Agilo RoboCuppers: RoboCup Team Description

Sebastian Buck, Robert Hanek, Michael Klupsch, Thorsten Schmitt

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

Abstract. This paper describes the *Agilo RoboCuppers* [1] team of the image understanding group (FG BV) at the Technische Universität München. With a team of four Pioneer 1 robots, equipped with CCD camera and a single board computer each and coordinated by a master PC outside the field we participate in the Middle Size League of the fourth international RoboCup Tournament in Melbourne 2000. We use a multi-agent based approach to represent different robots and to encapsulate concurrent tasks within the robots. A fast feature extraction based on the image processing library HALCON provides the data necessary for the on-board scene interpretation. All robot observations are fused to one single consistent view. Decision making is done on this fused data.

1 Introduction

The purpose of our RoboCup activities is to develop software components, frameworks, and tools which can be used flexibly for several tasks within different scenarios under basic conditions, similar to robot soccer. Our work is also used for teaching students in vision, machine learning, robotics and last but not least in developing large dynamic software systems. For this reason, our basic development criterion is to use inexpensive, easy extendible standard components and a standard software environment.

2 Hardware Architecture

Our RoboCup team consists mainly of four Pioneer 1 robots [1] each equipped with a single board computer. They are supported by a master PC (coach), which is also used in order to display the robot's data and states. The single board computers are mounted on the top of the robots. All robot computers are linked via a 10 Mbps radio Ethernet network [4, 5]. A master computer is located outside the soccer field and is linked to the radio Ethernet, too. It can also be used for debugging purposes, monitoring the robots' decision states and feature extraction processes. The operating system of the computers is Linux. Figure 1 demonstrates the hardware architecture.

---

[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.

P. Stone, T. Balch, and G. Kraetzschmar (Eds.): RoboCup 2000, LNAI 2019, pp. 567-570, 2001.

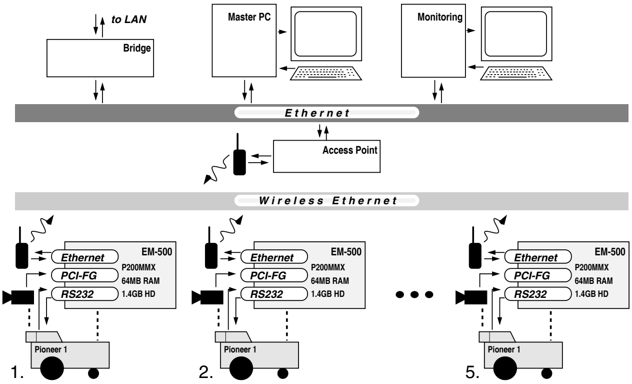

Fig. 1. Hardware architecture.

Figure 2 (a) shows one of our Pioneer 1 robots. Each of them measures $45\,\mathrm{cm} \times 36\,\mathrm{cm} \times 56\,\mathrm{cm}$ in length, width, and height and weighs about 12 kg. Inside the robot a Motorola microprocessor is in charge of controlling the drive motors, reading the position encoders, for the seven ultrasonic sonars, and for communicating with the client. In our case this is a single board computer (EM-500 from [2]) which is mounted within a box on the topside of the robot. It is equipped with a Pentium 200 MHz processor, 64 MB RAM, 2.5" hard disk, onboard Ethernet and VGA controller, and an inexpensive BT848-based [7] PCI video capture card [3]. PC and robot are connected via a standard RS232 serial 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. For better ball guidance we mounted a simple concave-shaped bar in front of each robot. A kicking device enables the robot to push the ball in direction of the robot's current orientation.

3 Fundamental Software Concepts

The software architecture of our system is based on several independent modules. Software agents control the modules, they decide what to do next and are able to adapt the behavior of the modules they are in charge for according to their current goal. For this, several threads run in parallel. The modules are organized hierarchically, within the main modules basic or intermediate ones can be used. The main modules are image (sensor) analysis, robot control, information fusion, and decision making. The latter runs on the master PC outside the field, the others on the single board computers on the robots.

Beside the main modules there are some auxiliary modules (monitoring the robots' states). For the communication between different modules, we strictly distinguish between controlling and data flow. One module can control another by sending messages to the appropriate agent. Data accessed by various modules is handled in a different manner. For this, a special sequence object class was

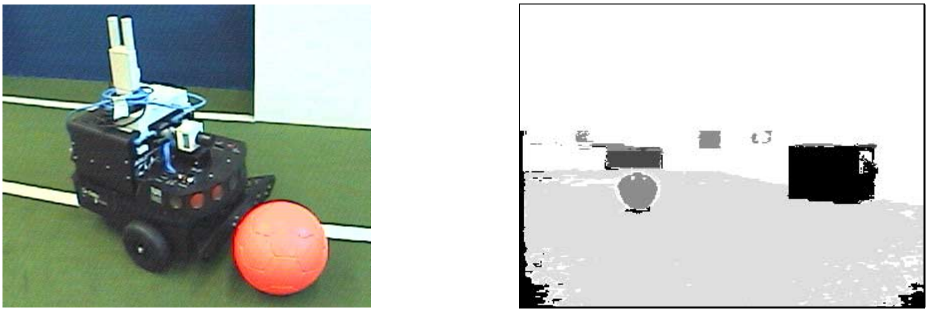

Fig. 2. (a) Odilo – one of our Pioneer 1 robots – and (b) what he percepts of the world around him.

defined. This offers a consistent concept for exchanging dynamic data between arbitrary components [11].

4 Vision

The vision module is a key part of the whole system. Given a raw video stream, the module has to recognize relevant objects in the surrounding world and provide their positions on the field to other modules. This is done with the help of the image processing library HALCON (formerly known as HORUS [9, 6]). The frame-grabber interface was extended to features for capturing gray scale images and color regions at the same time. For this we use the RGB-image data provided by the video capture card. The color regions can be achieved very fast by a three-dimensional histogram-based classifier. Gray scale images, color regions and the extracted data are provided by sequence objects as described in section 3. As a compromise between accuracy and speed we capture the images with half the PAL resolution clipping the upper 40 percent. This results in a resolution of 384 × 172 with a frame rate of about 12 images per second.

In general, the task of scene interpretation is a very difficult one. However, its complexity strongly depends on the context of a scene which has to be interpreted. In RoboCup, as it is defined in the present, the appearance of relevant objects is well known. For their recognition, the strictly defined constraints of color and shape are saved in the model database and can be used. These constraints are matched with the extracted image features such as color regions and line segments [10] (see Fig. 2 (b)).

Besides recognizing relevant objects with the help of the color regions, a second task of the image interpretation module is to localize the recognized objects and to perform a self-localization on the field. To localize objects we use the lowest point of the appropriate color regions over the floor in conjunction with a known camera pose relative to the robot. Self-localization is performed by matching the 3D geometric field model to the extracted line segments of the border lines and – if visible – to a goal. A sub-pixel accurate edge filter performed on the gray-scale image supplies contours from which, after removing radial distortions, straight line segments are extracted. Both, an absolute initial localization as well as a successive refinement, compensating the error of the odometric data have been implemented.

5 Decision Making

Decision making is done on the fused world model. Choosing an appropriate action relies on this data. A number of possible actions such as go2ball, shoot2goal, dribble, pass, cover... are defined. For all robots and each of those actions a_i success rates $P(a_i)$ and gains $G(a_i)$ are estimated (see [8] for details). This is done by manually defined feature-based functions so far. But we are working on more sophisticated parametric learning techniques. From all promising actions ($P(a_i)$ exceeds certain threshold) the one assigned to the highest $G(a_i)$ is chosen to be executed on a robot. All known global environment information is exploited for the computation of P and G.

To execute an action a planning algorithm creates a sequence of states which leads from the actual state over to the target state. Another algorithm maps the state changes to low level robot commands using neural networks. Therefore we measured state changes of robots according to different executed low level commands.

Naturally this approach depends on a stable connection between all robots. In case of an unstable interconnection the robots may also work properly but use less good environment information and therefore can only achieve a limited degree of cooperation.

References

1. ActivMedia Robotics, *http://www.activmedia.com/robots/*
2. Lanner Electronics Inc., *http://www.lannerinc.com/*
3. Videologic Inc., *http://www.videologic.com/ProductInfo/capt_pci.htm*
4. RadioLAN Inc., *http://www.radiolan.com*
5. Delta Network Software GmbH, *http://www.delta-net.de*
6. MVTec Software GmbH, *http://www.mvtec.com*
7. Brooktree: BT848 Single-Chip Video Capture Processor, *http://www.brooktree.com/intercast/bt848.html*
8. Buck S., Riedmiller M., *Learning Situation Dependent Success Rates of Actions in A RoboCup Scenario*, submitted to PRICAI2000.
9. Eckstein W., Steger C., *Architecture for Computer Vision Application Development within the HORUS System*, Electronic Imaging: 6(2), pp. 244–261, April 1997.
10. Hanek R., Schmitt T., *Vision-Based Localization and Data Fusion in a System of Cooperating Mobile Robots*, submitted to PRICAI2000.
11. Klupsch, M., *Object-Oriented Representation of Time-Varying Data Sequences in Multiagent Systems*, 4th International Conference on Information Systems and Synthesis – ISAS '98, pp. 33–40, 1998.