BabyTigers-98: Osaka Legged Robot Team

Noriaki Mitsunaga and Minoru Asada and Chizuko Mishima

Dept. of Adaptive Machine Systems, Osaka University, Suita, Osaka, 565-0871, Japan

Abstract. The Osaka Legged Robot Team, BabyTigers-98, attended the First Sony Legged Robot Competition and Demonstration which was held at La Cite La Villeta, a science and technology museum in Paris in conjunction with the second Robot Soccer World Cup, RoboCup-98, July, 1998. This article describes our approach to the competition. The main feature of our system is to apply a direct teaching method to behavior acquisition for four-legged robots as one of the robot learning issues. Since the robots recognize objects in the field by color information, the color calibration is another issue. To cope with color changes due to lighting conditions, we developed an interactive color calibration tool with graphic user interface. Based on the color calibration and the learning by teaching methods, the robot successfully exhibits basic skills such as ball approaching and kicking.

1 Introduction

The Osaka Legged Robot Team, BabyTigers-98, attended the First Sony Legged Robot Competition and Demonstration which was held at La Cite La Villeta, a science and technology museum in Paris in conjunction with the Second Robot Soccer World Cup Competition and Conference, RoboCup-98, July 2-9, 1998.

The final goal of the Legged Robot Project in our group is to establish the methodology to acquire behaviors for team cooperation in the RoboCup context from the interactions between the legged robots through multi sensor-motor coordinations. The desired behaviors can be categorized into three levels; a basic level, a basic cooperation level, and a higher team cooperation level. In this article, we briefly explain our first step for the first level skill acquisition with preliminary results. That is behavior acquisition by direct teaching.

The most fundamental feature of the legged robot is that it moves by its four legs (12 DOFs), which is quite different from conventional mobile robots (2 or 3 DOFs). From a viewpoint of sensor-motor learning and development, multi modal information and multi DOFs control should be established simultaneously, that is, affecting each other, the sensory information is abstracted and the multi joint motions are well coordinated at the same time [1]. However, it seems very difficult for artificial systems to develop the both together. Our goal is to design such a method.

From our experiences on robot learning, we realized that the number of trials by real robot is limited and a good trade-off between computer simulations and real robot experiences is essential for good performance. However, the computer simulation of the legged robots seems difficult to build, then we decided to adopt a direct teaching method in order to reduce the number of trials by real robot.

Since the robot detects objects in the field using color information such as a orange ball, aqua-blue and yellow goals, color calibration is another issue for the robot to robustly detect objects in the field and then to learn by teaching. Since color calibration during the game seems difficult to realize due to the limitation of on-board computation power, we adopt an off-line color calibration method. To make the calibration process much easier, we developed an interactive color calibration tool with graphic user interface.

This article is structured as follows. First, we show an interactive off-line color calibration system we developed. Next, we describe the method of robot learning by direct teaching. Finally, we show some experimental results and discuss the future issues.

2 Vision System

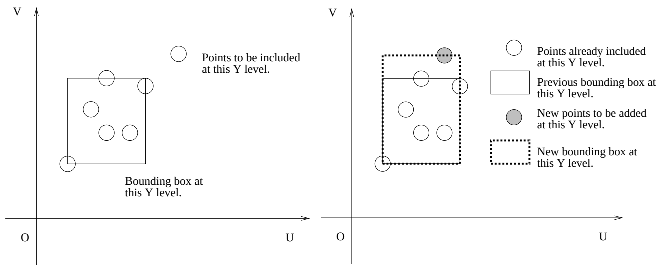
2.1 Color detection

In the legged league field [3], 7 colors (aqua-blue, yellow, orange, blue, red, pink, green) are used and robots need to detect all of them. The Sony's legged robot has the color detection facility in hardware that can handle up to 8 colors at frame rate (60-80[ms]). To detect colors with this facility, we need to specify each color in terms of subspace in YUV color space. YUV subspace is expressed in a table called Color Detection Table(CDT). In this table, Y are equally separated into 32 level and in each Y level we specify one rectangle $(u_{\min i}, v_{\min i})$,$(u_{\max i}, v_{\max i})$ $(i = 1, ..., 32)$.

The color observed by the robots dynamically changes due to the robot and/or object motions. The reason is that it depends on various kinds of parameters such as, the color of object itself, the color of light source, the surface material of object, the surface orientation of object, the angle between the view line of the camera and the beam of the light and so on. In order to cope with color changes, YUV subspaces should be large enough to involve the changes, but small not to include different colors. Then we developed an interactive tool so that we can make and modify CDTs that can handle expansion and reduction of the CDT easily.

Basic procedures are:

1. To make a CDT, we specify multiple pixels that should be treated as the same color in an image taken by the legged robot. Each pixel has a pair of (y, u, v) value or a point in YUV space. According to the value of y, each point is classified into 32 Y level. In each y level, $(u_{\min i}, v_{\min i})$,$(u_{\max i}, v_{\max i})$ $(i = 1, ..., 32)$ are set to indicate the bounding box that includes all points in that y level (see Fig.1(a)).
2. To add a point in YUV to the CDT which should be treated as the same color, expand the bounding box of the corresponding y level to include the point (see Fig.1(b)).

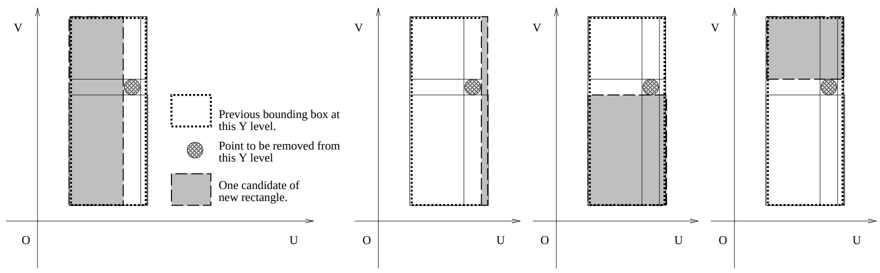


(a) Make a bounding box of the CDT

(b) Add a point to the CDT

Fig. 1. Making a CDT first time and expansion of a CDT

3. To delete a point in YUV from the CDT which should not be treated as the same color, shrink the bounding box of the corresponding y level to exclude the point. To exclude the point, we have four options (see Fig.2) and we take the one which is the least reduction of the area (in this case Fig.2(a)).

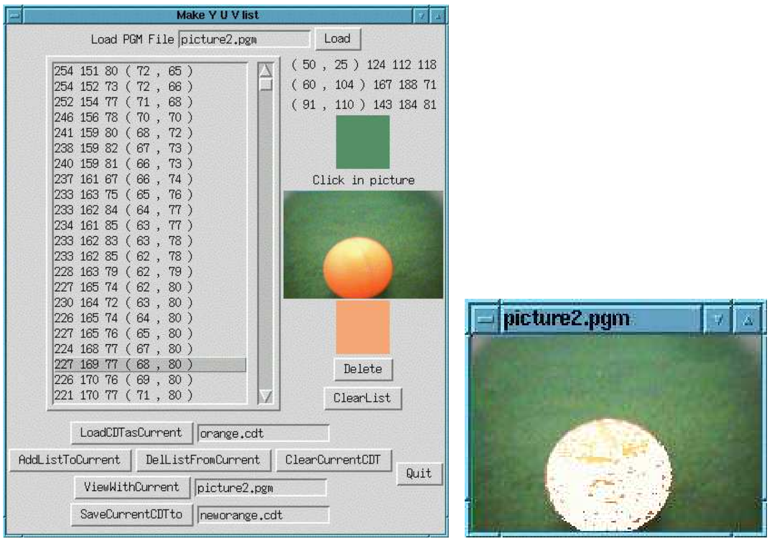


(a) (b) (c) (d)

Fig. 2. Deleting a point from CDT(four options)

We developed a GUI tool written in Tcl/TK to realize the above procedures easily. The appearance of the tool and emulated color detection is shown in Fig.3. We show an example of CDT construction with images shown in Fig.4, which shows ball images observed by the robot from different positions.

First we specify some pixels in Fig.4(a) and make a CDT. The distribution in



(a) making color detection table with the tool

(b) emulation of color detection

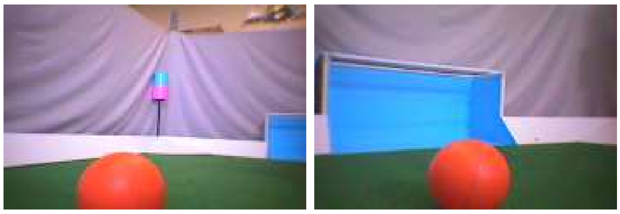
Fig. 3. GUI tool to make CDTs

YUV space projected to *UV* plane (at y level 16) is shown in Fig.5 (the bounding box is also shown). Distributions in *YUV* space of the ball color in the picture Fig.4(a) and (b) are shown in Fig.6. It indicates that the distribution in Fig.4(a) does not cover the one in Fig.4(b). That is, the CDT constructed by Fig.4(a) does not cover the color distribution of the ball in Fig.4(b) (see Figs.8(a) and (b)). Then, we specify some pixels that do not seem to be detected in Fig.4(b) and add to the CDT. Fig.7 shows the expansion of the bounding box in y level 16. After all of addition, the robot can detect Fig.4(b) as Fig.8(c).

2.2 Object detection

Since the legged league field includes multiple objects, some of which have the same color, and observed images are noisy, object detection is carried out with noise reduction based on a priori knowledge of the environment.

First, using the color detected image, pixels in the same color are connected into regions by their 8-neighbors. At the same time, area sizes, centroids and bounding boxes of the regions are calculated. Next, the small regions less than 4 pixels are deleted because the minimum size of the smallest object (ball) in the image plane is 4. Then, the order of object detection is determined, consid-



(a) ball picture (A) (b) ball picture (B)

Fig. 4. Pictures of a ball

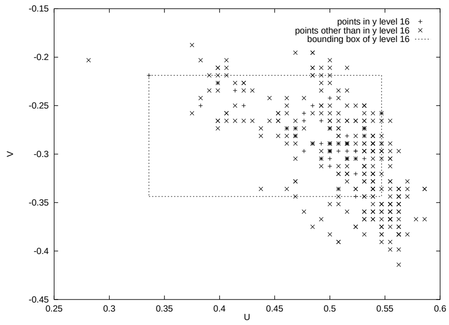


Fig. 5. Distribution of the ball color (A) in UV space

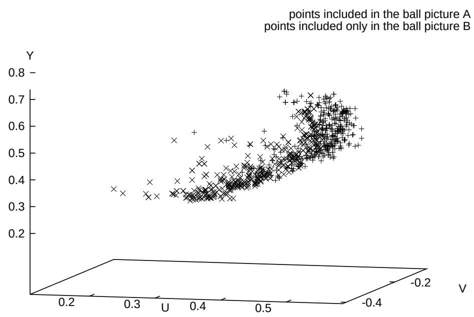


Fig. 6. Distribution of the ball color in YUV space

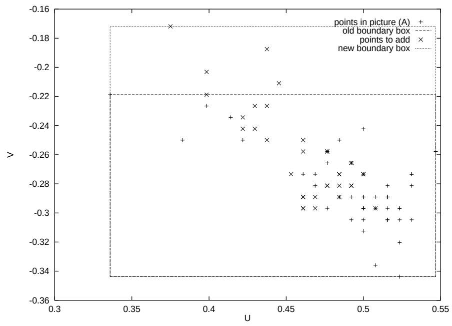


Fig. 7. Distribution of the ball color in UV space y level 16

(a) ball picture (A): color detection with the CDT made with the ball picture (A)

(b) ball picture (B): color detection with the CDT made with the ball picture (A)

(c) ball picture (B): color detection with the CDT made with the ball picture (A) and (B)

Fig. 8. Result of color detection with CDTs (emulation by software)

ering the importance of object, easiness of color discrimination and visible constraints of multiple object colors. Constraints means that maximum number of objects that can be observed simultaneously is known, orange(1), aqua-blue(3), yellow(3), green(3), pink(3), blue(3) and red(3). This can also be used to noise reduction. The order we used is as follows.

```
1. ball
2. poles: (top color)pink      (bottom color)aqua-blue
                     aqua-blue               pink
                     pink                    yellow
                     green                   pink
                     pink                    green
```

```
3. goals:                  aqua-blue goal
                           yellow goal
4. robots:                 blue robots
                           red robots
```

The color of robot's body and head was blue or red, of which saturation was low and difficult to detect. That is why we put them in low priority in object detection.

3 Behavior Acquisition

In the competition, we assigned the roles of attacker and goal keeper to three robots, that is, two attackers and one goal keeper. In the following, we explain how the behaviors for the goal keeper and the attackers are obtained.

3.1 Goal-keeper

We adopted a strategy for the goal keeper to save the goal by positioning itself in front of the goal. The angles of the visible area is 53 degrees in width and 41 degrees in height. It means that the robot cannot see as many goals and landmark poles to determine its position. The result of color detection is small in size (88×59 pixels), and the poles are small (diameter is 0.1[m] and height is 0.2[m]). So, the robot rotates its head to enlarge the visible field, and decides its movement depending on rough estimation of its position. Since the landmark poles near the opponent goal is difficult for it to observe, the robot compares the angles between its own goal and four poles near own goal in front of its own goal, and determines which direction to move. Further, the robot tracks the ball for a while when it is near the ball.

3.2 Attacker

The behavior of two attackers is obtained by direct teaching. More precisely, the robot learns its behavior by classifying the data taught by a human trainer from a viewpoint of information theory. A human trainer can teach the robot via serial console with PC.

First, the robot collects test data, pairs of an action command given by human trainer and the sensory information during the action execution in every 300ms. Next, C4.5 [2] is applied to the test data to extract rule sets. Then, the validity of the rule sets are checked against test data by applying the rule sets. Specifications of the data for shooting skill acquisition are as follows:

– **action command** forward, backward, left-shift, right-shift, left-rotation, and right rotation (20 degs/sec): these abstracted action commands will be decomposed into more primitive motor commands in future.

– **sensory information** head direction (rad.) and image features of both the ball and the goal in the observed 88x59 image: area (pixels), position (x-y coordinates), bounding rectangle (x-y coordinates of corners), height, and width. See Figure 9.
– **training position and sampling rate:** initial positions of direct teaching by serial line connection are evenly distributed in the field heading the goal, and the sampling rate is 300ms. One trial from the initial position to the goal takes about 10 seconds.

In the experiments, 740 pairs of an action command and sensory information for training data to make rule sets, and 500 pairs for test data to check the validity of the rule sets are given. The both data are obtained in the same manner starting from the similar initial positions, but individual pairs are different from each other trial by trial. The number of rules obtained is about 30, and typical ones for forward(F), left-rotation (LR), and right-shift (RS) are shown below. Figure 9 shows the typical situation of right-shift motion.

```
Forward: BallArea>56  Left-     BallArea>49  Right- BallArea<=40
         HeadDir>-.14 Rotation: HeadDir>.52  Shift: BallYcen>8
         HeadDir<=.37           HeadDir<=1.10       BallXmin>3
         GoalXmin<=11           BallWidth<=11       HeadDir<=-.24
                                                    GoalArea<=665
                                                    GoalXmax<=64
```

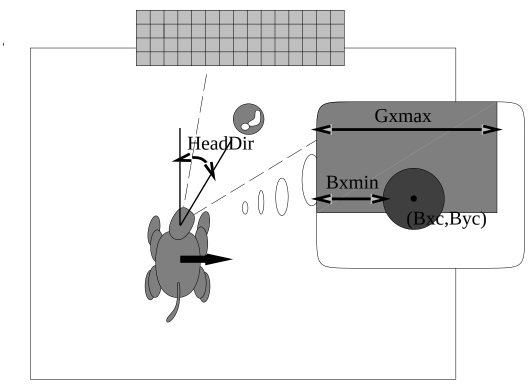


Fig. 9. Typical situation to take Right-Shift motion

The following table indicates a confusion matrix showing where the miss-classification of the training cases occur. Due to the wrong teaching by a human trainer (actually, she has not got used to teaching), this matrix includes many miss-classifications, especially `LEFT-TURN` does not have any correct classifications. We are planning to skill up teaching and also to use more training data to construct robust classification. Generalization is one of the big issue of direct teaching, and EBL seems one alternative to solve the problem. These are under investigation.

classified as:	LEFT	LEFT-TURN	RIGHT	RIGHT-TURN	FORWARD	BACK
	111	13	3		4	
	14			2	9	
			8	16	9	
			20	151	34	
	1		3	13	91	

4 Conclusion

This paper has described the implementation of vision system and strategies used in BabyTigers-98. Although it was not the best, the resultant behaviors were encouraging. The followings are future issues.

1. The current color calibration system includes an off-line interactive process, therefore, it cannot cope with dynamic changes in color during the match. Full autonomous on-line color calibration system should be developed.
2. Direct teaching seems useful under the condition that the robot sensation can be informed to the human trainer. However, it does not generally true, and often perceptual aliasing happens. As a short range research issue, the robot should reconstruct its state space so that the instructions given by the trainer can be consistent and effective. As a longer one, both the robot and the trainer should learn each other, that is, the robot build the trainers model and vice versa. Through the model developing process, the robot learns how to behave according to the instructions, and the trainer learns how to teach it.
3. Currently, only the basic skills have been obtained and not the cooperative ones yet. Direct teaching should be useful for such behavior acquisition. Mutual learning (co-evolution) process is indispensable to obtain the cooperative and competitive behavior in dynamic multi-agent environment.

References

1. Asada, M., "An Agent and an Environment: A View on "Having Bodies" - A Case Study on Behavior Learning for Vision-Based Mobile Robot," *Proceedings of 1996 IROS Workshop on Towards Real Autonomy,* pp.19-24, 1996.
2. Quinlan, J. R., "C4.5: Programs for Machine Learning," *The Morgan Kaufmann Series in Machine Learning*, 1993.
3. Veloso, M., Uther, W., Fujita, M., Asada, M. and Kitano, H., "Playing Soccer with Legged Robot," *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 437–442, 1998.

This article was processed using the LaTeX macro package with LLNCS style