

The Kyushu United Team 2003 in the Four Legged Robot League

Hiroki Najima, Yoshihiro Yasutake, Takeshi Kato, Takayuki Kadowaki,
Kazuteru Matsumoto, Kentaro Oda, Toshiyuki Ishimura, Takeshi Ohashi

Dept. of Artificial Intelligence, Kyushu Institute of Technology
najima@mickey.ai.kyutech.ac.jp

Abstract. This paper presents our approach to the Sony Four Legged Robot League of the RoboCup 2003. Our system consists of inter-robot communication, meta planning, base-level planning, vision, localization, behavior and walking modules. We introduce several techniques for these modules: the using multiple color tables and the object detection with scanline bring about robust vision system. The inter-robot communication allows us to build highly collaboration among the robots. The simulator allows us to test a robot's program effectively in the ideal environment where specified condition can easily be reproduced.

1 Team Development

1.1 Team members

[Kyushu Institute of Technology (KIT)]

Professors

Dr. Takeshi Ohashi, Dr. Takaichi Yoshida

Graduate Students

Yoshihiro Yasutake, Kentaro Oda, Toshiyuki Ishimura, Yoichiro Furutani,
Takeshi Kato, Atsushi Hashimoto, Kazuteru Matsumoto, Takayuki Kadowaki,
Kana Imamura, Masayuki Nitta, Yasuyuki Ogihara

Undergraduate Students

Hiroki Najima, Keiichi Kii, Hideo Tateiba, Go Kobayashi

[Fukuoka Institute of Technology (FIT)]

Professors

Dr. Takushi Tanaka, Dr. Masaru Ishii, Dr. Hiroaki Shirakawa

Undergraduate Students

Shou Kawakami, Takahiro Kudo, Tadayasu Oda, Yoshitake Furuie,
Yousuke Someki, Takashi Ono, Tomoyuki Fukunaga, Kenji Sasaki,
Hideo Shinoda, Kouji Nakano, Michihiro Sueda, Hidetaka Harada,
Yuusuke Hayashi

1.2 Team introduction

We, the highly involved members of the united team, are from Kyushu Institute of Technology (KIT) and Fukuoka Institute of Technology (FIT). The KIT team consists

of experts in distributed systems and object-oriented database systems field who involve applying their techniques for robotics field.

For more information, please visit our web page:

<http://www.asura.ac/>

The year 2001 four legged league of the RoboCup was our first participation, and our aim was to reach the world competitive level. As the year 2000 winner, the UNSW team had successful result due to their technical advances in locomotion, vision, localization and repertoire of behavior [1], our basic system was based on their successful sophisticated system. Since it was not sufficient, we introduced some techniques and tools. The last year we became the best 8 in the RoboCup 2002, and this year 2003 in RoboCup Japan open held in Niigata, we successfully got the first position.

The strategy of the year 2003 was to build highly collaboration among the robots, and fast development and tune-ups of strategies and behavior. So we introduced some new techniques and tools.

2 Vision & Localization

In the color segmentation, we adopt both TSL and YUV color spaces and provide a color table management framework that enables to select color tables, created from the each color space, statically and dynamically. In addition, we adopt the scanline that traces the pixels to support the recognition of objects.

2.1 Using Multiple Color Tables

The basic colors, appeared in the four-legged league field of RoboCup 2003, are orange for the ball, blue, pink and green for the markers, yellow and blue for the goals and dark blue and dark red for the robots. In the real field, the orange ball in the yellow goal and the ball under the head (very close to the body) are difficult to recognize by using the same color segmentation as the ideal one. In addition to these colors, the black and white colors have to be included to recognize the soccer ball used in the RoboCup Challenge 2003. Because black and white colors are the achromatic, and it is difficult to apply the same color space to classify achromatic and chromatic colors.

Therefore we consider that it is useful to prepare multiple color tables based on different color spaces and select them depending on the situation. It is also worth preparing the tables that focus on the specific colors. To realize these, we provide the color table managing framework that enable to select tables both statically and dynamically. Using a suitable color table makes it simple to segment the image and results in the improvement of the color segmentation.

We adopt both TSL and YUV color spaces. In TSL color space, we can classify a given color accuracy with determining 6 parameters: minimum and maximum thresholds of T, S and L components (linear separation). Through minimizing the number of parameters, we can adapt the parameters without complex tasks such as optimizing functions and learning algorithms. This advantage gives us the faster adaptation and manually tune-ups of the color table to a new lighting environment. In YUV color

space, we adopt the method presented by CMPack-02 [3]. This method uses threshold learning and suits for the images including colors that cannot handle with simple rectangle distribution.

For dynamic color table switching, we use the framework for managing the color tables enables to select and switch to an arbitrary color table at a time. For localization, the task that the robot turns its head to localize its position can select the color table that better classify the markers and goals, and the task that search the soccer ball also can select the purpose-built color table that detects the black and white color. Reversely, it is possible to identify the current color table. Then the strategies and object recognition algorithm can utilize the information which color tables are used in the current color segmentation.

2.2 Object Detection with Scanline

The fundamental algorithm of the object recognition adopts the color segmentation and it is effective in the environment that is about equal to the ideal environment. However, there are some cases that a color is difficult to use, and the same color is used on the different objects. In these cases, there is worth providing additional features for the object recognition, such as the color pattern appeared in a specific object. For example, we present the black and white soccer ball detection and goal detection.

Soccer ball detection. The soccer ball used in the RoboCup Challenge 2003 has the white and black colors that look like a regular soccer ball. White is used in the walls and lines in the field and the black is often appeared in the shades of the robot and the edge between the wall and floor. Then it is possible to create many blobs against a single color from an image and analyze their size, position and so on. As a support for the correct recognition, color pattern and color histogram are adopted. For example, the white-black-white pattern may show the soccer ball and the white-green-white pattern may show lines or the center circle. As a similar way, the proportion of the white, black and other color on and around the soccer ball can be estimated.

Goal detection and localization. The localization performed on the basis of a single goal requires distance and heading to the goal. Then we use the goal and goal line (a field edge on the goal side drew with a dotted line in Fig.1). The goal is detected by the color segmentation and recognized as a bounding box. The distance to the goal is expected by the goal height. Accutually we use the distance measured from the goal post (side edge of the goal) height. The bottom point of goal post also can be used, but we select the goal height because it is accurate enough to estimate the distance.

The heading (showed in Fig.2) is calculated from the goal line information. The goal line exists on boundary line between white wall and green field is detected using scanline. The scanline is used to detect this bound. To avoid a fail to detect the goal line, the scanline is made from bottom up on the image. And to skip the goal area line, the color pattern is detected. We use the color table that is specialized for this method. For example, field's green should be flat, the noise on line is minimized and a clean edge between white and green is better. Through this process, we calculate the distance between two points (a-r, b-r and a-b). Using these three edges, we can estimate the heading (showed in Fig.2).

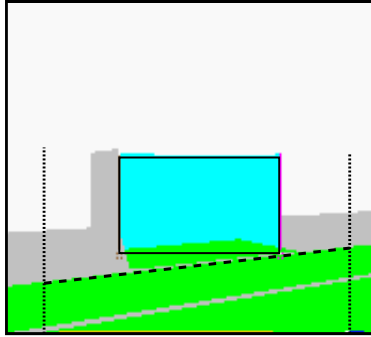


Fig. 1. Detect the goal and the goal line

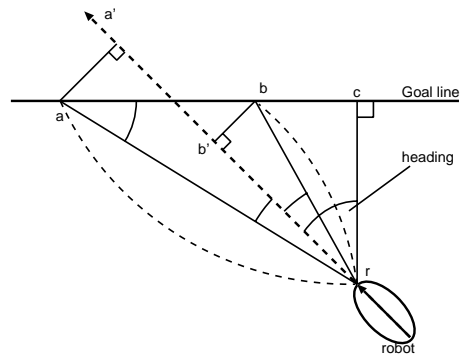


Fig. 2. Detect the goal and the goal line

3 Inter-Robots Communication

Inter-robots communication via Wireless LAN is important element to coordinate the robots as a team. The robots are able to perform sophisticated strategy with the information obtained from the inter-robots communication. The robots send the information such as the position of ball and an internal status to each other. The status contains a global position in the field, status of penalties and a current role. The robots use the information to determine their strategy. As a matter of fact, we use the shared information for global position coordination, dynamic role change according to some penalties, obstacle avoidance and a global ball discovery. However, the information obtained from inter-robots communication is not so reliable, because it contains accumulated errors originated from mixing information and it is past information due to network delay. Then the information obtained from inter-robots communication is useful in the case that the robots lack the information gathered by their camera and sensor.

3.1 Sending Information

Each robot sends information as a single message at the end of overall processing. Messages are broadcasted to all robots in a team via TCPGateway. The information sent by our robots is as follows:

- Position of the ball
- Position of the robot
- Status of penalties
- Role of the robot (Attacker, Defender, etc...)

When a robot receives message from TCPGateway, it passed to MessageManager object. MessageManager object decomposes this message into each information, and then the object sends it to the objects that process each information.

3.2 Ball Discovery

Discovering a ball is a most important task for soccer robots. Discovering is difficult if the ball exists far from the robot, or other robots hide the ball. If a ball is in the field of view of a robot, teammates are able to use the ball information detected by the robot and recognize the ball position.

3.3 Positioning

In the four-legged league, approaching the ball is important task for robots. However, multiple robots approaching causes collisions with each robot. To avoid collisions with teammate robots, the cost evaluation based on the ball position is important.

If a ball is in the field of view of more than one teammate robots, other robots evaluate costs for approaching the ball. For example, when robot A and robot B find the ball, each robot evaluates costs using the information obtained from inter-robot communication and then decides whether to approach the ball or not. If the cost of

robot B is relatively inexpensive, robot B tries to approach the ball and robot A moves to a position where the ball might come to.

Each robot evaluates approaching cost using the information such as the distance and direction from each robot to the ball. The information obtained from inter-robots communication includes network delay. Therefore, robots have to consider network delay to evaluate accurate costs. The influence of network delay is modeled as a threshold. For example, when robot A and robot B find the ball and the cost of robot A is lower than robot B and the difference between their cost is greater than a threshold, then robot A decides its own cost is lower than others, and tries to approach the ball.

3.4 Dynamic Role Change

In our system, each robot have a role. The purpose of which robots have a role is giving robots personality and making them do a different behavior. But it is not an optimal choice to fix his role, so we constructed the dynamic role changing system.

Our robots' roles classified into four-types as follows:

- Striker
- Libero
- Defender
- Goalie

Striker, Libero and Defender are field players. Libero is a field player that moves around the all area in the field. Striker is a field player that moves around half area in opponent side, and Defender moves around moves around half area in team side. With the separation of work area, if the ball goes to opposite side, at least one robot exists in the side where the ball exists. The robots decides own role by some information such as penalty and position of the team robots, position of the ball, point score, and remaining time.

Penalties decrease the number of teammate in the field. Whenever it occurred same, the remaining members should reorganize each role, and hold out against opponents' attack. For that, our robots send information about penalties. After returning penalized robots, the other robots go back to own role.

For example, opponents and a ball close by team goal, and our Libero and Defender pay penalty, so only Striker is in the field. Without this system, Striker can't defend goal, because of the team side isn't included striker's work area, so Striker only can see opponents' attack on the half line. By this system, this undesirable situations are avoidable.

4 Simulator

Development robot's programs are difficult because of real world uncertainty. Considering environment changes, for example, the changes of the lighting condition affect robot's behavior seriously in general. To provide the ideal environment that can be reproduced, to get rid of any limitations of resources, and to minimize the overheads, we develop a simulator [2]. The purpose of our simulator is to improve the efficiency of the development of vision-based robots' strategies.

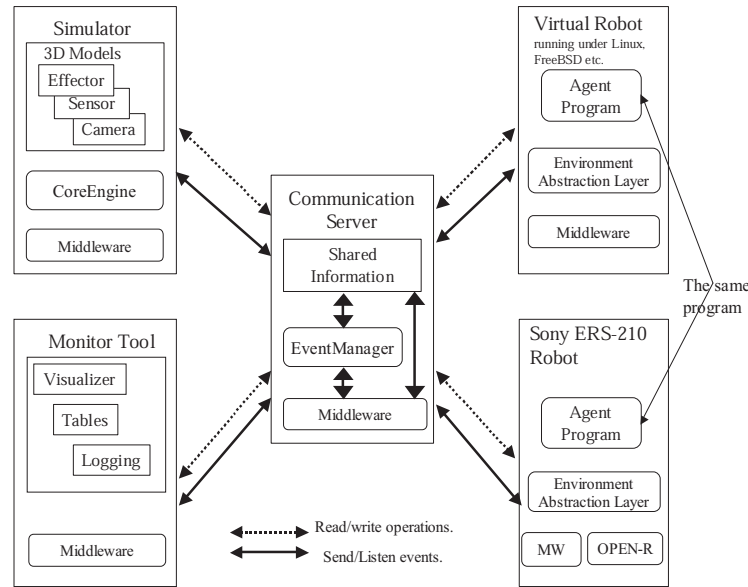


Fig. 3. The Architecture Overview

4.1 Architecture

The main idea to unify the both real and virtual environment is the introduction of an environment abstraction layer, a common communication protocol, communication hub and a vision-based simulator. We also developed the monitoring tool to visualize local information acquired from virtual and real robots. Fig. 3 shows an example of configuration both virtual and real robots reside on at the same time.

Simulator manages all objects in the virtual environment and synthesizes virtual camera images and simulates the robots' effector. The camera holds parameters such as view angle, resolution and so on, synthesized images by the Core Engine. The effector keeps current value and the range of value, and also can hold any sub-effector and camera as children.

Environment Abstraction Layer hides details on a underlying layer and gives an illusion of a modeled environment.

Communication Hub and *Middleware* manages communication between the *Simulator* and clients and can capture messages through it. Multiple heterogeneous robot agents implemented in various program languages can participate to this environment because each client lies on the common communication middleware and connects the *Simulator* via the TCP/IP network. If the simulation environment communicates with real robots, it can accommodate both real and simulated robots in the same simulation environment.

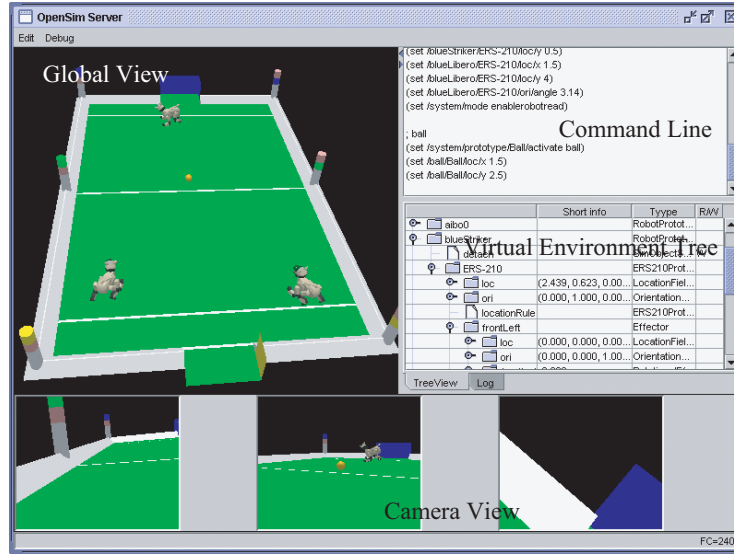


Fig. 4. The Simulator Server

4.2 Implementation

To implement the communication hub, simulator and monitoring tool, we use Java, Java Advanced Imaging API and Java 3D API. For generating/parsing XML file in the communication hub, we use the SAX parser. For an agent programming we use C++ and partially scheme, a lisp dialect. The agent program can be run both Sony ERS-210 platform and Linux and FreeBSD with gcc.

The *Simulator* shown Figure 4 consists of four components; global view, command line panel, the tree view, camera views. On the global view, user can change own view by mouse operations. The command line panel allows us to interact with the simulator by the script language described in the above. The tree view shows information about all objects in the simulation environment. We also implemented a library for communication between the server and the client, which supports C++ and Java.

To evaluate our method, we implemented a simulator for the Robocup Sony 4-Legged League on this environment. Then we succeed in migration from real robot's agent program to the simulator with a little bit modification. By using robot communication functionality of the *Simulator*, this simulator can simulate communication between robots instead of wireless LAN.

5 Conclusion

The using multiple color tables allows us to detect objects on purpose. The object detection with scanline provides us complex objects detection as a regular soccer ball, and improves accuracy of the vision based localization. The techniques of the inter-robots

communication provide advanced and efficient strategy. The dynamic role change allows us to arrange the robots effectively and alleviate the influence of penalties as a team. The simulator provides us to enormous and ideal resource such as robots and field, and it contributes to improve the efficiency of the development.

A few our source code originally developed by the UNSW team, and we would like to express our sincere thanks to the UNSW team efforts.

References

1. Bernhard Hengst, Darren Ibbotson, Son Bao Pham, John Dalglish, Mike Lawther, Phil Preston and Claude Sammut, The UNSW RoboCup 2000 Sony Legged League Team, in RoboCup 2000: Robot Soccer World Cup IV, Springer-Verlag, pp.64-75, 2000
2. Toshiyuki Ishimura, Takeshi Kato, Kentaro Oda and Takeshi Ohashi, An Open Robot Simulation Environment, RoboCup International Symposium 2003
3. Manuela Veloso, Scott Lenser, Douglas Vail, Maayan Roth, Ashley Stroupe and Sonia Chernova, 2002 Technical Report, http://www.openr.org/robocup/code2002SDK/CMU/cmu_teamdesc.pdf