AGILO RoboCuppers 2001
Utility- and Plan-based Action Selection based on Probabilistically Estimated Game Situations

Thorsten Schmitt, Sebastian Buck and Michael Beetz

Forschungsgruppe Bildverstehen (FG BV) – Informatik IX
Technische Universität München, Germany
{schmittt,buck,beetzm}@in.tum.de
http://www9.in.tum.de/robocup/

Abstract. This paper describes the *AGILO RoboCuppers* [1] the RoboCup team of the image understanding group (FG BV) at the Technische Universität München. With a team of four Pioneer I robots, all equipped with CCD camera and a single board computer, we've participated in all international middle size league tournaments from 1998 until 2001. We use a modular approach of concurrent subprograms for image processing, self localization, object tracking, action selection, path planning and basic robot control. A fast feature extraction process provides the data necessary for the on-board scene interpretation. All robot observations are fused into a single environmental model, which forms the basis for action selection, path planning and low-level robot control.

1 Introduction

The purpose of our RoboCup activities is to provide a software and hardware platform for (1) pursuing research in robot vision, probabilistic state estimation, multi-robot cooperation, experience based learning, and plan-based control; (2) supporting undergraduate and graduate education in computer vision, artificial intelligence, and robotics; and (3) performing software engineering of large dynamic software systems.

2 Hardware Architecture

We aim at realizing the AGILO RoboCup team on the basis of inexpensive, off the shelf, easily extendible hardware components and a standard software environment.

The team consists of four Pioneer I robots [1] each equipped with a single onboard linux computer (Pentium 200 MHz CPU, 64 MB RAM, 2.5" hard disk, on-board ethernet, and VGA controller, and an inexpensive BT848-based PCI video capture card). They are supported by an additional PC which resides outside the playing field. It is used to fuse the observations made by the individual team members and to monitor the robots' data and states. All robot computers and the additional PC are linked via a 10 Mbps wireless ethernet (5.8 GHz) [3, 2].

Fig. 1 (a) shows two[2] of our Pioneer 1 robots. Inside the robot a Motorola microprocessor controls the motors, reads the wheel encoders and the seven ultrasonic sonars, and communicates with the single board computer EM-500, which is mounted within a box on the topside of the robot. PC and robot are connected via a standard RS232 serial

[1] The name is derived from the Agilolfinger, which were the first Bavarian ruling dynasty in the 8th century, with Tassilo as its most famous representative.
[2] All attackers and defenders have identical shape and equipment.

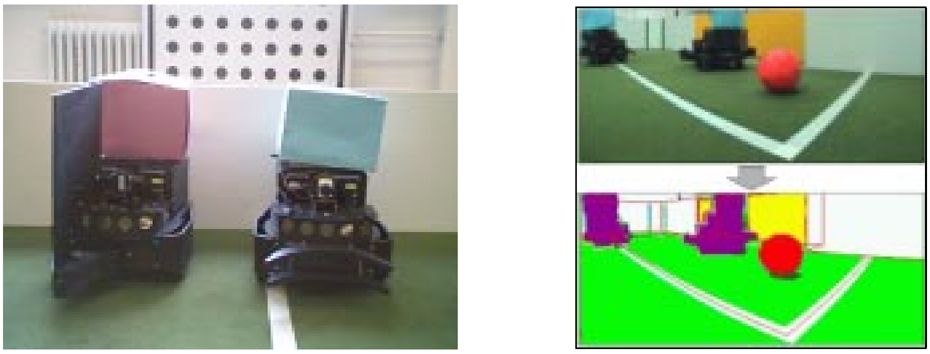

Fig. 1. (a) Theodo (goal keeper), Hugibert (attacker) and (b) what they perceive of the world around them.

port. A PAL color CCD camera is mounted on top of the robot console and linked to the S-VHS input of the video capture card. Gain, shutter time, and white balance of the camera are adjusted manually. Currently RGB-16 images are captured with a resolution of 384×172 (bottom 60% of half PAL resolution). For better ball guidance we've mounted a simple concave-shaped bar in front of each robot. A custom made kicking device enables the robot to kick the ball in direction of the robot's current orientation.

3 Fundamental Software Concepts

The software architecture [9] of our system is based on several independent concurrent modules. The modules are organized hierarchically into main, intermediate, and basic modules. The main modules are image (sensor) analysis, information fusion, action selection, path planning, and robot control. Beside the main modules the system employs auxiliary modules for monitoring the robots' states. The key software methods employed by the AGILO RoboCuppers software are:

1. vision-based cooperative state estimation for dynamic environments,
2. synergetic coupling of programming and experience-based learning for movement control and situated action selection, and
3. plan-based control of robot teams using structured reactive controllers (SRCs).

3.1 Self Localization, Object Tracking and Data Fusion

The vision module is a key part of our system. Given a raw video stream, the module has to estimate the pose of the robot and the positions of the ball and opponent robots. Low/medium-level image processing operations are performed with the help of the image processing library HALCON 5.2.3 (formerly known as HORUS [7]).

We have developed a probabilistic, vision-based state estimation method for individual, autonomous robots. This method enables a team of mobile robots to estimate their joint positions in a known environment and track the positions of autonomously moving other objects. All poses and positions of the state estimation module contain a description of uncertainty, i.e. a covariance matrix. The state estimators of different robots cooperate to increase the accuracy and reliability of the estimation process. This cooperation between the robots enables them to track temporarily occluded objects and to faster recover their positions after they have lost track of them. A detailed description of the self localization algorithm can be found in [8] and the algorithms used for cooperative multi-object tracking are explained in [13, 12].

Our vision algorithms can process up to 25 frames per second (fps) on a 200 MHz Pentium PC. The average number of images processed during a match is between 12 and 17 fps. This is due to computational resources being shared with the path planning and action selection modules.

3.2 Experience Based Learning for Situated Action Selection, Path Planning and Movement Control

Another major field of our research activities is *automatic robot learning based on experiences gained from exploration.* Experience based learning provides a powerful tool for the automatic construction of high-performance action selection and low-level robot control. In this respect experience based learning can effectively complement other methods for developing such controllers, in particular the hand coding of controllers. We use learning from experience in several parts of our system such as low level robot control, path planning and action selection.

In low level robot control we represent the state of a Pioneer I robot as a quintuple $\zeta = \langle x, y, \varphi, v, w \rangle$, where x and y are coordinates in a global system, φ is the orientation of the robot and v and w are the translational and rotational velocities, respectively. The low-level robot controller accepts commands of the form $\xi = \langle v, w \rangle$. A neural network maps the desired state changes to low level robot commands: $Net : \langle \zeta_0, \zeta_{target} \rangle \mapsto \xi_0$. To train this network we measure a huge number of state changes according to different executed low level commands [6]. Doing so our neural controller is based on nothing but experience *not* making any assumptions.
In order to find the optimal path planning algorithm for our RoboCup robots we statistically evaluated different methods and found out that there is *no optimal* algorithm but a number of navigation problem classes each performed best with a certain algorithm/parameterization [6]. These classes are defined with the help of a feature language. In order to select the best method for the given situation we've learned a decision tree [11]. The training data is obtained from accurate robot simulations where a huge number of path planning problems were performed with different algorithms each.

The selection of an appropriate action is performed on the basis of a fused environmental model. A set of possible actions A such as go2ball, shoot2goal, dribble, block... is defined. For all robots and each of those actions a_i success rates $P(a_i)$ and gains $G(a_i)$ are estimated [5]. From all promising actions $P(a_i)$, which exceed a predefined threshold Θ_{a_i} the one a_{ex} with the highest gain is chosen to be carried out.

3.3 Plan-based Action Control

While our situated action selection aims at choosing actions that have the highest expected utility in the respective situation it does not take into account a *strategic* assessment of the alternative actions and the respective *intentions* of the team mates. This is the task of the plan-based action control.

In order to realize an action assessment based on strategic consideration and on a considerations of the intentions of the teammates, we develop a robot soccer playbook, a library of plan schemata that specify how to perform individual team plays. The plans, or better plays, are triggered by opportunities, for example, the opponent team leaving one side open. The plays themselves specify highly reactive, conditional, and properly synchronized behavior for the individual players of the team.

The high-level controller is realized as a structured reactive controller (SRC) [4] and implemented in an extended RPL plan language [10]. The high-level controller works as follows. It executes as a default strategy the situated action selection that we have described in Sec. 3.2. At the same time, the controller continually monitors the estimated game situation in order to detect opportunities for making plays. If an opportunity is detected, the controller decides based on circumstances including the score and the estimated success probability of the intended play whether or not to perform the play.

Our research goals in the development of the high-level controller include the development of a computational model for plan-based control in very dynamic multi-robot applications and for the integration of learning processes into the plan-based control.

References

1. ActivMedia Robotics. http://www.activmedia.com/robots/.
2. Delta Network Software GmbH. http://www.delta-network-systems.com/, http://www.delta-net.de.
3. RadioLAN Inc. http://www.radiolan.com.
4. M. Beetz. Structured reactive controllers. *Journal of Autonomous Agents and Multi-Agent Systems*, 4:25–55, 2001.
5. S. Buck and M. Riedmiller. Learning Situation Dependant Success Rates of Actions in a RoboCup Scenario. In R. Mizoguchi and J. Slaney, editors, *Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, number 1886 in Lecture Notes in Artificial Intelligence, page 809. Springer-Verlag, 2000.
6. S. Buck, U. Weber, M. Beetz, and Th. Schmitt. Multi robot path planning for dynamic evironments: A case study. In *5th International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)*, 2001 submitted.
7. Wolfgang Eckstein and Carsten Steger. Architecture for Computer Vision Application Development within the HORUS System. *Journal of Electronic Imaging*, 6(2):244–261, April 1997.
8. R. Hanek and T. Schmitt. Vision-based localization and data fusion in a system of cooperating mobile robots. In *International Conference on Intelligent Robots and Systems (IROS)*, 2000.
9. M. Klupsch. Object-Oriented Representation of Time-Varying Data Sequences in Multiagent Systems. In N.C. Callaos, editor, *International Conference on Information Systems Analysis and Synthesis (ISAS'98)*, pages 833–839, Orlando, FL, USA, 1998. International Institute of Informatics and Systemics (IIIS).
10. D. McDermott. A reactive plan language. Research Report YALEU/DCS/RR-864, Yale University, 1991.
11. R. Quinlan. Induction of decision trees. In *Machine Learning 1 (1)*, 1986.
12. D. Reid. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control, 24(6):843–854, 1979.
13. T. Schmitt, R. Hanek, S. Buck, and M. Beetz. Cooperative probabilistic state estimation for vision-based autonomous soccer robots. In *5th International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)*, volume 5 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001 submitted.