AllemaniACs 2008 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 2008 RoboCup@Home competitions. We present important modules of our robot control software which allows 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.

In the sequel we describe our hardware platform in Section 2. We present some important modules of our control software in Section 3, before we give an example of a service robotics application of our high-level control language Readylog from the 2004 RoboCup Technical Challenge in Section 4.

2 AllemaniACs @Home Robot

The mobile robot platform we use in the 2008 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 has a size of 40 cm × 40 cm × 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. On-board we have two Pentium III PCs at 933 MHz running Linux, one equipped with a frame-grabber for a Sony EVI-D100P camera mounted on a pan/tilt unit. Our other sensor is a 360° laser range finder 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.

Fig. 1: Robot platform and its main components

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].

Since early 2007 we additionally have an anthropomorphic robotic arm called Katana6M180 which we use for manipulation tasks. The Katana is equipped with six motors providing six degrees of freedom. In our current configuration the joint connecting the gripper is mounted in a straight fashion. All six axes are equipped with "Harmonic Drive" gears which allow precise movements and a high repeatability. 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. Furthermore, we installed a stereoscopic camera on the very top of the robot in order to be able to acquire additional sensory input. This is then used for visual servoing of the arm. The camera is a Bumblebee 2 providing images in VGA resolution at 15 Hz from which depth information can be computed. In lack of a pan-tilt unit for the Bumblebee we mounted another camera on the very top of the robot. It is an iSight web cam from Apple that we use for face detection and recognition.

To meet the increased demands in computational power we additionally installed a Core 2 Duo computer running at 2 GHz with 2 GB RAM. It is mainly used for disparity image computation and to control the manipulator. Image processing for face detection and recognition is done on this machine as well.

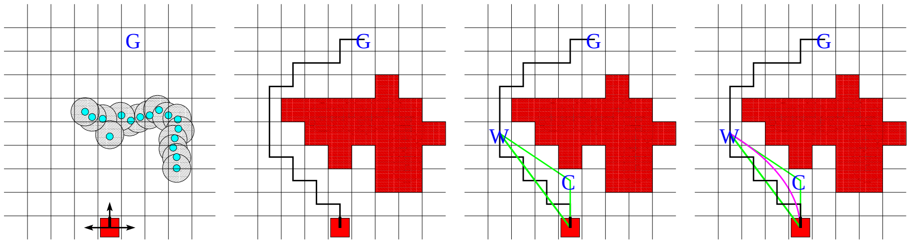
3 Control Modules

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

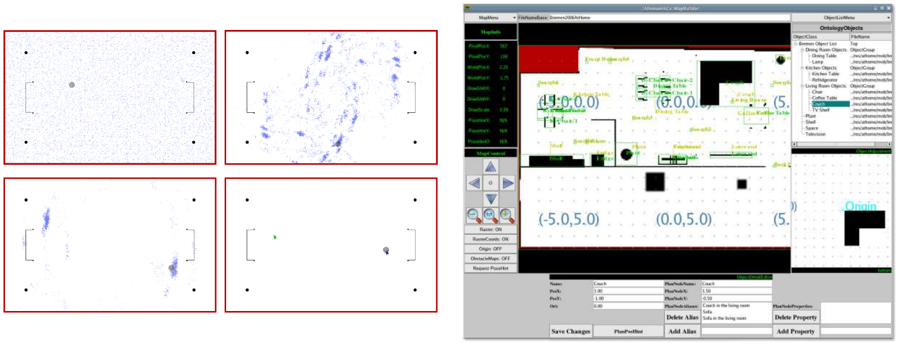
The collision avoidance module performs an A^* search over an occupancy grid [3] generated from the laser scanner inputs. The robot is positioned in the middle (origin) of the grid. Next, the collision-free path from the current location to a given target point must be calculated. We perform an A^* search from the robot's current location to the given target point. If the target point is located outside the grid range, we project the target point onto the border of the grid. To alleviate the search we extend the occupied cells by the size of our robot. Thus, the robot can be regarded as a mass point. The possible actions for the search are $A = \{N, S, W, E, NW, SW, ...\}$, i.e. the robot can move to any neighboring cell. To apply A^* we need to provide a cost function and a heuristic function. The cost function is the Euclidean distance between grid cells, as heuristic function we use the Manhattan distance to the target point.

The path A^* calculates must be translated into motor commands. Thus, we need a curve from which we can derive the appropriate commands sent to the motors. We approximate the steering commands by applying an A^* search over the velocity space. This search yields appropriate translational and rotational velocities with which the robot drives to the given target point. The general procedure is illustrated in Figure 2.



(a) obstacles (b) path search (c) way-point (d) velocity

Fig. 2. Different steps of the collision avoidance module.



(a) Sample distribution in MCL (b) Map Builder

Fig. 3. Sample distribution in localization on a RoboCup soccer field and a screen-shot of our map building application.

3.2 Localization

Our self-localization uses the Monte Carlo Localization algorithm [4]. 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 [5]. The approach, which was inspired by the RoboCup setting, works also very well for indoor navigation even in large environments. An exemplary run is depicted in Figure 3(a).

Object Classification For localizing the robot in the environment we build an occupancy grid map of the environment. With this map, we are able to determine dynamic obstacles in the environment when new laser readings arrive: every object which is not represented in the map is assumed to be a dynamic obstacle. To be able to distinguish between different dynamic objects, we use the laser signature of the objects. In the soccer setting we are able to distinguish between our own robots and opponents, and even humans can be told apart.

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 screenshot of the map builder is shown in Figure 3(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 a face detection module based on OpenCV [6,7]. Further, we have developed an integrated approach for face detection and recognition using random forests [8] 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 [9] and SURF [10].

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. It was developed at Carnegie Mellon University and features a simple interface to pass text which is then synthesized as speech. The initial FESTIVAL system is documented in [11].

Speech Recognition. We are using the Sphinx software system from Carnegie Mellon University for speech recognition. An overview of an early version of Sphinx is given in [12]. 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 quite reliable. To model the interaction we realized a simple dialog system which represents a task in a tree-like structure.

3.6 Sound Source Localization

One of our current research topics is sound source localization for mobile robots. We therefore 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].

4 Readylog

For specifying our high-level control we use a variant of the logic-based high-level agent programming language GOLOG [1]. GOLOG is a language based on the situation calculus [16]. Over the past years many extensions like dealing with concurrency, exogenous and sensing action, a continuous changing world and probabilistic projections (simulation) [17,18,19] made GOLOG an expressive robot programming language. We integrated those features in our Readylog interpreter [20]. 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 [21] for reading on MDP and to [22] on integrating MDPs into GOLOG). Readylog can not only be used for behavior specification but also to formalize domain knowledge such as soccer theory [23]. For further information on Readylog and other extensions integrated we refer to [24,25].

In 2005 we developed a qualitative abstraction of the world model for the Mid-Size domain [26]. 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.

In the RoboCup Mid-Size Technical Challenge 2004 we already presented a service robotics application: the robot was to drive autonomously to one particular soccer field which was chosen by one of the referees. The robot calculated the shortest way to the field, announcing historic sights of the exhibition hall like

(a) Part of the RoboCup 2004 Site

```
proc( pathPlan( Goal, H ),
        solve( while( not( mapNode = Goal ),
                      pickBest( child, childrenOf( mapNode ),
                      gotoMapNode( child ))),
              H, beAt( Goal ))).

function( beAt( Goal ), V,
           if( mapNode = Goal, V = 100, V = 0 )).

proc( gotoMapNode( Node ),
        [goto_global(ownNumber, nodeCoord( Node ), 1),
         playSound( Node )]).
```

(b) The Readylog program for the 2004 challenge.

Fig. 4. Technical Challenge 2004

the stand of the fields or the pillars of the hall on the way. Fig. 4(a) shows the occupancy map of the rear part of the exhibition hall. In the upper part you can probably detect two of the Mid-Size fields. The high-level control program for the challenge is shown in Fig. 4(b). The parameter *Goal* specifies the target location. Inside the *solve* statement the planning takes place: as long as the robot has not reached the target, it chooses the best neighbor to the current node in its topological map (which is not shown here) applying the *pickBest* statement. The reward function we use here is the *beAt* function, which gives a high reward at the target position and 0 otherwise. The cost function for an action which is also not presented here is defined as the Euclidean distance between nodes in the topological map. Thus, the robot finds the shortest sequence of nodes to the target position. At each node the robot calls the *gotoMapNode* procedure and plays a sound file to announce the exhibit.

5 Summary

In this paper we presented the AllemaniACs 2008 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. 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
4. 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)
5. Strack, A., Ferrein, A., Lakemeyer, G.: Laser-Based Localization with Sparse Landmarks. In: Proc. RoboCup Symposium 2005. (2005)
6. 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
7. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proceedings of the International Conference on Image Processing (ICIIP), Rochester, USA, IEEE (2002) 900–903
8. Breiman, L.: Random forests. Machine Learning **45** (2001) 5–32
9. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of the International Conference on Computer Vision ICCV, Corfu. (1999) 1150–1157
10. 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
11. Black, A.W., Taylor, P.A.: The Festival Speech Synthesis System: System documentation. Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Scotland, UK (1997)
12. Huang, X., Alleva, F., Hon, H.W., Hwang, M.Y., Rosenfeld, R.: The SPHINX-II speech recognition system: an overview. Computer Speech and Language **7** (1993) 137–148
13. Doostdar, M., Schiffer, S., Lakemeyer, G.: Robust Speech Recognition for Service-Robotics Applications. In: Proceedings of the RoboCup Symposium 2008. (2008) to appear.
14. Calmes, L., Lakemeyer, G., Wagner, H.: Azimuthal sound localization using coincidence of timing across frequency on a robotic platform. **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. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press (2001)
17. Giacomo, G.D., Lésperance, Y., Levesque, H.J.: ConGolog, A concurrent programming language based on situation calculus. Artificial Intelligence **121** (2000)
18. Grosskreutz, H., Lakemeyer, G.: cc-Golog: Towards more realistic logic-based robot controllers. In: Proc. AAAI'00, AAAI Press (2000) 476–482
19. Grosskreutz, H.: Probabilistic projection and belief update in the pGolog framework. In: Second International Cognitive Robotics Workshop. (2000)
20. Dylla, F., Ferrein, A., Lakemeyer, G.: Specifying multirobot coordination in ICPGolog – from simulation towards real robots. In: Proc. Physical Agents WS on IJCAI'03. (2003)
21. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley (1994)
22. Boutilier, C., Reiter, R., Soutchanski, M., Thrun, S.: Decision-theoretic, high-level agent programming in the situation calculus. In: AAAI'2000. (2000)
23. 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)
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. Schiffer, S., Ferrein, A., Lakemeyer, G.: Qualitative World Models for Soccer Robots. In: Qualitative Constraint Calculi, Workshop at KI 2006. (2006) 3–14