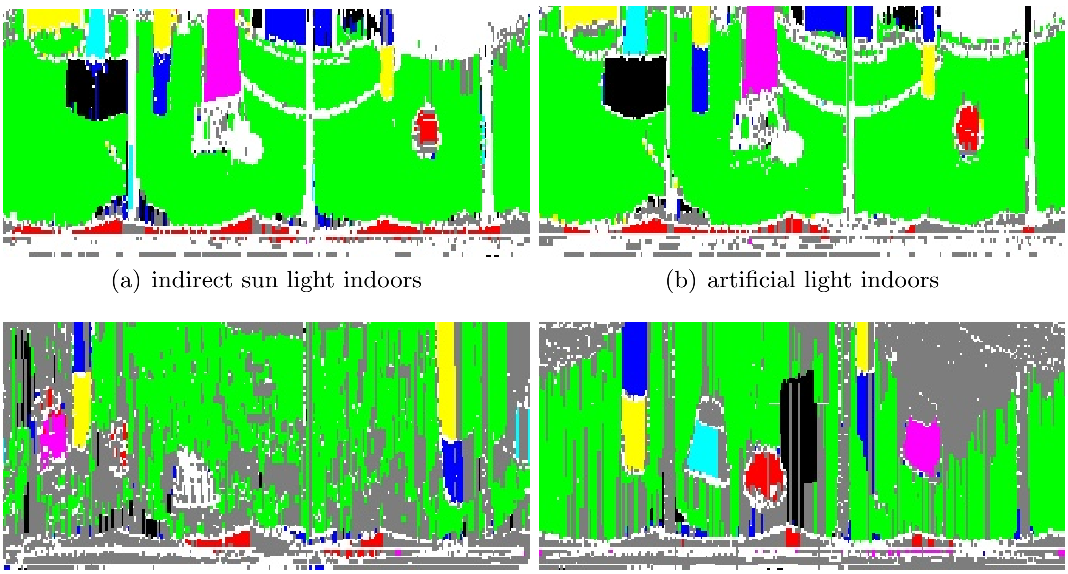
The red color, for example, has a much lower saturation value V when the camera is outdoors. It appears that it is not the kind of illumination, but the high intensity and the limited color range of the camera sensor that is responsible for this effect. The color is much brighter outdoors; since the YUV space is of conical shape, this results in a lower range of possible saturation values.

Still, the color distributions do not overlap, which indicates that a correct classification of colors should be possible.

3.1.4.2 Classification results

First, we have a look at the mean values and standard deviations of the reference color distributions, since these form the basis for the color classification step. In Fig. 3.9 lower right and Fig. 3.10 right, these regions are drawn as ellipses around the distribution of the respective colors. The images show the projection of the 3-dimensional ellipsoids on the UV-plane. The drawn ellipses represent the borders of 2-σ, 3-σ and 4-σ areas. Since the ellipsoids differ in the Y-values they cover, they do actually not overlap in the way the image of their projections may suggest.

The drawing of the ellipsoids indicates which thresholds to use to retrieve a binary classification result. Since the majority of pre-measured color pixels should be included, at least 3 σ seems reasonable. For a more robust identification towards unexpected light variations, a higher value could be useful. But as this can result in more false positive classifications, a compromise must be found. For the following classification experiments, a threshold of 3 σ was chosen.



(a) indirect sun light indoors (b) artificial light indoors (c) sunshine outdoors (d) shadow outdoors

Figure 3.11: Classification results with PI controller in indoor and outdoor environments

Fig. 3.11 shows the classification results in multiple light conditions. In general for all situations the classification algorithm shows good performance. In the indoor environment all objects are recognized with their correct colors, and only very few false positive classifications exist. In the outdoor environment the method has problems with very dark pixels resulting from the high differences in intensities due to sunlight and shadow.

To summarize the results, the PI controller provided enough color constancy to be able to fuse the distribution under different light conditions and to generate reference color models for indoor and outdoor situations. These reference color models have shown to provide a robust basis for color classification under a variety of different light conditions. The big difference of color distribution in indoor and outdoor suggest the use of separate reference models for these two cases. The vast illumination range occurring outdoors within one image has shown the physical limitations of the camera.

3.2 A 3D-LIDAR system for VolksBot and autonomous vehicles

Laser range finders (LIDARs) have already proven to be a highly relevant sensor technology for mobile robotics and autonomous driving in the past. The main reasons for this are the high accuracy, the adjustable and possibly high measuring range and the robustness towards various environment conditions compared to other approaches like stereo-vision, radar or sonar. Still, some drawbacks exist as surface- and perspective-dependent errors in measuring results may occur. This is the reason why LIDAR data is often used in combination with range information from other sensors, and their data is fused using probabilistic approaches to reduce the error [85]. A wide range of 2D-LIDARs is commercially available; a very common one is e.g. SICK LMS-200[1]. Recent developments by Hokuyo[2] offer miniaturized and light-weight alternatives with measuring ranges up to 30m. IBEO[3] focuses directly on driver assistance and aims for wide-range and low-cost LIDARs specifically for the automotive domain.

In many robotic applications and environments, 2D-sensing does not provide sufficient information, as it greatly limits the field of perception. World-modeling in 3D on the basis of LIDAR data has been a research track for years. Thrun et al. [86] used two 2D-LIDARs fixed on the robot and shifted by 90° to generate 3D-information by use of odometry while the robot is driving.

[1]SICK LMS-200: http://www.sick.com
[2]Hokuyo Laser range finders: http://www.hokuyo-aut.jp
[3]IBEO LIDARs: http://www.ibeo-as.de

Besides approaches using fixed 2D-LIDARs and ego-motion, 3D-LIDARs provide a 3D point cloud instantaneously without having to move the sensor. These systems can be divided into scanners that use an additional vertical mirror to acquire 3D data [87] and systems that use a mechanical arrangement to rotate an array of lasers or a factory-build 2D-LIDAR [88]. [89] extends a standard 2D laser range finder by a low-cost tilting module based on a servo motor. This *3DLS* LIDAR acts as the basis for the development of the presented *3DLS-K* system.

Velodyne[1] developed a high definition 3D-LIDAR *HDL-64E* which is currently being used by many research groups for autonomous driving. The *HDL-64E* rotates an array of fixed point laser units measuring the distances by use of the *Time-of-Flight* principle. A total of 64 laser units are mounted on upper and lower blocks, and the entire unit spins at velocities up to 900 RPM. The lasers are employed with each laser/detector pair aligned at predetermined vertical angles, resulting in a 26.8° vertical field of view with a vertical resolution of 0.4°. The horizontal field of view is 360°. The rotation velocity is user selectable between 300 and 900 RPM resulting in maximum horizontal resolution up to 0.09°. The *HDL-64E* provides about 1.3 million points per second by 100 Mbps Ethernet interface.

3.2.1 Hardware development of 3DLS-K

On the basis of the experiences gained with the previously mentioned *3DLS* LIDAR system, in 2007, I conducted the development of the *Fraunhofer 3DLS-K* LIDAR system as sensor component for the *VolksBot* and for autonomous urban driving. Two industrial laser range finders rotate around the vertical axis of the system, acquiring depth and intensity information for a 360° field of view (see Fig. 3.12).

The system consists of two *SICK LMS 291-S05* laser range finders mounted on an angular adjuster plate. The scanners have an apex angle of 180° and a resolution of 1 to 0.25°. Depending on the resolution, the response time to acquire one two-dimensional scan is from 13 up to 53ms. The maximum scan range is 80m. Being able to adjust the roll-angle of the scan planes allows one to increase the scan resolution or to increase the rotation speed while cutting off the usually irrelevant top and the lower part of the scanned sphere. The adjuster plate continuously rotates around the z-Axis being driven by a brushless *Maxon flat EC45 50W* motor[2] via a 89:1 gear. Three hall sensors acquire the position feedback from the motor sent to the *Maxon EPOS P 24/5* motor controller. The *Maxon EPOS P* contains an integrated micro controller for

[1]Velodyne HD LIDAR http://www.velodyne.com/lidar
[2]Maxon Motors: http://www.maxonmotor.com

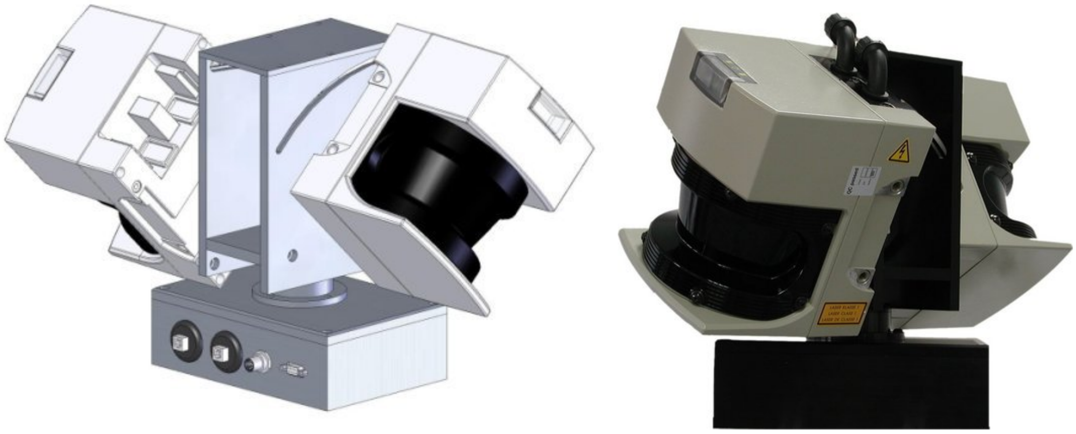


Figure 3.12: CAD drawing and image of the Fraunhofer 3DLS-K scanner

speed control and positioning, operating at 33 MHz. It is equipped with an RS232 and a CAN interface, inputs for the hall sensors and some general purpose digital inputs. The firmware of the motor controller is programmed in the IEC programming languages, which starts the motor after power-on and sets a predefined velocity. Communication with the firmware program is done via RS232 to set the velocity and to receive the motor position information.

Sliding contact rings mounted on the shaft are used to power the laser scanners with 24VDC, 0.9A each and to transmit the sensor data via RS485 from the scanners to the 2 USB plugs in the control box. For this, two RS485 to USB converters are integrated. Rotation speed can be set via an RS232 which also gives feedback from an inductive proximity switch which is used to obtain the precise zero position of the scanner rotation. Figure 3.13 shows the internal layout of the *3DLS-K* control box. The entire system is IP65 water resistant, weighs 13kg and has a size of 290x330x250mm. Power consumption is 2.2A at 24V DC. The scan resolution depends on the rotation speed and the angular adjustment of the scanners. At an angular adjustment of 60° and a rotation of 0.45Hz, a vertical resolution of 0.5° and a horizontal resolution of 1.7° with an update rate of 0.9Hz is obtained. An external control PC is used for post-processing the LIDAR data. Communication with the internal motor controller is done via an RS232 interface, and the data from the LIDARs is acquired via USB. The control PC constantly polls the 2D scanners for new scans and uses the position of the mounting plate shaft to allocate a two-dimensional scan slice to the global coordinate frame. The generation of 3D scans is explained in detail in the following subsection.

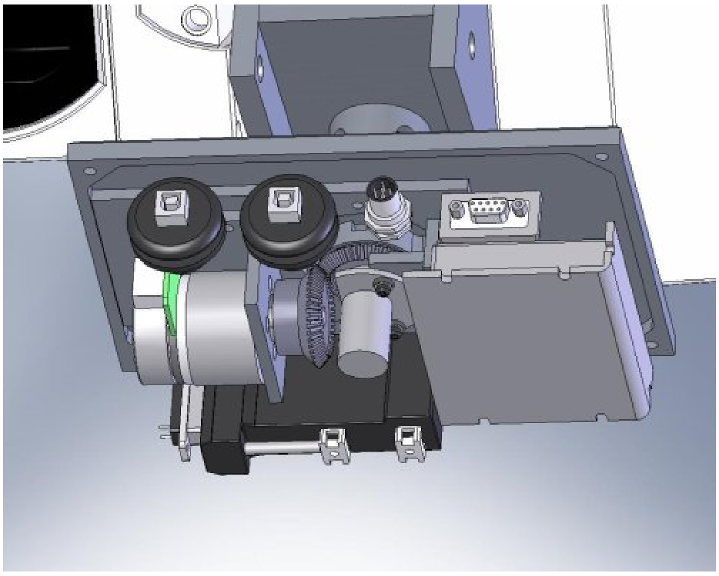


Figure 3.13: Internal layout of the 3DLS-K control box

3.2.2 Generation of 3D scans

While the two scanners rotate, range data is acquired continuously. The application software on the control PC receives 2D scans in the form of arrays of double values represented in polar coordinates, where the value of an element represents the distance and its index represents the angle in degree. Transforming from Polar to Cartesian coordinates and adding an offset to the rotation center, the 2D Cartesian coordinates in the scan plane in cm can be obtained:

$$x = d \cdot sin(\alpha) + 10y = d \cdot cos(\alpha) \tag{3.6}$$

, with α being the index of the array.

With an angular adjustment of the scanners roll-angle θ (in this case $\theta = 60°$), the 2D points in the scan plane have to be transformed into 3D points, taking into account the rotation of θ about the x-axis.

$$\begin{array}{lcrcl} x' & = & x & = & d \cdot sin(\alpha) + 10 \\ y' & = & -y \cdot sin(\theta) & = & -d \cdot cos(\alpha) \cdot sin(\theta) \\ z' & = & y \cdot cos(\theta) & = & d \cdot cos(\alpha) \cdot cos(\theta) \end{array} \tag{3.7}$$

The next transformation is addressing the continuous rotation of the scanners with yaw-angle β about the z-axis. β is the absolute angle of the shaft that rotates the adjuster plate with the scanners mounted on it.

$$\begin{array}{lcl} x'' = x' \cdot cos(\beta) - y' \cdot sin(\beta) & = & (d \cdot sin(\alpha) + 10) \cdot cos(\beta) + d \cdot cos(\alpha) \cdot sin(\theta) \cdot sin(\beta) \\ y'' = x' \cdot sin(\beta) + y' \cdot cos(\beta) & = & (d \cdot sin(\alpha) + 10) \cdot sin(\beta) - d \cdot cos(\alpha) \cdot sin(\theta) \cdot cos(\beta) \\ z'' = z' & = & d \cdot cos(\alpha) \cdot cos(\theta) \end{array} \tag{3.8}$$

Due to the continuous rotation of the scanners, the yaw-angle β is not constant during a single 2D scan. Therefore β needs to be determined precisely for each element of a 2D scan in order to avoid distortion. β is determined by combination of the internal hall encoders of the EC motor, which provide a relative position information, and an additional inductive proximity switch at the shaft, which defines an absolute zero position for each revolution. The resolution of the rotation is determined by the number of hall sensors and the number of pole pairs. The Maxon EC Flat Motor hosts three hall sensors and 8 pole pairs which results in a resolution of $\frac{1}{24}$ per revolution of the motor shaft. Multiplied by the motor gear ratio of $\frac{1}{89}$, this leads to a resolution of $\frac{1}{2136}$ per revolution of the scanners which corresponds to a horizontal resolution of 0.17°. This resolution is sufficient to construct the 3D scan. An example of a resulting 3D scan with *3DLS-K* mounted on the roof of a car is shown in figure 3.14. The distance values are color coded. Figure 3.15 shows the integration of the *3DLS-K* on a *VolksBot* robot.

Figure 3.14: A resulting 3D scan from 3DLS-K with color coded distance values

Figure 3.15: 3DLS-K as *VolksBot* sensor

3.2.3 Classification of flat terrain with 3DLS-K

The first application of the 3DLS-K scanner was to classify flat terrain such as roads that an autonomous robot could drive on. Detecting edges in range images is a common approach of addressing this problem. The *Scan Line Approximation* algorithm by Jiang and Burke [44] provides edge strength measures that have a geometric interpretation and supports a classification of edges into several subtypes which appeared to be a viable method for the given problem. It is based on the approximation of a scan line by a set of bivariate polynomials.

$$f(x,y) = \sum_{i+j\leq k} a_{ij} x^i y^j \qquad (3.9)$$

where f is an approximated polynomial function with $k = 2$. Every scan line is considered two-dimensional curve and partitioned into curve segments using an algorithm described by Duda and Hart [90]. Based on the midpoint and the two endpoints, a scan line is described by an approximation function. Then, the largest error e_{max} between the approximation function and the scan line is calculated. If this error exceeds a preselected threshold ϵ, the scan line is split into two parts at the location where e_{max} occurs. The splitting algorithm proceeds recursive until the approximation error e_{max} does not exceed the threshold ϵ. This single parameter ϵ controls the approximation accuracy.

Being able to cope with sensor noise is an important feature for the applicability of edge detection algorithms. The *Scan line approximation* algorithm detects noise by counting the number of points of the curve segment. Typical noise results in a curve segment with only one point. This is addressed by merging this point into the adjacent curve segments. With the resulting curve segments, edges can be detected and classified. Only the end points (x_1, x_2) of curve segments are considered potential edge points. For each potential edge point x_1 from the curve segment c_1, a *discontinuity strength* is defined: Let x_2 be the end point of the adjacent curve segment c_2 and $f_{1,2}(x)$ the function of the two curve segments c_1 and c_2. Then a suitable discontinuity strength for edges is given by

$$|f_1(\bar{x}) - f_2(\bar{x})|, \tag{3.10}$$

where $\bar{x}$ is the midpoint $\bar{x} = (x_1 + x_2)/2$. This describes the *jump edge* classifier by Jiang and Burke. Other classifiers are *crease-*, *convex-* and *concave edges* [44] which could be applied if required.

Fig. 3.16 shows the application of the *Scan Line Approximation* algorithm for the classification of flat terrain. Before applying the algorithm, occluded areas caused by the scanner and the platform itself have to be discarded in the scan. Assuming that the platform is on drivable road, edge detection is started from the origin of the scanner. If an edge is detected, the rest of the scan is classified as non-road. In the figure, points classified as road are marked in blue. Most of the road is correctly classified even if there are some false detections (figure 3.16 top) at the curb. These false detections could be filtered by discarding detected line segments which are too short. For these first experiments no special edge classifications and no filtering on edge strength has been applied, which is expected to further improve the results.

3.2.4 Application to autonomous urban driving

In 2007, our team *Team Berlin* [32] participated in the DARPA Urban Challenge[1] with the autonomous car *Spirit of Berlin*. In order to drive autonomously in an urban environment, one of the many tasks was to master intersection situations. Here, the vehicle has to decide if it can pass an intersection or has to yield to other cars according to official traffic rules. In order to make this decision, the system needs to perceive the environment - detect the road, obstacles and other cars. For the *Site Visit*, a qualification event of the DARPA Urban Challenge, the Fraunhofer *3DLS-K* LIDAR was successfully integrated as sensor for the *Spirit of Berlin* and used for the intersection

[1] DARPA Grand Challenge Website: http://www.darpa.mil/grandchallenge

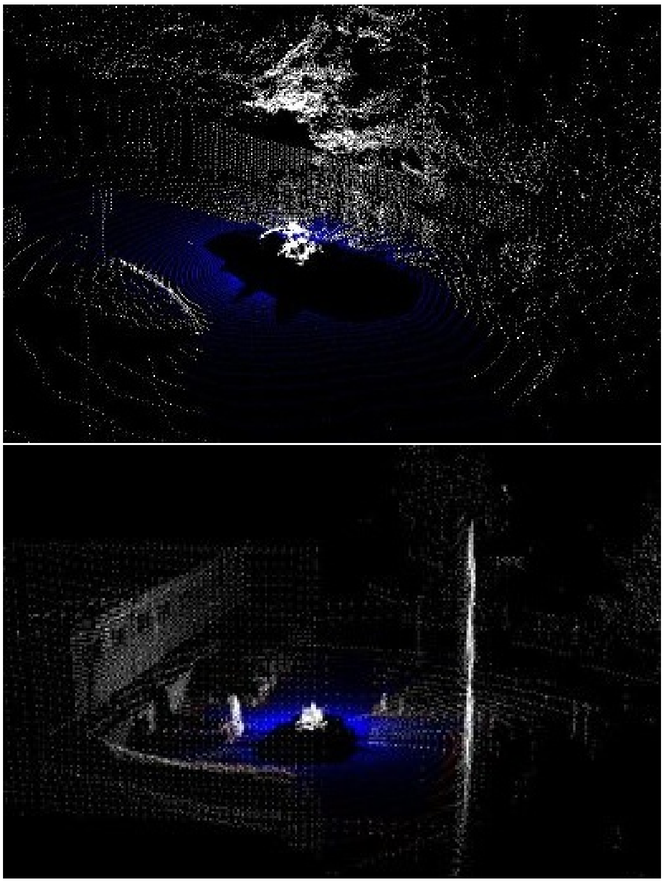


Figure 3.16: Classification results of flat terrain in 3D scans

analysis (see Fig. 3.17). Later on, the system was replaced by a *Velodyne HDL-64E* LIDAR system as it provided a higher resolution and update rate, which is essential especially when driving at higher speeds. Still, due to its modular architecture, the original software for the intersection analysis could be easily adapted to and used with the new sensor. The processing steps of the intersection analysis on the basis of 3D-LIDAR data are described in the following.

3.2.4.1 Obstacle detection

As a first step, the Cartesian sensor data of the 3D-LIDAR system is processed in order to extract points which are part of an obstacle. Here, a heuristic assumption is used, which is based on a geometrical property of flat roads: adjacent scan points on the road have nearly the same z-coordinates. Therefore neighboring points in the x-y plane that show a big difference in the z-coordinate indicate the presence of an obstacle. Because a direct comparison of all scan points with each other would be computationally very expensive, all scan points are first sorted in a map. This map is a 2D-grid in the

Figure 3.17: Spirit of Berlin with integrated Fraunhofer 3DLS-K LIDAR

x-y plane. With this, the associated grid cell can be accessed efficiently by the x-y coordinates of a 3D-point.

To compare the z-coordinate of point P with the z-coordinates of its neighbors, the mean $\bar{z_M}$ value over the z-coordinates of all points in P grid cell is calculated. The function $O(P_x, P_y, P_z) : \{0, 1\}$ determines if P is an obstacle point.

$$O(P_x, P_y, P_z) : \begin{cases} 1 & z_M(\bar{P_x}, P_y) \geq P_z * H \\ 0 & z_M(\bar{P_x}, P_y) < P_z * H \end{cases} \tag{3.11}$$

, with P_x, P_y, P_z being the metric x, y and z-coordinates in the car's coordinate system, and $z_M(\bar{P_x}, P_y)$ is the mean z-coordinate of the grid cell P is sorted in and H is a parameter that controls the responsiveness of this function to obstacles. If H is considered low, the number of points that are defined as obstacles is high. In order to filter false positives, for example possible obstacle points caused by a curb, H was parameterized between 50cm and 100cm. The result of this filtering method is depicted in figure 3.18. The figure shows a 3D scan from the Velodyne laser scanner. Points that are determined as *obstacle points* are marked red. The parameter H in figure 3.18 was set to 70cm. One can see that the edges of the approaching car are marked as *obstacle points* as well as edges of buildings, pillars and trees. The number of *obstacle points* is not very high. The approaching car results in about 50 *obstacle points* and decreases to 10 *obstacle points* if the car is 30 meters away. Even if this number of *obstacles points* seems to be low in contrast to the number of points in one 3D-scan, the number of *obstacles points* of the car is very constant over several scans.

The number of points in a grid cell also depends on the size of the cell. Larger grid cells will result in a larger number of points in one cell and therefore fewer *obstacle points*

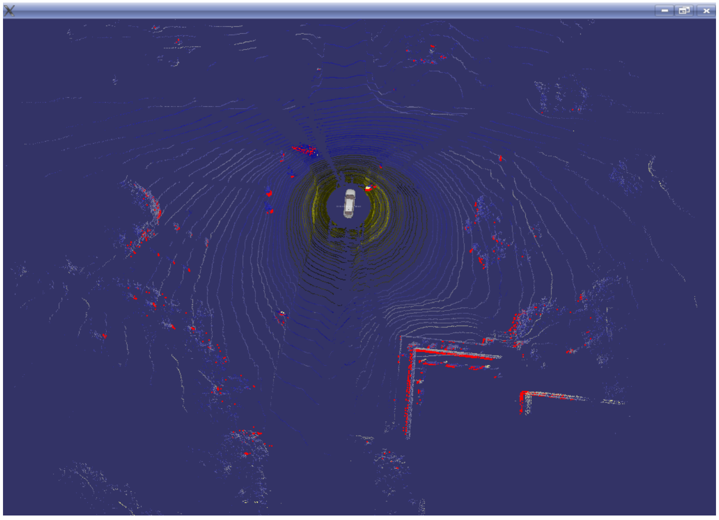


Figure 3.18: Detected obstacle points (red) in a 3D scan

in total. Smaller grid cells will result in a higher responsiveness of the *obstacle point* function (equation 3.11) and therefore increase the number of false positive *obstacle points*. Therefore, setting the dimension of the grid cells is a trade-off. A grid cell size of $10 \times 10cm$ showed the best results in the experiments.

3.2.4.2 Intersection structure

Intersections are defined in a Route Network Definition File (*RNDF*)[1] provided by DARPA. The RNDF defines a route network, which the vehicle has to use to complete a mission. The route network is defined as the set of accessible roads and areas in which an autonomous vehicle may travel. It specifies accessible road segments and provides information on way points, stop sign locations, lane widths, checkpoint locations and parking spot locations. One road segment consists of a set of lanes, and each lane consists of a set of way points. The RNDF provides world coordinates to each of these way points. Figure 3.19 shows a sample T-intersection which connects two road segments consisting of two lanes each. Way point F is the exit way point of lane 1.1 and therefore a possible entry point to the intersection. A car that approaches the intersection over lane 1.1 could either stay on lane 1.1 and pass the intersection by using the entry way point A, or it could change to lane 2.1 by using the lane entry way point C.

[1] RNDF: http://www.darpa.mil/grandchallenge/docs/RNDF_MDF_Formats_031407.pdf

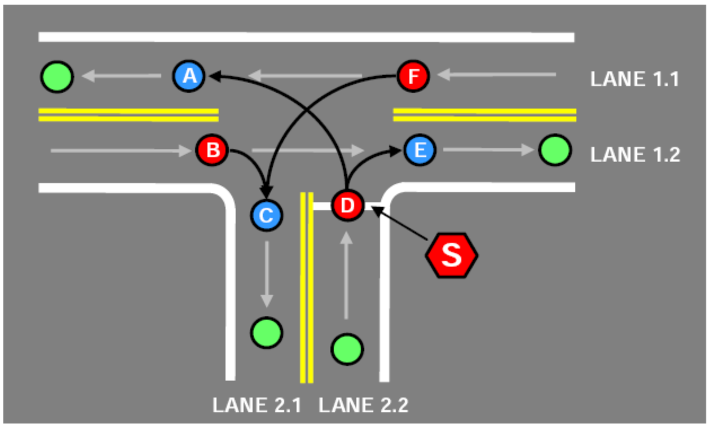


Figure 3.19: Example T-intersection of an RNDF file

3.2.4.3 Regions of interest of an intersection

To analyze the status of an intersection, i.e. to check if there is incoming traffic, *Regions of Interest* (ROI) are defined by use of bounding boxes. A bounding box is defined by the coordinate of the entry or exit way point x,y, by width w, length l and orientation α of the lane. The width of the bounding box is proportional to the lane width of the way points lane, which is provided in the RNDF. It was scaled by a factor of 0.8 to discard the curbs along the road. The length of the bounding box is set according to the type of way point. To determine if a point P is in this bounding box, the bounding box geometry and the point are transformed to the coordinate system origin. Therefore the point P is translated by the negative position vector of the way point W.

$$P = P - W \tag{3.12}$$

Then the point is rotated back against the orientation of the lane α

$$P = R^{-1} * P \tag{3.13}$$

where R^{-1} is the inverse rotation matrix

$$R^{-1} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}$$

.

After translating and rotating the point to the origin, relational operators can be used to determine if P is in the bounding box.

$$inRegion(P_x, P_y) : \begin{cases} 1 & (P_y < \frac{1}{2} * w) \quad \&\& \quad (P_y > -\frac{1}{2} * w) \\ & (P_x < l) \quad \&\& \quad (P_x \geq 0) \\ 0 & else \end{cases} . \tag{3.14}$$

where l and w are the length and width of the bounding box. Figure 3.20 shows the model of a four-way intersection with the regions of interest marked as rectangles. The red rectangles to the outside of the intersection label the described bounding boxes. The entry and exit way points are marked as white dots next to the rectangles. The yellow rectangles and the circle are used to determine if a car is in the center of the intersection. The yellow circle is defined by the center of gravity (white dot in the center of the circle) over all entry and exit way points. The radius of the center circle is the mean value of all distances from the center to the way points of the intersection, scaled by the factor 0.7. A point P is in the center circle, if

$$inCenterCircle(P_x, P_y) : \begin{cases} 1 & ||P|| < r \\ 0 & else \end{cases} . \tag{3.15}$$

where $||P||$ is the vector norm of P, and r is the radius of the center circle.

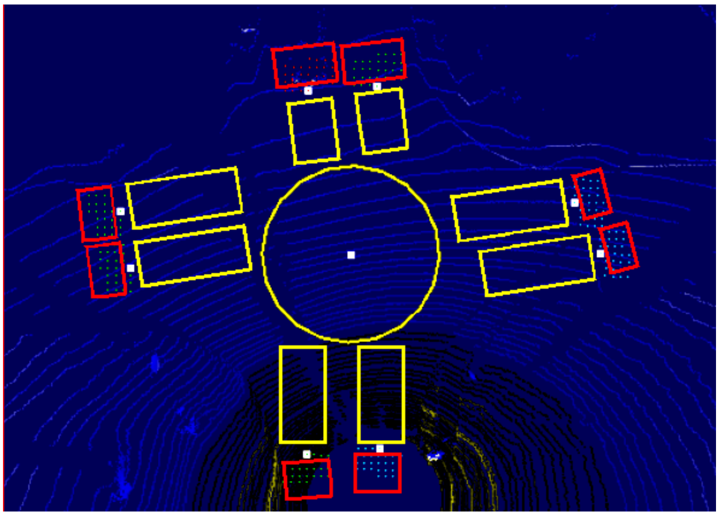


Figure 3.20: Regions of interest in a four-way intersection

3.2.5 Determining the ROI status

After defining the regions of interest in an intersection, the system has to determine the status of the regions. The status is either *busy* if an obstacle or car is in the region or *free* if not. In order to do so, the number of *obstacle points* in each region are determined. Identifying the ROI status is done by a simple function $B(t, n) = \{0, 1\}$ over the number of obstacle points n and the time t.

$$B(t, n) : \begin{cases} 1 & n \geq T(t) \\ 0 & else \end{cases} \tag{3.16}$$

where $T(t)$ is a linear saturated function, which was used instead of a fixed threshold. $T(t)$ is defined as $T(t) = regionThreshold$ for $t < 10$ and grows linear up to $T(t) = 4 * regionThreshold$. The constant $regionThreshold$ is a parameter of the intersection analysis. It was determined by experiments and chosen to be 10 *obstacle points*, which is the average number of *obstacle points* of a car at a distance of 30 meters. The reason for using this linear function as threshold is to further reduce false positive detection at far distances.

Figure 3.21 shows the resulting region status identification. The scene shows a car that approaches the intersection from the opposite lane. The red color of the region means that it is correctly identified as busy.

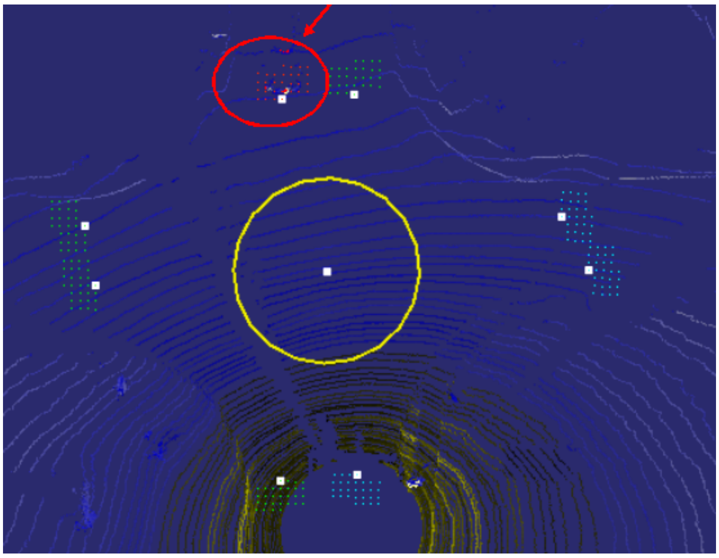


Figure 3.21: ROI status determination in a 3D scan

3.2.6 Intersection behavior

As a final step, an intersection behavior was implemented in order to decide whether the car can pass or has to yield for other cars. This behavior has to handle various kinds of intersection types, from simple four-way crossings to T-junctions, based on the Californian traffic rules. To implement this general *right of way* rule, a finite state machine was used. The state machine is depicted in figure 3.22. It has five states, eight transitions between the states and eight conditions. The conditions *leftTurnBusy*, *mainRoadBusy*, *peBusy* and *stopStillBusy* depend on the status of the *ROI*; *onMainRoad* and *leftTurn* are two special conditions which are dependent upon the ego state of the car. Lastly, there are two time-based conditions, *mainRoadTimerExceeded* and *crossingFreeTimerExceeded*.

The analysis starts in the *START* state, when it is invoked by the high-level behavior, and terminates by reaching the *CROSSING FREE* state. The *START* state is connected to the *INCOMING TRAFFIC* state by the condition *peBusy*. *peBusy* is true if one of the exit way points the system has to yield to is busy. This means if the car approaches an intersection and one exit point it has to yield is busy, the state machine changes to the state *INCOMING TRAFFIC*. As soon as the car at that exit point leaves the intersection, the condition *peBusy* changes to false.

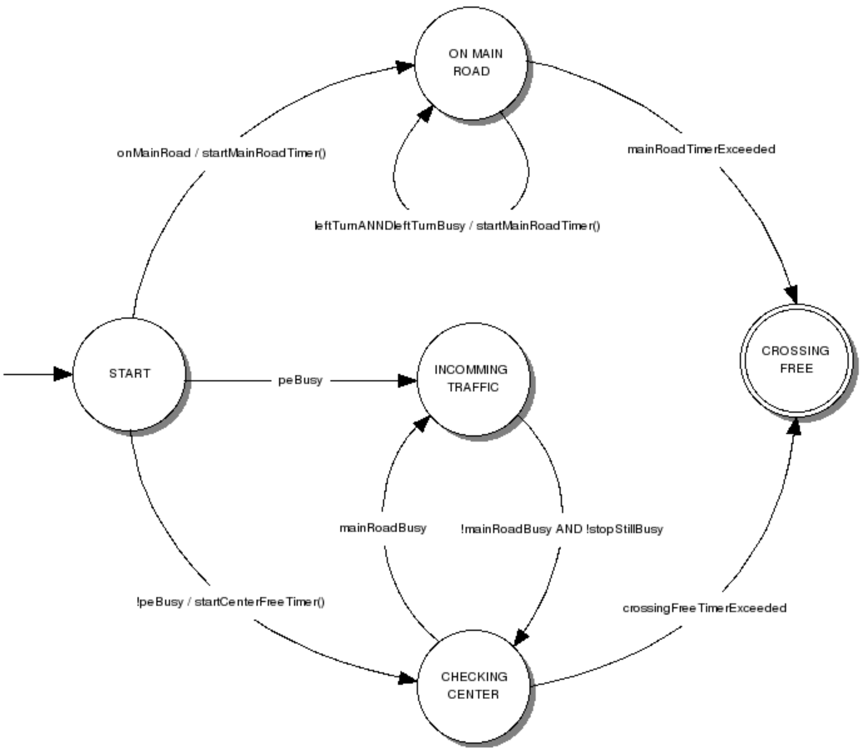


Figure 3.22: State machine containing the intersection behavior

The system was tested in simulation, on sensor traces and online on the *Spirit of Berlin* in various intersection situations with other participating cars. The final evaluation was done at the national qualification event and the semifinals of the *Darpa Urban Challenge* in Victorville, California, in 2007. Figure 3.23 shows the *Spirit of Berlin* facing a four-way intersection with three other cars during the semifinals. The system has shown to work correctly in many different cases including different types of intersections, different intersection geometries and different behaviors of the participating cars.

Figure 3.23: System evaluation at a four-way crossing during the semifinals of the DARPA Urban Challenge 2007

3.3 Summary and Conclusions

This chapter described the development and application of two sensor systems originally developed for the *VolksBot* construction kit, namely the *IAISVision* system and the 3D-LIDAR *3DLS-K*.

First, I presented the hardware development of the catadioptric vision system *IAISVision*, allowing for a wide range of system variants and applications. *IAISVision*, as well as its mirror, have been used by more than 30 research labs worldwide and in different robotic applications such as RoboCup Soccer(MSL), Robot rescue and surveillance or autonomous driving. Furthermore, I presented a robust color perception method for varying light conditions including PI control of camera parameters by use of reference colors, segmentation based on Markov Random Field and classification based on the Mahalanobis distance. The PI controller provided enough color constancy to be able to fuse the distribution under different light conditions and to generate reference color models for indoor and outdoor situations. These reference color models have shown to provide a robust basis for color classification under a variety of different light conditions. The big difference of color distribution in indoor and outdoor situations suggests the use of separate reference models for these two cases. The vast illumination range occurring outdoors within one image has shown the physical limitations of the camera. Future work will investigate the possible use of attention-based mechanisms to choose from different parameter sets for different light situations.

In the second half of this chapter, the system development and application of the 3D-LIDAR system Fraunhofer 3DLS-K was discussed. After detailing the hardware development of the sensor and the acquisition of 3D data, the sensor application to the domain of autonomous urban driving was presented. In particular, a method for the classification of flat terrain by use of the *Scan Line Approximation* algorithm and a method for scene analysis of the 3D data at intersections, including an according behavior system on the basis of a finite state machine, were developed. The system was tested in simulation, on sensor traces and on the autonomous car *Spirit of Berlin* in various intersection situations with other participating cars. The final evaluation was done at the national qualification event and the semifinals of the DARPA Urban Challenge in Victorville, California, in 2007, and the system worked correctly and reliably under various conditions. The approach and the results were presented at the workshop on 3D-Mapping at the IEEE International Conference on Intelligent Robots and Systems (IROS2008) [91].

The next chapter will provide insight into the conceptual design and implementation of the *RoboCup*@Home initiative, a standard benchmark and international competition for autonomous service robots, which aims to advance applied research and development in the domain.

4

RoboCup@Home: Advancing service robotics through benchmarking in scientific competitions

After having discussed approaches for application oriented robot prototyping and related sensor development, this chapter focuses on the problem of providing standard benchmarks allowing for quantitative performance evaluation of mobile service robots in real world settings.

Together with my former colleague Tijn van der Zant, I developed and initiated the *RoboCup*@Home competitions [92][64][93] in 2006 which has become the largest annual international service robotic competitions world-wide. Being part of the *RoboCup* initiative, the *RoboCup*@Home league targets the development and deployment of autonomous service and assistive robot technology being essential for future personal domestic applications. The domain of domestic service and assistive robotics implicates a wide range of possible problems. The primary reasons for this include the large amount of uncertainty in the dynamic and non-standardized environments of the real world, and the related human interaction. Furthermore, the application orientation requires a large effort towards high level integration combined with a demand for general robustness, ease of use and safety of the systems. This chapter details the need for interdisciplinary community effort to iteratively identify related problems, to define benchmarks, to test and, finally, to solve the problems. The concepts and the implementation of the *RoboCup*@Home initiative as a combination of scientific exchange, benchmark and competition is proposed as an efficient method to accelerate and direct technological and scientific progress in the domain of domestic service robots. Finally, the progress in terms of performance increase in the benchmarks and in technological advancements is evaluated and discussed.

4.1 Challenges in service robotics

The general idea of personal Domestic Service Robotics (DSR) has been around for a long time, but it is a comparably young research topic. The aim of creating useful, autonomous, multipurpose personal assistant robots which can interact with humans and objects in the real world in a natural way poses a large number of unsolved problems across many scientific disciplines.

There have been many successful and impressive demonstrations of robot technology in the past. In DSR, one focus (and one of the main difficulties) is the interaction with the real world, instead of operating under constrained settings and strictly defined environmental conditions as opposed to e.g. industrial robotics. These systems must cope with a large amount of uncertainty. A natural home environment, for example, is not specified in size, shape, appearance, the kind of objects contained in it, lighting and acoustic conditions, the kind and number of residents, etc. Furthermore, as objects and people can move, disappear and reappear, the environment is dynamic. A personal assistant robot must be able to manipulate objects in various locations and from different heights, and it needs to be capable of locomotion on different terrains. When interacting with humans, the system should possess some basic (social) intelligence and should be able to distinguish different people. Last but not least, safe and robust operation of these systems in such uncertain and dynamic environments is a fundamental requirement for their future acceptance and general applicability.

The creation of such autonomous systems requires the integration of a large set of abilities and technologies. Examples include human-robot interaction (speech, gesture, person, face recognition and person tracking, among others), navigation and mapping, reasoning, planning, behavior control, object recognition, object manipulation or tracking of objects. With regard to artificial intelligence, the systems should contain adaptive but robust behavior and planning methods, social intelligence, and learning capabilities. Intuitive programming methods (instead of entering computer code) are required for a broad acceptance and usability. Appropriate procedures should, for instance, enable the robot operator to teach new behaviors and environments via voice or gesture commands. As future households will most likely contain more intelligent electronic devices capable of communicating with each other, ambient intelligence, including the use of the Internet as a common knowledge base, will certainly play a more important role.

Just very recently, progress in these research fields, as well as progress and standardization in related hardware and software development, has led to an increase in availability of required methods and components for DSR. This includes the availability of software frameworks for robot control (e.g. Carmen [4], Player/Stage [5], MRPT[1], MRS[2]), simulation (e.g. USARSim [9]), and open source software libraries containing algorithms for computer vision (e.g. OpenCV[3]) with diverse applications as shown in [7]) or robot control (e.g. Orocos [6]). On the hardware side, robot construction kits like e.g. VolksBot [94] (see chapter 2) and base platforms (e.g. ActivRobots[4]), faster and energy efficient computation or light weight manipulation devices (e.g. Katana[5]) as well as miniature sensors (e.g 2D LIDAR[6]) are available.

In sum, increased availability, accessibility and compatibility of these essential robot components enables research groups not only to address a small subset of the mentioned above challenges in DSR, but also to address the problem as a whole. Obviously, DSR is not solely about integrating existing solutions. But the consequent reuse of existing technology can help to save time and effort, so researchers can focus on a particular research field while maintaining a fully operable robot platform.

This is also confirmed by the presence of some rather specialized service robotic applications on the market. Such applications include floor cleaning (e.g Roomba and Scooba[7]), lawn mowing (e.g. Robomow[8]) and surveillance (e.g. Robowatch[9]). Still, these service robots do not possess the properties of a multipurpose autonomous and intelligent domestic service robot. Prominent examples of domestic and personal assistant robot research projects include ReadyBot[10], and PR2[11]. Wakamaru[12] and PaPeRo[13] focus more on social interaction studies. Many of these projects address relevant aspects of DSR. Still, what appears to be missing is a joint, international and multidisciplinary research and development effort which also includes the aspect of application-oriented benchmarking of systems in DSR. This was the motivation to initiate the *RoboCup*@Home competitions.

[1]The Mobile Robot Programming Toolkit (http://babel.isa.uma.es/mrpt/index.php/Main_Page)
[2]Microsoft Robotics Studio (http://msdn.microsoft.com/en-us/robotics/default.aspx)
[3]The Open Computer Vision Library (http://sourceforge.net/projects/opencv)
[4]ActivRobots (http://www.activrobots.com)
[5]Katana robotic arm (http://www.neuronics.ch)
[6]Hokuyo Sensor Technology (http://www.hokuyo-aut.jp)
[7]iRobot (http://irobot.com)
[8]Robomow (http://www.friendlyrobotics.com)
[9]Robowatch (http://robowatch.com)
[10]ReadyBot (http://www.readybot.com)
[11]PR2 (http://www.willowgarage.com)
[12]Wakamaru (http://www.mhi.co.jp/kobe/wakamaru/english)
[13]PaPeRo (http://www.nec.co.jp/robot/english/robotcenter_e.html)

The *RoboCup*@Home league targets the development and deployment of autonomous service and assistive robot technology as being essential for future personal domestic applications. It is part of the international *RoboCup* initiative, and currently is the largest annual service and home robotic competition worldwide. The *RoboCup*@Home tournaments are organized in independent test sets, which are used to benchmark the robots' abilities and performance in a realistic non-standardized home environment. More specifically, *RoboCup*@Home aims to offer a combination of interdisciplinary community building, scientific exchange and competition, which iteratively defines benchmarks and performance metrics on which service robots can be evaluated and compared in a realistic, dynamic and non-standardized domestic environment.

Since the real world is not standardized, measuring the performance of non standardized robots acting in it is a difficult task. The experimental paradigm to evaluate the complex robotic systems has to use consequent scientific analysis to improve on itself. Measuring the performance of the robots requires continuous reconsideration of the methodologies used since both the robots (their capabilities) and their operation environment (and the robot's tasks) will definitively change over time. This co-evolutionary development process, the feedback and refinement procedure, is a key element of the *RoboCup*@Home league. In our case, the tools are statistical benchmarks which test certain robot abilities and the measurement of the robots' performance.

RoboCup@Home also measures, in a scientific and quantifiable manner, the performance of complex systems. I firmly believe that creating and applying this experimental paradigm can greatly improve DSR developments. This chapter thus addresses the problem of benchmarking DSR through scientific competitions by presenting the approach followed in the *RoboCup*@Home initiative. The chapter contains several contributions:

- it presents an overview of benchmarking through competitions, describing other existing competitions and highlighting the unique features of *RoboCup*@Home;
- it describes the underlying concept of the @Home competition and its implementation into a framework for benchmarking in DSR which aims to be a common testbed for application development;
- it provides a detailed analysis of the results from different viewpoints that are of importance for assessing the actual performance of DSR and for planning future tests and other competitions.

The remainder of the chapter is organized as follows: The next section gives an overview of the state of the art in robotic benchmarking and DSR. Then, the concept and the implementation of the *@Home* competition are presented. Section 4.4 will evaluate the benchmarking results of the past several years and discuss the observed increase in performance, the scientific achievements and the importance of a vital community. The chapter concludes with an outlook on short and mid-term goals.

4.2 Benchmarking Domestic Service Robots

Benchmarking has been recognized as a fundamental activity to advance robotic technology [14, 15], and many activities are in progress. Some projects and special groups are working on defining standard benchmarking methodologies and data sets for many robotic problems, like Human-Robot Interaction (HRI), SLAM, or navigation. Examples for such initiatives are the EURON Benchmarking Initiative[1], the EURON guidelines on good experimental methodologies and benchmarking[2], the international workshops on Benchmarks in Robotics Research and on Performance Evaluation and Benchmarking for Intelligent Robots and Systems, held since 2006[3], the Rawseeds project[4], which aims to create standard benchmarks especially for localization and mapping, and the RoSta project[5], which focuses on standardization and reference architectures.

Benchmarking can be distinguished in two classes: *system benchmarking*, where the robotic system is evaluated as a whole, and *component benchmarking*, where a single functionality is evaluated. Component benchmarking is integral for comparing different solutions to a specific problem and for identifying the best algorithms and approaches. Among the many examples, much effort has been put on mapping and SLAM (e.g. [16, 17]), and navigation (e.g. [18, 19, 20]). While component benchmarking is useful for directly comparing different techniques of solving a specific problem, it is not sufficient for assessing the general performance of a robot with respect to a class of applications. Indeed, the best solution for a specific problem may be unfeasible or inconvenient when integrated with other components that compose a robotic application. On the other hand, system benchmarking offers an effective way to measure the performance of an entire robotic system in the accomplishment of complex tasks, as such tasks require the cooperation of various sub-systems or approaches. In this kind of benchmarking,

[1] http://www.euron.org/activities/benchmarks/index
[2] http://www.heronrobots.com/EuronGEMSig/Downloads/GemSigGuidelinesBeta.pdf
[3] All these workshops are summarized in http://www.robot.uji.es/EURON/en/index.htm
[4] http://www.rawseeds.org
[5] http://www.robot-standards.eu

a standard reference environment, reference tasks and related performance metrics are to be defined. Examples of system benchmarking are given in the fields of interactive robots [21] and of socially assistive robots [22].

When defining standard benchmarks, two common problems arise:

- The difficulty of defining a benchmark that is commonly accepted by the community (this is due to differing viewpoints on a problem from separate research groups);
- The risk of fostering the development of specialized solutions for an abstracted, standardized setting.

To avoid these problems, scientific competitions have proven to be a very adequate method because:

- benchmarks are usually discussed and then accepted by all the participants;
- participants are usually required to solve multiple benchmarks. These benchmarks vary over the years, thus providing for a disadvantage in using solutions that are too specialized.

Moreover, competitions provide an effective means of interaction and communication among research groups because they are often associated with scientific conferences or workshops and provide participants a large audience for their research efforts. Finally, annual competitions provide regular feedback on performance increases and allow for establishing medium-term projects.

Among the many robotic competitions, the AAAI Mobile Robot Competitions were one of the first, being established in 1992 [23]. RoboCup (founded in 1997) [24] currently has the largest number of participants (e.g. 440 teams with more than 2,600 participants from 35 countries in 2006). The DARPA Grand Challenge is probably the most recognized in terms of public and media attention and the one that is most directly application-oriented. Furthermore, educational contests, such as EUROBOT[1] or RoboCup Junior[2], are organized with the main goal of presenting robotics to young students. Thus, they deal with simpler tasks and robotic platforms.

All of these competitions have obtained very relevant results, which are analyzed in the following:

[1] http://www.eurobot.org
[2] http://rcj.sci.brooklyn.cuny.edu

AAAI AAAI Mobile Robot Competitions are held in conjunction with the AAAI and (sometimes) IJCAI Conferences on Artificial Intelligence. Thus, it offers great visibility within the AI scientific community. Many important scientific and technological achievements demonstrated during these competitions have been reported [23]. Although these competitions offer a relevant suite for benchmarking AI and robotics technology with relevance for real-life applications, their focus and benchmarks change heavily on a yearly basis. This change of focus makes it difficult to approach the problems in a continuous and iterative way and to build up a community with a long-term goal.

RoboCup Soccer The ultimate goal of the RoboCup project is: "By the year 2050, develop a team of fully autonomous humanoid robots that can play and win against the human world champion soccer team." Moreover, as opposed to AAAI competitions, RoboCup events put the main focus on the competition and offer the possibility to discuss scientific achievements in a small and more focused RoboCup Symposium. RoboCup has proven to provide an efficient means of interaction and communication among research groups. It combines scientific research, competition, benchmarking and reality checks on various concepts. Performance is measured on a yearly basis. However, having a specific focus on soccer also presents some limitations. The main limitation is the danger of over-specialization of solutions due to more or less fixed environmental conditions and rules. For example, in the middle-size league, where the design of the robot is a major issue, all the teams rapidly converged towards the same hardware architecture (catadioptric cameras and omnidirectional driving robots) which was highly optimized on the provided scenario. Although this causes an immediate improvement of the average performance in the competition, it contains the danger of running into a single suboptimal solution which can not be applied to a real-world setting.

Robot Rescue Another example is given by Search and Rescue Robotics. Rescue competitions started in 2000 within the AAAI Mobile Robot Competition [95] and since 2001 within the RoboCup Rescue initiative [96]. RoboCup Rescue competitions have defined standard rescue arenas and tasks for benchmarking robotic search and rescue missions and for measuring an increase in performance of the rescue robotic technology in a standardized abstracted environment. The concepts of the rescue robot initiative with respect to benchmarking are similar to those proposed in @Home. Within the Rescue competitions, common metrics for HRI have been defined [97] and effective evaluation of HRI techniques have been carried out [98, 99], with a specific focus on the interfaces used by operators to interact remotely with the rescue robots. Nevertheless, this indirect kind of HRI via an operator station involving semi-autonomy and remote control is different to what is required in most DSR tasks, where the focus is a direct and more natural interaction and full autonomy. Still, one can think of certain DSR applications where such kind interaction is desired, e.g. to monitor and communicate with nursing cases remotely.

DARPA Grand Challenge The DARPA Grand Challenges[1] were the most prominent robotic competitions to date. GPS navigation together with multimodal sensor data fusion were commonly used to face the uncertainties and dynamics of real-world application scenarios. In the Urban Challenge, even real traffic rules were applied, but the complexity was limited by simplifying the cognition tasks. Contextual information was entered into a predefined map, the route network definition file (RNDF), which consisted of waypoints with GPS coordinates, connection types, traffic signs, number of lanes, width of lanes, etc. Participating in these challenges required a lot of effort, as the joint work of different research groups and industries with complementary competencies was a critical factor. At this time, it is uncertain if this initiative will be continued in the future and to what capacity.

Service Robotic Competitions Initiatives that are directly related to Domestic Service Robotics mainly aim at a single specific task. For example, the AHRC Vacuum Contest[2] and the 2002 IROS Cleaning Contest[3] [25] are focused only on floor cleaning, while ROBOEXOTICA[4] concentrates only on robots preparing and serving cocktails. A more general initiative is given by the ICRA HRI Challenge[5], and it is motivated by the fact that "the effectiveness of a robot engaging in HRI must be evaluated by human users who got the chance to interact with the robot for a sufficiently long period of time." However, it is still in preliminary stages because evaluation criteria for benchmarking the performance have not been defined.

Although these initiatives are very relevant to the field, it shows that there is no major international annual competition in the field of Domestic Service Robotics that can be considered a continuous integrated system benchmarking activity. This fact was

[1] http://www.darpa.mil/grandchallenge/index.asp
[2] http://www.botlanta.org
[3] http://robotika.cz/competitions/cleaning2002/en
[4] http://www.roboexotica.org/en/mainentry.htm
[5] http://lasa.epfl.ch/icra08/hric.php

the main motivation for the design and the implementation of the *RoboCup@Home* competitions, which will be discussed in detail in the following sections.

4.3 The *@Home* initiative

RoboCup@Home is an effort to compare and evaluate integrated, application oriented, systems by means of a competition. It is a is a *system benchmarking* activity for domestic service robotics implemented as an annual competition. The focus on application in combination with the aim of creating multipurpose robots requires integration and testing of many abilities. The aim on testing HRI in natural non-standardized environments contradicts with maintaining precisely predefined conditions while evaluating. In @Home, each test assesses a set of abilities and each ability is rewarded with a predefined amount of points. This section describes the conceptual design, the rules and the implementation of the *RoboCup@Home* competition.

4.3.1 Concept

The following considerations and criteria act as the basis of a common agreement for the *RoboCup*@Home initiative.

Uncertainty To reflect the uncertainty immanent in every real-world setting, the rules should not specify or limit any more qualities of a task than (absolutely) necessary. This complies with the aim of providing a lean set of rules. Moreover, it encourages robust solutions that remain functional over a wide number of particular situations under as many circumstances as possible. This way, object positions or environmental characteristics, such as lighting conditions, are not specified and the scenario setup is changed frequently.

Extendable framework for benchmarking With the aim of benchmarking myriad robot capabilities for DSR with many of them yet to be developed, the framework for the competition needs to allow for constant evolution and modular enhancement of itself. The framework consists of an initial set of independent tests all benchmarking an individual set of relevant capabilities in DSR. Over time, when an increase in performance in the individual tests is observed, these tests are either enhanced by making the tasks more difficult or the tasks are merged together to form a more integrated, and therefore more realistic, application scenario.

Autonomy Robots in the *@Home* league are required to be fully autonomous and mobile. That is, robots must complete benchmarks without being controlled remotely. To lower the demands for on-board computers, external computation is permitted as long as nobody interacts with the external computer during the test. To foster demonstration and use of new approaches, external devices the robot can interact with (external cameras, sensors, etc.) are allowed in certain tests. It may appear that instructions given by a human acting with the robot are a kind of tele-operation, but the execution of a given high-level task such as "Bring me object A" incorporates autonomy in terms of task decomposition, decision making, perception, task planning, and task execution. We enforce this autonomy by the uncertainty inherent in the environment and strict time limits for the setup of the robots.

Natural interaction In order to inhibit control of a robot by keyboard commands, interaction with a robot must be *natural* in all tests. This means that the interaction is either done via natural language or gestures (no keyboard control). Other modes of interaction like the use of touch screens or advanced remote controls can and should be demonstrated in the form of technical challenges (see *Open Challenge*, *Demo Challenge* and *Finals* in Section 4.3.3), where these restrictions are not applied. Then, corresponding solutions are to be integrated and allowed in future competitions. Moreover, haptic interaction (touching the robot) should further foster development of intuitive modes of control and interaction (instead of using a standard computer keyboard) and consider future use by the target audience: the general public, laymen in robotics, or elderly and disabled people.

Benchmarking in uncertain conditions The home environment in which the benchmarks take place is not standardized to represent a realistic setting. It contains common natural objects and varies over the years. Examples of previous competition environments are given in figure 4.1. The degree of uncertainty contained in the benchmarks is high as the environment is hardly specified in size, shape, contained objects, kind of walls or floor, lighting and sound conditions, etc. and changes from competition to competition. Especially the interaction with randomly selected people adds to this uncertainty. The non-standardized conditions under which Human-Robot-Interaction is tested include different persons of different size, different nationalities, different accents and voices, different gender, different clothes, different sound conditions, background noise, and demand for distinguishing between active persons and the many spectators. Still, it is of importance to maintain a certain range of difficulty for all participants while

Figure 4.1: The *@Home* scenario in 2006 (top left), 2007 (top right), and 2008 (bottom)

boundary conditions may slightly change, so that the performance can be compared. This requires a common agreement and careful definition of the level of uncertainty. To solve the discrepancy between desired uncertainty and comparability of performance measures, statistical benchmarking is used as a method. As the competition consists of multiple tests and these tests consist of multiple tasks which are evaluated, currently, more than fifty data points are collected per team on about a dozen abilities which allows comparison of the teams performances. Every team faces the same variability in the environment conditions and using multiple samples per team per ability allows for a statistical performance analysis.

Fostering a wide range of approaches and solutions The rules should be kept as unrestrictive as possible, and the benchmarks are to be defined in such way that the solutions for the given problems are not implicit. This approach requires a high level of common sense and agreement from the teams and the community, as trivial and undesired solutions to certain problems can not be completely avoided, e.g., having a robot approaching an object with an open-loop control instead of using sensor feedback to adjust to position changes. Also, participants should have the choice to select certain benchmarks according to their research background, skills, and their robot's capabilities. Besides having predefined benchmarks the teams can select from, the competition also offers the possibility of demonstrating new abilities and scientific results or applications not yet covered in the tests. These new aspects can later be used to enhance the benchmarks in the future.

Multidisciplinary community Putting few restrictions on the robots participating and providing the freedom to select benchmarks and approaches should motivate teams from different research backgrounds to participate in and to contribute to a growing community, one which fosters the exchange of multidisciplinary scientific and technological knowledge. Furthermore, the development of a common vision, common goals, as well as common sense and fair play in the competition, are required. Feedback from the teams is further needed to iteratively enhance the competition.

Generating public awareness The competition also aims to generate interest from a non-technical, public audience by demonstrating usefulness in daily life, future applications and social relevance. This way public awareness for DSR should be increased, and new links between research and industry should be established.

4.3.2 Defining key features

Before starting with the implementation of the benchmarks, an initial set of robot key features (abilities and properties) was derived from the analysis of the state of the art in DSR and robotic competitions in the beginning of this chapter. These features help to design the benchmarks and the score system for the competition. Furthermore, these features allow for a later analysis of the teams' performances and help to develop and later enhance the competition in a structured way. As the competition and its benchmarks are expected to evolve over time, the key features and their weights in the competition are also expected to be adapted. The key features are divided in two groups: *functional abilities* and *system properties*.

4.3.2.1 Functional abilities

Functional abilities include specific functionality that must be implemented on the robot in order to perform decently in the tests. Each test requires a certain subset of these abilities, as they are also directly represented in the score system. Teams must thus decide which of these abilities to implement depending on their background and the kind of tests they intend to participate in. *Functional abilities* currently are:

- *Navigation*, the task of path planning and safely navigating to a specific target position in the environment, avoiding (dynamic) obstacles

- *Mapping*, the task of autonomously building a representation of a partially known or unknown environment on-line
- *Person recognition*, the task of detecting and recognizing a person
- *Person tracking*, the task of tracking the position of a person over time
- *Object recognition*, the task detecting and recognizing (known or unknown) objects in the environment
- *Object manipulation*, the task of grasping or moving an object
- *Speech recognition*, the task of recognizing and interpreting spoken user commands (speaker dependent and speaker independent)
- *Gesture recognition*, the task of recognizing and interpreting human gestures.

4.3.2.2 System properties

System properties include demands on the entire robotic system that are considered of general importance for any domestic service robot. They can be described as "soft skills" which must be implemented for effective system integration and successful participation in the *@Home* competition.

Initial system abilities are:

- *Ease of use* - Laymen should be able to operate the system intuitively and within little amount of time.
- *Fast calibration and setup* - Simple and efficient setup and calibration procedures for the system.
- *Natural and multimodal interaction* - Using natural modes of communication and interaction such as, e.g. using natural language, gestures or intuitive input devices like touch screens.
- *Appeal and ergonomics* - General appearance, quality of movement, speech, articulation or HRI.
- *Adaptivity / General intelligence* - Dealing with uncertainty, problem solving, online learning, planning, reasoning.
- *Robustness* - System stability and fault tolerance.

- *General applicability* - Solving a multitude of different realistic tasks.

Although some of these properties cannot be benchmarked as directly as the *functional abilities*, they are considered as integral and an implicit part of the competition and tests.

4.3.3 Implementation of benchmarks

In the following section, I am going to elaborate the implementation of the *RoboCup*@Home competition as a set of benchmark tests for service robots in domestic environments. This implementation is based on the concepts mentioned in the previous section.

The competition is organized in a multistage system. All qualified teams (currently up to 24) participate in the first stage called *Stage I*. It consists of a set of benchmarks with a focus on testing basic tasks and checking for a small set of key features with a limited amount of uncertainty involved. Then, the ten best teams advance to the second stage called *Stage II* where the benchmarks are more demanding, more realistic and involve more uncertainty and a higher level of system integration. In the *Finals*, the performance of the five best teams is evaluated by a jury. A combination of the jury score and the previous score from *Stage I* and *Stage II* determines the ranking.

The tests themselves comprise realistic and useful tasks for a domestic service robot. Each test evaluates certain key features. A tabular overview of the functional abilities required in each test can be found in Table 4.1. An overview of benchmarks where certain system properties are tested is given in section 4.4.1.2. The implementation of the competition described in the following reflects the situation of the competition in 2008.

4.3.3.1 Score system

Two types of test exist: Regular tests are specified in terms of the task to solve and the scoring. In open tests, teams can either freely choose what to show (the Open Challenge and the Finals), or a topic is given according to which teams can do a demonstration (Demo Challenge). Since the scoring in the open tests is based on an evaluation by a jury, it is partially subjective. However, for every open test there is a list of criteria, which the jury bases its decision on. The criteria will be discussed in the test descriptions below. The scoring in the regular tests mainly reflects the key features mentioned earlier.

To keep the entry level for the competition reasonably low, while still aiming for high top level performance, a so-called *partial score system* was introduced in 2008. With this, a team receives a part of the total score for showing a part of the task's specification. Each of the partial scores is connected to one or more of the functional abilities and/or system properties. This does not only allow for assessing the fulfillment of these features individually, but it is also an incentive for teams to participate in a test even if they know that they cannot solve it completely.

Referees are provided by opponent teams. It is their duty to provide the same difficulty level for all teams in a certain test according to common sense and fair play. This involves, for example, the selection of random people, the definition of paths or the selection and placements of objects. As this is a critical procedure for the entire competition, it is closely monitored by the Technical Committee of the @Home league. It is important to notice that the referees only ensure the proper execution of a task according to the rules, while they do not evaluate the performance nor assign any subjective score. For each test in *Stage I* a maximum of 1000 points, and in *Stage II* 2000 points, can be scored.

4.3.3.2 Stage I Tests

The overall theme of tests in *Stage I* is to benchmark essential abilities and properties that any robot in *@Home* should exhibit. During the setup days before the competition, a set of ten randomly chosen and previously unknown objects is provided to the teams. A subset of these objects is then used for certain tests.

Introduce In the *Introduce* test, a robot has to autonomously enter the scenario and move to a position in front of the audience. There, the robot has to introduce itself and the team using speech, gestures, slides or multimedia. Afterwards, it must leave the scenario on its own. The performance of the robot is evaluated against certain criteria such as smoothness and flow in both movement and presentation, as well as the general appearance of the robot. This test calls for interaction abilities such as speech synthesis, articulation and expression of moods. Figure 4.5 (right) shows an *@Home* robot expressing an emotion. Since there are no regulations on the content and the and the procedure of the presentation, teams are free to show what they consider to be useful, attractive and potentially innovative means for a robot to convey information. Team leaders of the competing teams evaluate the performances. This ensures that the teams get to know each other, their research background and their robots already at the very beginning of the competition, thus fostering exchange of knowledge in the community. The variety in focus and interest among the different teams also ensures diverse feedback in the evaluation of the performing team.

Fast Follow A robot's task in the *Fast Follow* test is to follow a person from one entrance of the environment to the other. Two teams compete against each other in this test starting from opposite ends. They need to pass by one another on a common path through the scenario. The most important capabilities evaluated in this test are detection and tracking of humans and safe navigation in a dynamic environment, a task which naturally includes obstacle avoidance. By letting the teams' paths cross one another the robots further need to discriminate their human leader from the opponent's one. Partial scores are awarded for passing a check point at around half of the track, passing the opponent, completing the track, being the fastest team, and not touching any object.

Fetch & Carry In the *Fetch & Carry* test, the robot has to find and retrieve a certain object which it then needs to return to the human instructor. The robot is instructed to get the object using natural language. Teams are allowed to give the robot a hint on the item's location. Thus, speech recognition and natural language processing are essential to succeed in this test. Human-robot interaction by means of joint activities in common physical space is emphasized, since for the robot to understand the hint, it needs to be capable of interpreting the given spatial description. Partial scores are awarded for understanding the command, finding the object, manipulation of the object, successful delivery, and autonomously leaving the scenario in time.

Who is Who? The main theme in the *Who is Who?* test is the detection and recognition of people. The robot has to find three persons (two of them unknown to the robot) spread out around an area near to the entrance of the scenario. The robot then needs to find these people, introducing itself to every person found. Each person has to be either identified (if known already) or learned (if unknown). Beside sufficient navigation capabilities this test checks for capabilities in person detection and recognition (mostly face recognition). It also calls for skills in engaging and conducting interaction with humans such as speaker independent speech recognition. Furthermore, basic conversation capabilities are required in order to instruct the unknown person on its behavior during the recognition process. Figure 4.2 on the left shows an example from the 2007 competition. Partial scores are awarded for detecting a person, discriminating

Figure 4.2: Team UT Austin doing Who is Who (left), team eR@sers teaching in an object (right)

between known and unknown persons, learning and recognizing previously unknown persons, and autonomously leaving the scenario on time.

Competitive Lost & Found For the *Competitive Lost & Found* test, two teams compete against each other at the same time. The assignment is to find and identify as many out of three objects as possible. The referees pick the objects randomly from the set of objects and distribute them randomly throughout the scenario (in a way that the robot actually has a chance of finding the object) just before the test starts. The major abilities tested here are object detection and recognition. An additional focus is put on reliable and fast navigation, since the fastest of the two teams finding all objects receives an extra bonus. Having two robots compete in the same scenario simultaneously accentuates the need for safe navigation in a dynamic environment. Partial scores are awarded for finding an object, identifying an object, leaving the scenario, and being the fastest team.

Open Challenge In order to allow all participating teams to freely demonstrate their scientific achievements and their unique robot features or capabilities the *Open Challenge* concludes *Stage I*. Here, no restrictions on the kind of performance, kind of used devices, or kind of interaction are applied. The Open Challenge is meant as a means of iteratively enhancing the @Home competition by integrating relevant and innovative aspects demonstrated by the teams in future tests. This test consists of a presentation given by the team (an example is depicted in figure 4.3) and a demonstration of their robot. The performance is evaluated by the opponent team leaders according to a list of predefined criteria. These criteria are as follows:

Presentation The quality and the content of the presentation part is evaluated.

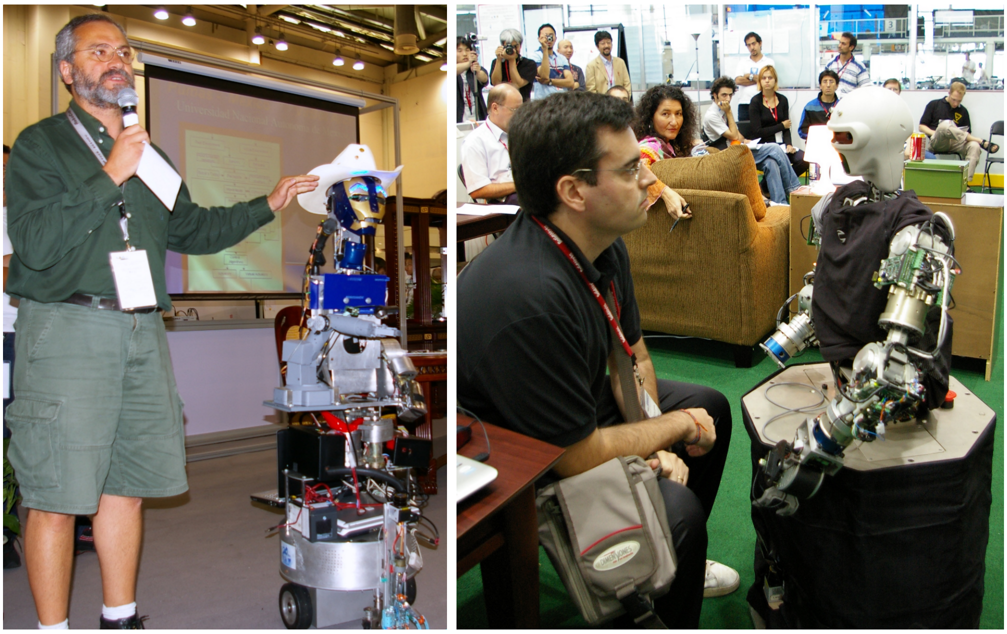


Figure 4.3: Presentations from team Pumas (left) and team PAL(right) during the *Open Challenge*

Social relevance / Usefulness for daily life of the demonstration is evaluated.

Human-Robot interaction The quality and kind of human-robot interaction in the demonstration is evaluated.

Autonomy The grade of autonomous behavior during the demonstration is evaluated. This is to avoid open loop behavior.

Difficulty and success The level of difficulty and success of the robot performance is evaluated.

Appeal/Relevance for @Home The Appeal of the demonstration and the relevance for the @Home initiative are evaluated. Should elements of the demonstration be integrated in future *@Home* competitions?

Scientific value / Jury questions Scientific value of the presentation and answers to jury questions are evaluated.

4.3.3.3 Stage II Tests

In contrast to the challenges in *Stage I*, *Stage II* comprises tests that are more complex, involve more uncertainty, and which check for the integration of several features in a more realistic, application-like setting.

PartyBot The *PartyBot* test is an elaborated version of the *Who is Who?* test from *Stage I* where the robot's task is to find, recognize, and/or remember multiple unknown persons randomly distributed throughout the entire environment (standing and sitting) and to tell them apart later on when serving a drink. Besides a focus on interaction capabilities, especially when having to get to know previously unknown persons, navigation, object detection, and manipulation are necessary to pick up and deliver a cup to a particular person. Two robots about to grasp a cup and a bottle are shown in figure 4.4. Partial scores are awarded for detecting and navigating to the persons, navigation to the cup, grasping and carrying the cup, handing it over, and autonomously leaving the scenario.

Figure 4.4: AllemaniACs (left) and B-IT Bots (right) robots grasping and delivering a drink

Supermarket To address the possible future application of assisted shopping, the *Supermarket* test was introduced. The robot needs to retrieve certain household objects from a shelf for a person randomly chosen from the audience that does not know how to operate the robot. This demands that the robot has the ability to explain its own modes of operation and to report on the robot's internal models to a layman. Furthermore, it requires speaker-independent speech recognition, and it enforces the ease of use proclaimed as a system property, since the operation by laymen raises uncertainty both in input and reaction. The lean specification allows for multimodal input such as gestures and speech. The interactive character further adds a demand for joint work-space concepts to be employed. Handling objects requires object detection and recognition as well as manipulation abilities. Partial scores are awarded for understanding which object to get, finding the object, retrieving the object, object manipulation on different heights, delivery of the object, and multimodal input (i.e., using speech and gestures).

Walk & Talk The task of the *Walk & Talk* test is to introduce a robot to a new environment and make it remember a set of places. A human leader guides the robot through the scenario that was completely rearranged beforehand (and is therefore unknown to the robot) and has to teach specific locations only using natural language. The robot then has to prove that it has correctly learned those locations by having to navigate to certain places in random order after a speech command is given. To accomplish this test a robot does not only need to recognize and track a human, but it also has to model and map the so far unknown environment to be able to navigate in it later on. Further, human-robot interaction capabilities such as speech recognition and gesture recognition are an indispensable means to meet the demands posed by the above setting. Partial scores are awarded for following the person to the locations, autonomously navigating to the locations learned previously, navigating back to the start position, and autonomously leaving the scenario.

Cleaning Up In the *Cleaning Up* test, a robot needs to collect a set of five unknown objects (i.e., not from the set of known objects) dispersed throughout the scenario. Objects can be anything that can be expected to lie around in a household. Restrictions are put on the size so that objects are not to small to be overlooked and not too big so they can still be handled by a robot by pushing or grasping. To solve this task, the robot has to find potential objects first (effective search and object detection). Then the robot is expected to test the assumption of having found an object by trying to manipulate it. After having figured out how to handle the objects, the goal is to move them to a predefined area in the scenario. Partial scores are awarded for correct detection of objects, having no false positive in the detection, delivery of objects to a designated area, and autonomously leaving the scenario.

Demo Challenge The *Demo Challenge* is an open demonstration similar to the *Open Challenge*, as no restrictions on the kind of interaction or the kind of external devices are applied. In contrast to the *Open Challenge* the topic of the demonstration is

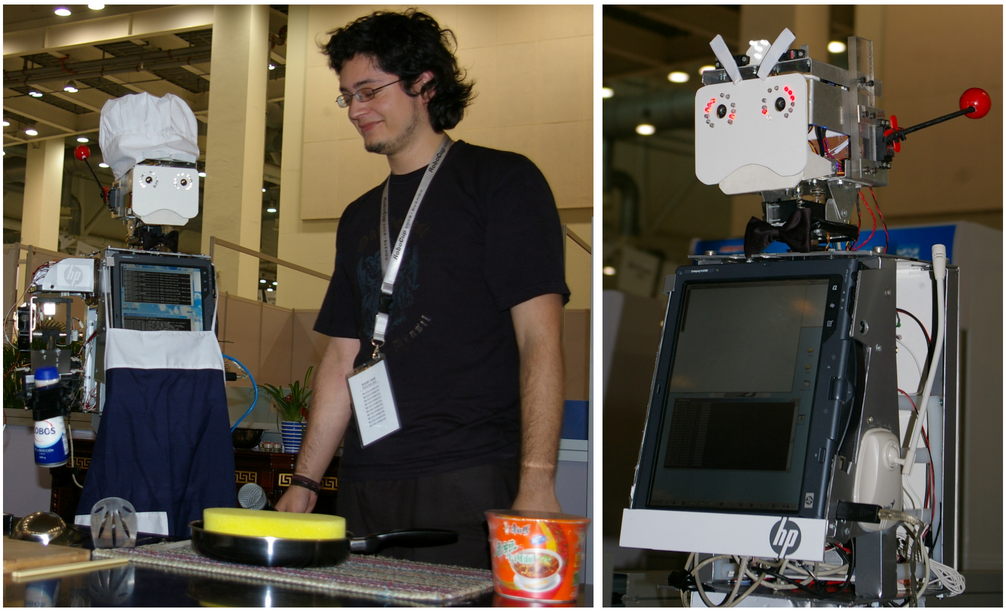


Figure 4.5: HomeBreakers robot Bender assisting cooking (left) and showing emotions (right)

pre-defined and varies from year to year. It is meant to foster development in a certain area or on a particular theme with a strong relation to real applications and daily-life situations. It should provide a showcase of the current state of the art in home robotics and inspire both the community and the public. In 2008 the theme was "cooking", i.e., the robot should assist a human in preparing a meal. The task was not formulated in any concrete specification. Possible means to assist were, for example, fetching a recipe from the Internet and retrieving ingredients necessary for the same. Figure 4.5 (left) shows a robot participating in the 2008 demo challenge. Evaluation was done by a jury consisting of the organizers of the @Home competition. The evaluation criteria in 2008 were: assisting and interacting with the human, ambient intelligence and object manipulation.

4.3.3.4 Finals

The competition concludes with the *Finals*, where as in the *Open Challenge*, each team can demonstrate what they think is an important feature or capability of their robot. The idea, however, is to present a coherent story-like performance which is evaluated by an external jury according to a list of predefined criteria. Because teams that have reached the Finals have already proven to fulfill a variety of abilities, the criteria of the evaluation are slightly different from those in the *Open Challenge*.

Scientific contribution / Contribution to the community Amount, relevance and quality of the team's contribution to the @Home community.

Relevance for *RoboCup*@Home/ Usefulness for daily life of the demonstration.

Usability / Human-robot interaction and multimodality Ease of use, quality of HRI and multimodality during the demonstration.

Originality and presentation Originality of the demonstration, quality of the presentation.

Difficulty and success of the demonstration.

Previous performance during *Stage I* and *Stage II* (determined by previous score).

4.4 Evaluation and discussion

Two important objectives for an annual scientific competition are to provide a common benchmark to many teams, which allows for the measurement of performance advances over time, and to develop relevant scientific solutions and results. In this section the results obtained by the *RoboCup*@Home teams both in terms of performance in the tests and in terms of scientific achievements are described and discussed.

As for a team's performance, it is important to note that the score system of *RoboCup*@Home relates the desired abilities of the robots with the scores of the competition. In contrast to other competitions (e.g., RoboCup soccer), where the score hides many factors, the *@Home* score provides an actual way of measuring the performance of teams in terms of such abilities. This score consequently enables an analysis of performance in order to update the rules and drive technological and scientific progress.

In the remainder of this section, first, an analysis of the team performance in 2008 based on the relationship between key features and test scores is presented; second, the evolution of the league over time is discussed; then, the main scientific contributions related to *@Home* tests obtained by the teams are highlighted; and finally, the results from the *@Home* community are discussed.

4.4.1 Representation of key features in the benchmarks

In the following the representation of key features, i.e. the functional abilities as well as the system properties, in the benchmarks and in the competition score are shown.

4.4.1.1 Functional abilities

Table 4.1 relates the *functional abilities* defined in section 4.3.2 with the tests described above. It quantifies the maximum score distribution per test with respect to the contained functional abilities. For ease of notation, The following abbreviations are used. Tests include Fast Follow (FF), Fetch & Carry (FC), Who is Who (WW), Lost & Found (LF), PartyBot (PB), Supermarket (SM), Walk & Talk (WT), and Cleaning Up (CL). The abilities are Navigation (Nav), Mapping (Map), Person Recognition (PRec), Person Tracking (PTrk), Object Recognition (ORec), Object Manipulation (OMan), Speech Recognition (SRec), and Gesture Recognition (GRec). Note that for the *Introduce* test, the *Open Challenge*, the *Demo Challenge*, and the *Finals*, values are not indicated because teams can freely choose their performance and the focus on certain abilities themselves. This way, new abilities are expected to be demonstrated, which can be used to enhance the competition in the future.

Test	**Nav**	**Map**	**PRec**	**PTrk**	**ORec**	**OMan**	**SRec**	**GRec**	**Total**
FF	550	0	0	450	0	0	0	0	1000
FC	375	0	0	0	150	400	75	0	1000
WW	350	0	550	0	0	0	100	0	1000
LF	550	0	0	0	450	0	0	0	1000
PB	1000	0	700	0	0	300	0	0	2000
SM	0	0	0	0	400	1000	200	400	2000
WT	918	416	0	250	0	0	416	0	2000
CL	1000	0	0	0	550	450	0	0	2000
Tot	**4743**	**416**	**1250**	**700**	**1550**	**2150**	**791**	**400**	**12000**

Table 4.1: Distribution of test scores related to functional abilities

Since the competition involves mobile robots, navigation is currently the most dominant ability represented in the score. Object manipulation and recognition also play an important role since service robots are useful if they can effectively manipulate objects in the environment. Person recognition, tracking, and speech/gesture recognition are needed to implement effective human-robot interaction behaviors. As gesture recognition was introduced as a new (and optional) ability in 2008, its weight in the total score is still comparably low. Finally, mapping plays a more limited role; such an ability is used in the Walk & Talk test, where the environment is completely remodeled during the test, so that the robot enters into an unknown environment, while for other tests only minor modifications of the environment are made right before the tests. Thus, pre-computed maps (either built off-line by the robot or manually drawn) can be used.

This table is important in order to define the weight of each ability in a test and in order to distribute the abilities among the tests. Furthermore, one can analyze the performance of the teams and the difficulty of the tests after a competition. This allows for an iterative and constant development and improvement of the benchmarks.

4.4.1.2 System properties

Similar relationships between system properties and the tests exist. As previously mentioned, this relationship cannot be quantified in scores as easily, as the system properties are of more implicit meaning for the tests. However, on the basis of the objective of the tests, the importance of each of the system properties can be estimated. Table 4.2 relates tests with system properties by denoting a 'very important' relation with '++', an important relation with '+', and a minor relation with '-'. Note that these symbols are used only to indicate the importance of system properties in a test, rather than defining the score of a test. System properties are further represented in

Test	EUse	FCal	NInt	App	Adap	Rob	GAppl
IN	-	+	-	++	-	-	-
FF	-	+	-	-	-	+	+
FC	+	+	+	-	+	+	+
WW	+	+	++	-	+	+	+
LF	-	+	+	-	+	+	+
OC	-	+	+	+	+	-	+
PB	+	+	++	-	+	+	++
SM	++	+	++	-	++	+	++
WT	+	+	++	-	+	+	++
CL	-	+	-	-	++	+	++
Dem	+	+	++	+	+	-	++
Fin	+	+	+	++	+	-	++

Table 4.2: Importance of system properties in each test

the general rules, in overall requirements, and in special properties in certain tests. By using laymen to operate the robots in the Supermarket test, the Who is Who test, and the PartyBot test, *Ease of Use* (EUse) is enforced. The restrictions on setup time and procedures demands for *Fast Calibration and Setup* (FCal). *Natural Interaction* (NInt) and *Multimodal input* is rewarded in the supermarket test. *Appeal and Ergonomics* (App) are part of the evaluation criteria in the Introduce test, the Open Challenge, and the Finals. *Adaptivity* (Adap) is especially important in the Cleaning Up test. The limited number of specifications in the tests and the environment, and the fact that people who interact with the robot are chosen randomly in many tests, demands *Robustness* (Rob). Finally, a team can only reach the *Finals* if its robot performs well in many tests with different tasks to solve. This incorporates the aspect of *General Applicability* (GAppl).

4.4.2 Analysis of 2008 team performance

In the following, the performance of the teams in these abilities during the *Robo-Cup*@Home 2008 competition is analyzed.

Ability	Available scr	Achieved scr max	Achieved scr avg
Navigation	4743 (40%)	1892 (40 %)	1178 (25%)
Object Manipulation	2150 (18%)	75 (3%)	15 (1%)
Object Recognition	1550 (13%)	450 (29%)	125 (8%)
Person Recognition	1250 (10%)	400 (32%)	190 (15%)
Speech Recognition	791 (7%)	692 (87%)	293 (37%)
Person Tracking	700 (6%)	700 (100%)	570 (81%)
Mapping	416 (3%)	416 (100%)	183 (44%)
Gesture Recognition	400 (3%)	0 (0%)	0 (0%)
Total	**12000 (100%)**	**4909 (41%)**	**2554 (21%)**

Table 4.3: Available and achieved score for the desired abilities

Table 4.3 presents the scores actually gained by the teams during the competition and the percentage with respect to the total score available, related to each of the desired abilities. The third column shows the result obtained by the best team, while the fourth one is the average of the results of the five finalist teams. This table allows for many considerations, such as:

- Which abilities have been most successfully implemented by the teams?
- How difficult are the tests with respect to such abilities?
- Which tests and abilities need to be changed in order to guide development into desired directions?

From the table it is evident that teams obtained good results in navigation, speech recognition, mapping and person tracking. Notice that the reason for a low percentage score in navigation is not related to inabilities of the teams, but to the fact that part of the navigation score was available only after some other task was achieved. Speech recognition worked quite well, especially considering that the competition environment is much more challenging than a typical service or domestic application due to a large number of people and a lot of background noise. The achievements in mapping and person tracking may be explained instead by the limited difficulty of the corresponding tasks in the tests.

On the other hand, in some tasks, teams were not very successful. Object manipulation is difficult, especially when an object is not known in advance and calibration time is limited. Because a large proportion of the score was given for manipulation, many teams attempted it, but only a few were successful. A similar analysis holds for object and person recognition and reported slightly better results with the same difficulties arising from operating under natural environment conditions (i.e., lighting), with limited or null calibration time. Finally, gesture recognition was not implemented by teams, probably due to the small number of points available.

Table 4.4 summarizes the number of teams participating in each test and those which received a non-zero score. This table helps to evaluate team preferences and difficulty of the tests. Note that teams were not required to perform all the tests. Therefore, some of the zero scores in the table derive from a team's choice not to participate in a test.

An evaluation of system properties is more complicated since they are difficult to quantify precisely. Our current approach is to test for system properties through general requirements and to enforce the combination of functional abilities. An analysis

Test	**Participating Teams**	**Teams with non-zero score**
Introduce	12	12
Fast Follow	12	12
Fetch & Carry	9	5
Who's Who	8	4
Comp. Lost & Found	8	2
Open Challenge	13	13
Party Bot	5	2
Supermarket	3	3
Walk & Talk	10	10
Robot Chef	4	4
Cleaning	3	1

Table 4.4: Number of teams participating and gaining score for each test.

of these results is very helpful for the future development of the *@Home* competition. It gives direct, quantitative feedback on the performance of the teams with respect to key abilities and tasks. This allows us to identify abilities and respective tests which need to be modified, and to adjust the weights of certain abilities with respect to the total score. Possible modifications involve:

- Increasing the difficulty if the average performance is already very high
- Merging abilities into high-level skills, more realistic tasks
- Maintaining or even decreasing difficulty if the observed performance is not satisfying
- Introducing new abilities and tests

As the integration of abilities will play an increasingly important role for future general purpose home robots, this aspect should especially be considered in future competitions.

4.4.3 League progress

The results obtained so far by the *@Home* initiative can be measured on several levels:

- increased number of participating teams and of community members,
- increasing performance in the tests,
- increase of public awareness (media, press, Internet),
- increasing number and quality of scientific contributions.

For some of these measures, a quantitative analysis over the years is presented in the following.

Since 2006, a total of 25 teams distributed worldwide (12 from Asia, 8 from Europe, 4 from America, 1 from Australia), have participated in the three years of the *RoboCup*@Home world championship until 2008. Furthermore, national competitions have been established in China, Mexico, Germany, Iran and Japan. These events are useful not only to test team developments and rules, but also to possibly select teams that will participate to the world championship.

Table 4.5 describes the number of participating teams in the annual world championship. The second column shows the number of teams that pre-registered and delivered the necessary qualification material, such as videos and a team description paper. The third column shows the number of teams that qualified after a review from the Organizing Committee, and the fourth column shows the number of teams that actually participated in the competitions. Finally, the fifth column shows the number of new teams (i.e., teams that did not participate in the previous years). The last line refers to the 2009 competition, for which 26 teams from 14 countries preregistered so far.

Year	Pre-registration	Qualification	Participation	New teams
2006	20	17	12 (440; 2.72%)	12
2007	16	13	11 (321; 3.42%)	5
2008	18	17	14 (373; 3.75%)	8
2009	26	23	-	-

Table 4.5: Number of participating teams

The number of and the increase in participating teams must be also related to the general participation across all leagues. (The number of total teams and percentage of

@Home teams are given in parenthesis in the fourth column). Regardless of the drop in the total number of teams throughout all leagues in 2007 (in the US) and 2008 (in China), mainly due to high travel and shipping costs, as well as the difficulties in custom and visa affairs, the increase of percentage of *@Home* teams is a clear indication of the growth of the league. Moreover, the number of pre-registrations and qualifications for the 2009 competitions in Austria is very promising.

Furthermore, being part of the RoboCup community allows teams to exchange ideas and solutions, to plan long-term projects and to participate in the competition for several years. Indeed, it is interesting to see that some teams adapted their robots designed and built for other RoboCup Leagues to compete in *@Home*, and that one team in 2006 and 2007 used the same robot in both the soccer Four-Legged and the *@Home* leagues. One team in 2008 even used the same robot in both the Rescue and *@Home* leagues. Moreover, many teams participated multiple years. Three teams have participated in all three years of *RoboCup*@Home, (and they also plan to participate in 2009), and 6 teams have participated in two of the competitions.

Another important parameter to assess the results of the competition is the increase in performance. Obviously, it is difficult to determine such measure quantitatively. The main reason is that the constant evolution of the competition and the iterative modification of both the rules and the partial scores do not allow for a direct comparison.

However, it is possible to identify certain situations which indicate the success of the initiative in terms of general performance increase. Table 4.6 gives some examples for this increase over the last three years. The first row contains the percentage of unsuccessful tests, i.e., tests where no score was achieved at all, dropping from 83% in 2006 to 41% in 2008. The second row shows the increase in the total number of tests per competition. The third row indicates the average number of tests that teams participated in successfully (i.e., with a non-zero score). The enormous increase from from 1.0 tests in 2006 to 4.9 in 2008 is a strong indication of an average increase in robot abilities and in overall system integration.

Measure	**2006**	**2007**	**2008**
Percentage of 0-score performance	83%	64%	41%
Total number of tests	66	76	86
Avg. number of succ. tests p. team	1.0	2.5	4.9

Table 4.6: Measures indicating general increase of performance

4.4.4 Scientific achievements

In addition to numerical analyses of test performances, relevant scientific achievements have been obtained by teams participating in the competition. *RoboCup*@Home provides a suitable setting for developing and testing integrated solutions for mobile service robots. As a result, robot hardware and software architectures evolve over time.

This effort is demonstrated in scientific papers and in the teams' reports (Team Description Papers), which contain technical and scientific details on the hardware/software architectures and the implemented approaches and functionality. In particular, due to the nature of the *@Home* competition, in these architectures special focuses are put on Human-Robot-Interaction (e.g. [100]), on personal assistive robots (e.g. [101]) and on high level programming for domestic service robots (e.g. [102]).

Scientific advancements can be also identified in specific functionality. Speech recognition evolved from difficult interaction with headsets and portable laptops (2006-2007) to speaker-independent speech recognition with effective noise cancellation using onboard microphones (2008) [103]. Face recognition has been made robust in the presence of spectators standing around the edges of the scenario [104, 105] and tuned for real-time use [106] (Figure 4.2 left). Object recognition in *@Home* requires a more general approach than the color-based recognition used in the soccer leagues, and it offers a challenging testbed. Techniques using different feature extractors and different matching procedures have been tested (e.g. [107]), reaching a level in which the robot can reliably remember an object shown by a user (by holding it in front of the robot) and then recognize it among several others (2008, Figure 4.2 right). Gesture detection and recognition has also been studied in order to communicate with the robot, and uses an effective approach based on active learning [108]. Finally, object manipulation has evolved from gathering a newspaper from the floor (2006), to grasping cups from a table (2007), to grasping different objects at various heights (2008) (Figure 4.4). A list scientific publications from *RoboCup*@Home teams can be found in the league Wiki[1].

A measure of the scientific contributions is also given by the five papers (out of 56) related to *RoboCup*@Home presented to the International RoboCup Symposium 2008, including one that received the best student paper award [103]. In comparison with all the RoboCup leagues and sub-leagues, *@Home* ranked third out of ten with respect to the number of papers presented at the RoboCup Symposium (together with Soccer Middle-Size and Soccer Simulation).

[1]List of @Home publications: http://robocup.rwth-aachen.de/athomewiki/index.php/Publications

4.4.5 Community

RoboCup@Home does not only involve the aspect of competition, but it has also a strong focus on building a community exchanging knowledge and technology. This community plays a substantial role, because of the following reasons:

- The specifications of the tests and of the scenario are kept to a minimum to meet the aim of realistic tasks and the involvement of a defined amount of uncertainty. Therefore, the interpretation of the rules and a common vision on the goals to achieve must rely on common sense.

- The constant evolution and enhancement of the competition is mainly based on the input and feedback from the community towards new concepts and procedures.

- The large, real-world problem space in which the league is operating calls for interdisciplinary exchange of know-how, as problems can hardly be solved by a single group alone. This fosters the integration of existing components in combination with new specific approaches. The exchange, use and combination of standardized and modular system components from inside and outside the community is expected to accelerate technological and scientific progress significantly.

- Establishing contact and exchange between science and industry should accelerate product and application development in DSR.

RoboCup@Home makes use of standard Internet tools to exchange technical knowledge and organize information. The web site[1] is dedicated to the initiative containing both the current information about the next competition, as well as historical data. The mailing list[2] is used for general communication to and within the community, including organization information, rule discussions, technical help, calls for scientific contributions, etc. In addition, a Wiki[3] for the *@Home* initiative has been created with the goal of becoming a standard knowledge pool for international domestic service robotics research and development. The Wiki acts as as a platform for technological and scientific knowledge transfer on hardware, software, methods and abilities among the teams, and as a helpful starting point for new teams.

The community is growing fast. The mailing list currently has 277 subscribers (June 2009), and the number and kind of subscriptions indicate that the mailing list is

[1] http://www.robocupathome.org
[2] robocupathome@iais.fraunhofer.de, https://lists.iais.fraunhofer.de/sympa/info/robocupathome
[3] RoboCup@Home Wiki (http://robocup.rwth-aachen.de/athomewiki)

not only used by the teams but also by various people from research institutions, other communities, universities, media and companies.

Up to date, the *@Home* Wiki received about 34,000 page views and more than 400 page edits since it was set up at the end of 2007. The most popular pages are the software page (2,550 views) and the hardware page (2,170 views), which strongly indicates that knowledge is actually being exchanged in the community.

Finally, attention to the *RoboCup*@Home activities in the media and press has increased, thanks to the many worldwide and regional events in which the competition has taken place. Various videos[1] and images[2] of past *RoboCup@Home* events are available online.

4.5 Conclusion and Outlook

This chapter presented the *RoboCup*@Home initiative as a community effort to develop and benchmark domestic service robots through scientific competitions. To do so, *system benchmarking* is employed to evaluate a robot's performance in a realistic, complex and dynamic environment. The general setting is designed to exhibit a high degree of uncertainty that the robots have to deal with.

The rules of the competition aim to implement the benchmark by means of general rules and a set of specific tests. Evaluation is conducted along a set of key features. These features, divided into *functional abilities* and *system properties* need to be met in order to be successful in the competition. The modular and open character of the competition's framework allows for an iterative adaptation of features and tests according to the observed and measured benchmark performances. Special focus is put on establishing a community to foster interdisciplinary exchange of knowledge and technology. Furthermore, this community is essential to create common vision and understanding for the problems and goals of the *@Home* initiative, and to give feedback for the iterative development of the competition. Starting with the first competition in 2006, the overall development of the initiative with respect to performance increases, the growing community, knowledge exchanges and public awareness has been very promising over the past three years. *@Home* has become the largest international competition for domestic service robots, with currently five national competitions in China, Japan, Germany, Iran and Mexico besides the annual world championships. Competitions in South America and the US are expected to be introduced in 2009.

[1] Videos of the 2008 competition (http://www.youtube.com/user/RoboCupAtHome)
[2] Images of various @Home events (http://picasaweb.google.com/RoboCupAtHome)

The future development of the *@Home* competition is highly iterative, as it involves constant feedback from the community, adjustments on the focus of desired abilities and changes of the rules.

In general, the tests, functional abilities and desired system properties will evolve over the years and will be combined to form more realistic high-level tasks. New tests with different focuses and higher complexity will be added in the future, depending on the results of previous years. The discussion on how to ensure a comparable measure of performance in the benchmarks, in the presence of a high level of desired uncertainty should be intensified. Short, mid and long-term goals are necessary, as they help identify and approach the problem in the large, real-world problem space in a structured way. At the moment the focus is on physical capabilities such as manipulation, human recognition and navigation. In the future, more focus will be put on artificial intelligence and mental capabilities in the context of HRI. This includes situational awareness, online learning, understanding and modeling the surrounding world, recognizing human emotions and providing appropriate responses.

The increase of complexity in the competition from 2007 to 2008 was rather high. Therefore, the Technical Committee of the *@Home* league agreed to make only minor modifications to the rules in 2009. Rule changes for 2009 will involve an increased focus on HRI, e.g. combined use of speech and gestures, robot operation by laymen, or following previously unknown persons. Application scenarios will become more realistic, e.g. the demo challenge will involve robots serving drinks and food at a real party setting involving many people unfamiliar with the robots. Furthermore, uncertainty and dynamics in the environment are increased by changing object positions more frequently, having more people in the scenario, and leaving the scenario with the robots.

Further, an annual *@Home* camp is planned to be established. It will consist of a set of lectures and practical sessions from and for members of the community. Having a separate event exclusively for knowledge exchange in the absence of any competitive aspect is expected to foster exchange of knowledge even more. Also, new research groups and communities will be addressed and invited to join and share their knowledge with the *@Home* community. Midterm goals include the search, identification, design and use of a common robot software architecture or framework to better exchange and reuse software components already developed in the community and beyond. The same holds true for hardware, where companies or groups with relevant hardware components like sensors, actuators, or even standard robot platforms will be identified and asked to join and to support the community.

Another midterm goal is gradually testing the robots in the real world, e.g. going shopping in a real supermarket or taking public transportation. Moreover, usability and appearance of the robots will be of higher importance if one wants to increase their public acceptance. The future *@Home* scenario will contain more high-level and continuous interaction with humans living together with the robot and will evolve towards more synergistic human-robot teams, as depicted in the studies presented by Burke et al. [109]. Moreover, an increased use of ambient intelligence is planned, which the robots have to interact with. The use of the Internet as a general knowledge base, and the communication with household devices, TVs, or external video cameras are some examples.

In general, the competition will move towards a high-level integration of the identified abilities into more realistic and relevant applications. This is expected to increase attractiveness, to generate more public awareness and hopefully it will inspire and accelerate affordable and useful consumer product development for domestic service robotic applications in the near future.

5

Conclusions

In this thesis, I presented approaches and methods for applying standardization and benchmarking in the field of mobile service robotics which aim to advance and accelerate research and development, and to foster prototyping of new service robotics applications. The thesis covers three major aspects in this context: First, the definition and implementation of a concept for a modular robot construction kit in hardware and software for application prototyping that I called *VolksBot* (see chapter 2). Second, sensor and application development for mobile robots including the catadioptric vision system *IAISVision* and the 3D-LIDAR system *3DLS-K* (see chapter 3). Third, the definition and implementation of a set of standard benchmark tests for personal domestic service robots, which formed the basis of the international service robotics competition called *RoboCup*@Home (see chapter 4). These three aspects are summarized again in the following.

In chapter 2, I presented the definition and the design of a modular mechatronic construction kit used to prototype mobile service robot applications. After specifying general design goals and deriving design criteria for the implementation of the kit, I defined a modular multilayer hardware architecture, which follows the previously defined criteria. The architecture adheres to certain requirements, such as high reconfigurability, minimizing dependencies between modules and definition of interfaces between the components. According to this architecture, I implemented a component library in CAD which allows efficient iterative robot design via consequent reuse and high reconfigurability of standard hardware components. Regarding electric and electronic hardware, I composed a hardware component library which holds commercial products as well as in-house developments such as sensors, actuators or control hardware with defined interfaces in hardware and software. In software, I used a commercial framework for the visual composition of modular signal graphs, *ICONNECT* which was enhanced with software modules containing robot-specific functionality. The further implementation of an operating system independent software library called *FAIRlib* allowed for the reuse and exchange of robotic-related software modules beyond the *VolksBot* project and the use of *ICONNECT*. In the proceeding of the chapter, I presented the hardware enhancement of the construction kit to obtain higher physical performance and flexibility in the robot design. This enhancement allowed the implementation a variety of rough terrain, outdoor and heavy duty robot platforms, *VolksBot RT*. To account for high mobility, I applied an approach for evolutionary design optimization on the robot's morphology in simulation, further resulting in the physical implementation of the *VolksBot XT* variant capable of moving over rugged terrain and climbing staircases. Various application prototypes have been created on the basis of the *VolksBot* construction kit. Applications to RoboCup include Rescue, Soccer and Domestic Service Robotics. Further service robotic applications include autonomous transportation, a fuel cell powered service robot and underwater robotics. In addition, the concept has been successfully applied in research and education. As further results, I demonstrated that the previously defined design goals have been met by the implementation of the construction kit on a large scale. Measures of the time required for platform assembly further support the claim for rapid prototyping. The chapter concludes with an overview of the distribution and use of the *VolksBot* kit and its components in the international robotic community. A total of about 200 *VolksBot* robots and 200 components have been built, distributed and used since 2003. Future work will include the consequent enhancement of the construction kit to allow for new robot variants and applications.

In chapter 3, I detailed the design and application of two sensor systems originally developed for the *VolksBot* construction kit, namely the *IAISVision* system and the 3D LIDAR system *3DLS-K*. First, I described the modular hardware design of the catadioptric vision system *IAISVision*. The iterative design approach is based on simulation by ray-tracing and allows one to optimize relevant geometrical system parameters according to the demands and the environment characteristics of the individual application. The *IAISVision* system, as well as its hyperbolic mirror, have been used by more than 30 research labs worldwide and in different robotic applications such as *RoboCup Soccer(MSL)*, robot rescue, surveillance and autonomous driving. Further on, I presented an adaptive color perception method which aims for color constancy under varying light conditions, including artificial and natural light. The method includes PI control of camera parameters, segmentation by Markov Random Field and classification based on Mahalanobis distance. The PI controller provided enough color constancy to be able to fuse the distribution under different light conditions and to generate reference color models for indoor and outdoor situations. These reference color models have been shown to provide a robust basis for color classification under a variety of different light conditions. The big difference in color distribution in indoor and outdoor situations suggests the use of separate reference models for these two cases. The vast illumination range occurring outdoors within one image has shown the physical limitations of the camera. Future work will investigate the possible use of attention-based mechanisms to choose from different parameter sets for different light situations.

In section 3.2, I presented the system development and application of the 3D-LIDAR system Fraunhofer *3DLS-K*. After detailing the hardware development of the sensor and the acquisition of 3D range data, I presented the sensor application in the domain of autonomous urban driving. In this context, two methods are discussed in detail: First, a method for classifying flat terrain by use of the *Scan Line Approximation* algorithm, and second, a method for scene analysis of the 3D data at intersections, including an correlative behavior system on the basis of a finite state machine. The system was tested in simulation, on sensor traces and on the autonomous car *Spirit of Berlin*, from FU Berlin, in various intersection situations with other participating cars. The final evaluation of the approach was done at the national qualification event and the semi-finals of the *DARPA Urban Challenge* in Victorville, California, in 2007. The system proved to work correctly and reliably in various intersection situations. The approach and the results were presented at the workshop on 3D-Mapping at the IEEE International Conference on Intelligent Robots and Systems (IROS2008) [91].

Chapter 4 describes an approach for defining, developing and benchmarking new personal service robot applications through scientific competitions. This has resulted in the conceptual design and implementation of the *RoboCup*@Home competitions. Established in 2006, it has become the largest international domestic service robotic competition to date. *Statistic system benchmarking* has been employed to evaluate a robot's performance in a realistic, dynamic and uncertain environment. The rules of the competition aim to implement the benchmark by means of general rules and a set of specific tests. Evaluation is conducted along a set of relevant key features for mobile service robots. The modular and open character of the competition's framework allows for an iterative adaptation of features and tests according to the observed and measured benchmark performances of previous years. Further focus is put on establishing a community to foster interdisciplinary exchange of knowledge, to iteratively improve the benchmarks through direct feedback and to create a common scope on problems and aims. This is essential due to the large problem space and the high grade of uncertainty the robots have to operate in. Comparability of the benchmark results is maintained by ensuring a defined range of uncertainty and by applying strict evaluation criteria on individual robot abilities, which are tested multiple times during a competition in different contexts. As a result, the increase in performance, the iterative development and improvement of the benchmarks, as well as the growth of the community and its scientific contribution, have been analyzed and discussed. Since 2006, 25 teams from Asia, Europe, the US, South-America and Australia have participated in the world championship, and national competitions in China, Mexico, Iran, Japan and Germany have been established. For 2009, 24 teams from 14 countries are qualified. Novel and relevant personal service robotic applications and technology, as well as scientific contributions, are beginning to emerge from within the growing community, which currently has 270 members.

In summary, this thesis proposes a combination of standardization, modularization, benchmarking and community building in mobile service robotic research and development as an integral approach to cope with the high level of related complexity in the domain. Especially during the last several years, significant progress has been made in this context: Standardized technology like platforms, hardware and software components have been made available, and scientific communities and benchmarking activities have been established and are growing in importance. This has already led to a significant increase in the robots' system performance. Moreover several consumer products have already been successfully placed on the market. This further indicates the high potential and future relevance of autonomous robot technology, which is also confirmed by experts from both scientific and economic fields. The current demographic development and the increase of public awareness further support this claim. By providing an approach for efficient robot prototyping for autonomous service robots - including application-oriented sensor development - and by establishing a standard benchmark and a community for the development of domestic service robots, I hope to have contributed to accelerate progress toward useful and widely accepted application development of intelligent mobile service robots in the near future.

6

Summary of contributions

In this thesis, I presented approaches and methods for applying standardization and benchmarking in the field of mobile service robotics which aim to advance and accelerate research and development of relevant robot applications. In summary, my main contributions in this thesis include the development and application of the *VolksBot* mobile robot construction kit and sensor development of the adaptive catadioptric vision system *IAISVision* and the 3D-LIDAR system *3DLS-K*. I presented the design and implementation of a new benchmark for domestic service robots, the *RoboCup*@Home initiative. In particular, the contributions are the following:

Standardization initiatives in mobile robot development and benchmarking In chapter 1 I summarized the state of the art in standardization efforts in mobile robot development and benchmarking. This includes the aspects of standardized software architecture, frameworks and middle-ware, simulation, robot construction kits and development platforms, as well as standard benchmarks and competitions. Additionally, section 2.1 provides further insight into the state of the art in modular robot design, section 2.4.1 details the current status in rough terrain robot locomotion and section 4.2 provides additional information on mobile robot benchmarking activities and competitions.

Several contributions concern the *VolksBot* robot construction kit presented in chapter 2:

Design goals and criteria for a mobile robot construction kit In section 2.2, I defined general design goals for a mobile robot construction kit considering the challenges and problems mentioned in section 1.1. From these design goals, I derived specific design criteria for the implementation of the modular robot construction kit I called *VolksBot*. Furthermore, the correlation between these goals and criteria is discussed. I further enhanced these design criteria in section 2.4.2 to meet the demands of rough terrain locomotion and high payloads. As a part of the results, I analyzed whether the initially defined design goals were met by the current implementation of the construction kit.

Specification of a multilayer hardware architecture for modular robot design In section 2.3.1 I specified a four-layer architecture for the *VolksBot* construction kit that fosters reusability of components and modules and that avoids dependencies between the modules. This way, a high configurability in robot design is obtained, allowing for the implementation of a wide range of robot variants, which are further presented in section 2.4.3.

Implementation of hardware component libraries After specifying this hardware architecture, in section 2.3.2, I implemented a CAD component library which allows efficient mechanical design of robot variants prior to the actual assembly process. Furthermore, in section 2.3.3 I specified a set of standard electric and electronic components to form a hardware component library for the *VolksBot* construction kit.

Modular software design In section 2.3.4 I presented the implementation of a modular software design approach for the construction kit. This approach involves the extension and use of the existing *ICONNECT* framework for modular composition of signal graphs, as well as the specification and implementation of a platform-independent software library for autonomous robot control called *FAIRlib*.

Enhancements for increased physical performance and flexibility In section 2.4, I introduced new hardware components such as the *Universal Drive Unit* which significantly enhance the physical performance of the construction kit in terms of mobility and payload. These enhancements allow for the development of a set of new VolksBot variants, *VolksBot Rough Terrain*, which have been introduced in section 2.4.3.

Evolutionary design optimization for enhanced mobility In section 2.5 I presented an approach for evolutionary optimization of physical robot design for rough terrain locomotion. This approach involves the evolution of specific geometric parameters of a *VolksBot* assembly in a physical simulation environment. The performance of the obtained solution in simulation has been verified by a physical implementation and performance evaluation of the platform.

Application prototyping and variant design with the VolksBot kit As evidence for the multipurpose character in the domain of application-oriented prototyping and the flexibility of the kit, I presented in total 14 different platform implementations including standard platforms and application prototypes on the basis of the *VolksBot* construction kit. Furthermore, I conducted the implementation of assembly manuals for five *VolksBot* standard variants.

Contributions regarding the development and application of the *IAISVision* system are the following:

Hardware design of the catadioptric vision system IAISVision In section 3.1.1 I presented an iterative approach for the custom hardware design of the catadioptric vision system *IAISVision*. This approach involves the optimization of geometrical system parameters in simulation (ray-tracing), custom design and manufacturing of the hyperbolic mirror.

Adaptive color camera calibration under changing light conditions In section 3.1.3.3 I presented an approach for camera auto-calibration aiming for maintaining color constancy under varying light conditions. The approach uses reference color rings placed on a catadioptric camera system and parallel PI-control of the intrinsic camera parameters.

Analysis of experimental results of the adaptive vision system In section 3.1.4, I presented an analysis of experimental results regarding color constancy and color classification which I conducted under varying light conditions indoors and outdoors using the adaptive color vision system.

Regarding the development and application of the 3D-LIDAR system *3DLS-K*, the contributions are the following:

Hardware development of the 3D LIDAR system 3DLS-K In section 3.2.1 I presented the hardware design and implementation of a novel continuously rotating 3D-LIDAR system called *3DLS-K* which is based on two industrial 2D laser range finders.

Synchronizing and referencing absolute and relative position information of the scanner rotation angle and the 2D range information allows the generation of 3D point clouds through transformation into the scanner's coordinate system (see section 3.2.2).

LIDAR based intersection analysis for autonomous urban driving Section 3.2.4 details the application of the 3D-LIDAR system 3DLS-K in the domain of autonomous urban driving. On the basis of the acquired 3D range information, I developed an approach for intersection analysis including obstacle detection and an a general behavior system which implements official traffic rules in the form of a finite state machine. The system was integrated in and tested in the autonomous car of *Team Berlin*, which reached the semi-finals of the *DARPA Urban Challenge* in 2007.

Contributions in the domain of benchmarking of autonomous service robots are the following:

Conceptual design of a framework for benchmarking autonomous service robots According to the demands and challenges discussed in section 1.1, in section 4.3.1, I specified a set of conceptual criteria which act as a general guideline for the further implementation of the *RoboCup*@Home initiative. On the basis of these criteria, I defined a set of key features (see section 4.3.2) which specify the technical and scientific scope for the individual benchmark tests to be implemented.

Implementation of a competition for benchmarking autonomous service robots Section 4.3.3 gives insight into the implementation of the *RoboCup*@Home competitions. It includes the specification of a scoring system, as well as general and individual rules for twelve benchmark tests. With a significant increase in participants, the number of regional events worldwide in benchmarking performance and scientific contribution (see section 4.4) *RoboCup@Home* has become the largest international competition for benchmarking domestic service robots to date.

Between 2004 and 2009, I wrote several articles which were or are to be published in journals and books, and for conferences and workshops. These publications are summarized in the following:

Publications Publications on the design of the *VolksBot* robot construction kit include a contribution at the International RoboCup Symposium 2005 [30], which introduces the general concept and presents initial results on the application of the concept to *RoboCup MSL* and *RoboCup Rescue*. Initial publications also include a contribution to the CLAWAR/EURON Workshop on Robots in Entertainment [110] and to the conference *Informatik 2005* [111]. A contribution to the journal *it - information technology* [29] focuses on the extended design and variants of the *VolksBot RT* series and application prototyping by example of the *PeopleMover* project. A contribution to the International Workshop on Safety, Security and Rescue Robotics in 2006 [55] details the expansion of the construction kit for rough terrain and high mobility and its application to the domain of rescue robotics. Finally, a chapter in the book *Robotic Soccer* [94] summarizes and discusses the approach and the results of the VolksBot project with a special focus on prototyping service robotic applications and the relevance for *RoboCup Soccer, Rescue* and *RoboCup@Home*.

A contribution to the International RoboCup Symposium 2007 [65] includes the development of the IAISVision system and its application to color vision under changing light conditions. Results on the development of the 3D LIDAR system *3DLS-K* and its application to autonomous urban driving were presented at the Workshop on 3D-Mapping at the IEEE International Conference on Intelligent Robots and Systems, IROS 2008 [91].

Regarding *RoboCup@Home*, the general, initial concept of a framework for benchmarking autonomous service robots was presented at the RoboCup International Symposium 2005 [92]. A chapter in the book *Robotic Soccer* [64] discusses the relevance of *@Home* for RoboCup and presents the initial implementation of the initiative. A contribution to the *Workshop on Home Robotics* [112] was made at the International Conference on Intelligent Robots and Systems 2008. Finally, a contribution to the Journal *Interaction Studies* [93] motivates, summarizes and analyzes the concept and the performance of the RoboCup@Home initiative to date.

References

[1] N. J. Nilsson, "Shakey the robot," Tech. Rep. 323, Stanford Research Institute, 1984. 3

[2] B. Gates, "A robot in every home," *Scientific American*, vol. 12, 2006. 3

[3] A. Cockburn, *Agile Software Development.* Addison-Wesley Professional, December 2001. 5

[4] M. Montemerlo, N. Roy, and S. Thrun, "Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (carmen) toolkit," in *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2436–2441, 2003. 6, 103

[5] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the 11th International Conference on Advanced Robotics*, pp. 317–323, 2003. 6, 103

[6] H. Bruyninckx, "Open robot control software: the orocos project.," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2523–2528, IEEE, 2001. 6, 103

[7] G. R. Bradski and V. Pisarevsky, "Intel's computer vision library: Applications in calibration, stereo, segmentation, tracking, gesture, face and object recognition," in *CVPR*, vol. 2, pp. 796–797, 2000. 6, 103

[8] R. Smith, "Open dynamics engine," tech. rep., Department of Computer Science, Lund University, 2004. 6, 31, 45, 47

[9] S. Balakirsky, "Usarsim: Providing a framework for multi-robot performance evaluation," in *Proceedings of the Performance Metrics for Intelligent Systems Workshop (PerMIS'06)*, pp. 98–102, 2006. 6, 103

[10] M. Ferrari, G. Ferrari, and R. Hempel, *Building Robots With Lego Mindstorms.* No. ISBN: 1928994679, Syngress, 1st ed., December 2001. 7, 12

[11] U. Gerecke, B. Wagner, and P. Hohmann, "A modular educational robotic toolbox to support university teaching efforts in engineering," in *Proceedings of the International Conference on Engineering Education (ICEE 2003)*, (Valencia, Spain), July 2003. 7, 13

[12] S. Enderle, S. Sablatnög, S. Simon, and G. Kraetzschmar, "Tetrixx: a robot development kit," in *First International Workshop on Edutainment Robots, 2000.* (T. Christaller, G. Indiveri, and A. Poigné, eds.), 2000. 7, 12

[13] U. Gerecke, P. Hohmann, and B. Wagner, "Concepts and components for robots in higher education," in *Proceedings of the World Automation Congress (WAC'2004)*, (Sevilla, Spain), July 2004. 7, 13

[14] A. del Pobil, "Why do We Need Benchmarks in Robotics Research?," in *Proc. of the Workshop on Benchmarks in Robotics Research, IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006. 7, 105

[15] S. Sabanovic, M. Michalowski, and R. Simmons, "Robots in the wild: observing human-robot social interaction outside the lab," in *9th IEEE International Workshop on Advanced Motion Control*, pp. 596–601, 2006. 7, 105

[16] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. 7, 105

[17] G. Fontana, M. Matteucci, and D. G. Sorrenti, "The RAWSEEDS proposal for representation-independent benchmarking of SLAM," in *RSS Workshop on Experimental Methodology and Benchmarking in Robotics Research*, 2008. 7, 105

[18] J. Baltes, "A benchmark suite for mobile robots," in *Proceedings of IROS-2000*, November 2000. 7, 105

[19] N. Munoz, J. Valencia, , and N. Londono, "Evaluation of navigation of an autonomous mobile robot," in *Proc. of Performance Metrics for Intelligent Systems Workshop (PerMIS)*, p. 1521, 2007. 7, 105

[20] D. Calisi, L. Iocchi, and D. Nardi, "A unified benchmark framework for autonomous Mobile robots and Vehicles Motion Algorithms (MoVeMA benchmarks)," in *RSS Workshop on Experimental Methodology and Benchmarking in Robotics Research*, 2008. 7, 105

[21] P. H. Kahn, H. Ishiguro, B. Friedman, T. Kanda, N. G. Freier, R. L. Severson, and J. Miller, “What is a human? toward psychological benchmarks in the field of humanrobot interaction,” *Interaction Studies*, vol. 8, no. 3, p. 363390, 2007. 7, 106

[22] D. Feil-Seifer, K. Skinner, and M. J. Mataric, “Benchmarks for evaluating socially assistive robotics,” *Interaction Studies*, vol. 8, pp. 423–439(17), 2007. 7, 106

[23] T. Balch and H. A. Yanco, “Ten years of the aaai mobile robot competition and exhibition: looking back and to the future,” *AI Magazine*, vol. 23, no. 1, pp. 13–22, 2002. 8, 106, 107

[24] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara, “RoboCup: A Challenge Problem for AI,” *AI Magazine*, vol. 18, no. 1, pp. 73–85, 1997. 8, 106

[25] E. Prassler, M. Hägele, and R. Siegwart, *Field and Service Robotics*, ch. International Contest for Cleaning Robots: Fun Event or a First Step towards Benchmarking Service Robots, pp. 447–456. 2006. 8, 108

[26] J. Won, K. DeLaurentis, and C. Mavroidis, “Rapid prototyping of robotic systems,” in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, 2000. 12

[27] G. Reshko, M. Mason, and I. Nourbakhsh, “Rapid prototyping of small robots,” Tech. Rep. CMU-RI-TR-02-11, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 2002. 12

[28] A. Birk, “Fast robot prototyping with the cubesystem,” in *Proceedings of the International Conference on Robotics and Automation, ICRA’2004*, IEEE Press, 2004. 12

[29] T. Wisspeintner and A. Bose, “The volksbot concept - rapid prototyping for real-life applications in mobile robotics,” *it - Information Technology*, no. 5, Schwerpunktheft Autonome Fussball-Roboter, pp. 274–281, 2005. Class-AA. 17, 143

[30] T. Wisspeintner, W. Nowak, and A. Bredenfeld, “Volksbot - a flexible component-based mobile robot system,” in *Proceedings of the International RoboCup Symposium 2005* (A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, eds.), Lecture Notes in Computer Science, pp. 716–723, Springer, 2005. 17, 143

[31] F. P. B Klaassen, K Zahedi, "A modular approach to construction and control of walking robots," in *VDI-Berichte 1841*, VDI, 2004. 25

[32] J. Rojo, R. Rojas, K. Gunnarsson, M. Simon, F. Wiesel, F. Ruff, L. Wolter, F. Zilly, N. Santrac, T. Ganjineh, A. Sarkohi, F. Ulbrich, D. Latotzky, B. Jankovic, G. Hohl, T. Wisspeintner, S. May, K. Pervoelz, W. Nowak, F. Maurelli, and D. Droeschel, "Spirit of berlin: An autonomous car for the darpa urban challenge hardware and software architecture," tech. rep., Rice University, Houston, TX, 2007. 27, 90

[33] A. Sicheneder, A. Bender, e. Fuchs, M. Mendler, and B. Sick, "A framework for the graphical specification and execution of complex signal processing applications," in *Proc. of the Conference on Accoustics Speech and Signal Processing*, 1998. 28

[34] R. Mandl and B. Sick, *Messen, Steuern und Regeln mit ICONNECT*. No. ISBN: 3528058129, Vieweg Verlag, June 2003. 28

[35] I. The MathWorks, *Simulink Users Guide*. The MathWorks, Inc, 2004. 28

[36] C. J. Kalkman, "Labview: A software system for data acquisition, data analysis, and instrument control," *Journal of Clinical Monitoring and Computing*, vol. 11, pp. 51–58, 1995. 28

[37] Object Management Group, "Common object request broker: Architecture and specification," tech. rep., Object Management Group, Framingham, Massachusetts, 1991. 30

[38] T. Jäger, H. & Christaller, "Dual dynamics: Designing behavior systems for autonomous robots," in *The Sixth International Symposium on Artificial Life and Robotics*, 1997. 30

[39] A. Bredenfeld and G. Indiveri, "Robot behavior engineering using dd-designer," in *IEEE/RAS International Conference on Robotics and Automation (ICRA 2001)*, 2001. 30

[40] G. Bradski, "The opencv library," *Dr. Dobbs Journal*, vol. Nov. 2000, pp. 120–126, 2000. 30

[41] D. J. Higham and N. J. Higham, *MATLAB Guide*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000. 31

[42] P. Besl and H. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992. 34

[43] H. Surmann, K. Pervölz, A. Nüchter, K. Lingemann, J. Hertzberg, and M. Hennig, "Simultaneous mapping and localization of rescue environments," *it - Information Technology*, vol. 47, pp. 282 – 291, 2005. 34

[44] X. Jiang and H. Bunke, "Edge detection in range images based on scan line approximation," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 183–199, 1999. 34, 89, 90

[45] F. Hardarson, "Locomotion for difficult terrain," tech. rep., Mechatronics Division, Department of Machine Design, Royal Institue of technology, Stockholm, Sweden., 1997. 34

[46] M. G. Dodson, S. L. Owsley, and S. J. Moorehead, "Goat (goes over all terrain) vehicle: a scaleable robotic vehicle," in *Proc. of SPIE*, 2003. 34

[47] H. Kitano, S. Tadokor, H. Noda, I. Matsubara, T. Takhasi, A. Shinjou, and S. Shimada, "Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research," in *Proceedings of the IEEE Conference on Systems, Men, and Cybernetics, 1999.*, 1999. 35

[48] S. Guccione, G.Muscato, G.Nunnari, G. Virk, A. Azad, A. Semerano, M. Ghrissi, T.White, and C. Glazebrook, "Robots for volcanos: The state of the art," in *Proceedings of the 3rd International Conference on Climbing and Walking Robots (CLAWAR 2000), pp.777-788, Madrid (Spain), 2-4 October 2000.*, 2000. 35

[49] S. K. Jun and V. N. Krovi, "Design considerations for an articulated leg-wheel locomotion subsystem," in *Proceedings of IMECE2004 2004 ASME International Mechanical Engineering Congress and RD&D Expo*, 2004. 35

[50] V. Krovi, G. Ananthasuresh, and V. Kumar, "Kinematic and kinetostatic synthesis of planar coupled serial chains," *ASME Journal of Mechanical Design, 124(2)*, pp. 143–155., 2002. 35

[51] R. Jarvis, "An articulated six wheeled drive robot for very tough terrain navigation," in *Proceedings of Australasian Conference on Robotics and Automation*, 2002. 35

[52] A. Halme, I. Leppänen, M. Montonen, and S. Ylönen, "Robot motion by simultaneously wheel and leg propulsion," in *Proceedings of the 4th International Conference on Climbing and Walking Robots (CLAWAR 2001), 24th-26th September 2001, Karlsruhe, Germany*, 2001. 35

[53] T. Estier, Y. Crausaz, B. Merminod, M. Lauria, R. Piguet, and R. Siegwart, "An innovative space rover with extended climbing abilities," in *Proceedings of Space & Robotics, the Fourth International Conference and Exposition on Robotics in Challenging Environments*, 2000. 42

[54] T. Estier, R. Piguet, R. Eichhorn, and R. Siegwart, "Shrimp, a rover architecture for long range martian mission," in *Proceedings of the Sixth ESA Workshop on Advanced Space Technologies for Robotics and Automation*, 2000. 42

[55] T. Wisspeintner, A. Bose, and P. G. Ploeger, "Robot prototyping for rough terrain applications and high mobility with volksbot rt," in *Proceedings of the International Workshop on Safety, Security and Rescue Robotics (SSRR '06)*, (Gaithersburg , Maryland , USA), August 2006. Conf-A. 43, 143

[56] P. J. Bentley, "Aspects of Evolutionary Design by Computers," in *Soft Computing - Engineering Design and Manufacturing, Springer-Verlag, London*, pp. 99–118, 1999. 45, 46

[57] M. Hülse, S. Wischmann, and F. Pasemann, "Structure and function of evolved neuro-controllers for autonomous robots," in *Connection Science*, vol. 16, pp. 249–266, Taylor and Francis Ltd, December 2004. 46, 48

[58] C. Leger, *Darwin2K An Evolutionary Approach to Automated Design for Robotics*. Kluwer Academic Publishers, 2000. 46

[59] P. Funes and J. Pollack, *Evolutionary Design by Computers*, ch. Computer evolution of buildable objects, pp. 387–403. Morgan Kaufmann, San Francisco, 1999. 46

[60] L. Lichtensteiger and P. Eggenberger, "Evolving the morphology of a compound eye on a robot," in *Proceedings of Eurobot'99*, 1999. 46

[61] C. Paul and J. Bongard, "The road less travelled: Morphology in the optimization of biped robot locomotion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, (Piscataway, NJ), pp. 226–232, IEEE Press, 2001. 46

[62] J. B. Pollack, H. Lipson, S. Ficci, P. Funes, G. Hornby, and R. A. Watson, "Evolutionary Techniques in Physical Robotics," in *ICES*, pp. 175–186, 2000. 46

[63] J. Borenstein and Y. Koren, "The vector field histogram fast obstacle avoidance for mobile robots," in *Proceedings of the IEEE/RAS 91*, 1991. 54

[64] T. van der Zant and T. Wisspeintner, *Robotic Soccer*, ch. RoboCup@Home: Creating and Benchmarking Tomorrows Service Robot Applications, pp. 521–528. I-Tech Education and Publishing, 2007. 57, 101, 143

[65] Y.Takahashi, W. Nowak, and T. Wisspeintner, "Adaptive recognition of color-coded objects in indoor and outdoor environments," in *Proceedings of the RoboCup International Symposium 2007*, 2007. 69, 143

[66] L. Maloney and B. Wandell, "Color constancy: A method for recovering surface spectral reflectance," *Journal of the Optical Society of America*, vol. 3, pp. 29–33, 1986. 72

[67] D. Forsyth, "A novel algorithm for color constancy," *Int. Journal of Computer Vision*, vol. 5, no. 1, pp. 5–36, 1990. 72

[68] K. Barnard, B. Funt, and V. Cardei, "A comparison of computational colour constancy algorithms; part i: Methodology and experiments with synthesized data," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 972–984, 2002. 72

[69] G. Finlayson and S. Hordley, "Improving gamut mapping color constancy," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1774–1783, 2000. 72, 73

[70] H. Austermeier, G. Hartmann, and R. Hilker, "Colour-calibration of a robot vision system using self-organising feature maps," in *ICANN'96 Proc. Int Conf on Artificial Neural Networks*, pp. 257–262, 1996. 72

[71] G. Mayer, H. Utz, and G. Kraetzschmar, "Playing robot soccer under natural light: A case study," in *Proceedings of the RoboCup International Symposium 2003* (D. Polani, B. Browning, and A. Bonarini, eds.), Lecture Notes in Computer Science, (Berlin), pp. 238 – 249, Springer, 2003. 73

[72] D. Austin and N. Barnes, "Red is the new black - or is it?," in *Proceedings of the 2003 Australian Conference on Robotics and Automation, Brisbane, Australia*, 2003. 73

[73] B. Funt, K. Barnard, and L. Martin, "Is machine colour constancy good enough?," in *Computer Vision - ECCV'98*, Lecture Notes in Computer Science, Springer, 1998. 73

[74] E. Grillo, M. Matteucci, and D. G. Sorrenti, "Getting the most from your color camera in a color-coded world," in *RoboCup 2004: Robot Soccer World Cup VIII* (D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, eds.), Lecture Notes in Computer Science, pp. 221–235, Springer, 2004. 73

[75] K. Gunnarsson, F. Wiesel, and R. Rojas, "The color and the shape: Automatic on-line color calibration for autonomous robots," in *Proceedings of the RoboCup International Symposium 2005* (A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, eds.), Lecture Notes in Computer Science, pp. 347–358, Springer, 2005. 73

[76] P. Heinemann, F. Sehnke, F. Streicher, and A. Zell, "Towards a calibration-free robot: The act algorithm for automatic online color training," in *RoboCup 2006: Robot Soccer World Cup X* (G. Lakemeyer, E. Sklar, D. Sorrenti, and T. Takahashi, eds.), Lecture Notes in Artificial Intelligence, Springer, 2006. 73

[77] M. Jüngel, J. Hoffmann, and M. Lötzsch, "A real-time auto-adjusting vision system for robotic soccer," in *Proceedings of the RoboCup International Symposium 2003* (D. Polani, B. Browning, A. Bonarini, and K. Yoshida, eds.), Lecture Notes in Computer Science, pp. 214–225, Springer, 2003. 73

[78] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley, the robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23(9), pp. 661–692, 2006. 73

[79] V. Chikane and C.-S. Fuh, "Automatic white balance for digital still cameras," *Journal for Information Science and Engineering*, vol. 22, pp. 497–509, 2006. 73

[80] A. de Cabrol, P. Bonnin, T. Costis, V. Hugel, P. Blazevic, and K. Bouchefra, "A new video rate region color segmentation and classification for sony legged robocup application," in *Proceedings of the RoboCup International Symposium*

2005 (A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, eds.), Lecture Notes in Computer Science, pp. 436–443, Springer, 2005. 74

[81] N. Lovell, "Illumination independent object recognition," in *Proceedings of the International RoboCup Symposium 2005* (A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, eds.), Lecture Notes in Computer Science, pp. 384–395, Springer, 2005. 74

[82] T. Kikuchi, K. Umeda, R. Ueda, Y. Jitsukawa, H. Osumi, and T. Arai, "Improvement of color recognition using colored objects," in *RoboCup 2005: Robot Soccer World Cup IX* (A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, eds.), Lecture Notes in Computer Science, pp. 537–544, Springer, 2005. 74

[83] F. Anzani, D. Bosisio, M. Matteucci, and D. G. Sorrenti, "On-line color calibration in non-stationary environments," in *Proceedings of the RoboCup International Symposium 2005* (A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, eds.), Lecture Notes in Computer Science, pp. 396–407, Springer, 2005. 74

[84] M. Sridharan and P. Stone, "Towards illumination invariance in the legged league," in *Proceedings of the International RoboCup Symposium 2004* (D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, eds.), Lecture Notes in Computer Science, pp. 196–208, Springer, 2004. 74

[85] J. Becker and A. Simon, "Sensor and navigation data fusion for an autonomous vehicle," in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000. 84

[86] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (San Francisco, CA), IEEE, 2000. 84

[87] L. Wang, J. Shu, T. Emura, and M. Kumagai, "A 3d scanning laser rangefinder and its application to anautonomous guided vehicle," in *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, 2000. 85

[88] K. C. Fuerstenberg, K. C. J. Dietmayer, and U. Lages, "Laserscanner innovations for detection of obstacles and road," in *Proceedings of AMAA 2003, 7th International Conference on Advanced Microsystems for Automotive Applications, May 2003, AMAA 2003*, 2003. 85

[89] H. Surmann, K. Lingemann, A. Nuechter, and J. Hertzberg, "A 3d laser range finder for autonomous mobile robots," *Proceedings of the 32nd ISR(International Symposium on Robotics)*, pp. 153 – 158,, 2001. 85

[90] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley-Interscience, New York, 1972. 89

[91] T. Wisspeintner, F. Maurelli, D. Droeschel, S. May, H. Surmann, and K. Pervoelz, "A 3d laser scanner system for intersection analysis and autonomous driving," in *Preceedings of IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008. 99, 137, 143

[92] T. van der Zant and T. Wisspeintner, "RoboCup X: A Proposal for a New League Where RoboCup Goes Real World," in *Proceedings of the RoboCup International Symposium 2005*, pp. 166–172, 2005. 101, 143

[93] T. Wisspeintner, T. van der Zant, L. Iocci, and S. Schiffer, "Robocup@home: Scientific competition and benchmarking for domestic service robots," *Interaction Studies*, vol. 10:3 (to be published), 2009. 101, 143

[94] T. Wisspeintner and W. Nowak, *Robotic Soccer*, ch. VolksBot - A Construction Kit for Multi-purpose Robot Prototyping, pp. 529–548. I-Tech Education and Publishing, 2007. 103, 143

[95] L. Meeden, A. C. Schultz, T. R. Balch, R. Bhargava, K. Z. Haigh, M. Bohlen, C. Stein, and D. P. Miller, "The aaai 1999 mobile robot competitions and exhibitions," *AI Magazine*, vol. 21, no. 3, pp. 69–78, 2000. 107

[96] H. Kitano and S. Tadokoro, "Robocup rescue: A grand challenge for multiagent and intelligent systems," *AI Magazine*, vol. 22, no. 1, pp. 39–52, 2001. 107

[97] A. Steinfeld, T. Fong, D. B. Kaber, M. Lewis, J. Scholtz, A. C. Schultz, and M. A. Goodrich, "Common metrics for human-robot interaction," in *Proc. of HRI*, pp. 33–40, 2006. 107

[98] H. A. Yanco, J. L. Drury, and J. Scholtz, "Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition," *Human-Computer Interaction*, vol. 19, pp. 117–149, 2004. 107

[99] J. L. Drury, H. A. Yanco, and J. Scholtz, "Using competitions to study human-robot interaction in urban search and rescue," *interactions*, vol. 12, no. 2, pp. 39–41, 2005. 107

[100] J. Savage, F. Ayala, S. Cuellar, and A. Weitzenfeld, "The use of scripts based on conceptual dependency primitives for the operation of service mobile robots," in *Proceedings of the International RoboCup Symposium 2008*, 2008. 129

[101] J. Ruiz-del-Solar, "Personal robots as ubiquitous-multimedial-mobile web interfaces," in *Proc. of 5th Latin American Web Congress (LA-WEB)*, pp. 120–127, 2007. 129

[102] S. Schiffer, A. Ferrein, and G. Lakemeyer, "Football is coming Home," in *Proc. of International Symposium on Practical Cognitive Agents and Robots (PCAR'06)*, University of Western Australia Press, 2006. 129

[103] M. Doostdar, S. Schiffer, and G. Lakemeyer, "Robust speech recognition for service robotics applications," in *Proceedings of the International RoboCup Symposium 2008*, LNCS, Springer, July 14-18 2008. to appear. 129

[104] M. Correa, J. Ruiz-del-Solar, and F. Bernuy, "Face recognition for human-robot interaction applications: A comparative study," in *Proceedings of the International RoboCup Symposium 2008*, 2008. 129

[105] W. B. Knox, J. Lee, and P. Stone, "Domestic interaction on a segway base," in *Proceedings of the International RoboCup Symposium 2008*, 2008. 129

[106] V. Belle, T. Deselaers, and S. Schiffer, "Randomized trees for real-time one-step face detection and recognition," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR'08)*, IEEE Computer Society, December 8-11 2008. 129

[107] P. Loncomilla and J. Ruiz-del-Solar, "Robust object recognition using wide baseline matching for robocup applications," in *Proceedings of the International RoboCup Symposium 2007*, LNAI 5001, Springer, 2007. 129

[108] H. Francke, J. Ruiz-del-Solar, and R. Verschae, "Real-time hand gesture detection and recognition using boosted classifiers and active learning," in *Advances in Image and Video Technology, Second Pacific Rim Symposium (PSIVT 2007)*, LNCS 4872, pp. 533–547, Springer, 2007. 129

[109] J. Burke, R. Murphy, E. Rogers, V. Lumelsky, and J. Scholtz, "Final report for the darpa/nsf interdisciplinary study on human-robot interaction," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, pp. 103–112, May 2004. 133

[110] A. Bredenfeld and T. Wisspeintner, "Volksbot - a hardware and software framework for modular mobile robots," in *1st CLAWAR/EURON Workshop on Robots in Entertainment, Leisure, and Hobby 2004*, 2004. 143

[111] T. Wisspeintner and A. Bredenfeld, "Volksbot - ein modularer roboterbaukasten für ausbildung und forschung," in *Informatik 2005 - Informatik LIVE!*, vol. 1, pp. 168–172, 2005. Conf-A. 143

[112] T. Wisspeintner, T. van der Zant, L. Iocci, and S. Schiffer, "Benchmarking domestic service robots," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008. 143

List of Figures

List of Tables

Erklärung

Hiermit versichere ich, für die Erstellung dieser Arbeit alle Hilfsmittel und Hilfen angegeben, und die Arbeit auf dieser Grundlage selbständig verfasst zu haben. Weiterhin versichere ich, dass die Arbeit noch nicht in einem früheren Promotionsverfahren in identischer oder ähnlicher Form eingereicht wurde.

München, den 12. Juni 2009

Thomas Wisspeintner

Appendix A

Zusammenfassung

Diese Arbeit zeigt Methoden und Ansätze auf, welche zum Ziel haben, Forschung und anwendungsorientierte Entwicklung im Bereich mobiler Serviceroboter durch Standardisierung in Hardware und Software sowie durch applikationsorientiertes Systembenchmarking zu beschleunigen. Die Arbeit beinhaltet drei Hauptbeiträge. Erstens, den Entwurf und die Umsetzung eines mechatronischen Baukastensystems für effiziente Entwicklung mobiler Serviceroboter namens *VolksBot*. Die Bedeutung und die Vielseitigkeit dieser Entwicklung wird mittels auf dem Baukasten basierender Roboterprototypen und -anwendungen sowie anhand der Verbreitung und Nutzung von *VolksBot*-Komponenten in der Robotikforschung und -entwicklung nahegelegt. Der zweite Beitrag bezieht sich auf Sensorentwicklungen für mobile Roboter und autonomes Fahren in städtischen Umgebungen. Zum Einen, das katadioptrische Kamerasystem *IAISVision* welches Robustheit gegenüber veränderter Beleuchtungssituationen aufweist, zum Anderen, ein kontinuierlicher 3D Laserscanner *3DLS-K* welcher als Sensorkomponente für autonomes Fahren während des Wettbewerbs *DARPA Urban Challenge 2007* eingesetzt wurde. Der dritte Beitrag bezieht sich auf die Entwicklung der *RoboCup*@Home initiative, welche der derzeit verbreitetste wissenschaftliche Wettbewerb für Service- und Haushaltsroboter ist. Die dargelegte Kombination aus Standardisierung und Rapid Prototyping bei der Entwicklung von Roboterkomponenten, dem Aufbau einer Gemeinschaft und der Definition und dem Benchmarking von mobilen Servicerobotikanwendungen zeigt vielversprechende Ergebnisse in Forschung und Entwicklung. Damit soll diese Arbeit zur zukünftigen Entwicklung relevanter Anwendungen und Produkten im Bereich Servicerobotik beitragen.

Appendix B

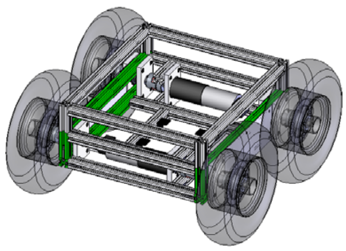
VolksBot RT4 assembly manual

The following pages contain excerpts of the *VolksBot* RT4 assembly manual in German. A separate assembly manual exists for the following *VolksBot* variants: RT3, RT4, RT6, Indoor v.2 and XT. The detailed stepwise description of the entire assembly process includes mechanical assembly, wiring of electronics and final testing of the entire system. This allows even untrained persons to assemble these robots. On the basis of these manuals, the assembly times mentioned in section 2.7.2 were measured.

Inhalt

8.2.1 Teil 1: Unterteil

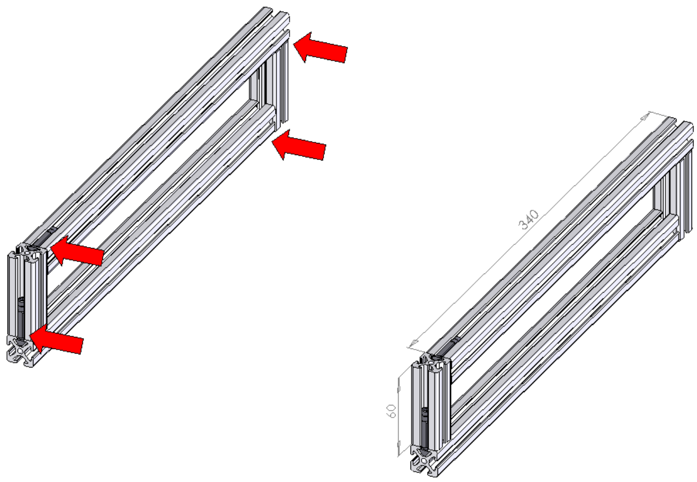
Bauteile für zwei Seitenteile:



Aluminiumprofile 20x20

- 4 Stück 60mm
- 4 Stück 340mm

- 8 Automatik-Verbinder



Einbau der Automatik-Verbinder auf Vorder- und Rückseite identisch.

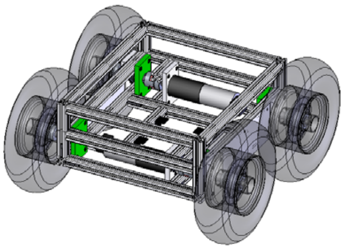
8.2.2 Rad- und Motor-Antriebseinheiten anbauen

Die Radlagerblöcke werden bündig mit der Außenkante fixiert.

Die genaue Lage des Motor-Lagerblocks ergibt sich aus der Länge der später gespannten Kette.

Zunächst werden die inneren Lagerblöcke der Antriebseinheiten montiert.

Bauteile für Antriebseinheiten:



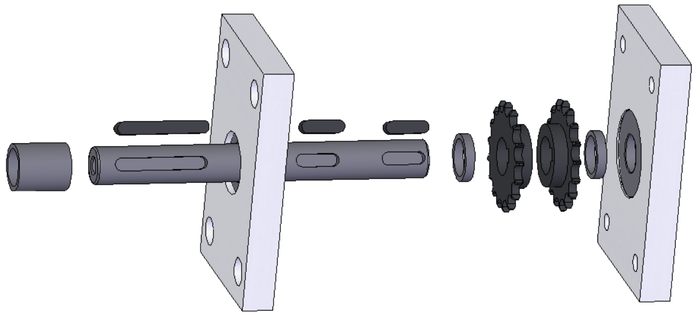
- 4 Radlagerblöcke (mit Kugellagern)
- 2 Motor-Radachsen (lang)
- 2 Radachsen (kurz)
- 8 Distanzringe, 3mm
- 8 Zahnräder
- 4 Passfedern, 14mm
- 2 Ketten
- 16 Zylinderkopfschrauben M5x10
- 16 Nutensteine M5

Hinweis: Die Achsen müssen exakt in die Kugellager eingepasst werden und dürfen dabei nicht verkanten.

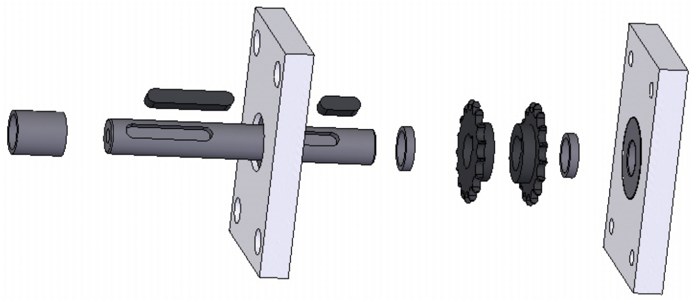
Die Achsen und Kugellager sind sehr exakt gefertigt und müssen gegebenenfalls mit einer Presse oder einem Kunststoffhammer zusammengefügt werden. Verwenden Sie keinen Metallhammer, um Stauchungen in den Achsen zu vermeiden.

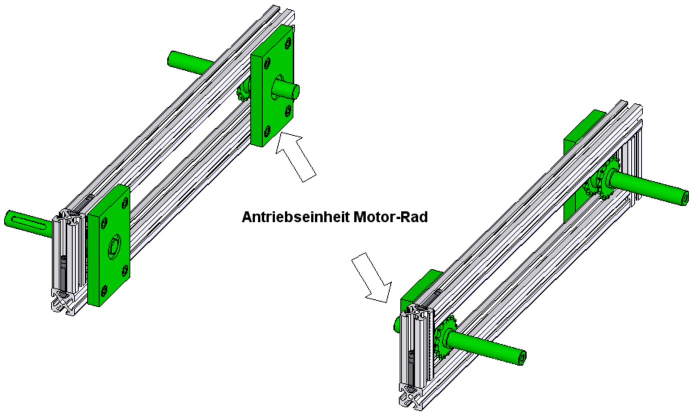
Gesamtaufbau der Antriebseinheiten (Explosionszeichnungen):

Antriebseinheit Motor-Rad (lange Achse)



Antriebseinheit Rad (kurze Achse)





Die Kette ist mit dem Kettenschloss sicher zu schließen.


1


2

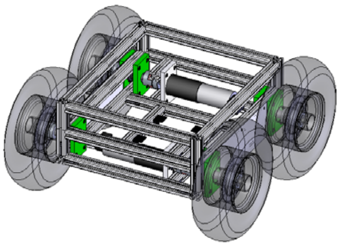

3


4


5

Die Zahnräder werden mit aufgezogener Kette (hier nicht abgebildet) auf die Achsen geschoben. Der Motor-Lagerblock wird so weit geschoben und dann fixiert, bis die Kette locker gespannt ist.

Abschließend werden die äußeren Lagerblöcke montiert.

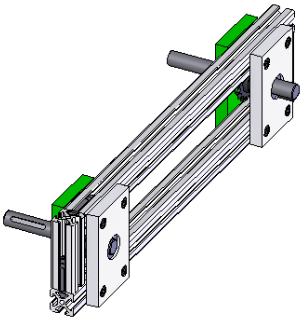


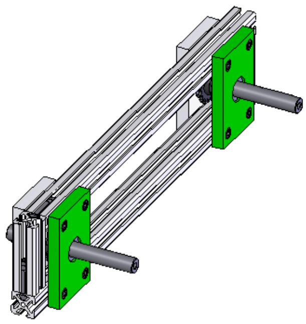
Bauteile:

- 4 Lagerblöcke mit Kugellagern
- 16 Zylinderkopfschrauben M5x10
- 16 Nutensteine M5

Hinweis: Die Kugellager in den Lagerblöcken müssen exakt über die Achsen geschoben werden und dürfen dabei nicht verkantet werden.

Die Achsen und Kugellager sind sehr exakt gefertigt und müssen gegebenenfalls mit einer Presse oder einem Kunststoffhammer zusammengefügt werden. Verwenden Sie keinen Metallhammer, um Stauchungen in den Achsen zu vermeiden.



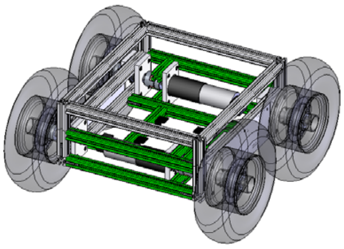


Motor- und Rad-Antriebseinheiten fertig montiert.

8.2.3 Teil 2: Querprofile + Motorenhalter

Die fertig montierten Seitenteile mit Motor- und Rad-Antriebseinheiten werden mit Querprofilen verbunden

Bauteile:

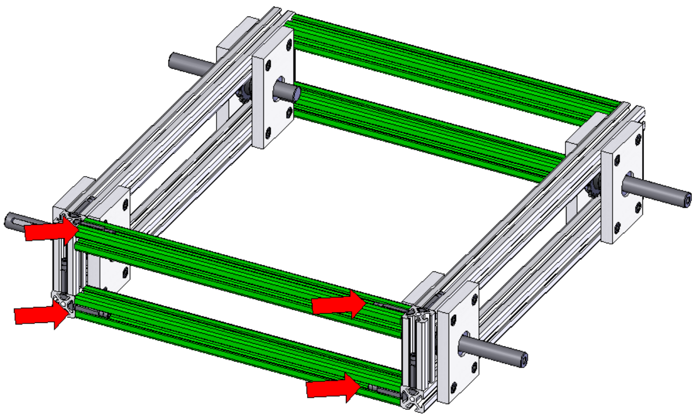


Aluminiumprofile 20x20

- 4 Stück 60mm
- 2 Stück 160mm
- 6 Stück 280mm

- 20 Automatik-Verbindersätze (Hülse, Schraube, Nutenstein)

Hinweis: Wenn eine Nute an beiden Enden einen Automatik-Verbinder enthält, müssen die Schrauben vor dem Eindrehen der Hülsen in die Nute eingelegt werden.



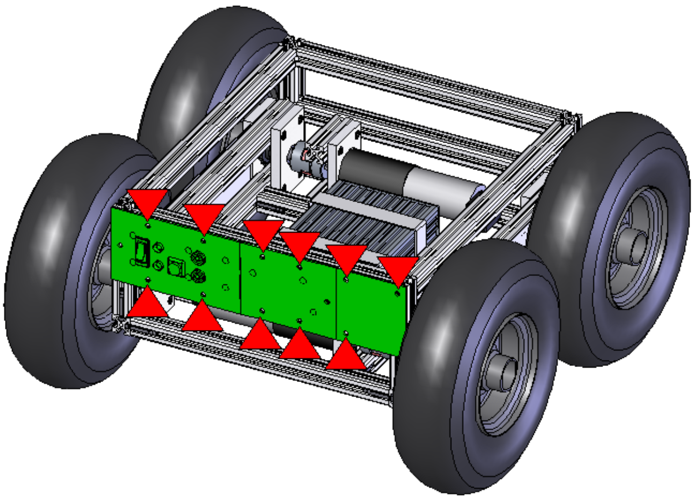
Einbau der Automatik-Verbinder auf Vorder- und Rückseite identisch.

8.3 Bedienfeld und Stromverteiler (optional)

Bauteile:

- 1 Bedienfeld
- 1 Stromverteiler
- 1 Verkleidung
- 11 Nutensteine M5
- 11 Halbrundschrauben M5x6

Das Bedienfeld und der Stromverteiler enthalten empfindliche elektronische Bauteile. Vorsichtig hantieren.

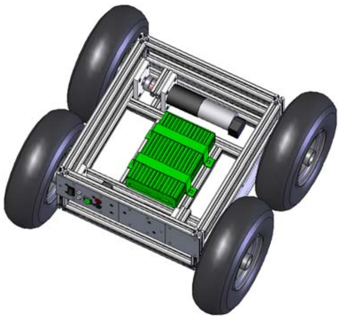


Legen Sie Nutsteine M5 in die Profilnuten hinter die Bohrlöcher von Bedienfeld, Stromverteiler und Abdeckung.

Schrauben Sie in die oberen und unteren Löcher jeweils Halbrundschrauben M5x6.

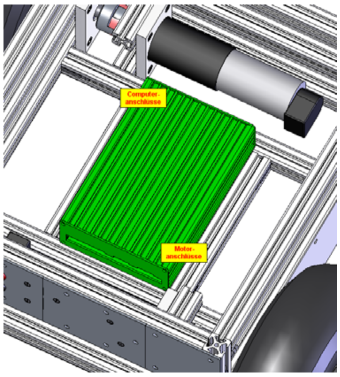
8.4 Motorcontroller

Bauteile:



- 1 Motorcontroller
- 2 Motorcontroller-Halter
- 4 Halbrundschrauben M5x10
- 4 Nutensteine M5

Legen Sie den Motorcontroller mittig auf die 4 Winkel. Die Motoranschlüsse zeigen nach vorne zum Bedienfeld (optional).



Sicht von oben

Legen Sie die Nutensteine in die Nuten und befestigen Sie die Motorcontroller-Halter mit den Halbrundschrauben an den Profilen.