

# The b-it-bots RoboCup@Home 2008 Team Description Paper

Dirk Holz, Jan Paulus, Thomas Breuer, Geovanny Giorgana, Ronny Hartanto,  
Walter Nowak, Johnny Jackanapes, Paul Ploeger and Gerhard Kraetzschmar

Bonn-Rhein-Sieg University of Applied Sciences, Computer Science Department,  
Grantham-Allee 20, 53757 Sankt Augustin, Germany  
<first name>.<last name>@inf.fh-brs.de

**Abstract.** This paper presents the b-it-bots RoboCup@Home team including the mobile robot platform for the RoboCup@Home 2008 competitions and its robot control architecture. Some components of particular interest will be described in detail.

## 1 Introduction

The b-it-bots RoboCup@Home team at Bonn-Rhein-Sieg University of Applied Sciences is the continuation of a former middle-size league team known as GMD Robots and AIS/BIT Robots. The b-it-bots team participates in @Home since RoboCup-2007 in Atlanta. Our research interests pursued in the team include mobile manipulation, navigation in cluttered and dynamic environments, and human/robot interaction.

The team consists of a mixture of Bachelor, Master and PhD students, which are advised by two professors from the Bonn-Rhein-Sieg University of Applied Sciences. The results of several research and development (R&D) as well as Master theses projects had already been successfully integrated into the former middle-size league team software system. In the same way, the complete RoboCup@Home control architecture and all of its components are developed by students in R&D projects and Master theses. Through this kind of graded course modules our RoboCup team is strongly interwoven with the Masters course Autonomous Systems offered at the Bonn-Rhein-Sieg University of Applied Sciences.

Our main research interests include mobile manipulation, environment modeling, computer vision and human/machine interaction. Our approach is to first identify and evaluate in each subfield the state-of-the-art and the best practice solutions currently available, then to develop new approaches.

The remainder of this paper is organized as follows: Our robot platform is described in Chapter 2 and its control software architecture in Chapter 3. In Chapters 4 and 5 we will provide some details about the components responsible for human/robot interaction and for mapping and navigation.

## 2 Robot Platform

Our robot is based on a modular mobile platform called *VolksBot* [9], which has been designed specifically for rapid prototyping and robot applications in education, research and industry. The *VolksBot* system is developed, manufactured and sold by the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS). It allows for easy access to and replacement of components such as motors, batteries and electronics.

The team uses a customized variant (see Fig. 1) with an integrated manipulator mounted in a way to provide good reachability and maneuverability. The overall platform size is  $(51 \times 51 \times 120)$ cm (W×L×H) and its weight is 60 kg. The drive unit used for locomotion uses a differential drive with two actively driven wheels, powered by two 150 W motors, and two castor wheels to enhance rotating and stability under load. The robot’s maximum velocity is 2 m/s.



**Fig. 1.** Robot platform "Johnny Jackanapes".

### 2.1 Manipulator

Our primary manipulator is a Neuronics Katana 6M180 robot arm. It is equipped with six motors providing five degrees of freedom w.r.t. the gripper’s position and orientation in its reachable workspace. It allows for precise movements with high repeatability. The sixth motor is used to open and close the two-fingered gripper, which is equipped with infrared reflectance as well as force sensors. The arm’s weight is  $\approx 4$  kg and it can handle a maximum payload of 500 g. It’s operation radius is 60 cm.

### 2.2 Range Sensors

As previously mentioned, part of the team effort involved the evaluation of different hardware components. This includes the evaluation of different range sensors, e.g. 2D and 3D laser scanners, as well as recent 3D time-of-flight cameras for perceiving spatial information about the robot’s workspace and constructing internal models of the environment.

2D laser scanners became the de facto standard range sensor in mobile robotics as they provide high resolution, accurate and fast range scans of the geometrical structure of a 2D plane in the surrounding environment. The laser scanner used in the setup shown in Fig. 1 is a SICK S300 2D laser scanner with a size of  $(102 \times 105 \times 152)$ mm (W×L×H) and a weight of 1.2 kg. The size of the apex angle limiting the scan plane is  $270^\circ$ , with an angular resolution of  $0.5^\circ$ .

However, the inherent drawback of 2D laser scanners for the purpose of obstacle avoidance and mapping is that objects not intersecting the 2D scan plane

are not detected by the laser scanner and, thus, cannot be perceived by the robot. Although this drawback can be neglected in many indoor robot applications, it plays an important role in a human's everyday environment. Here, many objects do not intersect the measurement plane, but still pose a threat to the robot, like for instance open drawers or small objects lying on the ground. Hence, 3D information becomes crucial.

One way to acquire 3D information is to mount a 2D scanner on a mechanical actuator to gain an additional degree of freedom [10]. Such a device is the Fraunhofer 3D Laser Scanner (3DLS), shown in Fig. 2. It is based on a SICK LMS 200 that supports a horizontal apex angle of  $180^\circ$  with an angular resolution of up to  $0.25^\circ$  and measurement frequencies of up to 75 Hz. To acquire three-dimensional information about the environment, the scanner is rotated around its horizontal axis, thereby providing a vertical angular range of up to  $120^\circ$  with a maximum resolution of  $0.25^\circ$ . In addition, we also evaluate the utility of a continuously rotating 3D laser scanner, the Fraunhofer 3DLS-K shown in Fig. 3, and of 3D time-of-flight cameras, like for instance the CSEM Swissranger devices.



**Fig. 2.** 3DLS



**Fig. 3.** 3DLS-K

### 2.3 Cameras

Up to now we have not yet decided on which visual sensors to use and experiment a lot with high-resolution 1394-cameras and standard webcams as well as commercial and custom-built stereo vision systems.

## 3 Software and Robot Control Architecture

Although we have started to develop a new control architecture and its components for the mobile service robot we use for @Home, we intend to re-use and port several existing software modules and components, like for robot navigation, object recognition and robot control, which had been developed for our former Middle Size League team.

Our control software consists of three fundamental parts, namely perception, task execution and actuation, that are run in parallel. The perception part consists of device drivers for sensors, feature extraction and sensor fusion mechanisms. The actuation part only provides device drivers for accessing different robot platforms and manipulators. The task execution component forms the fundamental part of the architecture thereby implementing a three-layered robot control architecture distinguishing three types of components: Behavior modules implement (reactive) controllers directly linking perception modules to actuation modules. They are themselves organized as a multilayered structure where a behavior can suppress or manipulate the output of behaviors on the lower layers. Note that the behavior modules are not necessarily purely reactive as they can access the robot's knowledge base.

Goal-directed action modules build upon behavior modules and activate or deactivate them in order to achieve a particular goal. They work in a fashion akin to schedulers, form the basis for abstract operators in higher-level deliberation components, and monitor the execution of behavior modules.

Task modules incorporate problem-specific deliberation components for planning the execution of actions. They also schedule the execution of action modules. In the context of RoboCup@Home, a task module can be interpreted as a piece of software for solving a specific test.

The task execution component offers an interface that allows for its control by e.g. task planners, remote control devices, or human-machine-interfaces.

## 4 Human/Machine Interface

Human interaction with a service robot is a largely unsolved problem and one of the focal areas of the RoboCup@Home competition. Up to now our human/machine interface includes three components: *speech recognition*, *speech synthesis*, and *person identification/recognition*.

In previous competitions, we used the Microsoft Speech API for recognizing spoken commands, which performed very well in Atlanta 2007. For future competitions, we are currently evaluating proprietary as well as open source systems for speech recognition with respect to their usefulness for RoboCup@Home and their performance on different untrained speakers. This evaluation includes, amongst others, the Microsoft Speech API, CSLU Speech Tools, CMU Sphinx, HTK, and ESMERALDA. How much training is needed in order to recognize commands with reasonable accuracy and reliability, whether and to which degree restricting the vocabulary of the speech recognition software allows for speaker independent control as well as how reliable the speech recognition behaves in noisy surroundings (using onboard microphones) form other important criteria in the evaluation process. For bidirectional communication we are currently evaluating several speech synthesis applications, like for instance Festival [3], Flite [4], FreeTTS [1] and the MAC OS X Speech Synthesis API.

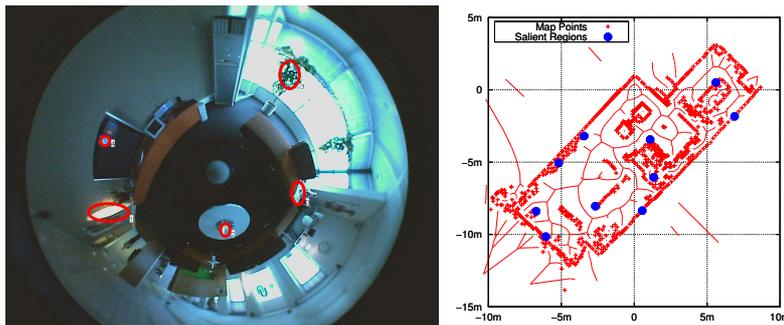
To recognize and identify the user, we currently apply a proprietary industry-based software for face recognition. This system is quite reliable, but it needs further testing with respect to its performance under different lighting conditions and its sensitivity to a person not directly facing the robot. With the long-term goal of low-cost service robots in mind, we are also searching for other state-of-the-art approaches and open source software.

## 5 Navigation and Mapping

The tests defined for the 2008 RoboCup@Home competition involve a significant amount of navigation by the robot in its environment, as a prerequisite to solve assigned tasks such as interacting with the environment or objects contained therein. For navigation to be safe and robust, but also sufficiently fast, a comprehensive model of the environment is needed. Our goal is to minimize

the amount of information that needs to be provided to the robot as a priori knowledge, and to let the robot autonomously acquire as much information for the environment model as possible. We distinguish three different phases in environment model acquisition, namely *exploration*, *correction* and *application*.

In the exploration phase the robot systematically explores its workspace and constructs an internal 2.5D geometric feature map. It thereby matches raw range data (from laser scanners or time-of-flight cameras) against an incrementally built map in a least mean square error sense. The matching techniques performing simultaneous localization and mapping (SLAM) for both 3 and 6 degrees of freedom are based on the Iterative Closest Point (ICP) algorithm [2], which searches for corresponding points in two data sets and returns a transformation that maps one set onto the other. We use the same transformation to correct an odometric estimation of the robot's pose shift. Points in a new range measurement that do not correspond to any point in the map during the matching process are added. In addition to constructing a geometric map, the robot also tries to construct a preliminary object data base by detecting salient regions in camera images, by means of a human-like visual attention system [5]. The feature vectors for each salient region are stored for each image taken, together with the robot's pose obtained by SLAM, and processed during the correction phase.

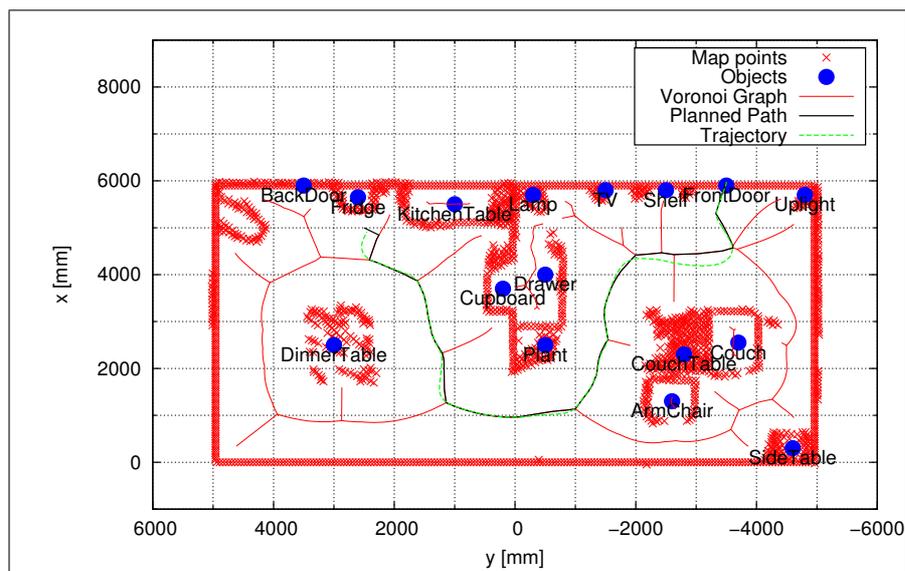


**Fig. 4.** Salient Regions in one camera image (left) and a geometric map augmented with a list of salient regions corresponding to the 10 most *interesting* objects. The constructed Voronoi Graph allows to directly plan paths to these locations.

In the correction phase we first identify interesting objects in the environment. We simply count how often image regions corresponding to one object are detected as being salient divided by the number of images where that very region was visible. These preliminary, unnamed objects are stored in a vector to augment the geometric environment model. Furthermore, they are used together with visual features [8] to detect loops in the robot's path during the exploration phase. In order to obtain a globally consistent map, this information is used to correct small errors that might have arisen in the mapping process. Having such a map, the robot extracts regions corresponding to e.g. rooms and corridors, that are just like the detected objects unnamed. For utilizing the gathered information in human/robot interaction, a human user has to assign names to detected ob-

jects and regions. Furthermore, the user can add undetected objects to the stored object vector and remove detected, but unnecessary objects. To demonstrate a typical environment representation obtained by the aforementioned techniques, Fig. 4 shows the results of applying the algorithms to a dataset provided in [11].

For navigation purposes in the application phase, we use a sweep line algorithm to generate a Voronoi graph based on the geometric information in the constructed environment model. The Voronoi graph can then be used for path planning using  $A^*$  graph search. For the actual navigation, i.e. motion planning and control, we distinguish whether the goal position is located in the robot's adjacencies or requires a longer trip. In the first case we apply a time-invariant motion-controller [6] to reach the goal position *and* orientation. For the second we assume that a path to the goal position exists (within the constructed Voronoi-graph) and then follow the shortest path found by  $A^*$  by applying a path following controller [7]. Both algorithms are especially designed for non-holonomic vehicles. While navigating or performing any other task the robot does not only use the internal environment model to re-localize itself, but also updates the model if a change in the environment is perceived. Furthermore, the robot performs behavior-based obstacle avoidance by means of 3D environment perception, thereby allowing for safe navigation in dynamic and cluttered workspaces.



**Fig. 5.** Map of the RoboCup German Open 2008 @Home arena.

Some results from applying the aforementioned methods and algorithms are shown in Fig. 5. Depicted is the geometric feature map constructed in a setup run during the RoboCup German Open 2008 together with the vector of known

locations. Note, that here the locations have been manually added to the object vector. Furthermore, the figure shows a path (black) from the “Front Door” to the ”Fridge” planned on the constructed Voronoi graph (red). The dotted (green) curve shows the trajectory resulting from applying the path following controller in [7].

## Acknowledgements

We gratefully mention the continued support of the team by the b-it Foundation.

## References

1. FreeTTS 1.2 - A speech synthesizer written entirely in the Java programming language. Available at <http://freetts.sourceforge.net>.
2. Paul J. Besl and Neil D. McKay. A method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239 – 256, 1992.
3. Alan W. Black and Paul A. Taylor. The Festival Speech Synthesis System: System documentation. Technical Report HCRC/TR-83, Human Communication Research Centre, University of Edinburgh, Scotland, UK, 1997.
4. Allan W. Black and Kevin A. Lenzo. Flite: a Small Fast Run-Time Synthesis Engine. In *Proceedings of the 4th ISCA Workshop on Speech Synthesis*, 2001.
5. S. Frintrop, E. Rome, A. Nüchter, and H. Surmann. A Bimodal Laser-Based Attention System. *J. of Computer Vision and Image Understanding (CVIU), Special Issue on Attention and Performance*, 2005.
6. G. Indiveri. Kinematic Time-invariant Control of a 2D Nonholonomic Vehicle. In *Proceedings of the 38th Conference on Decision and Control (CDC'99)*, 1999.
7. Giovanni Indiveri and Maria Letizia Corradini. Switching linear path following for bounded curvature car-like vehicles. In *Proceedings of 5th IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.
8. David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, 1999.
9. Thomas Wispeintner, Walter Nowak, and Ansgar Bredendfeld. Volksbot - a flexible component-based mobile robot system. In *Proceedings of the RoboCup Symposium*, 2005.
10. Oliver Wulf and Bernardo Wagner. Fast 3D-Scanning Methods for Laser Measurement Systems. In *International Conference on Control Systems and Computer Science (CSCS14)*, 2003.
11. Zoran Zivkovic, Olaf Booij, and Ben Kröse. From images to rooms. *Robot. Auton. Syst.*, 55(5):411–418, 2007.