RoboCup 2016 - homer@UniKoblenz (Germany)

Raphael Memmesheimer, Viktor Seib, Gregor Heuer, Patrik Schmidt, Darius Thies, Ivanna Mykhalchyshyna, Johannes Klöckner, Martin Schmitz, Niklas Yann Wettengel, Nils Geilen, Richard Schütz, Florian Polster, Dietrich Paulus

Active Vision Group
University of Koblenz-Landau
Universitätsstr. 1
56070 Koblenz, Germany
raphael@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 2016 in Leipzig, Germany. A special focus is put on novel system components and the open source contributions of our team. We have released packages for object recognition, a robot face including speech synthesis, mapping and navigation, speech recognition interface via android and a GUI. The packages are available (and new packages will be released) on http://wiki.ros.org/agas-ros-pkg.

1 Introduction

In 2015 *Lisa* and her team won the 1st place at RoboCup World Championship in the RoboCup@Home league in Hefei, China and were placed 2nd in the German Open.

Beside this sucess 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 demonstration award at RoCKIn Camp 2014 (Rome), 2015 (Pecioli), 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 the RoCKIn 2015 competition (Lisbon) team homer@UniKoblenz won the 1st overall rating together with SocRob, the Best Team Award, 1st place in the Navigation Challenge, 1st place in the Getting to Know my home task benchmark.

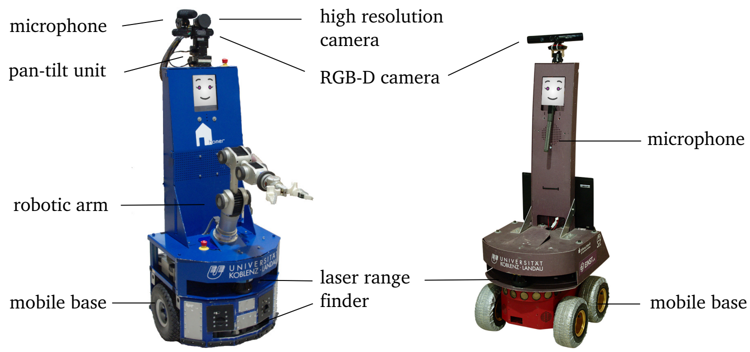

Fig. 1. Lisa (in blue) on the left is our main robot. The Lisa (right) serves as auxiliary robot.

In 2016 we plan to attend the RoboCup@Home in Leipzig, Germany, with two robots: 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 team is supervised by Viktor Seib and is lead by Raphael Memmesheimer.

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.

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* [12] 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

In this year's competition we will use two robots (Fig. 1). The blue Lisa is our main robot and is built upon a CU-2WD-Center robotics platform[1]. The old Lisa serves as an auxiliary robot and uses the Pioneer3-AT platform. Every robot is equipped with a single notebook that is responsible for all computations. Currently, we are using a Lenovo Thinkpad W520 equipped with an Intel Core i7-2670QM processor, 12 GB of RAM with Ubuntu Linux 14.04 and ROS Indigo.

Each robot is equipped with a laser range finder (LRF) for navigation and mapping. A second LRF at a lower height serves for small obstacle detection.

The most important sensors of the blue Lisa are set up on top of a pan-tilt unit. Thus, they can be rotated to search the environment or take a better view of a specific position of interest. Apart from a RGB-D camera (Asus Xtion) and a high resolution RGB camera (IDS UI-5580CP-C-HQ), a directional microphone (Rode VideoMic Pro) is mounted on the pan-tilt unit.

A 6 DOF robotic arm (Neuronics Katana 400HD) is used for mobile manipulation. It is certified for a safe operation around humans and is able to manipulate light-weight objects up to 0.5 kg. The end effector is a custom setup and consists of 4 Festo Finray-fingers.

Finally, a Raspberry Pi inside the casing of the blue Lisa is equipped with a 433 MHz radio emitter. It is used to switch device sockets and thus allows to use the robot as a mobile interface for smart home devices.

4 Software Contribution

We followed a recent call for chapters for a new book on ROS[2]. We want to share stable components of our software with the RoboCup and the ROS community to help advancing the research in robotics. All software components will are released on the Active Vision Group's ROS wiki page: `http://wiki.ros.org/agas-ros-pkg`. The contributions 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 2 shows an example of such a map.

[1] Manufacturer of our robotic platform: `http://www.ulrichc.de`

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

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 [14, 15], 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.

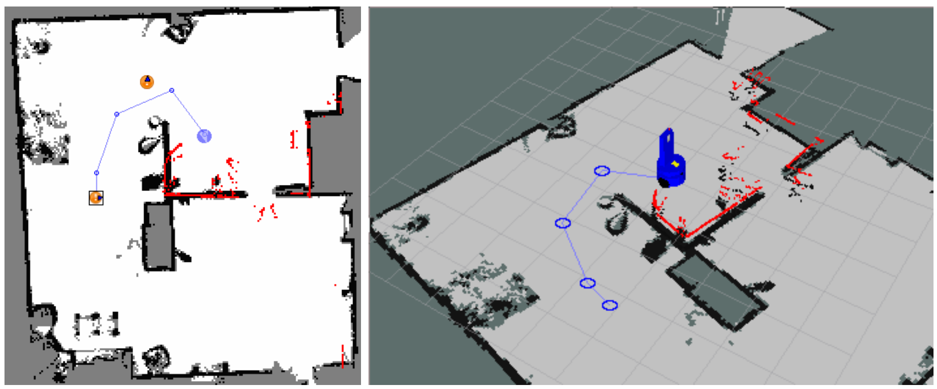

Fig. 2. 2D and 3D view of a map and a planned path (blue line). Red dots indicate the current laser scan, while orange points in the 2D map stand for navigation points.

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 [9]. With this object recognition approach we won the Technical Challenge 2012 (Figure 3).

Human Robot Interaction

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

Fig. 3. Object recognition results during the Technical Challenge 2012.

face expressions (Figure 4). 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.

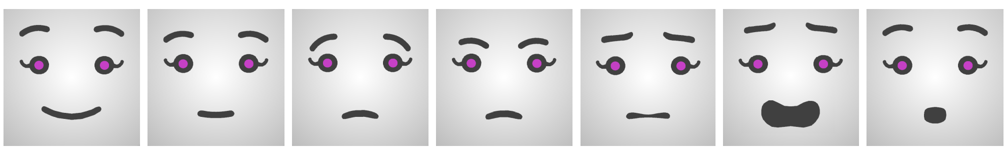

Fig. 4. Animated face of our service robot Lisa. The depicted face expressions are (from left to right): happy, neutral, sad, angry, disgusted, frightened, and surprised.

5 Technology and Scientific Contribution

5.1 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 3D Object Recognition

For 3D object recognition we use a continuous Hough-space voting scheme related to Implicit Shape Models (ISM). In our approach [10], SHOT features [13] 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].

5.3 Affordance Detection

Affordances have gained much popularity for object classification and scene analysis. Our current research focuses on analyzing scenes regarding sitting and lying possibilities for an anthropomorphic agent. Recently, we introduced the concept of fine-grained affordances [11] [8]. It allows to distinguish affordances on a fine-grained scale (e.g. sitting without backrest, sitting with backrest, sitting with armrests, lying) and thus facilitates the object classification process. Additionally, our approach estimates the sitting or lying pose with regard to the detected object (Figure 6).

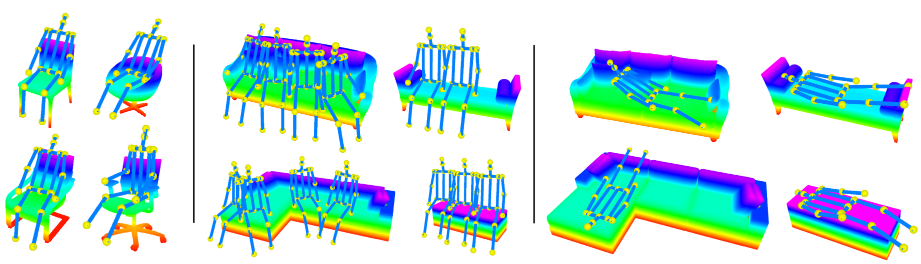

Fig. 5. Detected sitting and lying affordances and the corresponding pose of an anthropomorphic agent.

5.4 Speech Recognition

For speech recognition we use a grammar based solution supported by a academic license for the VoCon speech recognition software by Nuance[3]. 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.

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

5.5 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* 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.

References

1. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *ECCV*, pages 404–417, 2006.
2. Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.
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. 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.
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.
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.
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.
8. Viktor Seib, Malte Knauf, and Dietrich Paulus. Detecting fine-grained sitting affordances with fuzzy sets. In Nadia Magnenat-Thalmann, Paul Richard, Lars Linsen, Alexandru Telea, Sebastiano Battiato, Francisco Imai, and José Braz, editors, *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SciTePress, 2016.
9. 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.
10. 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.
11. 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.
12. 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.
13. 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.
14. Alexander Zelinsky. Robot navigation with learning. *Australian Computer Journal*, 20(2):85–93, 5 1988.
15. Alexander Zelinsky. *Environment Exploration and Path Planning Algorithms for a Mobile Robot using Sonar*. PhD thesis, Wollongong University, Australia, 1991.

Name of team homer@UniKoblenz

Member Raphael Memmesheimer, Viktor Seib, Gregor Heuer, Patrik Schmidt, Darius Thies, Ivanna Mykhalchyshyna, Johannes Klöckner, Martin Schmitz, Niklas Yann Wettengel, Nils Geilen, Richard Schütz, Florian Polster

Contact information raphael@uni-koblenz.de

Website `http://homer.uni-koblenz.de`

Hardware :

- Neuronics Katana 400HD
- Lenovo W520 Notebook
- Directed Perception D46-17.5 PTU
- Rode VideoMic Pro
- IDS UI-5580CP-C-HQ Camera
- Asus Xtion Pro Live
- CU2WD Robot platform
- Hokuyo Laser Scanner
- Raspberry Pi
- Pentax K30 DSLR

Software :

- ROS
- OpenCV
- PCL
- Festival TTS
- Ogre3D
- Nuance VoCon
- xtion_grabber
- strands
- Face++ cloud service (`http://www.faceplusplus.com`)
- Custom software for:
  - User Interface (homer_gui `https://gitlab.uni-koblenz.de/robbie/homer_gui`)
  - Object recognition (homer_object_recognition `https://gitlab.uni-koblenz.de/robbie/homer_object_recognition`)
  - Mapping / Navigation (homer_mapnav `https://gitlab.uni-koblenz.de/robbie/homer_mapnav`)
  - Robot face (homer_robot_face `https://gitlab.uni-koblenz.de/robbie/homer_robot_face`)
  - Speech Recognition / Speech synthesis (android_speech_pkg `https://gitlab.uni-koblenz.de/robbie/homer_android_speech`)