RoboCup 2015 - homer@UniKoblenz (Germany)

Viktor Seib, Stephan Manthe, Jana Holzmann, Raphael Memmesheimer, Arne Peters, Markus Bonse, Florian Polster, Baharak Rezvan, Katrin Riewe, Malte Roosen, Uzaira Shah, Timur Yigit, Andreas Barthen, Malte Knauf, Dietrich Paulus

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

Abstract. This paper describes the robot *Lisa* used by team homer@UniKoblenz of the University of Koblenz-Landau, Germany, for the participation at the RoboCup@Home 2015 in Hefei, China. A special focus is put on novel system components and the open source contributions of our team. Apart from the already available ROS packages for our object recognition and the robot face, we are currently preparing the release of our mapping and navigation, speech recognition interface and the GUI. The packages are available (and new packages will be released) on `http://wiki.ros.org/agas-ros-pkg`.

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, in Mexico-City, Mexico (2012), where it was awarded the RoboCup@Home Technical Challenge Award and in Eindhoven, Netherlands (2013). Further, we participated in stage 2 at the RoboCup@Home World Championship in Instanbul, Turkey (2011). Our team achieved several times the 3rd place in the RoboCup GermanOpen (2008, 2009, 2010 and 2013) and participated in the GermanOpen finals (2011, 2012 and 2014).

Apart from RoboCup, team homer@UniKoblenz won the best presentation award at RoCKIn Camp (Rome, 2014), the 1st place in the overall rating, as well as the 2nd place in the Object Perception Challenge in the RoCKIn Competition (Toulouse, 2014).

In 2015 we plan to attend the RoboCup@Home in Hefei, China, with two robot: the new Lisa in blue and the old Lisa in purple (Fig. 1).

Our team will be presented in the next Section. Section 3 describes the hardware used for Lisa. In Section 4 we present the software components that we contribute to the community. The following Section 5 presents our recently developed and improved software components. Finally, Section 6 will conclude this paper.

2 Team homer@UniKoblenz

The Active Vision Group (AGAS) offers practical courses for students where the abilities of Lisa are extended. In the scope of these courses the students design, develop and test new software components and try out new hardware setups. The practical courses are supervised by a research associate, who integrates his PhD research into the project. The current supervisor and team leader is Viktor Seib.

Each year new students participate in the practical courses and are engaged in the development of Lisa. These students form the team *homer@UniKoblenz* to participate in the RoboCup@Home. *Homer* is short for “home robots” and is one of the participating teams that entirely consist of students. The current team is shown in Figure 2.

Fig. 1. Setup of our 2 robots. The current setup of Lisa is shown on the left. With this setup, Lisa is participating in the RoboCup since 2014 and is now our main robot. An older version of Lisa is shown on the right. This setup is the auxiliary robot of our team.

Fig. 2. Team homer@UniKoblenz 2015. Back row (left to right): Florian Polster, Uzaira Shah, Arne Peters, Baharak Rezvan, Andreas Barthen, Stephan Manthe, Malte Roosen, Markus Bonse, Raphael Memmesheimer. Front row (left to right): Katrin Riewe, Timur Yigit, *Lisa*, Viktor Seib, Jana Holzmann. Not in the picture: Malte Knauf.

2.1 Focus of Research

The current focus of research is object recognition with visual and shape features. Novel approaches related to Implicit Shape Models (ISM) [4], as well as approaches for affordance detection are currently tested and integrated into the software collection of Lisa.

Additionally, with large member fluctuations in the team, as is natural for a student project, comes a necessity for an architecture that is easy to learn, teach and use. We thus migrated from our classic architecture *Robbie* [11] to the Robot Operating System (ROS) [6]. We developed an easy to use general purpose framework based on the ROS action library that allows us to create new behaviors in a short time.

3 Hardware

(New in 2015) Robot Platforms For the first time we use two *Lisa* robots. The blue Lisa is build upon the CU-2WD-Center platform, manufactured by UlrichC[1]. It is equipped with odometry sensors and has a 2 wheel drive. The auxiliary robot, purple Lisa, uses a MobileRobots Pioneer3-AT platform. However, the controller boards were replaced by our own, designed around an Arduino controller. On top of both platforms we have installed a prototype casing, 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.
The following hardware description applies to the main robot, the *Lisa* with the blue casing.

Notebook The software of the main robot runs on a Lenovo W520 notebook equipped with an Intel Core i7-2670QM processor and 12 GB of RAM using Ubuntu Linux 14.04 and ROS Indigo.

SICK LMS100 laser range finder A SICK LMS100 is mounted at the bottom of the casing 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.

(New in 2015) Hokuyo URG-04LX laser range finders Two Hokuyo laser range finders are mounted inside the robot platform (one at the front, the other one at the back). They have a range of 5.6 m and are used to detect small obstacles close to the robot.

DirectedPerception PTU-D46 pan-tilt unit A DirectedPerception PTU-D46 is mounted on top of the main robot's casing. It is able to rotate 159° in each direction and to tilt from +31° to −47° out of a horizontal position. Further sensors are attached on top of this unit.

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.

(New in 2015) IDS UI-5580CP-C-HQ camera The new sponsored IDS camera[2] is used to provide high resolution images to enhance object recognition. It is connected over ethernet and provides images with a resolution of 5 megapixels at 14 Hz.

(New in 2015) ASUS Xtion Pro live We replaced the Microsoft Kinect by an ASUS Xtion Pro live. Though both sensors provide the same data (depth and color images of size 640×480 pixels at 30 Hz), the Xtion camera does not need a separate power supply. Further, its new firmware is able to register color and depth images on the sensor, whereas the Kinect camera uses a software driver and thus increases the CPU payload.

Neuronics Katana 400HD The Katana 400HD is a 6 DOF industrial-grade robot arm. It is used to manipulate light-weight objects (up to 0.5 kg) on tables and other furniture of similar height. The end effector is a custom setup and consists of 4 Festo Finray-fingers. It is safe for interaction with humans.

(New in 2015) Raspberry Pi We integrated a Raspberry Pi in our robot setup. So far, we use it to switch remote-controlled socket outlets. This enables Lisa to switch on and of e.g. lights, a coffee maker, a hot plate and a TV to make your home smarter.

[1] Manufacturer of our new robotic platform: `http://www.ulrichc.de`
[2] IDS imaging: `http://www.ids-imaging.com`

4 Software Contribution

We followed the recent call for chapters for a new book on ROS[3]. We want to share stable components of our software with the RoboCup and the ROS community to help advancing the research in robotics. After finishing the Chapter, we plan to revise and test all components during the RoboCup GermanOpen and plan to release the software in May. All software components will be released on the Active Vision Group's ROS wiki page: `http://wiki.ros.org/agas-ros-pkg`. The contributions currently in preparation are described in the following paragraphs.

Mapping and Navigation

Simultaneous Localization and Mapping To know its environment, the robot has to be able to create a map. For this purpose, our robot continuously generates and updates a 2D map of its environment based on odomentry and laser scans. Figure 3 shows an example of such a map.

Navigation in Dynamic Environments An occupancy map that only changes slowly in time does not provide sufficient information for dynamic obstacles. Our navigation system, which is based on Zelinsky's path transform [13,14], always merges the current laser range scans into the occupancy map. A calculated path is checked against obstacles in small intervals during navigation. If an object blocks the path for a given interval, the path is re-calculated. This approach allows the robot to efficiently navigate in highly dynamic environments.

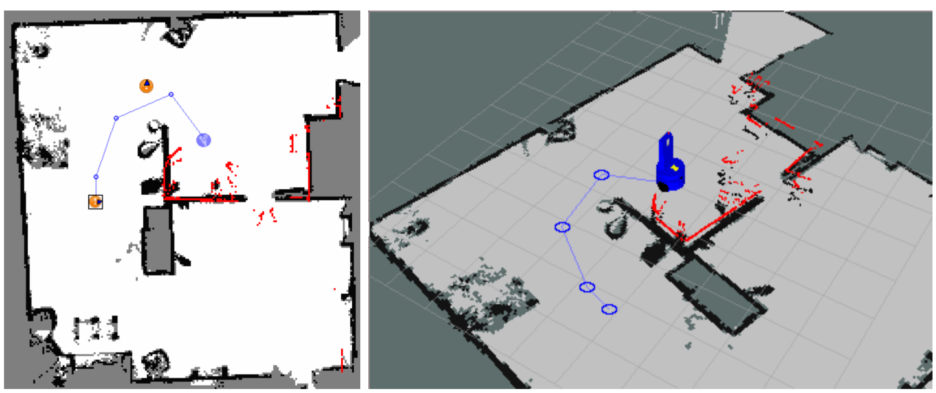

Fig. 3. 2D and 3D view of a map and a planned path (blue line). Red dots indicate the current laser scan, while orange points in the 2D map stand for navigation points.

Object Recognition

Object Recognition The object recognition algorithm we use is based on Speeded Up Robust Features (SURF) [1]. 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 neighbors 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 [3] and shown as suitable for fast training and robust object recognition. A detailed description of this approach is given in [8]. With this object recognition approach we won the Technical Challenge 2012 (Figure 4). A preliminary version of this software is already available online on our ROS package website.

[3] Call for chapters for a ROS book: `http://events.coins-lab.org/springer/springer_ros_book.html`

Fig. 4. Object recognition results during the Technical Challenge 2012.

Human Robot Interaction

Speech Recognition and Synthesis We developed an Android app [4] and a corresponding ROS package [5] that allow access to the Android Speech synthesis and recognition API and integrates with the ROS eco system. Thus, two important HRI problems (speech recognition and synthesis) are solved with this easy to integrate and low cost solution.

In comparison to our VoCon integration (see Section 5) this approach does not rely on a predefined grammar. The communication between ROS and the Android app is realized using TCP sockets. The Android speech synthesis and recognition can also be used without internet connection. However, recognition results are more reliable with an internet connection.

Robot Face We have designed a concept of a talking robot face that is synchronized to speech via mouth movements. The face is modeled with blender and Ogre3D is used for visualization. The robot face is able to show seven different face expressions (Figure 5). The colors, type and voice (female or male) can be changed without recompiling the application.

We conducted a broad user study to test how people perceive the shown emotions. The results as well as further details regarding the concept and implementation of our robot face are presented in [7]. The robot face is already available online on our ROS package website.

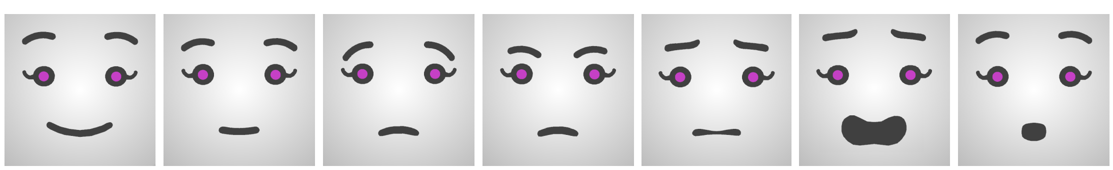

Fig. 5. Animated face of our service robot Lisa. The depicted face expressions are (from left to right): happy, neutral, sad, angry, disgusted, frightened, and surprised.

Graphical User Interface

Integrated GUI Although a graphical user interface (GUI) is not necessary for a robot, it provides a lot of comfort in monitoring or controlling the state of the system. ROS already has *rviz*, a powerful tool for visualization of sensor data that can be extended with own plugins. Thus,

[4] Preliminary version of the Android speech app: `https://github.com/airglow/android_speech_app`
[5] Preliminary version of the ROS speech package: `https://github.com/airglow/android_speech_pkg`

we did not aim at creating our own visualization tool. Rather, we include rviz's visualization as a plugin into our GUI. This means that our GUI is able to visualize everything that rviz does and can be adapted using the same configuration files. However, our GUI provides more than pure data visualization. It offers an easy possibility to create a map, define navigation points and monitor the navigation. Also, it enables the user to train and test new objects for object recognition in a convenient way and to monitor and define commands for human robot interaction. Additionally, basic state machine-like behavior and task execution can be defined in the GUI with only a few mouse clicks. The GUI can be run directly on the robot or on a different computer via WLAN. The user interface is created with Qt4 and OpenGL. A visualization example is given in Fig. 6.

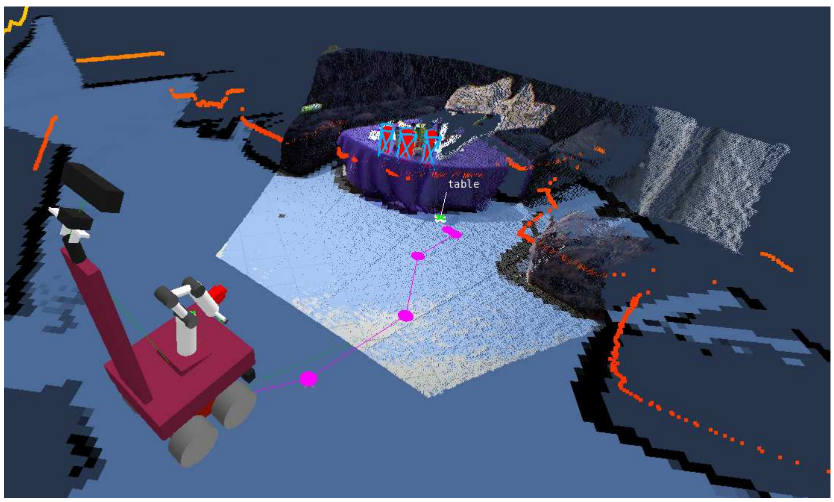

Fig. 6. Visualization of RGBD and laser sensor data. Further, a planned path to the point of interest "table" as well as grippable objects on the table are visualized.

5 Technology and Scientific Contribution

5.1 (New in 2015) General Purpose System Architecture

In the past years we have migrated step by step from our self developed architecture to ROS. Since 2014, our complete software is ROS compatible. To facilitate programming new behaviors, we created a architecture aiming at general purpose task executing. By encapsulating arbitrary functionalities (e.g. grasping, navigating) in self-contained state machines, we are able to start complex behaviors by calling a ROS action. The ROS action library allows for live monitoring of the behavior and reaction to different possible error cases. Additionally, a semantic knowledge base supports managing objects, locations, people, names and relations between these entities. With this design, new combined behaviors (as needed e.g. for the RoboCup@Home tests) are created easily and even students who are new to robotics can start developing after a short introduction.

5.2 Object Detection and Recognition

Objects for mobile manipulation are detected by first segmenting horizontal planes as table hypotheses from the point cloud of the ASUS Xtion camera. Subsequently, all points above the plane are clustered and the resulting clusters considered as objects. The IDS camera is used for object recognition because of its higher resolution in comparison to the Xtion RGB image. To enable object recognition with the high resolution camera, we had to find out the relative pose between the depth camera and the IDS camera using stereo calibration. With this information it is possible to project the bounding box of a detected object to the high resolution RGB image and perform object recognition.

Transparent objects (in our case drinking glasses) are detected by making use of one fault of the structured light sensor. The light emitted by the Xtion camera is scattered in transparent objects providing no valid depth information. We segment areas with no depth data and compare 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.

5.3 (New in 2015) 3D Object Recognition

For 3D object recognition we use a continuous Hough-space voting scheme related to Implicit Shape Models (ISM). In our approach [9], SHOT features [12] from segmented objects are learned. Contrary to the ISM formulation, we do not cluster the features. Instead, to generalize from learned shape descriptors, we match each detected feature with the k nearest learned features in the detection step. Each matched feature casts a vote into a continuous Hough-space. Maxima for object hypotheses are detected with the Mean Shift Mode Estimation algorithm [2].

5.4 (New in 2015) Object Manipulation

After detecting an object for grasping the motion sequence for the robotic arm is planned. We use an A* search algorithm within the arm configuration space. To avoid collisions, we run an on-the-fly collision detection in which we check an simplified arm model represented by a set of line segments against a kd-tree build on the depth image of the environment. The planned path is smoothed with a recursive line fitting algorithm in the configuration space. The resulting path consists of a minimal set of configurations which are directly connected by linear moves in the configuration space.

5.5 (New in 2015) Affordance Detection

Affordances have gained much popularity for object classification and scene analysis. Our current research focuses on sitting affordances to analyze scenes regarding sitting possibilities for an anthropomorphic agent. Recently, we introduced the concept of fine-grained affordances [10]. It allows to distinguish affordances on a fine-grained scale (e.g. sitting without backrest, sitting with backrest, sitting with armrests) and thus facilitates the object classification process. Additionally, our approach estimates the sitting pose with regard to the detected object (Figure 7).

Fig. 7. Detected sitting affordances and the corresponding sitting pose of an anthropomorphic agent.

5.6 (New in 2015) Speech Recognition

For speech recognition we use a grammar based solution supported by a academic license for the VoCon speech recognition software by Nuance[6]. We combine continuous listening with a begin and end-of-speech detection to get good results even for complex commands. Recognition results below a certain threshold are rejected. The grammar generation is supported by the content of a semantic knowledge base that is also used for our general purpose architecture.

5.7 Facial Expression Recognition

To 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 does not need prior calibration with a neutral face expression of the person of interest.

5.8 (New in 2015) People Detection and Tracking

People are detected by the combination of three sensors. The laser range finder is used to detect legs, while the RGB camera image provides data for face detection. We use the face detection algorithm implemented in the OpenCV library. Finally, the depth camera allows to detect silhouettes of persons.

For operator and people tracking we use rich RGB-D data from a depth camera. The sensor is mounted on a pan-tilt unit to actively follow the current person of interest. Our people tracker is based on the publicly available 3D people detector of Murano et al. [5] in combination with online appearance learning using adaboost classifiers on color histograms. We estimate the target position and velocity using a linear Kalman filter with constant velocity motion model. At every timestep, we select the detection with highest classification score inside the gating region for target association and update the classifier with positive and negative samples from the set of current detections accordingly. Occlusion detection is based on classification scores as well, i.e., we perform Kalman update and appearance learning only if the highest classification score exceeds a given threshold.

6 Conclusion

In this paper, we have given an overview of the approaches 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. Furthermore, we explained our system architecture, as well as approaches for 2D and 3D object recognition, human robot interaction and object manipulation with a 6 DOF robotic arm. This year we plan to use the blue *Lisa* for the main competition and the purple *Lisa* Lisa as auxiliary robot for open demonstrations. Based on the existing system from last year's competition, effort was put into improving existing algorithms of our system (speech recognition, manipulation, people tracking) and adding new features (encapsulated tasks for general purpose task execution, 3D object recognition, affordance detection) to our robot's software framework. Finally, we explained which components of our software are currently being prepared for publication to support the RoboCup and ROS community.

[6] `http://www.nuance.com/for-business/speech-recognition-solutions/vocon-hybrid/index.htm`

References

1. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *ECCV*, pages 404–417, 2006. 4
2. Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995. 7
3. 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. 4
4. Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV' 04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, 2004. 2
5. F. Basso M. Munaro and E. Menegatti. Tracking people within groups with rgb-d data. In *In Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2012. 8
6. Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009. 2
7. Viktor Seib, Julian Giesen, Dominik Grüntjens, and Dietrich Paulus. Enhancing human-robot interaction by a robot face with facial expressions and synchronized lip movements. In Vaclav Skala, editor, *21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2013. 5
8. Viktor Seib, Michael Kusenbach, Susanne Thierfelder, and Dietrich Paulus. Object recognition using hough-transform clustering of surf features. In *Workshops on Electronical and Computer Engineering Subfields*, pages 169 – 176. Scientific Cooperations Publications, 2014. 4
9. Viktor Seib, Norman Link, and Dietrich Paulus. Implicit shape models for 3d shape classification with a continuous voting space. In *Proceedings of International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2015. to appear. 7
10. Viktor Seib, Nicolai Wojke, Malte Knauf, and Dietrich Paulus. Detecting fine-grained affordances with an anthropomorphic agent model. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - Workshops of ECCV 2014*. Springer, 2014. 7
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. 2
12. Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Proc. of the European conference on computer vision (ECCV)*, ECCV'10, pages 356–369, Berlin, Heidelberg, 2010. Springer-Verlag. 7
13. Alexander Zelinsky. Robot navigation with learning. *Australian Computer Journal*, 20(2):85–93, 5 1988. 4
14. Alexander Zelinsky. *Environment Exploration and Path Planning Algorithms for a Mobile Robot using Sonar*. PhD thesis, Wollongong University, Australia, 1991. 4