RoboCup 2013 - homer@UniKoblenz (Germany)

Viktor Seib, Florian Kathe, Daniel Mc Stay, Stephan Manthe, Arne Peters, Benedikt Jöbgen, Raphael Memmesheimer, Tatjana Jakowlewa, Caroline Vieweg, Sebastian Stümper, Sebastian Günther, Simon Müller, Alruna Veith, Michael Kusenbach, Malte Knauf, Dietrich Paulus

Active Vision Group
University of Koblenz and Landau
Universitätsstr. 1
56070 Koblenz, Germany
vseib@uni-koblenz.de
http://robots.uni-koblenz.de/

Abstract. This paper describes the robot hardware and software used by team homer-@UniKoblenz of the University of Koblenz and Landau, Germany, for the participation at the RoboCup@Home 2013 in Eindhoven. A special focus is put on novel scientific achievements and newly developed features with respect to last year's competition. For the improvement of human-robot interaction we developed a generic face model that is synchronized to speech and can show seven different face expressions. This robot face is available as a ROS-node for other teams. Some of the novelties of this year is a new robotic platform, a new microphone, and digital camera for object recognition. Our object recognition algorithm will be published shortly as a ROS package. Further improvements were made in speech filtering, people detection and tracking as well as face expression recognition.

1 Introduction

Our team homer@UniKoblenz has already participated successfully as finalist in Suzhou, China (2008), Graz, Austria (2009) in Singapur (2010), where it was honored with the RoboCup@Home Innovation Award, and in Mexico-City, Mexico (2012), where it was awarded the RoboCup@Home Technical Challenge Award. Further, we participated in stage 2 at the RoboCup@Home World Championship in Instanbul, Turkey (2011).

Besides RoboCup@Home we competed in the RoboCup Rescue league with our robot *Robbie*, where our team won the Interleague Mapping Challenge award (2010) and became German Champion in Rescue Autonomy for the 5th time (2011).

In 2013 we will attend the RoboCup@Home World Championship in Eindhoven with our robot *Lisa* 1. Our team will be presented in the next section. Section 3 describes the hardware used for Lisa. In Section 4 our software architecture, autonomous navigation and human-robot interaction will be discussed. Finally, Section 5 will summarize this paper.

2 About our Team

The members of team *homer@UniKoblenz* are students from the University of Koblenz-Landau, Germany. They develop and improve our robots in practical courses.

2.1 Team Members and their Contributions

Team *homer@UniKoblenz* consists of the following members and their contributions.

Fig. 1. Current setup of our robot Lisa (right). Our new robotics platform (left).

Viktor Seib:	team leader *homer@UniKoblenz*, scientific supervisor
Florian Kathe:	team leader assistant, programming
Daniel McStay:	technical chief designer, programming
Stephan Manthe:	quality assurance, programming
Arne Peters:	hardware, programming
Benedikt Jöbgen:	hardware, programming
Raphael Memmesheimer:	infrastructure, programming
Tatjana Jakowlewa:	public relations, programming
Caroline Vieweg:	website, programming
Sebastian Stümper:	media, programming
Sebastian Günther:	flexible manpower, programming
Simon Müller:	bachelor's thesis - (Lisa learns drawing)
Alruna Veith:	bachelor's thesis (Face Expression Recognition)
Michael Kusenbach:	master's thesis - (People Detection)
Malte Knauf:	student research project - (Mapping and Navigation using ROS)

2.2 Focus of Research

Our interests in research are sensor fusion for person tracking, people detection, object, face and gesture recognition as well as object manipulation using a manipulator.

3 Hardware

3.1 Robot Platforms

We use a MobileRobots Pioneer3-AT as a platform. It provides sonar and odometry sensors and is equipped with four air-filled tires having a diameter of 21.5 cm. They can be controlled individually, allowing the robot to turn on the spot while maintaining a high degree of stability. Attached to the platform is a 2 DOF gripper, which is used to pick up objects from the floor. On top of the P3-AT, we have installed a prototype framework, which was designed and built by the Center of Excellence of the Chamber of Crafts in Koblenz and is able to carry additional sensors and a notebook running the control software.

(New in 2013) Currently, we are migrating to a new robotic platform, the CU-2WD-Center, manufactured by UlrichC[1]. It is equipped with odometry sensors and has similar proportions like the Pioneer3-AT. The main difference is the 2 wheel drive in contrast to the 4 wheel drive of the Pioneer. It allows the robot to turn on the spot with significantly less friction and thus, saves battery power and preserves the motors.

3.2 Sensors and additional Hardware

Notebook The software of the robot runs on a Lenovo W520 notebook equipped with an Intel Core i7-2670QM processor and 12 GB of RAM using Ubuntu Linux 12.04 as operating system.

SICK LMS100 laser range finder The SICK LMS100 is mounted at the bottom and generates 180° scans. It has an adjustable angular resolution, while its maximal measured distance is 20 m. It is used for mapping, localization and people tracking.

DirectedPerception PTU-D46 pan-tilt unit The DirectedPerception PTU-D46 is mounted on top of the robot's neck. It is able to rotate 159° in each direction and to tilt from $+31°$ to $-47°$ out of a horizontal position. The angular resolution is 0.012857°. Further sensores are attached on top of this unit.

(New in 2013) Rode VideoMic Pro Microphone The Microfon Rode VideoMic Pro is used for speech input. It is a light-weight microphone and is mounted on the pan-tilt unit of our robot. We use the Jack audio server[2] to capture speech and pass it either directly to the system or a noise filter. For filtering, the Quantile-based noise estimator [SFB00] is used. This audio setup and speech filtering was recommended by and tested in cooperation with team Golem from the UNAM, Mexico.

(New in 2013) Canon PowerShot SX100 IS In earlier experiments we have found out that because of their low resolution the color images provided by the Kinect are not suitable for reliable object recognition. Thus, this off-the-shelf digital camera is mounted on top of the pan-tilt unit to capture high resolution (8 megapixels) images for object recognition.

Microsoft Kinect The Microsoft Kinect ist attached to the pan-tilt unit and provides depth and color images of size 640x480 pixels at 30Hz. The depth sensor operation range lies between 0.8 m and 3.5 m at a horizontal field of view of 58° and a vertical field of view of 45°. It also has two microphones and a tilt motor for sensor adjustment. However, the robot is equipped with another microphone for speech recognition. Its tilt motor is not used because of the pan-tilt unit that the Kinect is mounted on.

Philips 1300NC color camera A color camera with 1.3 megapixels is mounted on the robot platform for object recognition. Is is used to capture images of objects reachable by the 2 DOF gripper.

Neuronics Katana 400HD The Katana 400HD is a 6 DOF industrial-grade robot arm. It is attached to our robot's body plate and used to manipulate objects on tables and other furniture of similar height. With an accuracy of 1 mm and a length of 90 cm, it enables us to perform delicate manipulation tasks on light-weight objects. The end effector is a standard pincher gripper and is safe for interaction with humans. Furthermore, it allows within its workspace to hand over items from the 2 DOF gripper.

[1] Manufacturer of our new robotic platform: `http://www.ulrichc.de`
[2] Jack audio server: `http://jackaudio.org/`

4 Technology and Scientific Contribution

4.1 System Core Architecture

Our Robbie software architecture [TSL+11] is based on a generic core that handles the forwarding of messages in the system. Messages are sent and received by modules and exchanged via a dispatcher. Modules use workers as a small set of program code which mainly provide computing functionalities. Devices provide access to hardware components. The system can be configured dynamically by a central registry that contains various profiles that store the required modules and configuration settings for a certain task.

This software architecture has an interface for using the open-source meta-operating system ROS[3]. It allows to use all services provided by ROS and meanwhile keep the current system core architecture of Robbie. The main idea is to use both systems in order to keep on the one hand the robust and extensive implementations of Robbie and on the other hand to develop new skills in ROS. However, in the long-term a complete changeover to ROS is intended to provide easier compatibility by using a widely spread platform and to offer open-source implementations to the community (please refer to Sec. 4.6 and Sec. 4.8).
(New in 2013) This year all of our hardware is integrated using either already available or self-developed ROS nodes.

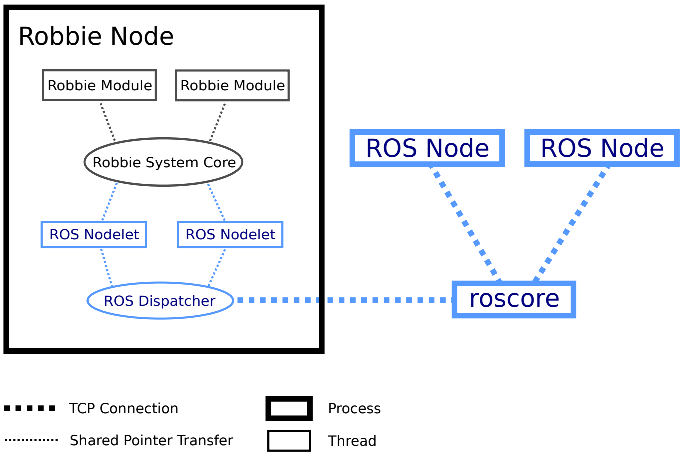

Fig. 2. Schematic description of the communication between Robbie and ROS.

Figure 2 gives a schematic overview of the communication between the Robbie software architecture and ROS. The changes in Robbie include an additional ROS dispatcher besides the Robbie dispatcher that manages the messages transfered by ROS nodelets within Robbie. Nodelets can subscribe and publish topics via shared pointer transfer within Robbie and in the meantime also receive and publish the common Robbie messages. They act as a linkage between the two systems. Beyond Robbie the common ROS functionalities can be used which is depicted here as ROS Nodes communicating with roscore.

4.2 Graphical Interface

The Robbie framework offers a GUI that can be run directly on the robot or on a different computer via WLAN. The user interface is realized using Qt4 and OpenGL. This feature has shaped up as a very important tool for understanding and improving the complex algorithms needed for a fully autonomous robot and for sensor data visualisation 3.

[3] Documentation of ROS `http://www.ros.org/`

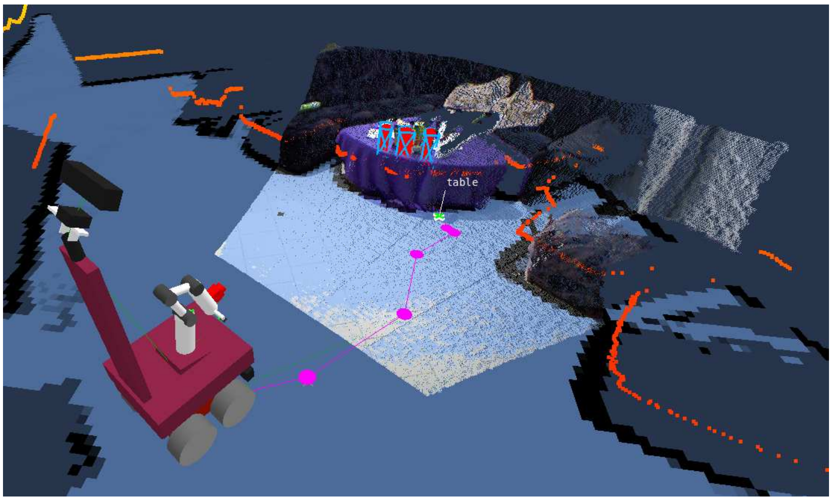

Fig. 3. Visualization of RGBD and laser sensor data. Further, a planned path to the point of interest "table" as well as grippable objects on the table are visualized.

4.3 General Purpose Task Planning

An abstraction layer in our software architecture allows for task planning aiming at general purpose task execution. For this purpose we encapsulated most common tasks (e.g. navigation to a given location or grabbing a specific object) in so called *task modules*. Every task module is designed for a single task and has the same interface, thus hiding the rather complex, task specific interfaces of each single task. Using this abstraction principle, complex behavior can be triggered by a speech command. The input string is processed and the required task modules are activated to perform the given actions. To support this task planner a new layer was added to our map. It not only allows to define points of interest for navigation, but also whole areas that can be assigned a name. Hence, rooms can be defined by areas rather than points as before.

4.4 Simultaneous Localization and Mapping

To enable users without technical knowledge to use the robot and to ease the setup procedure, it has to be able to create a map without the help of a human. For this purpose, our robot continuously generates and updates a 2D map of its environment based on laser scans. Figure 4 shows an example of such a map.
(New in 2013) To detect obstacles below and above the LRF-plane we use Kinect data to augment the occupancy map.

4.5 Navigation in Dynamic Environments

In real-life situations, the approach described above is not sufficient for navigating through an everyday environment, as due to the movement of persons and other dynamic obstacles, an occupancy map that only changes slowly in time does not provide sufficient information.

Thus, our navigation system, which is based on Zelinsky's path transform [Zel88,Zel91], always merges the current laser range scan as a frontier into the occupancy map. A once calculated path is then checked against obstacles in small intervals during navigation, which can be done at very little computational expense. If an object blocks the path for a given interval, the path is re-calculated. This approach allows the robot to efficently navigate in highly dynamic environments.

Fig. 4. Real-time maps of the RoboCup 2008 (left) and 2009 (right) @Home arena.

4.6 Human-Robot Interface

The robot is equipped with speakers and a microphone, which enables communication via speech interface. In addition, it has a small screen that is used to display facial expressions and state information. On the software side, we decided to use two open source libraries: pocketsphinx[4] for speech recognition and festival[5] for speech synthesis.

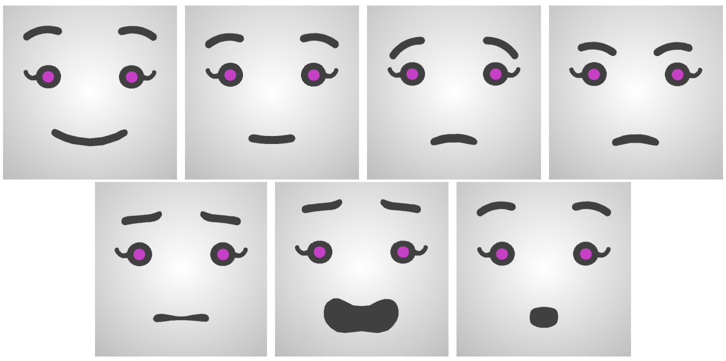

Fig. 5. Animated face of our service robot Lisa. The depicted face expressions are (from left to right): happy, neutral, sad, angry, disgusted, frightened, and surprised.

We have designed a concept of a talking robot face that is synchronized to speech via mouth movements. For this new feature we are using the open source library Ogre3D[6] for visualisation. Furthermore, the face is modelled with Blender[7] and exported via the Ogre Mesh Exporter[8] for the use with Ogre.

To include the robot face into our software, we have created a ROS node for this application. We extended the robot face to show seven different face expressions (Figure 5). Further, we provide two similar face meshes, a female and a male one. The colors and the voice (female or male) can be configured via a configuration file without recompiling the application. The robot face has been released as an open source package for ROS[9] so it is possible to apply it for any robot.

[4] Speech recognition system pocketsphinx `http://www.speech.cs.cmu.edu/pocketsphinx/`
[5] Speech synthesis system festival `http://www.cstr.ed.ac.uk/projects/festival/`
[6] Open-source graphics rendering engine Ogre3D `http://www.ogre3d.org/`
[7] Free open source 3D content creation suite Blender `http://www.blender.org/`
[8] Ogre Mesh Exporter `http://www.ogre3d.org/tikiwiki/Blender+Exporter`
[9] Robot Face: `http://www.ros.org/wiki/agas-ros-pkg`

(New in 2013) To further enhance communication with humans a robot has to detect their emotions. One step towards this goal is the recognition of facial expressions. We use an approach based on the facial action coding system and action units. In the first step features are extracted from the detected face and classified into action units. The subsequent step analyzes the obtained action units and assigns a facial expression. Both classification steps use a support vector machine. Our system is able to recognize the same 7 face expressions that Lisa can exhibit. This approach works without prior calibration with a neutral face expression of the detected person.

4.7 Object detection

Objects for mobile manipulation are detected by first segmenting horizontal planes as table hypotheses. Subsequently, all points above the plane are clustered and the resulting clusters considered as objects.

(New in 2013) Transparent objects (in our case drinking glasses) are detected by making use of one fault of the Kinect sensor. The structured light emitted by the Kinect is scattered in transparent objects providing no valid depth information. We segment areas with no depth data and compare them to holes in detected planes to extract contours and match them with drinking glass contour templates. Since the supporting table plane around the transparent objects has valid depth information, a size and location estimation of the transparent objects is obtained and used for grasping.

4.8 Object and Face Recognition

The object recognition algorithm we use is based on Speeded Up Robust Features (SURF) [BTVG06], which are local scale-invariant features of gray images. First, features are matched between the trained image and the current camera image based on their euclidean distance. A threshold on the ratio of the two nearest neighbours is used to filter unlikely matches. Then, matches are clustered in hough space using a four dimensional histogram using their position, scale and rotation. This way, sets of consistent matches are obtained. The result is further optimized by calculating a homography between the matched images and discarding outliers. Our system was evaluated in [DTPG10] and shown as suitable for fast training and robust object recognition.

With this object recognition approach we won the Technical Challenge 2012. Currently, we are implementing our object recognition as a ROS package and will make it available for all teams shortly[10]. A detailed description of the approach will be provided as well.

4.9 People Detection

People are detected by the combination of three sensors. The laser range finder is used to detect legs. The RGB image of the Kinect camera provides data for face detection. We use the face detection algorithm implemented in the OpenCV library. Finally, the Kinect depth camera allows to detect silhouettes of persons. For a person to be detected, not every sensor has to see the person. However, the more sensors see a person the higher the probability to really encounter a person at the position in question.

4.10 People Tracking

For the perception of people we combine the three methodes described in 4.9. In case that these sensors yield similar points, the output is merged. Thus, it is assumed that the points belong to the same person. The obtained person positions are used as input for the particle filter which estimates the movement and position of the people.
(New in 2013) To further imrove tracking we use the Tracking Learning Detection algorithm (TLD, [KMM10]).

[10] Object Recognition: `http://www.ros.org/wiki/agas-ros-pkg`

4.11 Object manipulation

Our system detects planes in the acquired 3D point cloud using RANSAC. This information is used to find euclidean point clusters on top of planes which fit into the gripper and are thus regarded as candidates for grasping.

If a specific object has to be grasped, one of the color cameras is adjusted to face the object cluster. The object detection algorithm is executed on the region of interest defined by the object cluster's bounding box. Depending on the position of the object, camera and manipulation device are selected. These are either the bottom camera with the 2 DOF gripper or the Kinect RGB-camera combined with the 6 DOF robotic arm. Additionally, a laser range finder on top of the 6 DOF arm help for precise gripping.

The movement planning for our robotic arm is performed using an approach operating directly in working space. Chaining motion primitives, our path planner builds a graph from the starting position to the goal. The planning can be optimized towards specific objectives. These are performing a smooth path or keeping a maximum distance from obstacles using heuristic and cost functions [CCL10].

5 Conclusion

In this paper, we have given an overview of the approach used by team homer@UniKoblenz for the RoboCup@Home competition. We presented a combination of out-of-the box hardware and sensors and a custom-built robot framework. We explained the fundamentals of our message-based robot architecture and its further enhancements, the use of well-established techniques like SLAM based on a particle filter and a grid map. Furthermore, we explained our approach for object recognition using matching and clustering of local invariant features and the ability to detect and manipulate objects with a 2 DOF gripper and 6 DOF robotic arm. Based on the existing system from last years competition, effort was put into enhancing the detection of people and the human-robot interaction abilities of our robot. 3D scans of the environment are aquired using a RGB-D sensor. A task planner facilitates the execution of tasks that are typical for service robots.

References

BTVG06. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *ECCV*, pages 404–417, 2006. 7

CCL10. Benjamin Cohen, Sachin Chitta, and Maxim Likhachev. Search-based planning for manipulation with motion primitives. Anchorage, Alaska, 2010. 8

DTPG10. Peter Decker, Susanne Thierfelder, Dietrich Paulus, and Marcin Grzegorzek. Dense Statistic Versus Sparse Feature-Based Approach for 3D Object Recognition. In *10th International Conference on Pattern Recognition and Image Analysis: New Information Technologies*, volume 1, pages 181–184, Moscow, 12 2010. Springer MAIK Nauka/Interperiodica. 7

KMM10. Z Kalal, J Matas, and K Mikolajczyk. P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. *Conference on Computer Vision and Pattern Recognition*, 2010. 7

SFB00. V. Stahl, A. Fischer, and R. Bippus. Quantile based noise estimation for spectral subtraction and wiener filtering. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1875–1878. IEEE, 2000. 3

TSL+11. S. Thierfelder, V. Seib, D. Lang, M. Häselich, J. Pellenz, and D. Paulus. Robbie: A message-based robot architecture for autonomous mobile systems. *INFORMATIK 2011-Informatik schafft Communities*, 2011. 4

Zel88. Alexander Zelinsky. Robot navigation with learning. *Australian Computer Journal*, 20(2):85–93, 5 1988. 5

Zel91. Alexander Zelinsky. *Environment Exploration and Path Planning Algorithms for a Mobile Robot using Sonar.* PhD thesis, Wollongong University, Australia, 1991. 5

Document date: 2013-02-08