AI Magazine Volume 19 Number 3 (1998) (© AAAI)

TRACKIES

RoboCup-97 Middle-Size League World Cochampion

Minoru Asada, Sho'ji Suzuki, Yasutake Takahashi, Eiji Uchibe, Masateru Nakamura, Chizuko Mishima, Hiroshi Ishizuka, and Tatsunori Kato

■ This article describes a milestone in our research efforts toward the real robot competition in RoboCup. We participated in the middle-size league at RoboCup-97, held in conjunction with the Fifteenth International Joint Conference on Artificial Intelligence in Nagoya, Japan. The most significant features of our team, TRACKIES, are the application of a reinforcement learning method enhanced for real robot applications and the use of an omnidirectional vision system for our goalie that can capture a 360-degree view at any instant in time. The method and the system used are shown with competition results.

Building robots that learn to perform a task in a real world has been acknowledged as one of the major challenges facing AI and robotics. Reinforcement learning has recently been receiving increased attention as a method for robot learning with little or no a priori knowledge and a higher capability for reactive and adaptive behaviors (Connel and Mahadevan 1993). In the reinforcement learning scheme, a robot and an environment are modeled by two synchronized finite-state automatons interacting in discrete-time cyclical processes. The robot senses the current state of the environment and selects an action. Based on the state and the action, the environment makes a transition to a new state and generates a reward that is passed back to the robot. Through these interactions, the robot learns a purposive behavior to achieve a given goal.

As a test bed for real robot applications of the reinforcement learning method, we selected soccer-playing robots. We started with simple tasks such as avoiding an opponent and shooting a ball into a goal (Asada et al. 1996), then shifted to more complicated tasks such as shooting while avoiding an opponent (Uchibe, Asada, and Hosoda 1996). The behavior of the opponent is scheduled for the learner to efficiently obtain the desired behavior. Currently, we are focusing on a problem of state-space construction through the robot experiences (Takahashi et al. 1996).

We participated in the middle-size robot league of RoboCup-97, held as part of the Fifteenth International Joint Conference on Artificial Intelligence. Our team consisted of four attackers, each of which has a normal vision system and one goalie with an omnidirectional vision system. In this article, we describe the milestone of our research efforts in our work for the RoboCup middle-size league competition. First, we give a brief overview of the reinforcement learning method and the problems in applying it to real robot applications; we then give our method of coping with these issues in the context of RoboCup. Finally, we show our system and the experimental results of RoboCup-97.

Applying Q-Learning to a Real Robot

First, we follow the explanation of Q-learning by Kaelbling (1993). For a more thorough treatment, see Watkins and Dayan (1992). Then, we show some problems of applying Q-learning to real robot tasks.

Basics of Q-Learning

We assume that the robot can discriminate the set S of distinct world states and can take the set $\boldsymbol{A}$ of actions on the world. The world is modeled as a Markov process, making stochastic transitions based on its current state and the action taken by the robot. Let $T(s, a, s')$ be the probability of transition to the state s' from the

current state-action pair (*s, a*). For each state-action pair (*s, a*), the reward *r*(*s, a)* is defined.

The general reinforcement learning problem is typically stated as finding a policy that maximizes the discounted sum of rewards received over time.[1] This sum is called the *return* and is defined as

$$\sum_{n=0}^{\infty} \gamma^n r_{t+n},$$

where r_t is the reward received at step t given that the agent started in state s and executed policy f. γ is the discounting factor; it controls to what degree rewards in the distant future affect the total value of a policy. The value of γ is usually slightly less than 1.

Given definitions of the transition probabilities and the reward distribution, we can solve for the optimal policy, using methods from dynamic programming (Bellman 1957). A more interesting case occurs when we want to simultaneously learn the dynamics of the world and construct the policy. Watkin's Q-learning algorithm gives us an elegant method for learning the dynamics of the world and constructing the policy.

Let $Q^*(s, a)$ be the expected return, or *action-value function,* for taking action a in a situation s and continuing thereafter with the optimal policy. It can recursively be defined as

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \max_{a' \in A} Q^*(s',a').$$

Because we do not know T and r initially, we construct incremental estimates of the Q-values online. With $Q(s, a)$ equal to an arbitrary value (usually 0), every time an action is taken,

$$Q(s,a) \Leftarrow (1-\alpha)Q(s,a) + \alpha(r(s,a) + \gamma \max_{a' \in A} Q(s',a')),$$

where r is the actual reward value received for taking action a in a situation s, s' is the next state, and α is a learning rate (between 0 and 1).

Problems of Applying Q-Learning to Robot

Traditional notions of state in the existing applications of the reinforcement learning algorithms fit nicely into deterministic state-transition models (for example, one action is forward, backward, left, or right, and the states are encoded by the locations of the agent). However, this is not always the case in the real world, where everything changes asynchronously (Mataric 1994). That is, one action does not always correspond to one state transition, and vice versa. Thus, we need to have the following principles for the construction of state and action spaces.

First is natural segmentation of the state and action spaces: The state (action) space should reflect the corresponding physical space in which a state (an action) can be perceived (taken).

Second is the real-time vision system: Physical phenomena happen continuously in the real world; therefore, the sensor system should monitor the changes of the environment in real time, which means that the visual information should be processed in video frame rate (33 microseconds [ms]).

The state and action spaces are not discrete but continuous in the real world; therefore, it is difficult to construct the state and action spaces in which one action always corresponds to one state transition. We call this *the state-action deviation problem* as one of the so-called *perceptual aliasing problems* (Whitehead and Ballard 1990) (that is, a problem caused by multiple projections of different actual situations into one observed state). The perceptual aliasing problem makes it difficult for a robot to take an optimal action. In the following, we first show how to construct the state and action spaces and then how to cope with the state-action deviation problem.

Learning Time

The long learning time is the famous *delayed reinforcement problem* because no explicit teacher signal indicates the correct output at each time step. To avoid this difficulty, we construct the learning schedule such that the robot can learn in easy situations at the early stages and later on learn in more difficult situations. We call this *learning from easy missions* (or LEMs).

Shooting Behavior Acquisition

The task for a mobile robot is to shoot a ball into a goal, as shown in figure 1 (Asada et al. 1996). We assume that the environment consists of a ball and a goal.

We define substates and action space for the robot to learn. *Substates* are defined according to the position and the size of the ball or goal, which are naturally and coarsely classified images (figure 2). The *action space* is defined to resolve the state-action– deviation problem, as follows: Each action executed during a fixed-time interval (usually short, say, 33 ms in our case) is regarded as an action primitive. The robot continues to take one action primitive at a time until the current state changes. This sequence of the action primitives is called an *action*.

We assign the reward value of 1 when the ball is kicked into the goal; otherwise, it is 0.

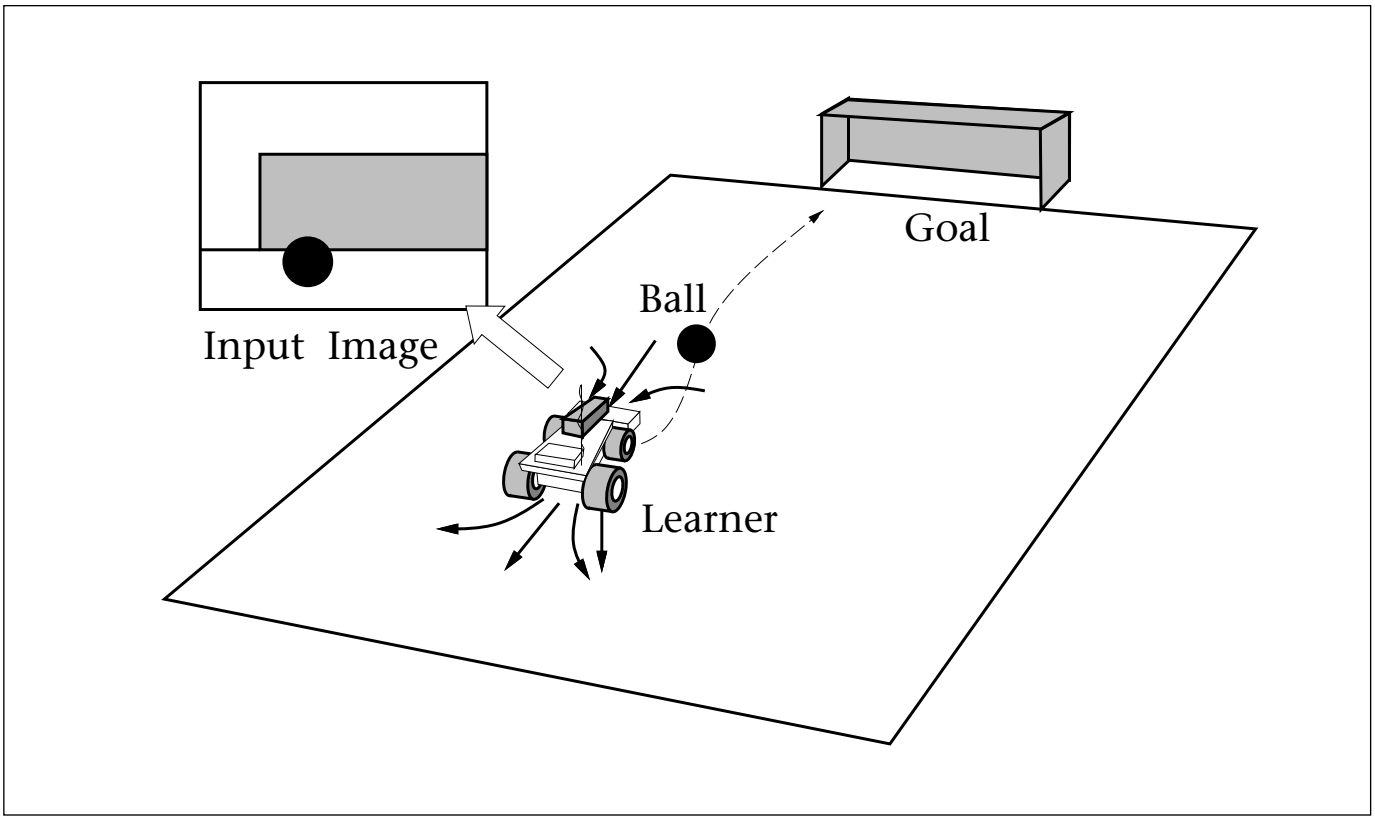

Figure 1. The Task Is to Shoot a Ball into a Goal.

This makes the learning time consuming because it takes the robot a large number of trials to reach the goal state. Although adopting a reward function in terms of distance to the goal state makes the learning time much shorter in this case, it seems difficult to avoid the local maxima of the action-value function Q.

A discounting factor γ is used to control to what degree rewards in the distant future affect the total value of a policy. In our case, we set the value at slightly less than 1 ($\gamma = 0.8$).

Shooting a Ball While Avoiding an Opponent

In the second stage, we set up an opponent just before the goal, that is, a *goalie*, and make the robot learn to shoot a ball into a goal while it avoids the goal keeper (figure 3). The basic idea is, first, to obtain the desired behavior for each subtask and then to coordinate two learned behaviors. For the first subtask (shooting behavior), we have already obtained the learned policy by using the state space shown in figure 2. For the second subtask (avoiding behavior), we add the substates for the opponent that consist of the size and its position in the image.

The time needed to acquire an optimal policy mainly depends on the size of the state space. If we apply the monolithic Q-learning to multiple goal tasks, the expected learning time is exponential in the size of the state space (Whitehead 1991). Therefore, a number of methods have been utilized to speed up learning in multiple tasks. One technique is to divide a multiple task into some subtasks and coordinate behaviors that are independently acquired. We have proposed a method that obtains a coordinated behavior consisting of different behaviors previously learned (Asada et al. 1994). The difficulty of the problem is to coordinate different behaviors that are concurrent and interfere with each other; therefore, action selection might be in conflict with the dynamic and complicated situations.

We consider three kinds of coordination: (1) simple summing of different action-value functions, (2) switching of action-value functions according to situations, and (3) learning of a new behavior given the previously learned policies. In the first two methods, the previously learned action-value functions are simply summed or switched. Therefore, these methods cannot cope with local maxima or hidden states caused by a combination of state spaces. Consequently, an action suitable for these situations has never been learned. To cope with these new situations, the robot

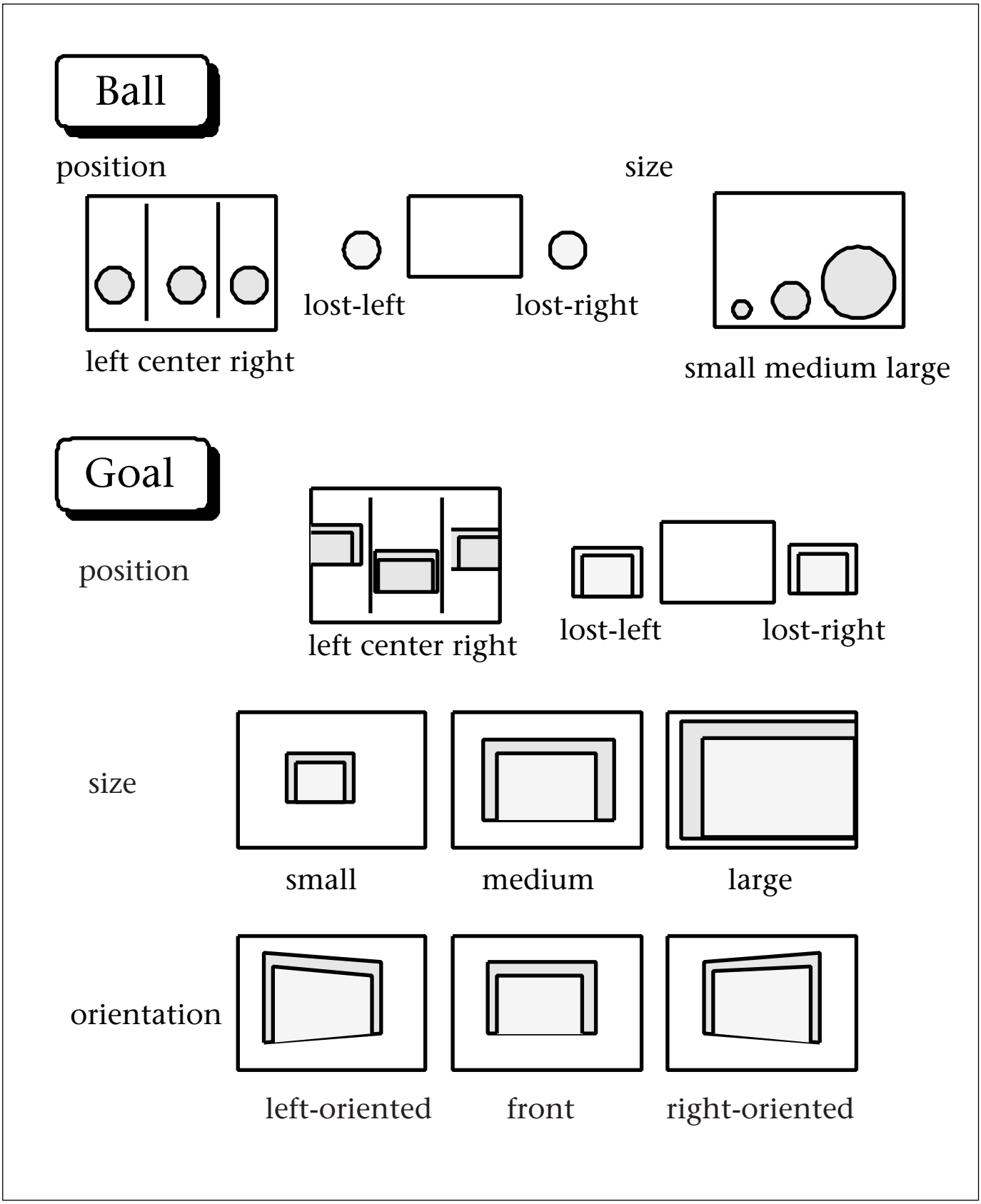

Figure 2. The Ball Substates and the Goal Substates.

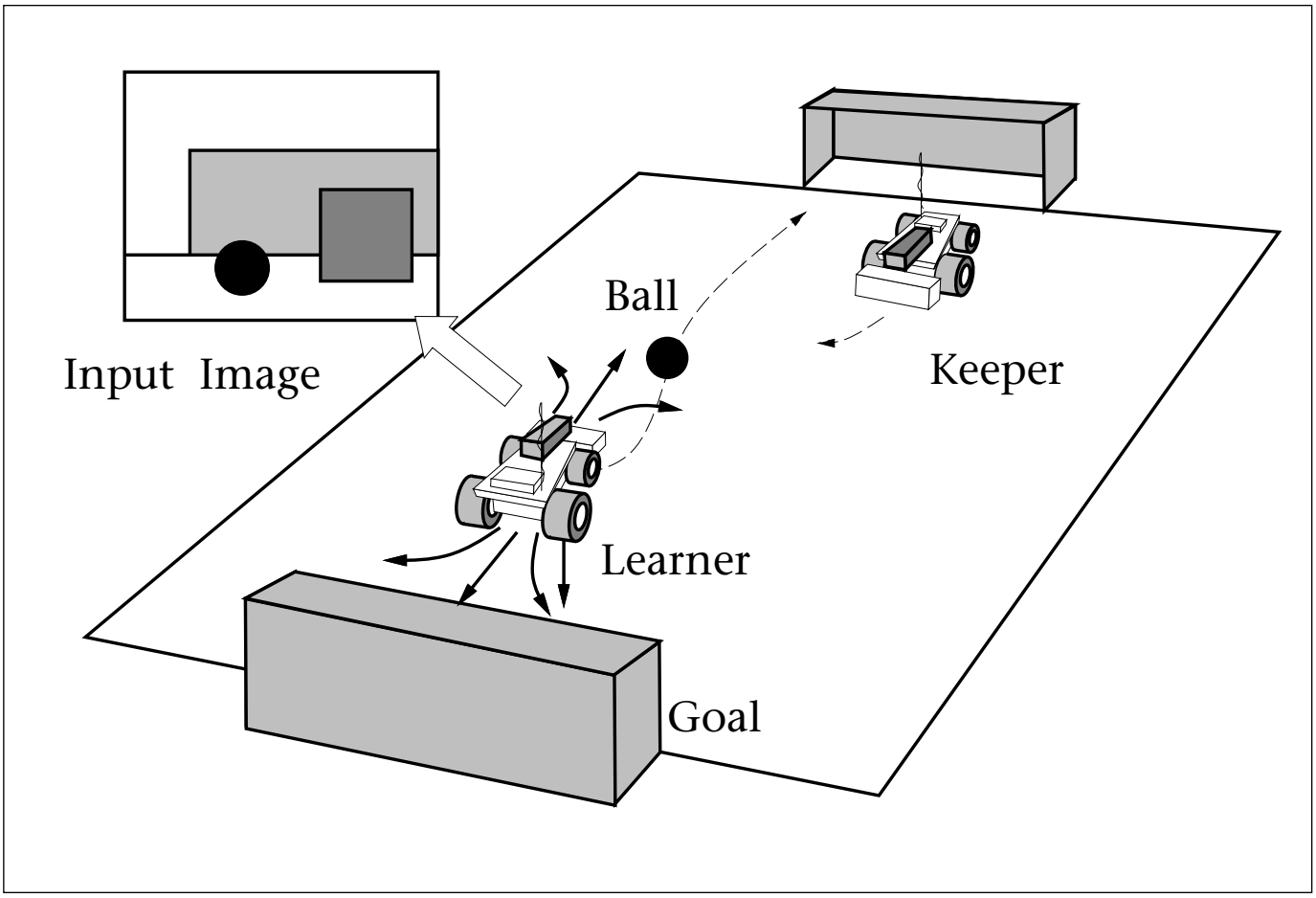

Figure 3. The Task Is to Shoot a Ball into the Goal Avoiding an Opponent.

needs to learn a new behavior by using the previously learned behaviors (see Asada et al. [1994] for more details).

Real Robot System

In the competition, we applied our methods to our five robots: Four of them are attackers with a normal vision system (figure 4), and the last one is a goalie with an omnidirectional vision system (figure 5) to look at the goal and ball coming from any direction at the same time. Every robot has a power-wheeled steering locomotion system, a single-color charge-coupled device camera, and a video transmitter using an ultrahigh-frequency band.

Figure 6 shows the configuration of the robot system. Each robot is controlled by a remote PC computer. The image taken by a CCD camera on the robot is transmitted to a UHF receiver and processed on the host computer. According to the learning results, action selection is done by the host computer, and a radio-controlled interface generates a control

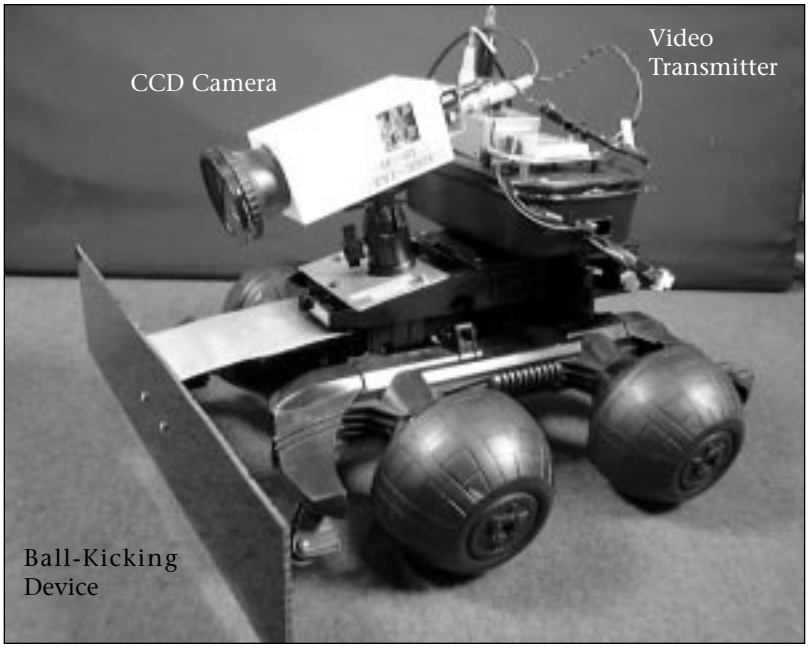

Figure 4. Attacker Robot.

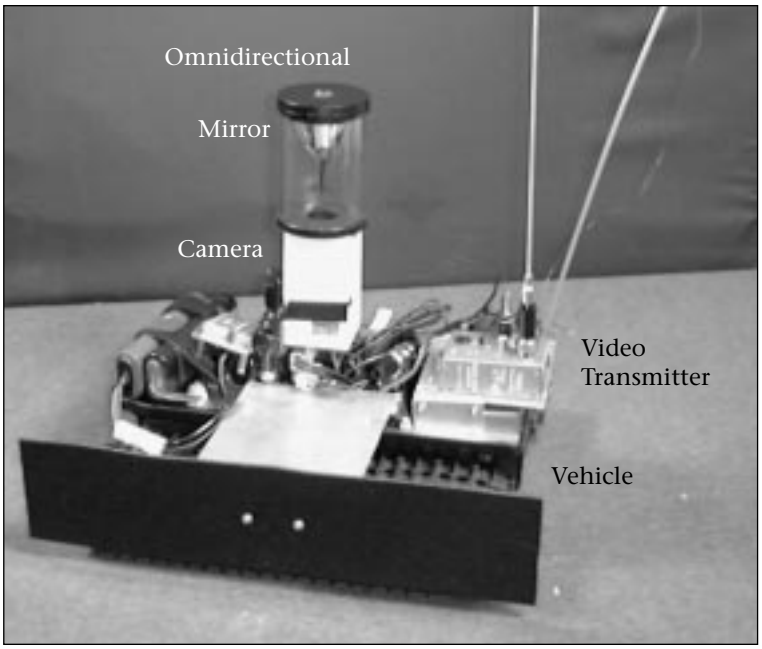

Figure 5. Keeper Robot.

signal to move the robot. Fujitsu color-tracking vision is used for image processing. It is used to detect a colored region in the image, which, according to RoboCup regulations, is defined as the ball; the goal; or the opponent in red, blue, and yellow, respectively.

Results of RoboCup-97

For the competition, we embedded the learned behaviors of shooting a red ball into a blue goal while avoiding yellow opponents into four attackers and goalkeeping into one goalie. The learning scheme for the goalie is simple. The reward is when the goalie locates itself between the ball and the goal and keeps the goal. The state and action spaces are similarly defined as the learning scheme for the shooting behavior.

We had two games in the preliminary round with the Royal Melbourne Institute of Technology RAIDERS and the University of Southern

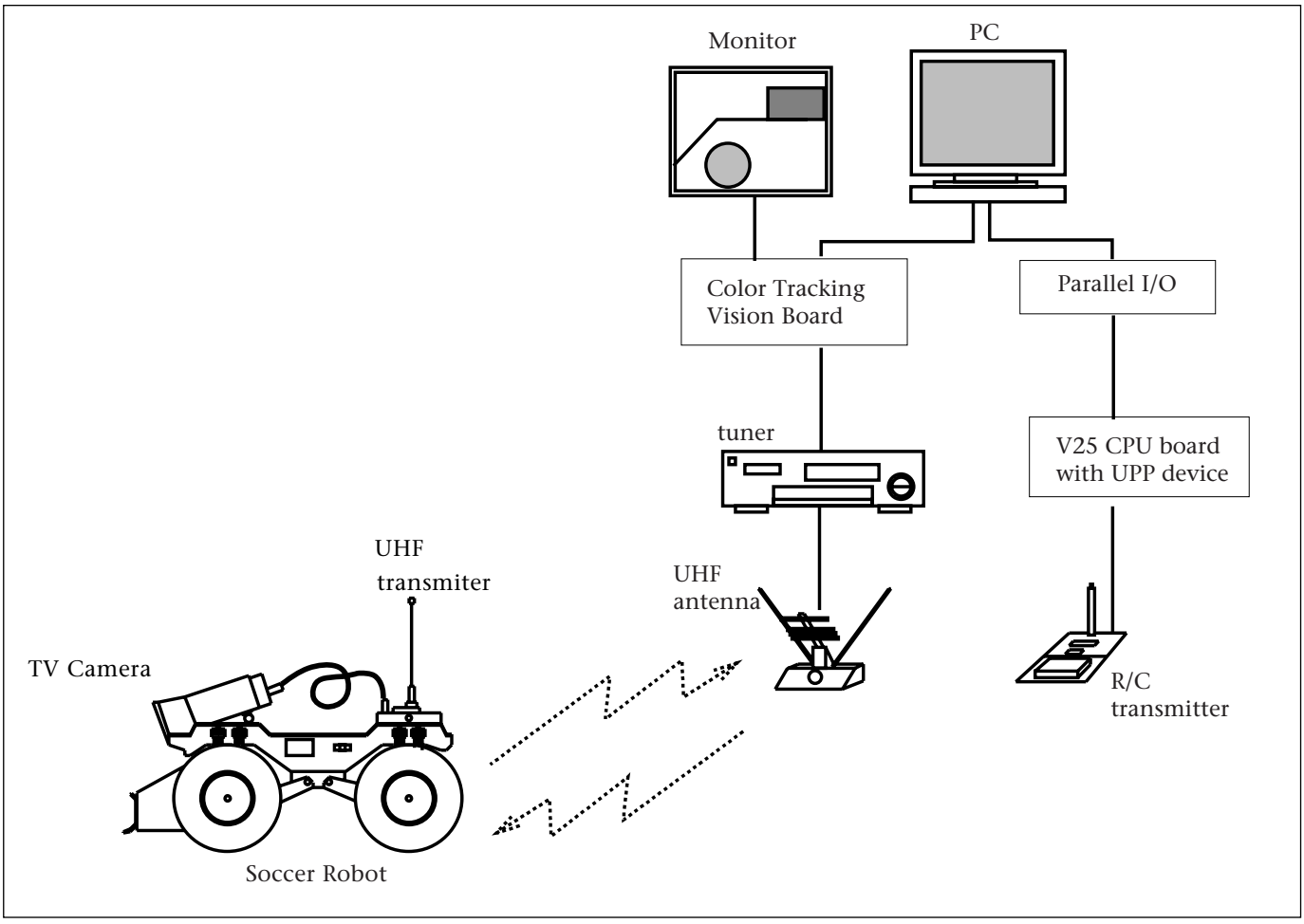

Figure 6. Configuration of Robot Controller.

California DREAMTEAM. RAIDERS has four omnidirectional moving robots controlled by one remote brain based on the global vision system set on the ceiling. During the game, the RAIDERS control system did not work well, and the robot motions seemed random. Fortunately, TRACKIES got a goal in the first period, and the final score was 1–0. Figure 7 shows a scene from this game.

The robot bodies used by DREAMTEAM were completely the same as TRACKIES, although the control methods of the two teams were different from each other. TRACKIES is based on the remote brain system, but DREAMTEAM is based on the self-contained autonomous system. The game score with DREAMTEAM was 2–2; all goals were made by DREAMTEAM (two goals and two own goals); TRACKIES could not get any goal.

TRACKIES and DREAMTEAM made it to the final game, which was also a draw (0–0). DREAMTEAM seemed to change its program approach from offensive to defensive because it had two own goals in the preliminary. Figure 8 shows robot views of four TRACKIES, including a goalie (top left), keeping the goal by blocking the red ball in front of the yellow goal. As we can see from the image, the omnidirectional view seems suitable for the goalie task.

During all the games, the radio situation in the competition site was bad, and TRACKIES could not work the remote brain system in both ways. Often, transmitted images were noisy, and sometimes there was no signal. In addition, the motor commands sent from the host computer could not correctly reach the robot body. Therefore, the robot motions sometimes seemed random and meaningless.

Figure 7. Osaka University Versus Royal Melbourne Institute of Technology.

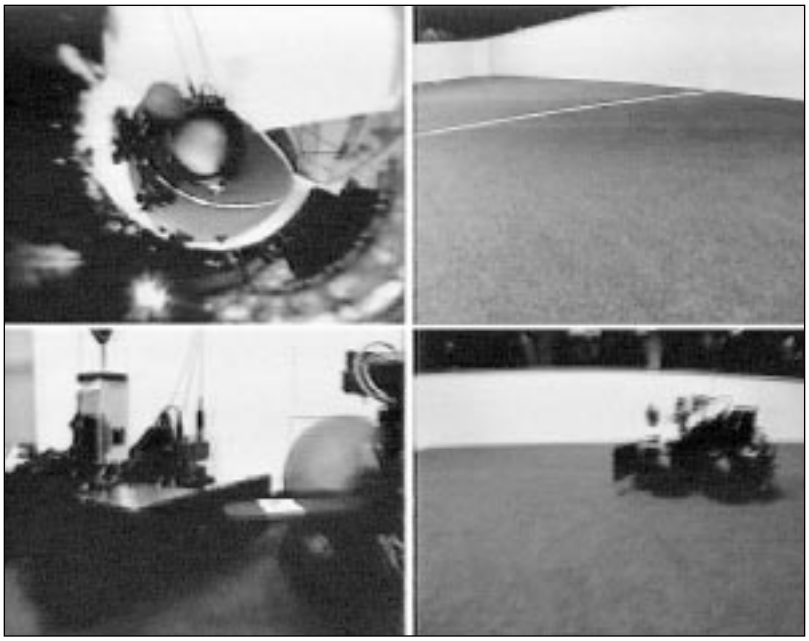

Figure 8. Four Robot Views from the TRACKIES Side.

Conclusions

At RoboCup-97, we did not implement any cooperated behaviors such as passing and shooting because of the lack of time. However, we have proposed a method (Uchibe, Asada, and Hosoda 1998). Also, the avoiding behavior only worked for RAIDERS because they wore yellow uniforms during the game. No other teams wore uniforms to be discriminated, including TRACKIES. The uniform should solve this discrimination problem in a clever way that is low cost and efficient.

Note

1. A policy f is a mapping from $\boldsymbol{S}$ to $\boldsymbol{A}$.

References

Asada, M.; Noda, S.; Tawaratsumida, S.; and Hosoda, K. 1996. Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning. *Machine Learning* 23:279–303.

Asada, M.; Uchibe, E.; Noda, S.; Tawaratsumida, S.; and Hosoda, K. 1994. Coordination of Multiple Behaviors Acquired by Vision-Based Reinforcement Learning. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 917–924. Washington, D.C.: IEEE Computer Society.

Bellman, R. 1957. *Dynamic Programming*. Princeton, N.J.: Princeton University Press.

Connel, J. H., and Mahadevan, S. 1993. *Robot Learning*. Norwell, Mass.: Kluwer Academic.

Kaelbling, L. P. 1993. Learning to Achieve Goals. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), 1094–1098. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Mataric, M. 1994. Reward Functions for Accelerated Learning. In *Proceedings of the Conference on Machine Learning,* 181–189. San Francisco, Calif.: Morgan Kaufmann.

Takahashi, Y.; Asada, M.; Noda, S.; and Hosoda, K. 1996. Sensor Space Segmentation for Mobile Robot Learning. Paper presented at the 1996 International Conference on Multiagent Systems Workshop on Learning, Interaction, and Organizations in a Multiagent Environment, 10 December, Kyoto, Japan.

Uchibe, E.; Asada, M.; and Hosoda, K. 1998. State-Space Construction for Behavior Acquisition in Multi-Agent Environments with Vision and Action. In Proceedings of the Sixth International Conference on Computer Vision (ICCV 98). Bombay, India: Narosa. Forthcoming.

Uchibe, E.; Asada, M.; and Hosoda, K. 1997. Vision-Based State-Space Construction for Learning Mobile Robots in Multi-Agent Environments. In Proceedings of the Sixth European Workshop on Learning Robots (EWLR-6), 33–41.

Uchibe, E.; Asada, M.; and Hosoda, K. 1996. Behavior Coordination for a Mobile Robot Using Modular Reinforcement Learning. In Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1329–1336. New York: IEEE Computer Society.

Watkins, C. J.; Watkins, C. H.; and Dayan, P. 1992. Technical Note: Q-Learning. *Machine Learning* 8:279–292.

Whitehead, S. D. 1991. A Complexity Analysis of Cooperative Mechanisms in Reinforcement Learning. In Proceedings of the Ninth National Conference on Artificial Intelligence, 607–613. Menlo Park, Calif.: American Association for Artificial Intelligence.

Whitehead, S. D., and Ballard, D. H. 1990. Active Perception and Reinforcement Learning. In Proceedings of the Workshop on Machine Learning 1990, 179–188. San Francisco, Calif.: Morgan Kaufmann.

Minoru Asada is a full professor in the Department of Adaptive Machine Systems in the Graduate School of Engineering at Osaka University. He is a chairperson of the RoboCup Japanese National Committee. His interests include cognitive robotics, artificial life, and robot vision.

Sho'ji Suzuki is a research associate in the Department of Adaptive Machine Systems in the Graduate School of Engineering at Osaka University. He has been developing vision-based mobile robots for RoboCup. His interests include mobile robot control and robot communication.

Yasutake Takahashi is a Ph.D. candidate in the Department of Computer-Controlled Machinery in the Graduate School of Engineering at Osaka University. He is interested in state-space construction for robot learning.

Eiji Uchibe is a Ph.D. candidate in the Department of Computer-Controlled Machinery in the Graduate School of Engineering at Osaka University. He is interested in learning cooperative behaviors based on vision and action.

Masateru Nakamura is currently with Mitsubishi Heavy Industry, Inc. He was a master course student in the Department of Computer-Controlled Machinery in the Graduate School of Engineering at Osaka University.

Chizuko Mishima is a master course student in the Department of Adaptive Machine Systems in the Graduate School of Engineering at Osaka University. She is interested in mechanical issues of mobile robots for RoboCup.

Hiroshi Ishiduka is a master course student in the Department of Adaptive Machine Systems in the Graduate School of Engineering at Osaka University. He is doing research on a mobile robot with an active camera head.

Tatsunori Kato is a master course student in the Department of Adaptive Machine Systems in the Graduate School of Engineering at Osaka University. He is doing research on an omnidirectional view on a mobile robot (goalee) for RoboCup.