Agilo RoboCuppers: RoboCup Team Description

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

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/

Abstract. This paper describes the *Agilo RoboCuppers* [1] – the RoboCup team of the image understanding group (FG BV) at the Technische Universität München [2]. With a team of five Pioneer 1 robots, equipped with a CCD camera and single board computer each and coached by a master PC outside the field we intend to take part in the medium size RoboCup league in Paris 1998. The system is controlled by several agents, each of them in charge for a specific task on a particular robot. For the scene recognition we use the image processing library HALCON, which is used for onboard feature extraction. These features as well as the odometric data are checked on the master PC with regard to consistency and plausibility, and the results are distributed to all robots.

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

[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] The FG BV research group is part of the Chair Image Understanding and Knowledge-Based Systems (Prof. Bernd Radig) at the faculty of computer science, Technische Universität München.

least, in developing large dynamic software systems. For this reason, our basic development criterion is to use unexpensive, 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. 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. It is also placed outside the field and linked to the ethernet. Figure 1 gives an overview about the hardware architecture.

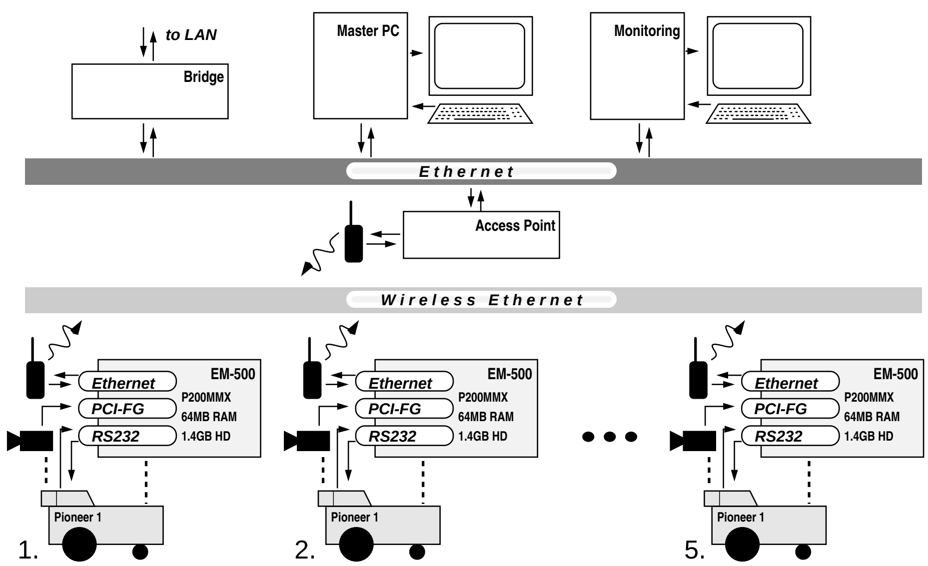

Fig. 1. Hardware Architecture

Figure 2 shows one of our robots. The Pioneer 1 robots are bought from ActivMedia Robotics. They measure 45 cm × 36 cm × 30 (22) cm (18" × 14" × 12" (9")) in length, width, and height (with/without computer), respectively. The weight of each (with our assembly and a larger battery – a 12 VDC, 12Ah sealed lead cell) is about 12 kg (25 lbs.). They are mainly black.

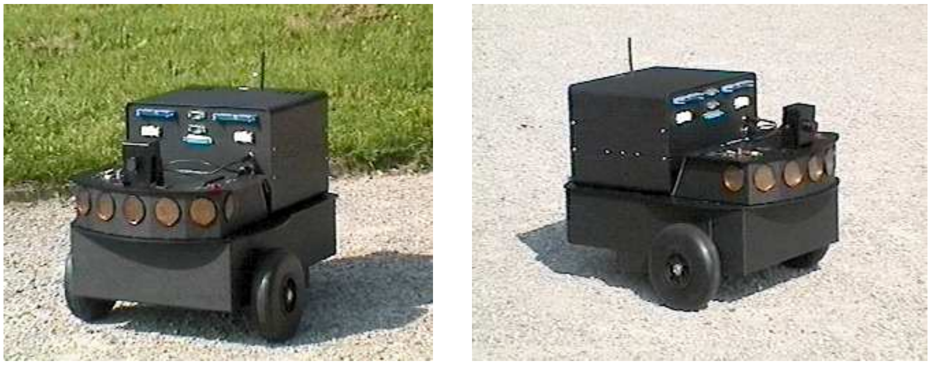

Fig. 2. Tassilo – one of our Pioneer 1 Soccer Players

Inside the robot a Motorola microprocessor is in charge for controlling the drive motors, reading the position encoders, for the seven ultrasonic sonars, which are attached to the console, and for communicating with the client. This is in our case a single board computer which is mounted within a box on the topside of the robot. The card is a small EM-500 all-in-one-pc from Lanner Electronics Inc., measuring 14.5 cm $\times$ 20 cm (5.75" $\times$ 8"). It is equipped with a 200 MHz MMX Pentium processor, 64 MByte RAM, 1.4 GByte 2.5" hard disk, onboard ethernet and VGA controller, and an inexpensive VideoLogic PCI video capture card (framegrabber card). PC and robot are connected via a standard RS232 serial port. A color CCD-camera is mounted on top of the robot console and connected to the video capture card.

All robots, the master, and the monitor PC are linked via ethernet which is constituted by a transparent WaveCell wireless LAN. The operating system for all computers is Linux, what offers stability combined with a large amount of flexibility and a familiar programming environment.

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 modules define fundamental robot behaviors, provide robot data, and realize different methods for extracting particular sets of vision data.

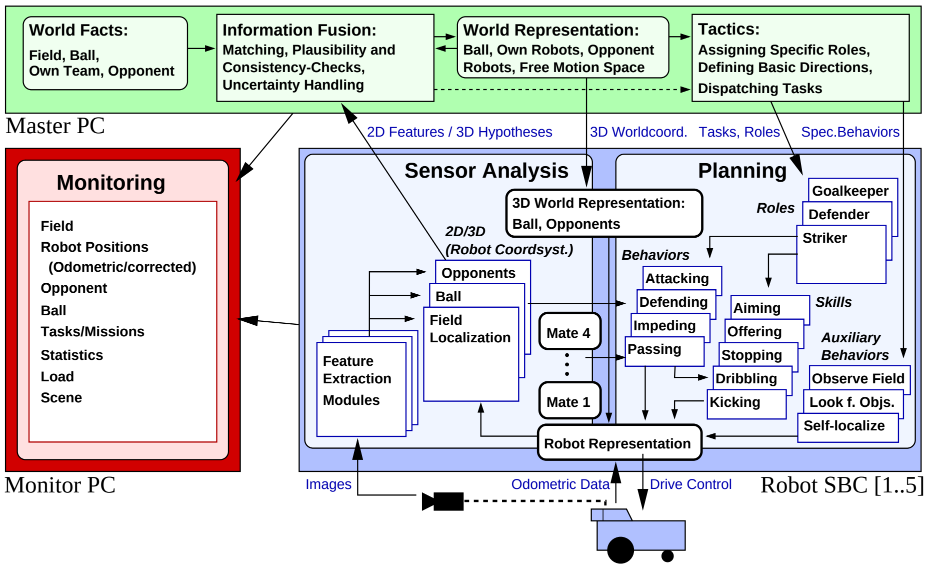

Fig. 3. Software Architecture

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 offers a consistent concept for exchanging data between arbitrary components [2]. The most important features of these objects are:

- They hold a list of arbitrary sequence values ordered according to their history. This allows to access values from the past. Values to old are "forgotten" from the sequence.
- All sequence values have a time stamped attached.
- The sequence objects "know" their functions for updating the sequence.
- It is possible to define side effect functions which are performed, if a sequence was updated, and others if the update failed.
- When asking the sequence for a new value, an update is automatically triggered.

- The sequence can be equipped with functions for interpolating or predicting values, which are called if a value is requested for a time between two available values or for a future time, respectively.
- These sequence objects are global and thread-save, so 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 during the running process, even by the process itself, if it is needed.

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

4 Components

4.1 Vision

The vision module is a key part of the whole system. It is in charge of collecting and interpreting information about the surrounding world. Given a raw video stream, the module has to recognize relevant objects in it and provide their positions on the field to other modules. This is done with the help of the complex image analyzing tool HALCON (formerly known as HORUS [1]). The extracted data is provided by sequence objects as described in section 3.

In general, the task of image interpretation and object recognition is a very difficult one. However, its complexity strongly depends on the context of a scene which has to be interpreted. If the appearance of objects has invariant salient properties such as color, geometry or scale, the recognition of such objects becomes significantly easier.

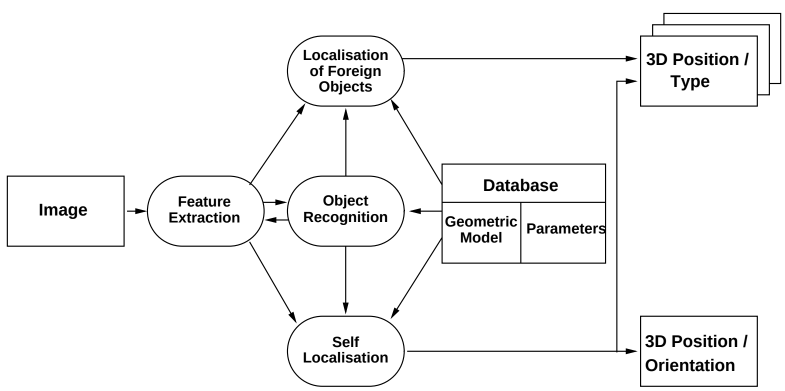

Fig. 4. Data Flow Diagram of the Vision Module

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 color and shape are saved in the modell database and can be used. These constraints are matched with the extracted image features such as color regions and line segments (see Fig. 4).

Beside 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. For the self-localization we use a 3D geometric model of a field. A predicted view of the virtual camera is matched with the extracted line segments. To localize foreign objects, we use their estimated size to determine their position relative to the robot.

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 first one is the geometrical one. It is build from the following elements: the five robots of our team, the (probably also five) robots of the opponent's team, and the ball. Each of the robots is characterized by its position in x/y coordinates, the orientation, and an velocity vector. The ball has only two coordinates and the speed vector. Another representation of the world 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. The planning module shall be able to influence the fusion component telling it which information is most important within the current situation. This will change the priority for calculating special features.

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. The core of the local planner consists of three concepts, namely *role selection*, *abstract role*, and *intelligent pilot* (see Fig. 5). The *role selection* and the *intelligent pilot* are implemented as classes. Each robot has its own view of the current situation and conjectured changes and may choose an appropriate behavior (role). The proper choice of the current role, which may change in the course of the game, is performed by the *role selection*. *Abstract role* is a uniform interface for the role selection and a number of different concrete roles (behavior patterns) can be derived from it: striker, forward, defender, obstacle (impeding the opponents), goalkeeper, observer (providing information for the global information fusion). The *role selection* will try to select the best role according to the current situation.

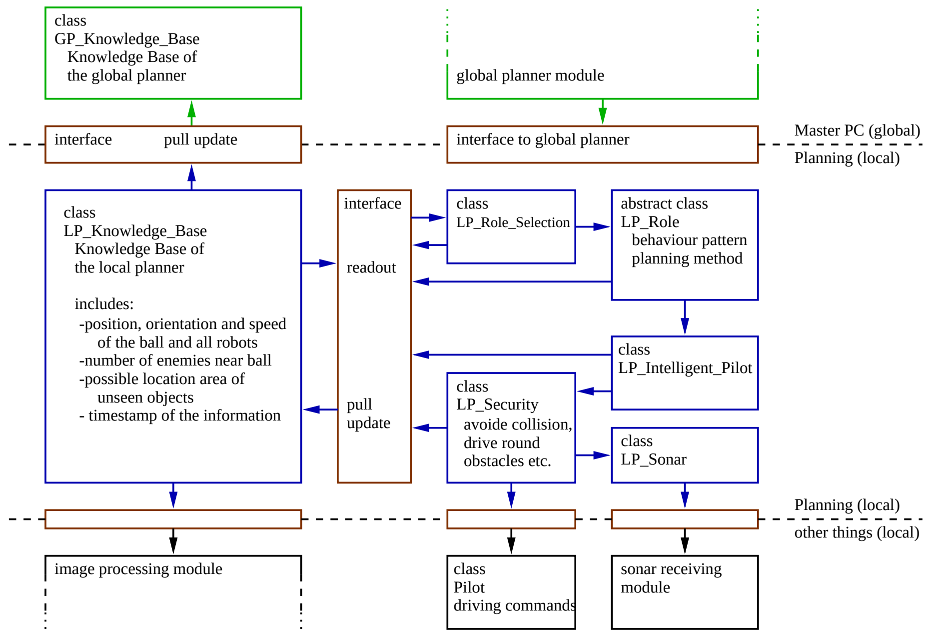

Fig. 5. Local Planning

Each role can use *skills* of the *intelligent pilot* such as *approach the ball* or *move to position x/y*. Thus the *intelligent pilot* provides sophisticated robot control and incorporates tasks that are common to all roles, e.g., obstacle avoidance and circumvention. It uses the commands of the security class. This class ensures that the robot does not collide with other robots or the wall. At this, it makes use of the built-in sonar sensors of the robot.

The main 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 the main forward and asks the others to change their role (e.g., to act as *obstacle*) 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. To avoid helpless robots in case of an instable or disturbed interconnection the robots may also work properly controlled only by their local planner.

References

1. Eckstein W., Steger C., *Architecture for Computer Vision Application Development within the HORUS System*, Journal of Electronic Imaging, April 1997.
2. Klupsch, M., *Object-Oriented Representation of Time-Varying Data Sequences in Multiagent Systems*, Accepted for publication on International Conference on Information Systems and Synthesis – ISAS '98.