Agilo RoboCuppers: RoboCup Team Description

Thorsten Bandlow, Robert Hanek, Michael Klupsch, Thorsten Schmitt

Forschungsgruppe Bildverstehen (FG BV) – Informatik IX
Technische Universität München, Germany
{bandlow,hanek,klupsch,schmittt}@in.tum.de
http://www9.in.tum.de/research/mobile_robots/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 five 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 Robot League of the Third International Workshop on RoboCup in Stockholm 1999. 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 onboard scene interpretation. In addition, these features as well as the odometric data of the robots are sent over the net to the master PC, where they are verified with regard to consistency and plausibility and fusioned to one global view of the scene. The results are distributed to all robots supporting their local planning modules. This data is also used by the global planning module coordinating the team's behaviour.

1 Introduction

The aim of our activities on robot soccer 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. This can be used for teaching students in vision, artificial intelligence, 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 five Pioneer 1 robots [1] each equipped with a single board computer. They are supported by a master PC or coach, and one monitor PC for displaying the robot's data and states. Since the team size was reduced to four robots, the fifth robot is used as a substitute. The single board computers are mounted on the top of the robots, firmly fixed –

[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.

M. Veloso, E. Pagello, and H. Kitano (Eds.): RoboCup-99, LNAI 1856, pp. 691–694, 2000.

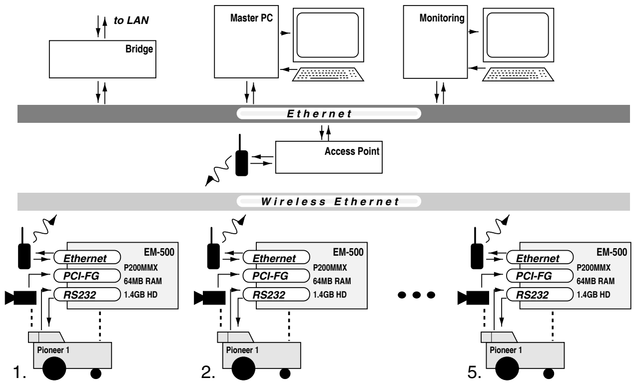

Fig. 1. Hardware architecture.

mechanically and electrically. 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' planning states and feature extraction processes. The operating system for all computers is Linux. Figure 1 gives an overview of the hardware architecture.

Figure 2 (a) shows one of our Pioneer 1 robots. Each of them measures 45 cm × 36 cm × 56 cm in length, width, and height and weighs about 12 kg. Inside the robot a Motorola microprocessor is in charge for 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.

3 Fundamental Software Concepts

The software architecture of our system is based on several independent modules, each these performs a specific task. 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. of our system. 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, local planning, information fusion, and global planning. The latter two run 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, one for monitoring the robots, extracted sensor data and

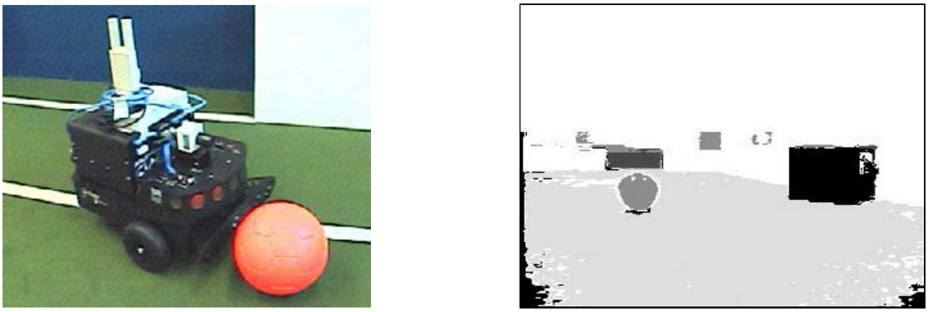

Fig. 2. (a) Odilo – one of our Pioneer 1 robots – and (b) what he percepts of the world around him.

planning decisions, one for interacting with the system or with particular robots, and one for supervising the running processes. A large number of basic functions define fundamental robot behaviors, provide robot data, and realize different methods for extracting particular sets of vision data. 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 defined. This offers a consistent concept for exchanging dynamic data between arbitrary components [9].

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 [8, 6]). This tool provides efficient functions for accessing, processing and analysing iconic data, including framegrabber access and data management. The framegrabber interface was extended to features for capturing gray scale images and color regions at the same time. For this we use the YUV-image data provided by the video capture card. The color regions can be achieved very fast by a two-dimensional histogram-based classifier, which describes color classes as regions in the UV-plane and uses a brightness intervall as an additional restriction. 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 7 to 10 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 appearence of relevant objects is well known. For their recognition, the strictly defined constraints of

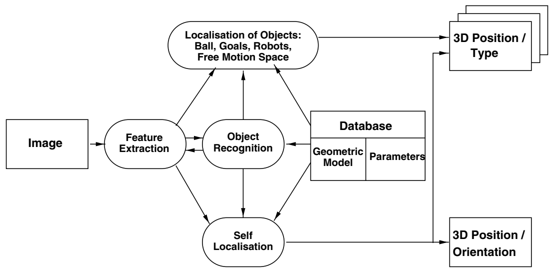

Fig. 3. Data flow diagram of the vision module.

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 (see Fig. 2 (b) and 3).

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 if needed. 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. From this we can determine their distance and position 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 subpixel accurate edge filter performed on the gray-scale image (Y-channel) 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.

Building up and maintaining a RoboCup team is a great challenge and needs huge personal efforts and a lot of time. Thus we hope that we will still have enough resources in future to continue our interesting and promising work.

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. Eckstein W., Steger C., *Architecture for Computer Vision Application Development within the HORUS System*, Electronic Imaging: 6(2), pp. 244–261, April 1997.
9. 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.