# Owaribito – A Team Description

Shinya Hibino[1], Yukiharu Kodama[1], Yasunori Nagasaka[2], Tomoichi Takahashi[2], Kazuhito Murakami[1], and Tadashi Naruse[1]

[1] Aichi Prefectural University, Nagakute-cho, Aichi, 480-1198 JAPAN
[2] Chubu University, Kasugai, Aichi, 487-8501 JAPAN

**Abstract.** In this paper, we describe image processing method for high-speed robot ID recognition with black and white markers, path generation algorithm which controls left and right wheel velocity independently, strategy and communication software with error correcting code (ECC).

## 1 Introduction

An autonomous robot is a product of the system engineering that is built on the accumulation of the wide range of technologies such as mechanical engineering, electronic engineering, information engineering and so on. One of the main issues of the robot system design is the balanced design of the robot components. We explore the balanced design as well as the study of component technology.

We have been developing the Owaribito system since 1998. The hardware of the robot [1] is not changed since then, while the software is newly developed. In this paper, therefore, we describe the main features of the Owaribito's software, i.e. image processing, path generation, strategy and communication software.

## 2 Image Processing

The size of the image given by the global camera is $640 \times 480$. From the image, we extract object regions such as a ball and robots by using color information. We used the method developed by CMU [2] for the extraction .

Top surface of each robot of us has a black and white plate shown in Figure 1. By discriminating the shape of top surface, the robot ID and the forward direction can be simultaneously detected. That is, we prepare, as a template, the table of positions and directions of $n$ equally divided points on the circle with radius $r$. The $r$ and $n$ are determined to satisfy that each point corresponds to a pixel. In case of the Owaribito, $n$ is 44. Applying the template to each robot region on the image, we find a point whose pixel value changes from black to white. Consulting with the table, the forward direction of the robot can be found. Moreover, finding the point changing from white to black gives an angle of the arc of white region. It determines the robot ID. In case of $n = 44$, the resolution of angle is about 8 degree. This method is so simple that it can detect IDs and directions in high-speed. In the real system, the image noise cannot be avoidable so that we also use color tags to stably detect IDs and directions.

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 579–582, 2002.

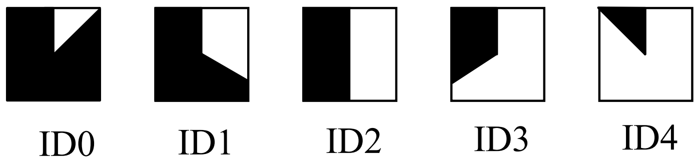


**Fig. 1.** Top surface of robot

## 3 Path Generation

Since our robot has two wheels, the degree of control freedom is two and the control of direction is limited. Considering this and the fact that the goal position where each robot should go changes from hour to hour, it is realistic the path generation which the robot approaches the goal position by asymptotically changing his direction. We call this a **direction converging path generation**. Figure 2 shows the paths generated by this method. In the figure, paths are superimposed on the field image. Each double circle with a number is a current position of our robot and the end point of each path is a goal position. The goal position with triple circle is a ball. The dotted circles are opponent robots. If opponent robots stand on the line connecting the current position of our robot and his goal position, subgoals are put near the opponents to avoid collision. In the case, the robot goes to the subgoals at first and then goes to the goal. Robots with number 2 and 3 have a subgoal and robots with number 0 and 1 do not. The goal keeper (number 4) does not move in this figure. Note that the path is generated only to determine the velocity for next $\Delta t$ time step[1]. The path is newly generated every $\Delta t$ time step.

Our path generation algorithm is given as follows.

*step1* Let $p(=(x,y)), (v_l, v_r), \mathbf{u}$ be position, velocity (of left and right wheel) and forward direction vector of a robot, respectively. Let $t$ be the current time. Calculate the curvature $\kappa$ and robot velocity $v = \frac{v_l+v_r}{2}$. See literature [3] for detail.

*step2* Get a goal position $p' = (ox, oy)$. This is given by the strategy algorithm.

*step3* Let $\overrightarrow{pp'}$ be a vector directed from the current position to the goal position and let $\theta$ be an angle between vectors $\overrightarrow{pp'}$ and $\mathbf{u}$.

*step4* Give a curvature at the time $t + \Delta t$ by $\kappa_{new} = \theta \times n_a / R_A$, where $R_A$ is a constant and $n_a$ is a variable depending on subgoal(s). (This equation generates a path which has a large curvature at first and a small as approaching to the goal. See Fig. 2.)

*step5* Putting $dr = 1/\kappa - 1/\kappa_{new}$, calculate a velocity variation of robot $|dv| = |S/dr|$, where $S$ is a constant. Let $v_m(\kappa)$ be a maximum robot speed when a curvature $\kappa$ is given. Give new velocity by $v_{new} = v + dv$ if $v_{new} < v_m(\kappa_{new})$, otherwise by $v_{new} = v - dv$. Then, calculate a new position at the time $t + \Delta t$.

[1] Image processing speed determines the $\Delta t$. In our system, $\Delta t = 33msec$.

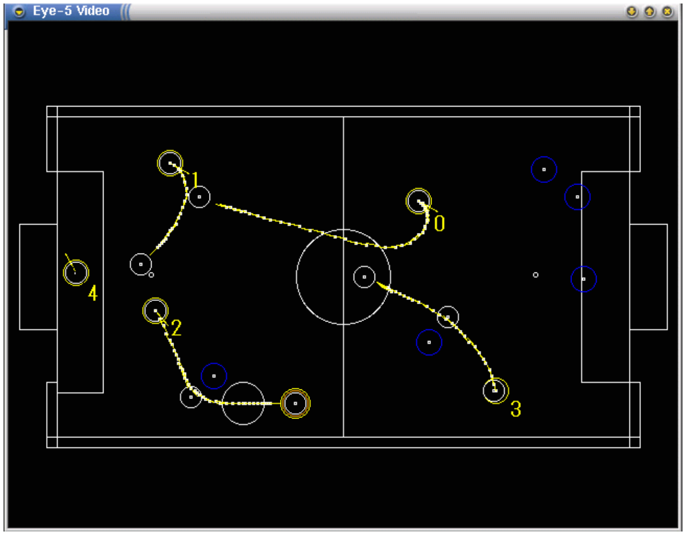


**Fig. 2.** An exapmle of path generation

*step6* Calculate repeatedly the steps from *step1* to *step5* and check whether the path reaches the given goal or not. (Fig. 2 shows the result of this calculation.) If the path reaches the goal, it is OK. If not (if over M times repeated), recalculate these steps by changing the constant $R_A$ until the path reaches the goal. This computation gives the robot velocity of next $\Delta t$ time period.

## 4 Strategy

Current Owaribito system has only a simple rule checker. The checker checks a multiple attacker, a multiple defender, a pushing, a contacting with the goalie and a 15 second stopping rule. Pushing wall or a mate is also checked. If a robot will fall in violation, an avoidance operation is applied. The aviodance operation is simple, it is a move to the direction that does not violate the rule.

## 5 Communication

We use a radio modem system of Futaba Corp. for communication between robots and a host computer. It is a 2.4 GHz band spread spectrum system. It has a direct communication mode, which realizes a low communication latency but a low communication quality. User is responsible for the communication quality.

The host processor dispatches a robot command, which gives wheel velocities of all robots, every 50 msec. The command consists of 17 bytes packet, i.e.

2 bytes for header, 10 bytes for velocities, 5 bytes for ECC, which supports communication quality. The communication latency is only 9 msec.

To achieve high communication quality, we realized a communication monitoring system. With an another receiver on the soccer field, the host processor can monitor, with high accuracy, the same command as the one which the robots received. Comparing the sent command with the monitored command, send-error can be detected. Moreover, since an accurate time that the command arrived at the robots can be calculated, more accurate path control can be possible.

In the RoboCup-01 competition, we monitored the communication throughout our matches. The result is shown in Table 1. It is clarified that the packet with ECC is sufficient for the communication in RoboCup if the competitions are managed with the system which a committee of RoboCup-01 did.

**Table 1.** Statistics of communication data

| owaribito *v.s.* | total bytes | normal bytes | corrected bytes | errored bytes |
|---|---|---|---|---|
| fu_fighter | 138960 | 138411 | 548 | 1 |
| fuomni/viper | 271950 | 270805 | 1145 | 0 |
| field ranger | 265020 | 263894 | 1125 | 1 |
| robosix | 181200 | 180504 | 696 | 0 |

## 6 Concluding Remarks

The Owaribito realized the high-speed detection of robot ID and direction, direction converging path generation, communication monitoring and so on. Implementing these functions gave a stable operation of the robot system in the RoboCup-01.

Owaribito has not a kicking device now, however, it is absolutely neccesary. We will participate in RoboCup-02 at Fukuoka with the robots having the kicking device and a omni directional drive mechanism and implementing the high level technology such as pass play operation.

## References

1. T. Naruse, T. Takahashi, K. Murakami, Y. Nagasaka, et al, " OWARI-BITO" in M. Veloso et al eds., Lecture Notes in Artificial Intelligence 1856, RoboCup-99: Robot Soccer World Cup III, pp. 675-678, Springer, 2000
2. J. Bruce, T. Balch, M. Veloso "Fast and Inexpensive Color Image Segmentation for Interactive Robots", Proc. 2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2061 - 2066, 2000
3. R. D'Andrea and J. Lee "Cornell Big Red Small-Size-League Winner", AI magazine, Vol. 21, No. 3, pp. 41 - 44, 2000