AllemaniACs Team Description RoboCup@Home

Stefan Schiffer and Gerhard Lakemeyer

Knowledge-Based Systems Group
RWTH Aachen University,
Aachen, Germany
{schiffer,gerhard}@cs.rwth-aachen.de

Abstract. This paper describes the scientific advances of the AllemaniACs team for the 2011 RoboCup@Home competitions. We present our robot *Caesar* and important modules of our robot control software which allow us to perform reliable service robotics applications in the @Home league. Furthermore, we report on our high-level programming language providing a powerful framework for agent behavior specification.

1 Introduction

While in the RoboCup soccer leagues the complexity of the task lies in fast accessing the sensors, quick decision making, and cooperation, the challenge in the @Home league is to build a system which enables a robot to robustly and safely navigate through human populated home environments. Since the new RoboCup@Home league focuses on service robotics applications another challenge is that of human-robot interaction. Tasks like following & guiding a human, learning and navigating within the environment, or manipulating objects are part of the @Home competition.

This means for one that the robot must be able to build an internal representation for arbitrary home environments. That is because the environment that the robot has to operate in for the competition is not known in advance. For another, the robot must be able to localize itself in this particular environment and it has to be able to navigate through it safely. This task surely demands for path planning and obstacle avoidance abilities. Our robot use a Monte Carlo approach with a laser range finder for localization. Furthermore, it employs an A^*-based collision avoidance algorithm and a path planner which ensures short paths between reachable points in the environment.

The high-level control is based on the language Readylog, a variant of the logic-based language GOLOG [1] which combines explicit agent programming as in imperative languages with the possibility to reasons about actions and their effects. In particular, we are interested in decision-theoretic planning in the Readylog framework which allows to generate optimal plans for complex tasks.

2 AllemaniACs @Home Robot *Caesar*

The mobile robot platform we use in the RoboCup@Home competitions is based upon the platform used in the AllemaniACs Mid-Size RoboCup Team until 2006. It features several improvements dedicated to the specific requirements in service robotics.

The robot *Caesar* has a size of 40 cm $\times$ 40 cm $\times$ 60 cm (Fig. 1). It is driven by a differential drive, the motors have a total power of 2.4 kW and are originally developed for electric wheel chairs. For power supply we have two 12 V lead-gel accumulators with 15 Ah each on-board. The battery power lasts for approximately one hour at full charge. This power provides us with a top speed of 3 m/s and 1000°/s at a total weight of approximately 60 kg. Our main sensor for navigation and localization is a 360° laser range finder (a Lase ELD-L-A[1]) with a resolution of 1° at a frequency of 10 Hz. For communication a WLAN adapter capable of using IEEE 802.11a/b/g is installed. This hardware platform was initially designed for soccer playing, but with almost no modifications we can easily also use it for service robotics applications. We report on our transition from the Mid-Size to the @Home league in [2].

Fig. 1: Robot platform and its main components

Since early 2007 we additionally have an anthropomorphic *robotic arm* called Katana6M180[2] from Neuronics which we use for manipulation tasks. The Katana is equipped with six motors providing six degrees of freedom. The arm's weight is around 4 kg and it has a maximal payload of 500 g. The arm is mounted on top of the mobile robot platform described above. To provide the arm with the required power, we mounted two additional 12 V lead gel accumulators on the robot.

On the very top of the robot we installed a Microsoft *Kinect sensor*. It is used for visual servoing of the arm as well as for face and object recognition. The camera has a field of view of 57° horizontally and 43° vertically. It can be tilted in a range of $\pm$ 27°. The camera provides 32-bit color images with a resolution of 640 $\times$ 480 and 16-bit depth information with a resolution of 320 $\times$ 240 both at 30 frames/s. The depth sensor ranges from 1.2 m to 3.5 m. The Kinect further also provides a 16-bit audio stream at 16 kHz. To increase flexibility and to better direct the robot's gaze for a particular task, the camera is mounted on a self-assembled *pan-tilt unit (PTU)*. It is made from two Tribotix

[1] http://www.lase.de/produkte/2dscanner/eld_l_a/en.html
[2] http://www.neuronics.ch/cms_en/web/index.php?id=244

Robotis Dynamixel RX-28 motors.[3] Communication happens via RS485; both motors can be controlled simultaneously over the same wire. To further improve on our sensing capabilities we additionally installed two URG-04LX-UG01[4] *laser range finders* from Hokuyo. They provide distance data in a range of up to 5.6 m in a 240° scan window at a resolution of 0.36° with 10 scans per second. With a size of only 50 mm × 50 mm × 70 mm and a weight of 160 g we could easily fit the two lasers on the robot, also because they run on 5 V and have a reasonably low power comsumption of 2.6 W.

To meet the increased demands in computational power we installed two Intel® Core™2 Duo computers running at 2 GHz with 2 GB RAM each.

3 Control Software

We are using the robot control framework Fawkes [3] for most of our components. Fawkes is open source software which we make freely available at `http://www.fawkesrobotics.org/`. However, some modules such as the localization and the navigation are still using our former software framework RCSoft. We use the 360° laser range finder as our main sensor for navigation, obstacle avoidance, and localization. In the following we describe the respective modules in more detail.

3.1 Collision Avoidance and Navigation

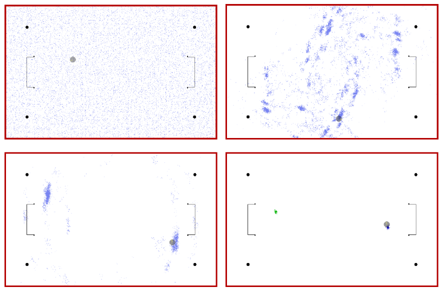
Our method for collision avoidance [4] was initially used in the RoboCup soccer domain, but it was designed to work in human populated environments such as office domains from the start. The collision avoidance module performs an A* search over an occupancy grid [5] generated from the laser scanner inputs. We perform an A* search from the robot's current location to the given target point. The path A* calculates must then be translated into motor commands.

3.2 Localization

Our self-localization uses the Monte Carlo Localization algorithm [6]. It works by approximating the position estimation by a set of weighted samples: $\mathbf{P}(l_t) \sim \{(l_{1,t}, w_{1,t}), \ldots, (l_{N,t}, w_{N,t})\} = \mathbf{S}_t$. Each sample represents one hypothesis for the pose of the robot. Roughly, the Monte Carlo Localization algorithm now chooses the most likely hypothesis given the previous estimate, the actual sensor input, the current motor commands, and a map of the environment.

To be able to localize robustly with the laser range finder we modified the Monte Carlo approach. To allow for the integration of a whole sweep from the LRF we use a heuristic perception model. With this we are able to localize with high accuracy in the RoboCup environment. The method is presented in detail in [7]. The approach, which was inspired by the RoboCup setting, works also

[3] `http://www.tribotix.com/Products/Robotis/Dynamixel/RX/RX28.htm`
[4] `http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html`



(a) Sample distribution in MCL

(b) Map Builder

Fig. 2. Sample distribution in localization on a RoboCup soccer field and a screen-shot of our map building application.

very well for indoor navigation even in large environments. An exemplary run is depicted in Figure 2(a).

3.3 Semantic Map Building

In order to be able to efficiently adapt to the frequent changes which are immanent in a home-like environment we developed a *semantic* map building application. It allows us to update the robot's world representation to the current situation very quickly. Our map builder uses a collection of semantically annotated objects that can be dragged and dropped to their specific location in a base-map. This simplifies the map building process to some few clicks. A screen-shot of the map builder is shown in Figure 2(b). Semantic annotations include a signature of the object as seen by the laser range finder, the area to be used in the obstacle server, and a name along with some common aliases. Additionally one could also include sample pictures of the respective object. The particular information for each object have to be provided beforehand, e.g. the signature of an object as seen by the laser range finder has to be drawn or recorded and pictures need to be taken and associated with the object. The items in the different low-level data structures are inter-referenced by their name. This way, each module can refer to an object or place by its name in human terminology.

3.4 Vision

Our vision system is able to perform object detection based on color segmentation and shape recognition. For visual servoing within manipulation tasks we additionally make use of 3D information we compute from the images of our stereo camera.

Face Detection and Recognition In order to work in a human environment a robot needs to have capabilities to detect humans and to tell them apart. Face detection and recognition is a feasible means to do so. Therefor, we employ an AdaBoost based method for face detection [8,9] which is available in OpenCV[1]. Further, we have developed an integrated approach for face detection and recognition using random forests [10] where face recognition can also be used separately. The recognition framework is able to integrate new identities to its database on the fly.

Object Recognition Object recognition becomes increasingly important, especially in service-robotics where the robot should be able to interact with objects in its environment. To improve our current object recognition capabilities we are integrating feature based methods for object recognition such as SIFT [11] and SURF [12].

3.5 Human-Robot Interaction

In a natural human environment interaction between the robot and the human beings around it is an integral part of the challenges in the @Home league. Therefore, we realize communication facilities in terms of a speech recognition module to process human instructions, requests, and questions and a synthesis module to generate spoken responses.

Speech Synthesis. For speech synthesis we make use of Festival[2]. It was developed at the University of Edinburgh and features a simple interface to pass text which is then synthesized as speech.

Speech Recognition. We are using the Sphinx[3] speech recognition software system from Carnegie Mellon University. We have build a robust speech recognition system [13] using Sphinx by first segmenting closed utterances potentially directed to the robot and then decoding with two different decoders in parallel. This allows for highly accurate recognition in restricted domains like RoboCup@Home. At the same time, false positives which are immanent in the noisy environments one is confronted with at RoboCup competitions can be filtered out reliably.

Gesture Recognition. Recently, we also developed a robust and flexible gesture recognition system. It uses a multi-staged approach inspired by the filter and refine technique. First, hands are detected in the image based on color. The color, however, is not trained in advance but taken from a face found in the image instead. The color is furthermore frequently updated to account for

---
[1] http://opencv.sourceforge.net/
[2] http://www.cstr.ed.ac.uk/projects/festival/
[3] http://cmusphinx.sourceforge.net/

changes, e.g., in illumination. After a hand is detected, its posture is classified using a random forest. Static gestures like pointing can already be recognized at this stage. To acocunt for dynamic gestures, the detected hands are tracked over time and then compared to a set of previously defined gesture prototypes.

Touchscreen Display. To further improve on the robot's interaction capabilities we mounted a 10.4 inch touchscreen display on the robot. It can be used to input commands and it display a face that can express basic emotions.

Spoken Command Interpretation. With the General Service Robot test in mind, we recently developed a system to interpret spoken human requests by means of decision-theoretic planning. Ambiguities in commands issued to the robot or errorneous language is resolved by the system automatically. The planning involves evaluating different possible interpretations and it may include queries for clarification to the user as well.

3.6 Sound Source Localization

One of our current research topics is sound source localization for mobile robots. We therefor developed a biologically inspired algorithm that uses interaural time differences to locate a sound source [14]. We also investigate whether and how useful fusion with other sensor modalities can be done [15].

3.7 Readylog

For specifying our high-level control we use READYLOG [16], a variant of the logic-based high-level agent programming language GOLOG [1]. GOLOG is a language based on the situation calculus [17]. Over the past years many extensions like dealing with concurrency, exogenous and sensing action, a continuous changing world and probabilistic projections (simulation) [18,19,20] made GOLOG an expressive robot programming language. We integrated those features in our READYLOG interpreter [21]. For the decision making, we further integrated a planning module into GOLOG which chooses the best action to perform by solving a Markov Decision Process (MDP) (we refer to [22] for reading on MDP and to [23] on integrating MDPs into GOLOG). For further information on READYLOG and other extensions integrated we refer to [24,25].

READYLOG can not only be used for behavior specification but also to formalize domain knowledge such as soccer theory [26]. In 2005 we developed a qualitative abstraction of the world model for the MID-SIZE domain [27]. The qualitative world model is integrated in the READYLOG language and used for abstract planning. The qualitative world model provides abstractions for positional information such as *left* or *right* as well as higher-level concepts like that of reachability which is fundamental in soccer. The qualitative spatial data provided are based on human cognition. Thus, they render useful especially when it comes to human-robot interaction since the robot can handle information which originate from human language more easily.

Later, we put this on a more formal basis in the situation calculus by introducing fuzzy representations [28] and control [29] to Readylog. The application of both, fuzzy representation and control, especially for domestic domains is reported on in [30].

To improve on the robustness of our system, we developed a method to integrate self-maintenance [31] in our high-level control. It accounts and takes care of run-time failures occuring in task execution. For example, if a module that provides some specific functionality needed for the current task is not loaded it is automatically started or if a particular component is in a wrong state this is being taken care of by appropriate maintenance actions.

4 Summary

In this paper we presented the AllemaniACs RoboCup@Home team. We described our robot platform and its main components at the present time. Then, we reported on various important modules within our control software. We also pointed to some of our current research topics. Finally, we gave an example on how our high-level control language Readylog can be used to implement service robotics applications.

References

1. Levesque, H.J., Reiter, R., Lesperance, Y., Lin, F., Scherl, R.B.: GOLOG: A logic programming language for dynamic domains. Journal of Logic Programming **31** (1997) 59–83
2. Schiffer, S., Ferrein, A., Lakemeyer, G.: Football is coming Home. In: Proc. of International Symposium on Practical Cognitive Agents and Robots (PCAR'06), University of Western Australia Press (2006)
3. Niemueller, T., Ferrein, A., Beck, D., Lakemeyer, G.: Design principles of the component-based robot software framework fawkes. In: Second International Conference on Simulation, Modeling, and Programming for Autonomous Robots, Darmstadt, Germany, Springer, Springer (2010)
4. Jacobs, S., Ferrein, A., Schiffer, S., Beck, D., Lakemeyer, G.: Robust collision avoidance in unknown domestic environments. In: Proceedings of the International RoboCup Symposium 2009 (RoboCup 2009). Volume 5949 of LNCS., Springer (2009) 116–127
5. Moravec, H., Elfes, A.: High resolution maps from wide angular sensors. In: Proc. of the IEEE Int. Conference On Robotics and Automation (ICRA). (1985) 116–121
6. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte Carlo localization for mobile robots. In: Proc. of the Int. Conf. on Robotics and Automation (ICRA). (1999)
7. Strack, A., Ferrein, A., Lakemeyer, G.: Laser-Based Localization with Sparse Landmarks. In: Proc. of the Int'l RoboCup Symposium. (2005)
8. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001). (2001) 511–518
9. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proc. of the Int'l Conf. on Image Processing (ICIIP), Rochester, USA, IEEE (2002) 900–903
10. Belle, V., Deselaers, T., Schiffer, S.: Randomized trees for real-time one-step face detection and recognition. In: Proc. of the 19th Int'l Conf. on Pattern Recognition (ICPR'08), IEEE Computer Society (2008) 1–4

11. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of the International Conference on Computer Vision ICCV, Corfu. (1999) 1150–1157
12. Herbert Bay, Tinne Tuytelaars, Luc Van Gool: SURF: Speeded Up Robust Features. In Leonardis, A., Bischof, H., Pinz, A., eds.: Proceedings of the ninth European Conference on Computer Vision, Springer (2006) 404–417
13. Doostdar, M., Schiffer, S., Lakemeyer, G.: Robust speech recognition for service robotics applications. In: Proceedings of the International RoboCup Symposium 2008 (RoboCup 2008). Volume 5399 of LNCS., Springer (2008) 1–12 Best Student Paper Award.
14. Calmes, L., Lakemeyer, G., Wagner, H.: Azimuthal sound localization using coincidence of timing across frequency on a robotic platform. Journal of the Acoustical Society of America **121** (2007) 2034–2048
15. Calmes, L., Wagner, H., Schiffer, S., Lakemeyer, G.: Combining Sound Localization and Laser based Object Recognition. In Tapus, A., Michalowski, M., Sabanovic, S., eds.: Multidisciplinary Collaboration for Socially Assistive Robotics, Papers from the AAAI Spring Symposium, Stanford CA, AAAI Press (2007) 1–6
16. Ferrein, A., Lakemeyer, G.: Logic-based robot control in highly dynamic domains. Robotics and Autonomous Systems, Special Issue on Semantic Knowledge in Robotics **56** (2008) 980–991
17. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press (2001)
18. Giacomo, G.D., Lésperance, Y., Levesque, H.J.: ConGolog, A concurrent programming language based on situation calculus. Artificial Intelligence **121** (2000)
19. Grosskreutz, H., Lakemeyer, G.: cc-Golog: Towards more realistic logic-based robot controllers. In: Proc. AAAI'00, AAAI Press (2000) 476–482
20. Grosskreutz, H.: Probabilistic projection and belief update in the pGolog framework. In: Second International Cognitive Robotics Workshop. (2000)
21. Dylla, F., Ferrein, A., Lakemeyer, G.: Specifying multirobot coordination in ICP-Golog – from simulation towards real robots. In: Proc. Physical Agents WS on IJCAI'03. (2003)
22. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley (1994)
23. Boutilier, C., Reiter, R., Soutchanski, M., Thrun, S.: Decision-theoretic, high-level agent programming in the situation calculus. In: AAAI'2000. (2000)
24. Ferrein, A., Fritz, C., Lakemeyer, G.: On-line decision-theoretic golog for unpredictable domains. In: Proc. of 27th German Conference on AI. (2004)
25. Ferrein, A., Fritz, C., Lakemeyer, G.: Using Golog for Deliberation and Team Coordination in Robotic Soccer. KI Künstliche Intelligenz (2005)
26. Dylla, F., Ferrein, A., Lakemeyer, G., Murray, J., Obst, O., Röfer, T., Schiffer, S., Stolzenburg, F., Visser, U., Wagner, T.: Approaching a formal soccer theory from behaviour specifications in robotic soccer. Computer in Sports. WIT Press (2008)
27. Schiffer, S., Ferrein, A., Lakemeyer, G.: Qualitative World Models for Soccer Robots. In: Qualitative Constraint Calculi, Workshop at KI 2006. (2006) 3–14
28. Ferrein, A., Schiffer, S., Lakemeyer, G.: A fuzzy logic semantics for qualitative fluents in the situation calculus. In: Proceedings of the International Conference on Intelligent Robotics and Applications (ICIRA'08). LNCS, Springer (2008) 498–509
29. Ferrein, A., Schiffer, S., Lakemeyer, G.: Embedding fuzzy controllers into golog. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'09), IEEE (2009) 894–899
30. Schiffer, S., Ferrein, A., Lakemeyer, G.: Fuzzy Representations and Control for Domestic Service Robots in Golog. In Iocchi, L., del Solar, J.R., van der Zant, T., eds.: Domestic Service Robots in the Real World. Workshop Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010), Darmstadt, Germany (2010) 183–192
31. Schiffer, S., Wortmann, A., Lakemeyer, G.: Self-Maintenance for Autonomous Robots controlled by ReadyLog. In Ingrand, F., Guiochet, J., eds.: Proceedings of the 7th IARP Workshop on Technical Challenges for Dependable Robots in Human Environments, Toulouse, France (2010) 101–107