

# Agilo RoboCuppers: RoboCup Team Description

Michael Klupsch, Maximilian Lückenhaus, Christoph Zierl, Ivan Laptev,  
Thorsten Bandlow, Marc Grimme, Ignaz Kellerer, Fabian Schwarzer

Forschungsgruppe Bildverstehen (FG BV) – Informatik IX  
Technische Universität München, Germany  
{klupsch,lueckenh,bandlow,schwarzf,zierl}@in.tum.de  
[http://www9.in.tum.de/research/mobile\\_robots/robocup/](http://www9.in.tum.de/research/mobile_robots/robocup/)

**Abstract.** This paper describes the *Agilo RoboCuppers*<sup>1</sup> – 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 a CCD camera and single board computer each and coordinated by a master PC outside the field we participated in the medium size RoboCup league in Paris 1998. 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 necessary data for the onboard scene interpretation. These features as well as the odometric data are checked on the master PC with regard to consistency and plausibility. The results are distributed to all robots as base for their local planning modules and also used by a coordinating global planning module.

## 1 Introduction

Our research group started working on robot soccer at the beginning of 1998 considering it a very challenging and interesting research domain for several reasons. The main challenge is to combine several complex computer domains, like vision, robotics, and artificial intelligence to one real system of several autonomous hard- and software components which perform together one common task. For this, multiple agents need to collaborate. They should be able to organize themselves, to learn how to act in specific situations, and to handle with uncertain data. The basic conditions are quite harsh: a dynamically changing environment is to be observed in real time and fast moving objects like ball and opponent must be recognized, tracked, and considered within planning methods for controlling the movement of the own robots.

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

---

<sup>1</sup> 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.

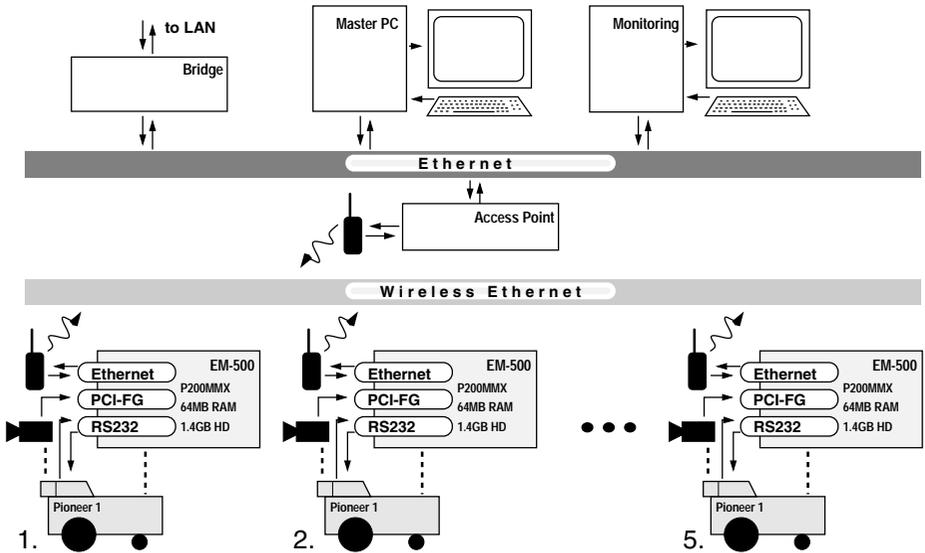


Fig. 1. Hardware architecture.

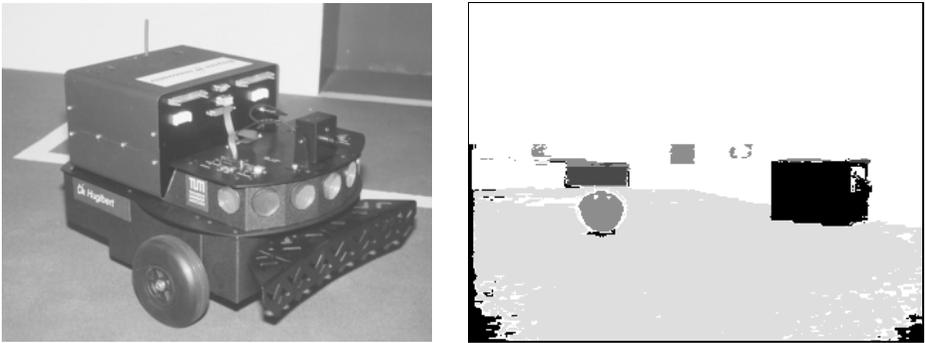
development criterion is to use inexpensive, easy extendible standard components and a standard software environment.

This paper is organized as follows. Section 2 gives an overview about the employed hardware, namely robots and computers. It follows in section 3 the description of some fundamental concepts concerning the overall system structure. Details on the design of the most important components are given in section 4.

## 2 Hardware Architecture

Our RoboCup team consists mainly of five Pioneer 1 robots, five single board computers and one master PC. The single board computers are mounted on the topside of the robots, firmly fixed – mechanically and electrically. All robot computers are linked together via a radio ethernet network[4]. The master computer is linked to the radio ethernet, too, and is located outside the soccer field. For debugging during software development and monitoring the robots' behaviors and features extracted from the sensor data, we use an additional monitoring computer. The operating system for all computers is Linux. Figure 1 gives an overview about the hardware architecture.

Figure 2 (a) shows one of our Pioneer 1 robots [1]. Each of them measures 45 cm  $\times$  36 cm  $\times$  30 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



**Fig. 2.** (a) Hugibert – one of our Pioneer 1 robots – and (b) a example view for the world according to him.

equipped with a P 200 processor, 64 MB RAM, 2.5” hard disk, onboard ethernet and VGA controller, and an inexpensive PCI video capture card [3]. PC and robot are connected via a standard RS232 serial port. A color CCD camera is mounted on top of the robot console. For better ball guidance we mounted a simple concave-shaped bar in front of each robot.

### 3 Fundamental Software Concepts

The software architecture of our system is based on several independent modules which perform each 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. In Figure 3 an overview about the software architecture of our system is shown.

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 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.

As 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

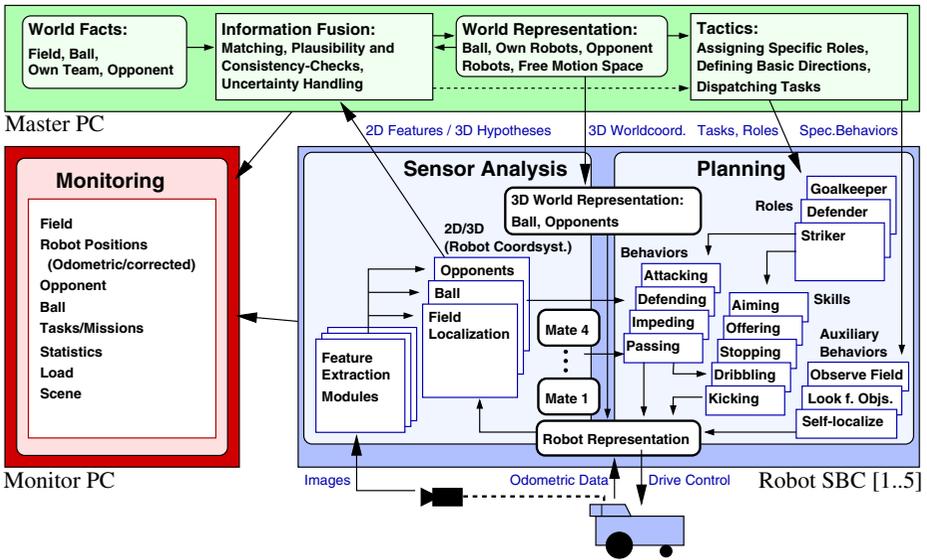


Fig. 3. Software architecture.

offers a consistent concept for exchanging data between arbitrary components [6]. The most important features of these objects are:

- A list of previous sequence values is held allowing to access values from the past. Values too old are “forgotten”.
- A uniform time stamp is attached to all sequence values.
- The sequence objects “know” their functions for updating the sequence.
- Side effect functions can be defined which are performed after updating a sequence, and others, if the update failed.
- An update is automatically triggered when querying for a new sequence value.
- Sequence values can be interpolated or predicted in the case that a value is requested for a time between two available values or for a future time, respectively.
- Sequence objects are global and thread-safe, i.e. one module can be in charge for setting the values appropriately and other modules can use their values at the same time.
- Sequence data can easily be made transparently global over a network. So all robots as well as the master PC can access the data of the other robots.
- They can be (re)configured online dynamically during the running process.

The agents are responsible for triggering the sequences, which are needed in the current situation and to configure them according to the actual robot task.

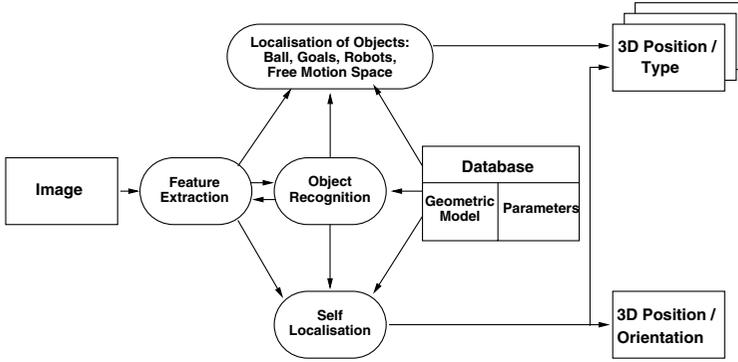


Fig. 4. Data flow diagram of the vision module.

## 4 Components

### 4.1 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 analyzing tool HALCON (formerly known as HORUS [5]). The extracted data is provided by sequence objects as described in section 3.

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 (see Fig. 2 (b) and 4).

Besides recognizing relevant objects, 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 their estimated size to determine their 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 the visible goals.

### 4.2 Information Fusion

The information fusion module has to combine the fragments of information, which are provided by our robots, and form a consistent view of the world. This component is supposed to have two representations of the world: The geometrical one consists of the position, orientation, and velocity vector of all robots and the ball. Another representation is based on a grid covering the soccerfield and its contents. This will mainly be used for consistency checks and for global planning.

Information concerning uncertainties of the robot data is provided by the robots in conjunction with the odometric data. It is represented by ellipsoids where the volume of the ellipsoid corresponds with its uncertainty.

### 4.3 Planning

Planning is done on two levels: A *central, global* planner coordinates the different team members and a *local* planner, that resides on each robot, controls the behavior of a single robot. Each robot has its own view of the current situation and incorporates global data from the information fusion module to choose an appropriate action (behavior or skill). This is done by evaluating the actual values of the sequence objects. The decision is also influenced by the current *role* of the robot that may change during the game (except for the goalkeeper).

The task of the *global planner* is to coordinate the robots. E.g., if several robots try to reach the ball, the global planner chooses the most promising of them as forward and asks the others to change their role so that they do not impede the selected forward. Decisions of the global planner are driven by the world view provided by the information fusion module. Therefore, the global planner depends on a stable connection to all robots. In case of an instable or disturbed interconnection the robots may also work properly controlled only by their local planner.

## 5 Experiences during RoboCup'98 in Paris

As a result of our participation, we summarize the following:

- We have made good experiences with concentrating on our main scientific goals – robot vision and intelligent control of distributed systems – by combining robust standard components for the robot hardware. However, one should not underestimate the effort to establish and maintain a running robot system.
- A good monitoring system is inevitable for debugging and analyzing within a distributed robot system. This is particularly true with a system sharing global data for coordination.
- The lack of a kicking device is a main disadvantage in terms of competition.
- The different interpretation of the rules (e.g. size restrictions, penalty shoot-out) by teams and referees made it difficult to enable a fair competition.

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 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](http://www.videologic.com/ProductInfo/capt_pci.htm)
4. AAeon Technology Inc. <http://www.aeon.com.tw/html/prod.htm>
5. Eckstein W., Steger C., *Architecture for Computer Vision Application Development within the HORUS System*, Electronic Imaging: 6(2), pp. 244–261, April 1997.
6. 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.