Middle Robots League, Team *Ulm-Sparrows*, pages 211–216
http://www.ep.liu.se/ea/cis/1999/006/34/

The Ulm Sparrows 99

Ulm-Sparrows

Stefan Sablatnög, Stefan Enderle, Mark Dettinger, Thomas Boss, Mohammad Livani, Michael Dietz, Jan Giebel, Urban Meis, Heiko Folkerts, Alexander Neubeck, Peter Schäffer, Marcus Ritter, Hans Braxmeier, Dominik Maschke, Gerhard Kraetzschmar, Jörg Kaiser, Günther Palm

SS, SE, MD, TB, ML, MD, JG, UM, HF, AN, PS, MR, HB, DM, GK, JK, GP: University of Ulm

Abstract. *In RoboCup-98, ULM-Sparrows team worked hard just to get both a simulation and a middle size robot team to work and to successfully participate in a major tournament. For this year, we were in a better position to start some more serious research work. Aside of improvements in the robot hardware and an extension of the vision processing capabilities, we implemented a more complete version of our soccer agent architecture and made some progress in the areas player localization, environment modelling, and basic playing skills. For the latter, we started to apply learning techniques.*

1 Motivation and Research Goals

ULM-Sparrows is a research effort seeking to investigate and solve open problems relevant to both the RoboCup Challenge [KAK+97] and a local interdisciplinary research effort called SMART [PK97][1]. Some research issues of particular interest to our team include *skill learning* in continuous domains, *adaptive spatial modeling* of highly dynamic environments, and *emergent multiagent cooperation* for achieving coordinated team play *without* explicit communication. We also have a general interest in studying *robot control architectures* for soccer agents and *neurosymbolic integration*, in particular the integration of symbolic and neural methods in robot control architectures. See [KESo99] for a more detailed account of our goals. We have both a simulation team and a middle size real robot team to pursue these goals.

[1] See **www.uni-ulm.de/SMART/** for more information.

2 Improved Robot Hardware

In RoboCup-98, we used three modified Pioneer-1 robots, a LEGO-based robot, and a custom-built goalie based on a toy tracked vehicle. The performance of our robot team suffered severely from mechanical problems and electrical instability. Using the same code as in Paris, but after some modifications improving the stability situation, we could successfully participate in the VisionCup-98 in September, where we could beat the teams from Munich and Tuebingen (2nd in Paris) and prove the feasibility and validity of our approach.

For RoboCup-99, we decided to make further enhancements to our hardware platform and designed and developed a modular, distributed architecure for our soccer robots. The principle idea (see [?]) is to locally couple sensors and actuators with comparatively cheap microcontrollers that perform varying amounts of processing. A CAN bus connects all *smart devices* and is used by *smart sensors* to deliver preprocessed results and by *smart actuators* to receive the required control inputs. This *smart device architecture* [?] brings – so to speak – computation closer to where the data is and reduces communication load while providing an immense amount of modularity and flexibility. As a result, we could structure the mechanical design of our soccer robot into several independent modules, which must be connected by just four pins: two for power supply, and two for the CAN bus.

3 Extended Vision Capabilities

Last year, our robots were only able to see the goals and the ball. This limitation was due to the small frame resolution and low frame rate which was caused by the port-based interface between frame grabber and the vision processing CPU. By replacing the main PC/104 board and frame grabber with a combination that uses a PCI bus interface, we hope to remedy this situation and extend our vision capabilities to recognize players as well and to help us in localization.

The behavior-based structure of the color-tracking vision processing software more or less remains the same: we just need to invoke extra behaviors for tracking different colors.

4 Implementation of Soccer Agent Architecture

Encouraged by the convincing demonstration of its feasibility and – considering our opponents' much stronger hardware capabilities – the surpising success at VisionCup-98, we continued with the implementation of our soccer agent architecture (see Fig. 1 and [KESo99] for details), which is applied both in the simulation and the real robot teams. We implemented a C++ software library that allows to quickly implement and modify the reactive layer of the architecture. Amongst other things, the library provides behavior and arbiter classes, which are easy to instantiate and ensure safe execution of behaviors and arbiters as parallel threads. The library also

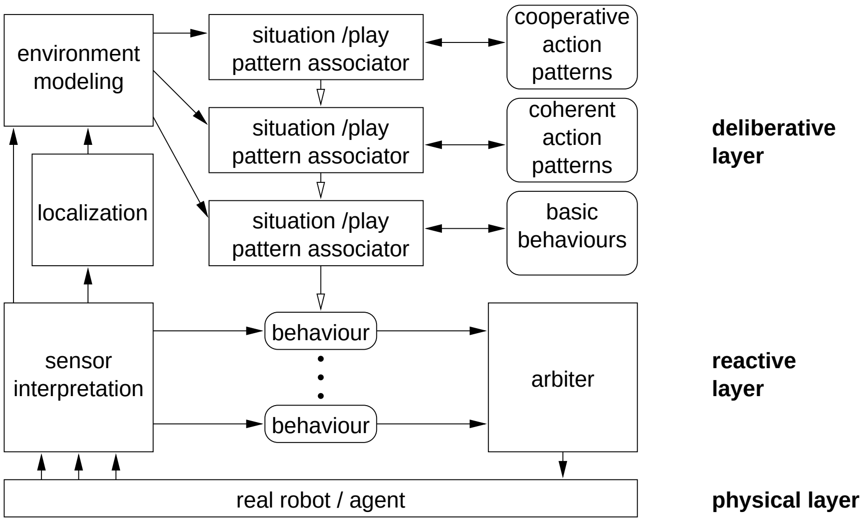

Figure 1: ULM-Sparrows soccer agent architecture

provides an abstract interface to a soccer agent or a real robot. We are currently working on extensions to make sensor interpretation, localization and environment modeling more behavior-oriented and to include appropriate classes in the software library. Furthermore, we are working on software that allows to construct coherent action patterns in a much simpler way. The programmer uses a graphical user interface to specify a temporal sequence of behavior sets and the required events and resulting signals to switch between them. From this graphical specification, the program code can be automatically generated.

5 Environment Modelling

We apply multi-layered spatial representations for modeling the environment. In our soccer agents, we currently employ a two-layer approach that is derived and adapted from the dynamo spatial representation architecture (see [KCES99]) developed in the smart project.

The lower layer consists of an *egocentric* representation. Contrary to smart, where we use two-dimensional probabilistic occupancy maps, we use multiple feature-based maps in robocup (see Fig. 2). Relevant features include position and/or distance of the ball, both of the goals, other robots, and field landmarks (flags in simulation; corner bars, wall markings, and possibly logos of advertisement in real robots). A focus of attention window is attached to the map to model the limited view of soccer agents. Updating of this representation is based upon sensory input in the focus of attention, while model-based extrapolation is applied to the remainder (this is currently available in simulation only). The egocentric representation is mainly used by low-level behaviors for rapid action selection.

The upper layer of the environment model is an *allocentric* spatial representation, which is constructed and maintained by integrating over time information present in the egocentric representation. Due to differences in the information available, the mechanisms differ in the simulation and robot

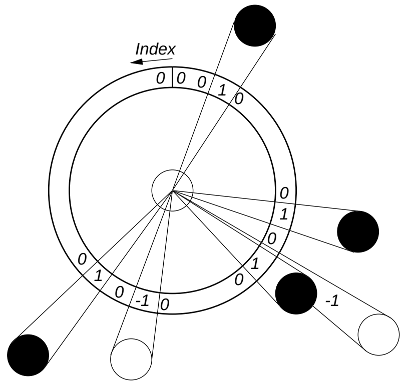

Figure 2: Egocentric feature map representation for soccer agents

teams: In the simulation league the soccer server almost always delivers sufficient information on global landmarks (flags) but provides essentially no odometry information. Because the position of flags is a priori known, the robot's position on the field is determined first. Then other players are mapped based on relative position information provided by the soccer server. The robots, however, do deliver odometry data but have a much more limited field of view; as a result, they often detect little or no global landmark information. Therefore, in the real robots the environment model must rely somewhat more on extrapolation based on odometry data and a more elaborated mechanisms for integrating localization and updating the environment model must be applied.

6 Player Localization

In the simulation league localization is an almost trivial problem, because almost always sufficient information is available to localize the agent with respect to a global reference frame. As pointed out above, the situation is quite different in real robots. We apply an approach called Monte Carlo localization, which was originally developed by Dellaert et al. [DFBT99] and was locally adapted to suit our needs in robocup. In the Monte Carlo localization approach a probability density function (PDF) is used to model the robot's location. Assuming a given PDF at some time step t_0, the update of the PDF to time t_1 is divided into the two steps prediction and update. In the prediction phase, the current PDF is changed according to the robot motion since the last update phase. Formally, the new state x_k is only dependent on the previous state x_{k-1}, and a known control input u_{k-1}. In the update phase the PDF is adapted based on sensory input received after the command has been performed.

The information we actually exploit as control input is not the command send to motor controllers, but the data provided by odometry sensors. Thereby we avoid the error made by the controller, but still have the error made by odometry. In Dellaert et al. the sensor input consists of laser or sonar scans. We apply this approach to feature maps extracted by the vision module (see above). In order to do this, we must be able to estimate feature maps given a specific position on the field, and we must have a similarity measure between feature maps. Update of the PDF is then performed based on sampling positions based on the predicted PDF and computing a similarity measure between the estimated and the sensed feature map.

7 Learning Basic Skills

As demonstrated very convincingly e.g. by CMUnited-98, it is absolutely necessary to provide robust low level behaviours. Although behaviors can be handcrafted, this is a very tedious process that must possibly be redone every time some system parameter changes. Such parameter changes have occurred almost every year in both leagues we compete in, e.g. in the soccer server the size of the physical agent and stamina model parameters haven been changed, as well as the field size and lighting conditions in the middle size league.

All our behaviors so far have been hand-crafted, but we are now starting to apply reinforcement learning techniques to this problem. Definite results are not yet available but will be included in the final version.

8 Conclusions

ULM-Sparrows team development follows the lines of research set out in our initial team description paper [KESo99]. We hope to make some substantial progress without endangering system stability, and hope to perform well in RoboCup-99.

References

[DFBT99] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Proceedings of IEEE International Conference on Robotics and Aautomation 1999 (ICRA-99)*, 1999. to appear.

[KAK+97] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara. Robocup a challenge problem for ai. *AI magazine*, 18(1):73–85, 1997.

[KCES99] Gerhard K. Kraetzschmar, Ralf Corsépius, Stefan Enderle, and Stefan Sablatng. Towards adaptive hybrid spatial representations for autonomous mobile robots. In *Proceedings of EUROBOT-99*, 1999. submitted.

[KESo99] Gerhard K. Kraetzschmar, Stefan Enderle, Stefan Sablatng, and other. The ulm sparrows: Research into sensorimotor integration, agency, learning, and multiagent cooperation. In Minoru Asada, editor, *RoboCup-98*, volume TBD of *Lecture Notes in Artificial Intelligence*. Springer, 1999. to appear.

[PK97] Gnther Palm and Gerhard Kraetzschmar. Integration symbolischer und subsymbolischer Informationsverarbeitung in adaptiven sensomotorischen Systemen. In Matthias Jarke, Klaus Pasedach, and Klaus Pohl, editors, *Informatik 97: Informatik* *als Innovationsmotor*, Informatik aktuell, pages 111–120, Heidelberg, Germany, 1997. GI, Springer.