

Object Recognition and Self-Localization for Interactive Soccer Robots

Gen'ichi Yasuda¹ and Bin Ge²

¹Department of Mechanical Engineering, Nagasaki Institute of Applied Science, Nagasaki, Japan
yasuda@csce.nias.ac.jp

²Department of Informatics, Dalian University, Dalian, China

Abstract—This paper presents an experimental study on object recognition and self-localization for multiagent robotic soccer. A color image segmentation method based on YUV color information of each pixel of the color image is applied to the ball recognition and self-localization of a small mobile robot. First the lower and higher threshold values of Y, U, and V are determined for the sample color markers. Then color image segmentation is performed using the tables that transform each value of Y, U, and V in a pixel to membership in several color classes. The position of objects on the field is efficiently and accurately calculated using a monocular camera. Self-localization is accomplished using the distances and orientations of two landmarks. Experimental results of real-time movement control are also presented.

Index Terms—Color image segmentation, object recognition, self-localization mobile robots, robotic soccer.

I. INTRODUCTION

In recent years, there have been many studies on vision-based motion control in various robotic applications. We are now developing a multiagent robotic system for soccer game [1]. In robotic soccer, each of robots must know its surroundings such as the position of the ball and goals when the robots carry out cooperative tasks. Furthermore a robot must keep its own position to estimate the arrangement of objects and share it among its teammates. The reactivity of soccer robots requires a vision system with a high processing cycle time. The soccer rules specify well defined colors for different objects in the field, and these are used as the major cue for object detection, for example, several color patches on the top of the robots, an orange golf ball, and a green color field with specific white markings. In such applications, vision systems employing region segmentation by color are crucial.

In this paper, we focus on object recognition and self-localization for small mobile robots using only a monocular camera system with the limited viewing angle. Fast color image segmentation is performed based on YUV color information of each pixel of the color image. The position of an orange ball on the field is efficiently and accurately calculated using a monocular camera. A suitable self-localization method for soccer robots is presented.

II. CONTROL SYSTEM ARCHITECTURE

The robot system is constructed of a control unit (1.5GHz Pentium 4 CPU) and a mobile base unit. The hardware configuration is shown in Fig.1. The mobile base unit uses differential drive mechanism with two wheels. It is equipped with a 16 bit CPU (Hitachi, H8) for controlling wheels and sensor units. The CPU communicates with the control unit via wireless modem (RS-232C interface, 9600bps). It acts as the controller of the primitive actions ordered by the control unit, and sends sensor data when the control unit requests them.

The control unit is equipped with an image processing board (Hitachi IP5005) and executes image processing to detect and localize objects in the field. It works for recognizing the circumstances around the robot, planning its behaviors, transmitting commands and communicating with other robots.

The robot mounts a monocular video camera to observe the field or workspace, target object, and all robots, and sends NTSC video signal to the image processing board, which converts the video signal to 256 by 220 color digital image, and extracts and measures objects in every 33msec. The image processing software is implemented using Visual C++ and the software library of IP5005 under Windows NT4.0.

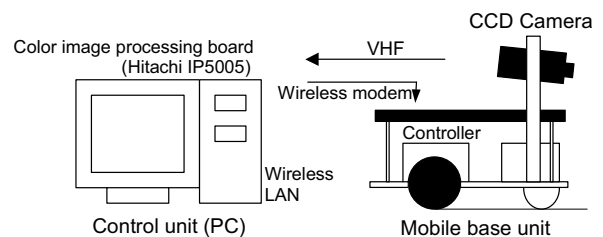


Fig.1 Hardware configuration of the robot

III. COLOR IMAGE SEGMENTATION

A fast color image segmentation algorithm using inexpensive image capture and CPU hardware was proposed [2][3]. The method enables a pixel to be classified into one of a number of color classes based on color space transformation. The YUV color space has the advantage that chrominance is

coded in two (U and V) of the color space dimensions, while intensity is coded in the third (Y). A particular color can be described as a column spanning all intensities, so such color spaces are often more useful than RGB for robotic applications.

The recognition of position and orientation of objects using the color image segmentation method is performed according to the flowchart as shown in Fig.2. In the segmentation method, each color class in the YUV color space is specified as a set of six threshold values: two for each dimension in the color space.

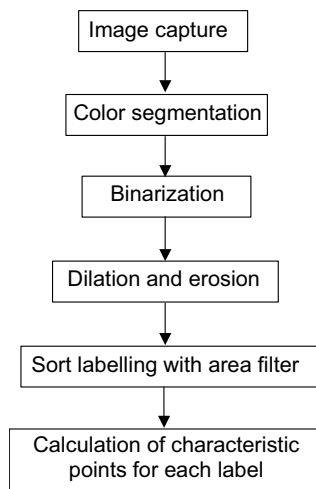


Fig.2 Flowchart of object recognition.

A color class is represented as the product of three Boolean valued functions, one along each of the axes in the space, which are stored in arrays. A pixel's membership in multiple color classes is determined if each element in an array were a more than two bit integer. In our implementation, the number of color classes is 8, and the YUV color space was discretized to 256 levels in each dimension. So three 256×8 tables, one along each of the axes in the space, are experimentally constructed. Membership in 8 distinct color classes can be evaluated at once with two bit-wise AND operations. In color image processing, each value of Y, U, and V in a pixel is transformed to 8-bit data using the table, then its membership is determined by bit-wise AND operations, while the conventional comparison approach requires 8×6 operations per pixel for the same classification because 6 operations (conditional branches) are required to determine membership in one color class for each pixel.

After the color segmentation, the regions are binarized, then they are separated based on color into separate lists using a 4-neighborhood labeling algorithm. Next, they are filtered and sorted by area size so that the smaller noisy regions are removed and the larger regions are chosen by high level processing algorithms. The centroid of each region is calculated using a library function of IP5005.

In order to examine the performance of the color extraction method, experiments of localization of a robot using a CCD camera over-looking the field were conducted. The robot mounts a standard red color marker for position recognition and a special blue color marker for orientation measurement on the

top of the body. These colors can be differentiated reliably in the YUV color space. The color markers are rectangular; the red color marker 90mm square, while the blue color marker $90\text{mm} \times 50\text{mm}$. Camera calibration was accomplished to compensate any distortions on the edge of the screen.

We examined values of Y, U, and V for some sample color markers. Fig.3 shows the experimental results of values of U and V for a red marker and a blue marker, a gray color field, and an orange golf ball.

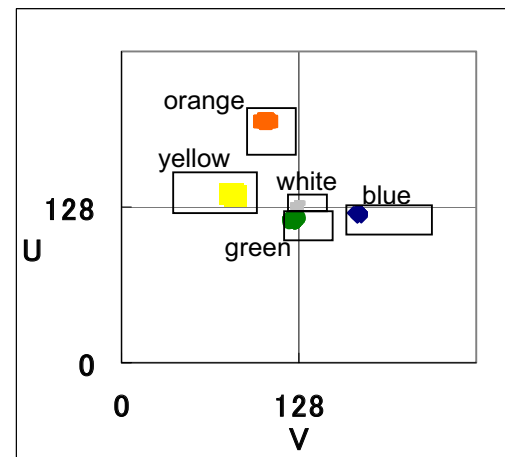


Fig.3 Values of U and V of colors for robot marker, ball and field.

Fig.4 shows an example of color segmentation of a robot. Small color marks of size not matching the ones on the robots are discarded. The blue color marker for orientation measurement is matched to the pink color marker by their distance. Two distance-matched markers are recognized as a robot. The variation of position measurements was about 1 pixel, satisfying the required accuracy for position control.

The system can classify each pixel and find regions of up to 8 colors, and outputs their centroid and area at 30 Hz on general purpose commodity hardware; the image processing time was shortened by prefetching the next frame image during processing the current image and by restricting the processing area. Position and orientation recognition of several mobile robots in the field can be simultaneously executed within 33 ms using the library functions provided by IP5005.

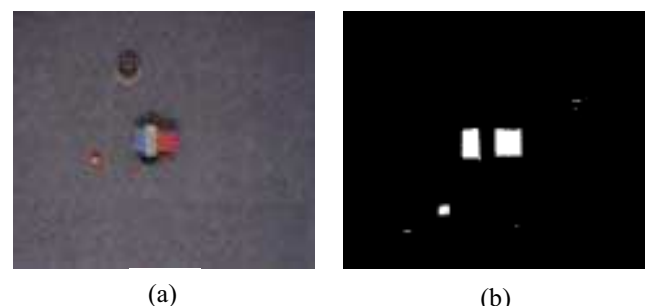


Fig.4 Color segmentation of a robot and an orange ball: (a) original, (b)after binarization.

IV. OBJECT RECOGNITION USING MONOCULAR CAMERA

The color segmentation software was applied to object recognition and self-localization using an onboard monocular camera of a mobile robot. After binarization dilation and erosion operators to the color regions are applied using the 4-neighborhood, in order to remove small disturbances and make the regions more compact. Figs. 5 and 6 show example experimental results applying the operators for object recognition. Due to noise, reflections on the top of the ball or other disturbances the segmentation process may provide more than one orange segment as candidates for the ball. Using the ball's distance the radius of the ball's projection onto the image plane is determined. A segment of too small size is rejected as ball hypothesis.

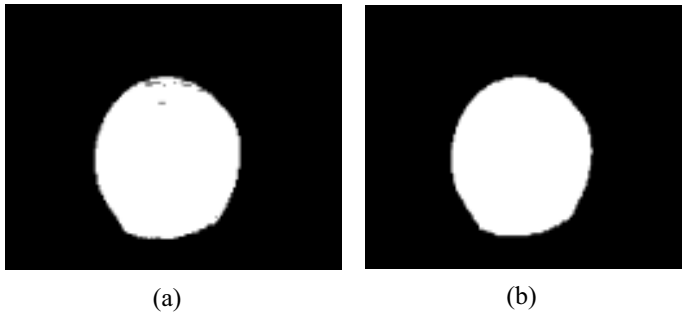


Fig.5 Experimental result of noise removal by dilation and erosion operators:
(a) original binary image, (b) after dilation and erosion.

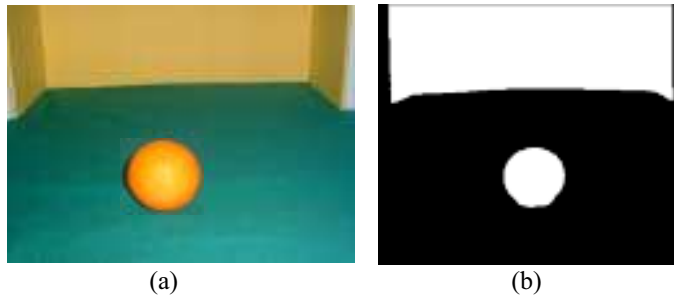


Fig.6 Color segmentation of an orange ball and a goal:
(a) original, (b) after dilation and erosion.

Besides recognizing relevant objects, further tasks are to localize the recognized objects and to perform the self-localization of the robot on the field. For this the intrinsic camera parameters as well as the external ones relative to the robot were determined by calibration.

The location of an object is calculated using the formula of a monocular camera system [4]. All relevant objects are located on the ground of the field. The restriction onto the ground provides an one-to-one correspondence between a point on the ground plane and its observation in the image plane. In order to

estimate the location of an object the lowest point in the corresponding segmentation result is determined. This point and the optical center of the camera define a viewing ray. Its intersection with the ground plane yields an estimate of the object's distance and its direction in camera coordinates. With the help of the intrinsic camera parameters pixel coordinates are converted into image coordinates. The 3-D point corresponding to the observed image point is calculated by the formula specifying the relationship between the ground plane and the image plane. The focal length of the camera, its height, the angle of inclination are also used as parameters. The point given in camera coordinates can easily be expressed in the coordinate system of the robot. Fig.7 shows the relationship between the ground plane and the image plane.

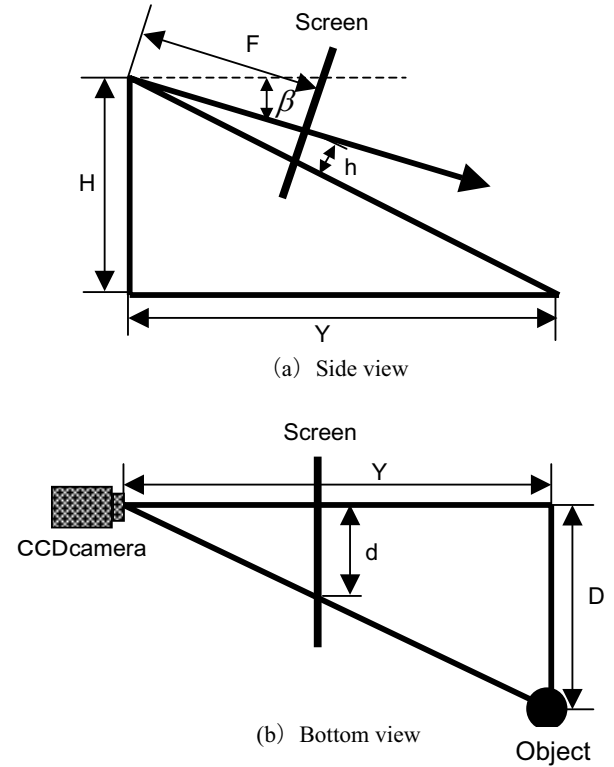


Fig.7 Relationship between ground plane and image plane.

The distances estimated by the monocular camera system are formulated as follows:

$$Y = \frac{H(F - h \tan \beta)}{F \tan \beta + h} \quad (1)$$

$$D = \frac{d(Y \cos \beta + H \sin \beta)}{F} \quad (2)$$

where

H : height of the camera (=121 mm)

F : focal length of the camera (=8mm)

β : angle of inclination (=8°)

h : Y coordinate in image plane

d : X coordinate in image plane

Experiments of ball localization were conducted. The positions of the orange golf ball in front of the robot were depicted in Fig.8. The result of the experiment is shown in Table1. The row Δx shows the average estimation errors in x coordinate, and the row Δy shows those in y coordinate. The estimation errors are shown in [mm]. A variation of measurement is sufficiently small for movement control in robotic soccer as shown in Fig.8.

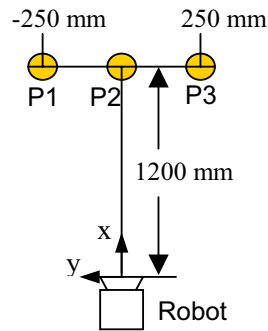


Fig.8 Positions of ball.

Table1: Experimental result of ball localization

	P1	P2	P3
Δx	18.89	11.02	21.55
Δy	7.73	0.14	6.23

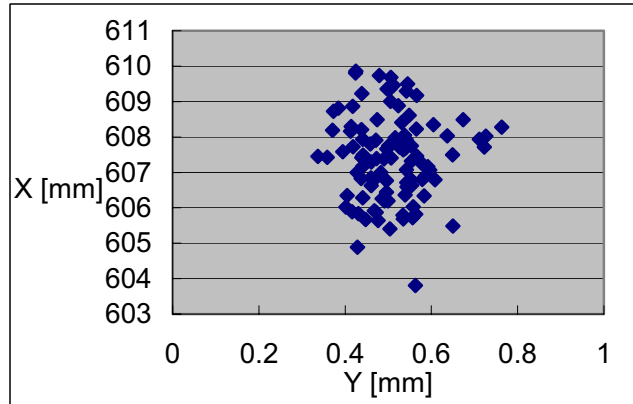


Fig.9 A variation of measurement of ball location.

V.. SELF-LOCALIZATION

Typical two self-localization methods based on triangulation were examined in order to develop the suitable method for soccer robots with a monocular camera. The first one uses the orientations of the three landmarks. The second one uses the distances and orientations of the two landmarks. It makes two circles using them, gets the intersection of the circles, and determines the position. This method is more favorable than other methods in the point that two landmarks tend to be

observed in a single image frame. The robot does not need to move its camera or itself to find out the landmarks.

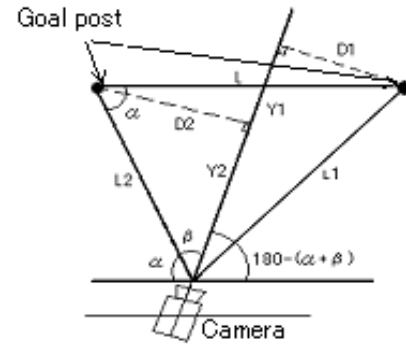


Fig.10 Estimation of position and orientation with respect to the goal post

Fig.10 shows the self-localization method utilizes the distances and orientations of the two landmarks. First, the distances $(Y1, D1)$ and $(Y2, D2)$, and the angle β are directly calculated using the formula of the monocular camera system. Next, the distances $L1, L2$, and the angle α are calculated. Thus the position and the orientation of the robot are estimated with respect to the goal post.

In experiments of self-localization, the positions of the robot were shown in Fig.11. Fig 12 shows the binary images at each position. The borders of the goal and the ground are detected as goal post positions. The result of the experiment is shown in Table2. l denotes the distance from the robot to the center of the goal, and θ denotes the orientation of the robot. The row Δl shows the average estimation errors in x coordinate, and the row $\Delta \theta$ shows those in y coordinate. The estimation errors of Δl are shown in [mm], and the estimation errors of $\Delta \theta$ are in $^\circ$.

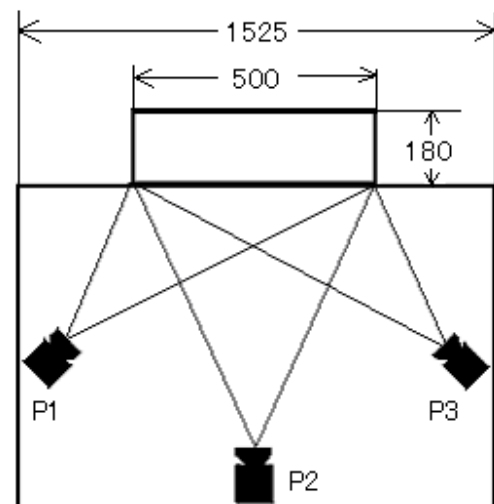


Fig.11 Positions of the robot.

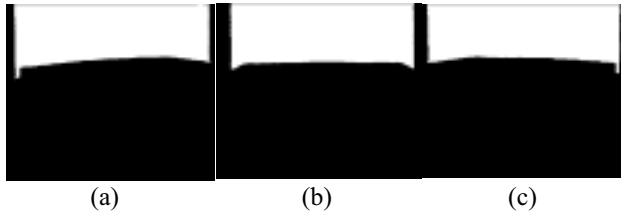


Fig.12 Binary images: (a) P1, (b) P2, (c) P3.

Table2: Experimental result of self-localization

	P1 ($l = 1000$ $\theta = 76$)	P2 ($l = 1200$ $\theta = 90$)	P3 ($l = 1000$ $\theta = 94$)
Δl	20.32	22.31	10.55
$\Delta \theta$	0.26	0.14	0.74

We adopted the method that usually uses the position obtained by odometry, and uses the estimated position obtained by using the landmarks searched actively when the odometry becomes unreliable. The goals and other objects are expressed in the coordinate system of the robot, and used for path planning.

It is difficult for each of robots to get and keep the information on its surroundings separately if the robot has only one monocular camera system with the limited viewing angle. The environment map is constructed using local visual information of each robot, and is used to accomplish the cooperative behaviors; for example, each of the robots stays at the favorable position instead of being bunching up together.

VI. EXPERIMENTS OF MOVEMENT CONTROL

In robotic soccer, a robot must coordinate its actions to push the ball into the goal. The motion control algorithm must allow a general nonholonomic mobile robot to accurately reach a target point with a desired orientation in a highly dynamic environment. The point-to-point (PTP) control method is used to control the robots to follow transient subgoals for ball handling.

An experimental small mobile robot was constructed for robotic soccer, and its movement control was examined using the object recognition system. A view of the robot is shown in Fig.13. The robot is of size 210 mm $\times$ 250 mm, and uses differential drive mechanism with two wheels due to its simplicity. The driving motor and wheel sets are located on both sides, and two caster wheels are located in front and rear of the robot. Steering control is derived from the velocity difference between two driving wheels. Each DC motor is geared to a wheel with speed reduction of 1:80, and the resolving power of movement is 0.036mm with the photo encoder attached to each wheel. The control software was implemented on a microcontroller-based control system (Hitachi 16MHz H8-3048 CPU), with the interface circuits for the internal and external sensors.



Fig.13 View of experimental small mobile robot.

First the dynamics of the two-wheeled mobile robot was experimentally identified, and the delay due to the overall image processing and the communication between the control unit (PC) and the onboard controller was estimated. By the experimental result of step response of the mobile robot, where the PWM duty cycle of two motors is changed from 0 to 50, the estimated delay time was about 150 ms.

The flowchart of command generation for movement control executed in the control unit is shown in Fig.14. In movement control, the turn angle of the robot towards the target is calculated in terms of robot coordinates.

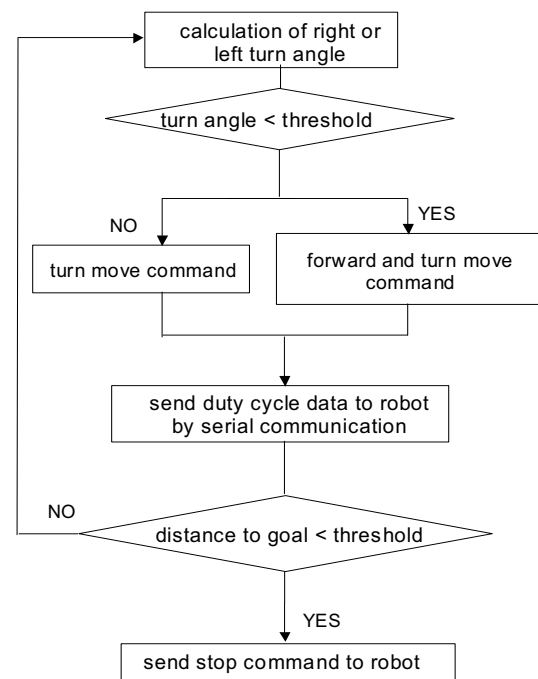


Fig.14 Flowchart of generation of command for movement control in the control unit.

For motion control, the control software calculates the following at the control period of 33 ms. First, it determines the desired translational and rotational velocities using the position and orientation errors to the goal position, compensating the

feedback delay due to color image processing using the estimated delay time and time constant of the controlled object.

Fig.15 shows the robot kinematics for the point-to-point (PTP) control method to control the robots to follow transient subgoals. We adopted the following feedback control law:

$$\begin{aligned} v &= k_{\rho} \rho \\ \omega &= k_{\alpha} \alpha + k_{\beta} \beta \\ (\beta &= \alpha + \theta) \end{aligned} \quad (3)$$

where v : the tangent velocity of the robot, ω : the angular velocity of the robot, ρ : the distance between the center of the robot's wheel axle and the goal position, α : the angle between the forward direction of the robot and the vector connecting the center of the wheel axle with the goal position, θ : the pose error.

The control parameters k_{ρ} , k_{α} , k_{β} were determined [5] so that the closed control system is locally exponentially stable as follows:

$$k_{\rho} > 0, k_{\alpha} - k_{\rho} > 0, k_{\beta} < 0 \quad (4)$$

Furthermore, the parameters were adjusted so that the robot can be smoothly guided to face the goal [6][7].

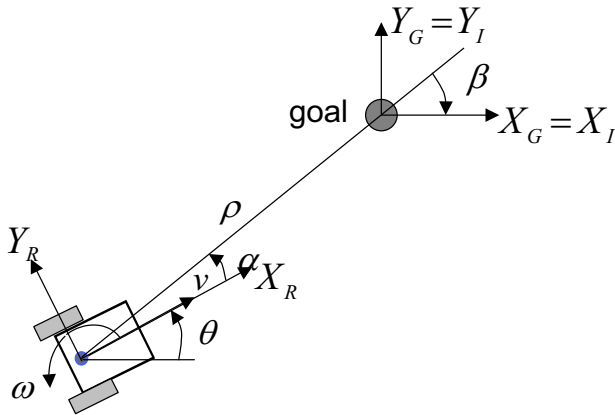


Fig.15 Kinematics and movement control of a mobile robot.

Then it calculates the desired angular velocities of the right and left wheels by a simple, invertible linear transformation. Finally, it sends 8-bit control data (4-bit data for each motor) to the onboard computer with wireless modem. The onboard computer executes multitasking processing and responds to the commands from the host quickly. The angular velocity control loop is executed on the microcontroller and run every 1 ms to compensate for differences between desired and estimated velocity of the wheels, considering nonlinear characteristics, such as frictional dead zone and velocity saturation, between the duty cycle and the angular velocity.

Fig. 16 shows an example robot trajectory of movement control by the feedback control, where the desired orientation at the target position is 90° . The mobile robot smoothly reached the target at the desired orientation.

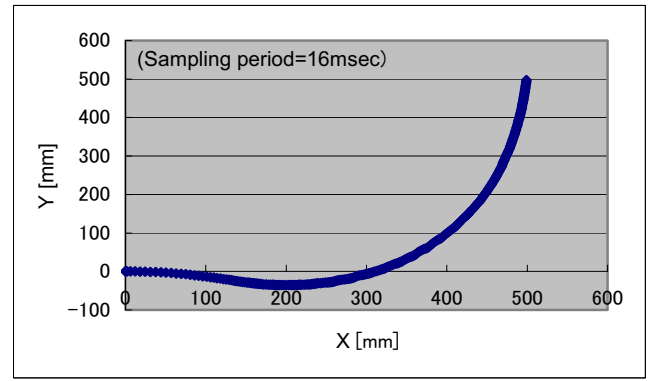


Fig.16 An example of robot trajectory by feedback control.

VII. CONCLUSIONS

We have implemented the color image segmentation method on general purpose, inexpensive hardware, and applied to object recognition and self-localization suitable for soccer robots. The accuracy of object localization was satisfactory for movement control in robotic soccer. The self-localization method chooses two landmarks most suitable for knowing its own position actively. We are planning to develop a method to construct an environment map by integrating the visual information of teammates as the future work. Using the map, the robot can know the situation outside its view, and perform advanced behaviors such as role sharing and effective pass work.

We have realized movement control of mobile robots using the localization method. The feedback control law determines the translational and rotational velocities using the position and orientation errors to the goal position. Experiments show that the robot can be smoothly guided to face the goal.

REFERENCES

- [1] G. Yasuda, "Modular distributed control architecture for cooperative soccer-playing robot agents," *Journal of Robotics and Mechatronics*, vol. 12, no. 1, pp. 29-34, Feb. 2000.
- [2] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in *Proceedings of IROS'00*, 2000, pp. 2061-2066.
- [3] G. Yasuda, and B. Ge, "Position control of mobile robots using global vision with fast color image segmentation," in *Proceedings of the 3rd China-Japan Mechatronics Symposium*, 2002, pp. 191-196.
- [4] T. Kubota, H. Hashimoto, and F. Harashima, "Path Searching for Mobile Robot by Local Planning," *Journal of Robotics Society of Japan*, vol. 7, no. 4, pp. 267-274, Aug. 1989 (in Japanese).
- [5] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Cambridge, MA, MIT Press, 2004.
- [6] K. Han, and M. Veloso, "Reactive visual control of multiple non-holonomic robotic agents," in *Proceedings of ICRA'98*, 1998.
- [7] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A stable tracking control method for a non-holonomic mobile robot," in *Proceedings of IROS '91*, 1991, pp. 1236-1241.