Research Subject

Information Processing for Learning and Acquisition of Behaviors (Asada Group)

Our research group at Graduate School of Engineering, Osaka University, has been seeking for the methods of behavior learning to accomplish the given task. The approach is not task-specific but expectant of being meaningful from a viewpoint of information processing in our brain, and we have been attacking several kinds of issues by constructing a model and verifying it through real robot experiments. In this section, the following three issues are explained with implementation details and achievements:

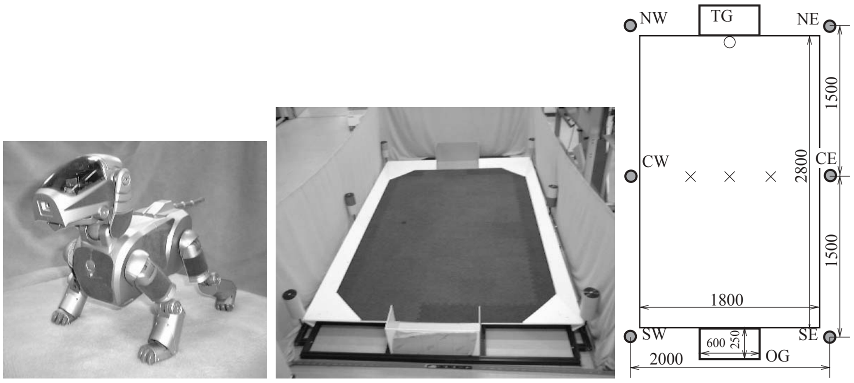
1. observation strategy learning for decision making of small quadruped based on information theory,

2. multi-layered learning systems for vision-based behavior acquisition of a real mobile robot, and

3. vision-based reinforcement learning for humanoid behavior generation with rhythmic walking parameters.

(1) Goal and summary

1. Observation strategy learning for decision making of small quadruped based on information theory Mobile robots are often equipped with vision sensors that provide huge amount of data, which demands methods to appropriately extract information for action decision such as attention control.



(a) the robot (b) the field (c) size of the field

Figure 1: Robot and the experimental field (same as the one for RoboCup 1999 SONY legged robot league). Cross and circle marks indicate the starting position and the ball position.

The aim of this research is to propose an efficient observation strategy for action decision of a small quadruped robot. We define the efficiency by the time used for observation

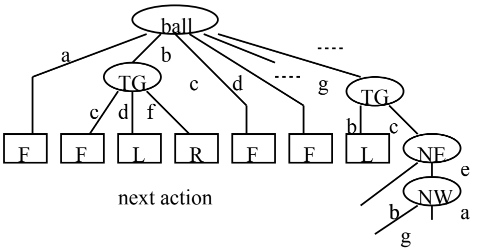


Figure 2: A part of the action decision tree. F, L, and R mean forward, left forward, and right forward respectively.

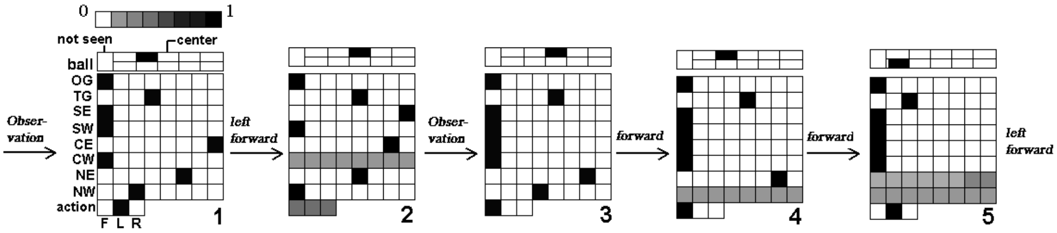


Figure 3: An example of the changes of the probability distribution.

to make a decision. We compare the contribution of the observation by the information gain. The observation strategy we propose is to do observations in the order of the information gain. First, we proposed a method which requires a robot to stand still to observe and make a decision. Then, we proposed an extension which enables observation during walking motion.

We used the information gain as a measure to determine which target (landmark) should be observed, and realized the efficient observation strategy for action selection [1]. To compress and memorize the training data in the manner of an action decision tree, we used a kind of classifier tree construction method by the information gain. A training data set is given to the robot with the direction to observe the landmark and the action to be taken just after the observation. By the occurrence probabilities of actions in the training data, it calculates the Shanon's information gain and constructs the action decision tree which instructs to observe landmarks in the decreasing order of the information gain.

Figure 1 shows a robot and its soccer field that we used in the experiments. A ball was put at the circle in front of the goal. The task was to put a ball into the goal starting from one of the three positions in the field (cross marks in the figure). Figure.2 shows a part of the action decision tree constructed by the method. This tree instructs the robot that when the ball is observed in the left direction ('b'), check the direction of the target goal ('c', 'd' or 'f') and decide an action. Figure 3 shows a sequence of the movements of the robot with action decision tree. It started from the center right position in the field and took the following actions: (1) turn left, (2) observe landmark, (3) move forward, (4) move forward, and then (5) turn left. We see that at (2), it could not determine action because of the ambiguity in the action probabilities. However at (4) and (5), one of the action probabilities was high and it could determine its action in spite of the ambiguity in observation for one or two landmarks. These show typical situations we intended, in

Table 1: Comparison of average number of gaze directions, and time to make a decision in the experiments.

	# of data	average gaze directions	average time[s]
pre-quantized	34	3.1	3.3
info. gain	43	3.5	3.3
info./time	35	1.3	0.85

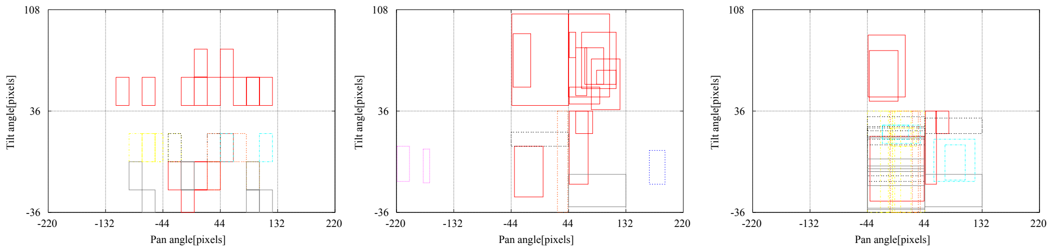
which observation is done not for self-positioning but for action decision.

Next, we extended the method so that it can handle not only discrete values but also continuous sensor values [2]. Generally, sensors equipped on a robot output continuous values. Then, we often quantized sensor spaces in advance to convert them to discrete values. However, there are a number of problems such as loosing the order inside the tessellated unit, the granularity of the tessellation, and so on. Therefore, autonomous quantization of the sensor values is suitable for decision making.

C4.5 is a well known method which autonomously quantizes continuous values during construction of a classifier tree. C4.5 divides a value with a threshold into two discrete values so that the information gain is maximized by the devision. However, to know a landmark's observed direction in divided space, one must observe several directions with a limited view angle camera. We proposed to use an attention window for a division. An attention window is a range of a continuous value in which landmark is observed. To know whether a landmark is in an attention window or not, one has to observe only one direction. Then, we proposed a method to construct an action decision tree by comparing the information gain to know whether a specific landmark is observed in an attention window. Quantizations of continuous values are done by attention windows during the tree construction. The constructed action decision tree instructs the landmark and the direction to observe. We also proposed to use the information gain per time in cases where the time to gaze depends on the gaze direction.

We show the experimental result with a task of navigation in a robot soccer situation. Table 1 shows the average gaze directions and the average observation time used for an action decision. The 'pre-quant' in the table indicates the case where we quantized sensor space in advance and the action decision tree was constructed by information gain. The 'info. gain' in the table indicates the case where quantizations were done by the proposed attention windows and the action decision tree was constructed by information gain. The 'info./time' in the table indicates the case where quantizations were done by the proposed attention windows and the action decision tree was constructed by information gain per time. We see that the average observation time drastically decreased with the use of the information gain per time. Figure 4 shows the created attention windows in each case. We see that attention windows of different sizes are created with needs and attention windows concentrates in the center direction with use of information gain per time, which reduce the observation time.

Next, we proposed the extension for observation during walking [3]. To realize more efficient observation strategy, observation during walking is important.



(a) pre-quantized (b) info. gain (c) info./time

Figure 4: Generated attention widows.

There are issues on observation and action decision during walking,

1) large sensor errors because of shakes by walking,

2) movements between the sensor capture and the end of vision sensor processing, movements among several gazes,

3) appropriately stop or slow down when there are too much ambiguity for action decision to avoid bumping and so on.

For the extension of our previous methods, we need, 1) a compensation method for shaking due to walking, 2) an integration method of observations from different positions and different gazes, and 3) a measure of ambiguity of action selection in order to appropriately stop walking.

For the problem 3) we propose to use the expected information gain in addition to action probabilities for the measure of ambiguity. When the expected information gain or action probabilities do not meet the thresholds, the robot stops and observes. Also we propose an image compensation mechanism for shakes by walking and movements between gazes and solve problems 2) and 3). Compensation values are calculated only from the image sequences. We calculate the observation probabilities at current position when the robot observes landmarks standing still by the observation during walking with the compensation values.

Figure 5 shows the shakes by walking and the result of the compensation. Figure 6 shows the sequence of observation and action decision in a navigation task by proposed method.

2. Multi-Layered Learning Systems for Vision-based Behavior Acquisition of A Real Mobile Robot In the machine learning area, several approaches have tried to make agents learn purposive behaviors autonomously to achieve their goals through agent-environment interactions. Especially, reinforcement learning has recently been receiving increased attention as a method for behavior learning with little or no a priori knowledge and higher capability of reactive and adaptive behaviors.

In order to realize an autonomous robot which acquires various behaviors by itself in real world, it is necessary to be able to manage a wide range of state and action variables

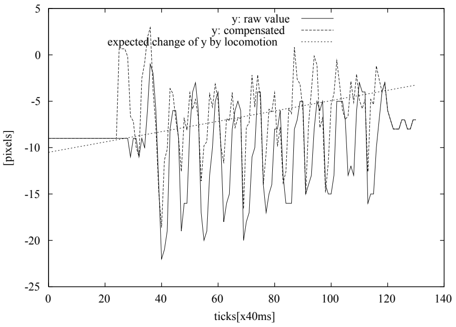


Figure 5: Image compensation to the y axis by proposed method.

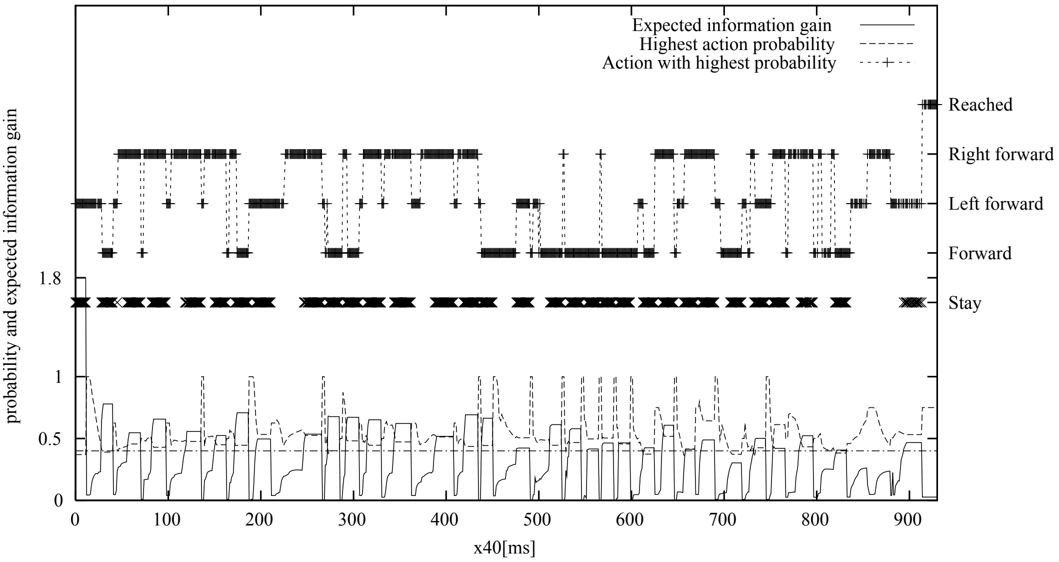


Figure 6: Changes of expected information gain, the highest action probability, and actions. One of which has the highest action probability, and the other is the one taken by the robot.

according to situations, to keep the spaces as small as possible, and to learn/control behaviors based on the small state and action spaces. It is almost impossible or impractical that robot acquires the various behaviors for the given tasks based on a huge monolithic state/action space which consists of all sensors' information and actuators commands, because the computational resources are limited and learning time is not eternity from a practical viewpoint.

Another approach to the problem of the curse of dimension and the perceptual aliasing is to adopt a hierarchical structure within leaning control system. That is, the system

1. prepares learning/control modules of one kind, each of which deals with a subspace divided from a whole state/action space,

2. abstracts situations and behaviors based on the acquired learning/control modules, and

3. acquires higher level, new behaviors based on the state and action spaces constructed from already abstracted situations and behaviors.

This approach can suppress the explosion of the state and action spaces since the higher level learning/control system manages adequately small size spaces which are abstracted in the lower levels.

We proposed a mechanism which constructs learning modules at higher layers using a number of groups of modules at lower layers. The modules in the lower networks are self-organized as experts to move into different categories of sensor value regions and learn lower level behaviors using motor commands. In the meantime, the modules in the higher networks are organized as experts which learn higher level behavior using lower modules. We applied the method to a simple soccer situation in the context of RoboCup[4], and showed the the validity of this method.

Figure 7 shows a picture of a mobile robot we designed and built, a ball, and a goal, and an overview of the robot system. It has two TV cameras. One has a wide-angle lens and is tilted down in front of the body in order to capture the ball image as large as possible. Other has a omni-directional mirror and is mounted on the robot. A simple color detection method is applied to detect objects around the robot on the images in real-time. The driving mechanism is PWS (Power Wheeled System), and the action space is constructed in terms of two torque values to be sent to two motors that drive two wheels. These parameters of the system are unknown to the robot, and it tries to estimate the mapping from sensory information to appropriate motor commands by the method. The environment consists of the ball, and the goals, and the mobile robot.

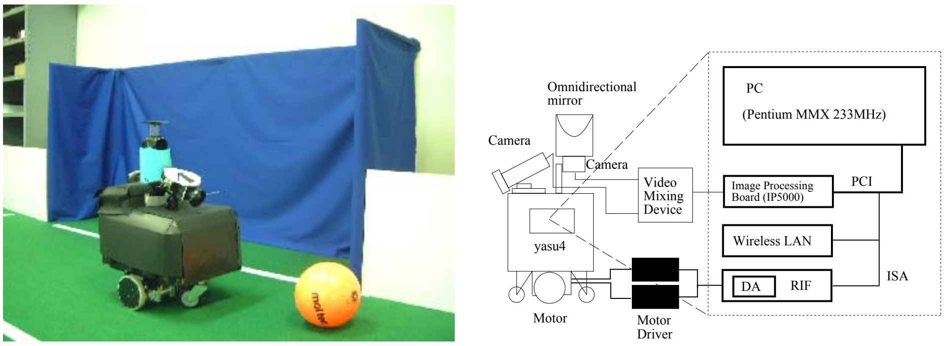


Figure 7: A mobile robot, a ball, and a goal (left), and an overview of the robot system (right)

The robot prepares learning modules of a kind, makes a layer with the modules, and constructs a hierarchy with the layers. The hierarchy of the learning modules' layers seems to play a role of task decomposition (Figure 8). The lower learning modules explore small areas, and learn lower level, fundamental behaviors. In contrast, the upper learning modules explore a large area, and learn higher level, more abstracted behaviors based on the learning modules at the lower layer [5].

The proposed architecture of the multi-layered reinforcement learning system is shown in Figure 9, in which (a) and (b) indicate a hierarchical architecture with two levels, and

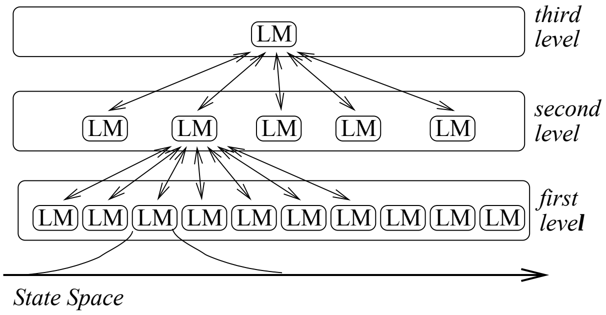
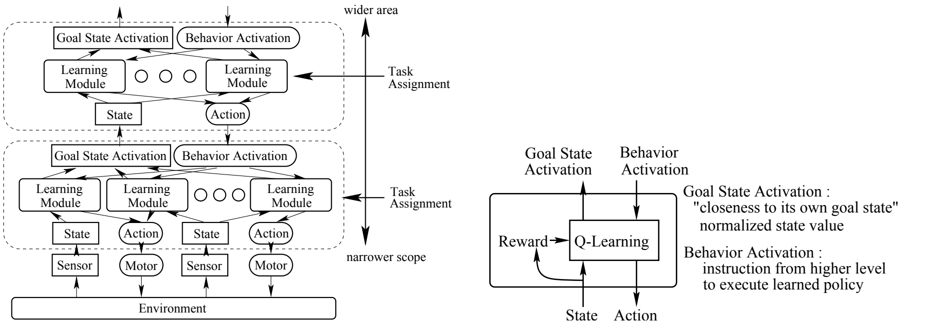


Figure 8: An overview of a hierarchical learning architecture : LM stands for learning module

individual learning module embedded in the layers. Each module has its own goal state in its state space, and it learns the behavior to reach its goal using Q-learning method. The state and the action are constructed using sensory information and motor command, respectively, at the bottom level.



(a) A whole system (b) A behavior learning module

Figure 9: A hierarchical learning architecture

This multi-layered learning system defines situations/behaviors based on the modules of lower layers, defines state and action spaces using them, and acquires new abstracted behaviors on them. The left side of figure 10 shows a rough sketch of this idea. The system assigns learning modules on the state space of a certain layer. Each learning module acquires the behavior to reach its own goal specified on the state space. The another layer on it regards a region assigned to a lower learning module as a situation, and a motion to a close region as a behavior.

The system learned and constructed the four layers and one learning module exist at the top layer (the left side of figure 10). The state space of the lower layer is constructed in terms of the centroid of the goal images, and the action space is constructed in terms of two torque values to be sent to two motors that drive two wheels. The state and action spaces at the upper layer are constructed by the learning modules at the lower layer which are automatically assigned.

The basic idea for the self-distribution of learning modules is “to assign the goal state of each learning module in the state space uniformly”. Because it seems difficult to define

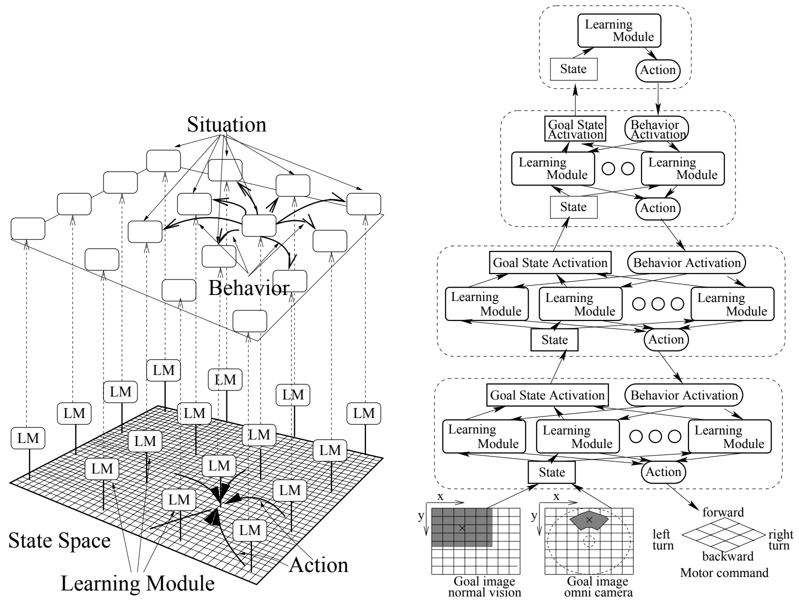


Figure 10: The concept of abstraction from learning modules to situations and behaviors (left), and the implemented hierarchy architecture of learning modules (right)

a distance function in the state space in advance, we use the state value function as the distance function that estimates the distance to its own goal state because we can regard that state value represents how close the robot is to the goal if the robot received reward only when it reach its goal. Figure 11 shows the distribution of goal state activations of learning modules at the bottom layer in the state spaces of wide-angle camera image (left) and omni-directional mirror image (right), respectively. The x, y axes indicate the centroid of goal images.

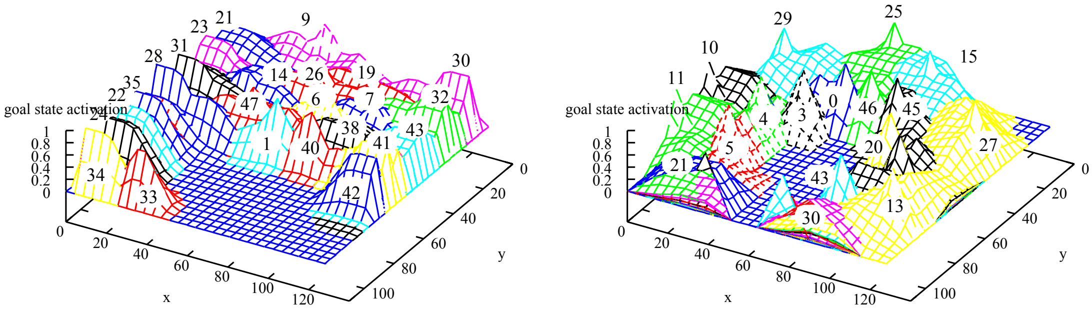


Figure 11: The distributions of learning modules at lower layer on the normal camera image (left) and the omni-directional camera image (right)

When the higher layer constructs its state-action space based on situations and behaviors acquired by the modules of several lower layers, it should consider that the layers are independent of each other, or there is dependence between them. The layer might be basically independent from each other when each layer's modules recognize a different object and learn behaviors for it. On the other hand, there might be dependence between the layers when modules on all layer recognize the same object in the environment and learn the behavior against it. For example, the system would regard that both layers are independent from each other if the modules on one layer acquire several navigation behaviors, and the modules on the other layer acquire object manipulation behaviors; in the case of robot in the RoboCup field, one layer's modules could be experts for ball handling and the other layer's modules ones for navigation on the field. On the other hand, it will recognize that there is a certain relationship between the layers when the system captures a number of data which represent one certain object with different sensor devices. In such a case, the system can recognize the situation complementarily using plural layers' outputs even if one layer loses the object on its own state spaces. Now, we proposed "a multiplicative approach" for the former, and "a complementary approach" for the latter [6], and implemented to the real robot (Figure 12). At the lowest level, there are four learning layers, and each of them deals with its own logical sensory space (ball positions on the perspective camera image and omni one, and goal position on both images). At the second level, there are three learning layers in which one adopts the multiplicative approach and the others adopt the complementary approach. The multiplicative approach of the "*ball pers.*×*goal pers.*" layer deals with lower modules of "*ball pers.*" and "*goal pers.*" layers. The arrows in the figure indicate the flows from the goal state activations to the state vectors. The arrows from the action vectors to behavior activations are eliminated.

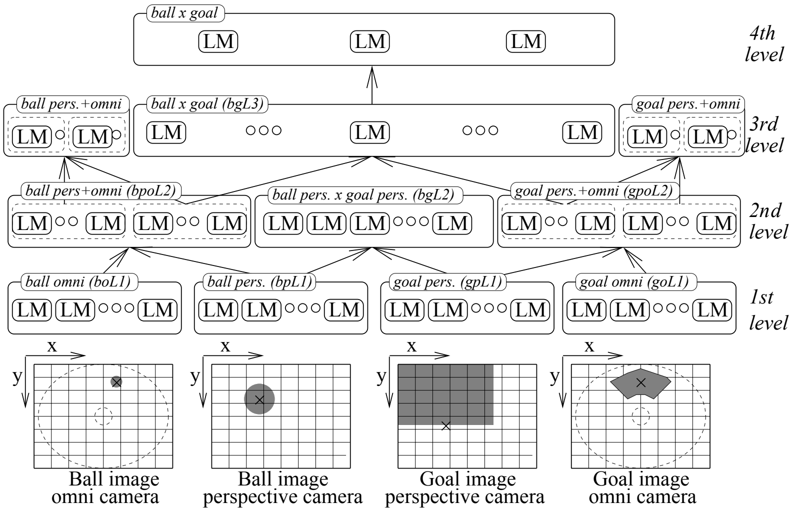


Figure 12: A hierarchy architecture of learning modules

We let our robot shoot a ball into the goal using this multi-layer learning structure. The target situation is given by reading the sensor information when the robot pushes the ball into the goal; the robot captures the ball and goal at center bottom in the perspective camera image. As an initial position, the robot is located far from the goal, faced opposite direction to it . The ball was located between the robot and the goal. The left side of Figure 14 shows the time development of the goal state and behavior activations of learning modules at all levels while the robot shoots the ball into the goal. The arrows on the top of each series indicate the behavior activations, and the others indicate the goal state

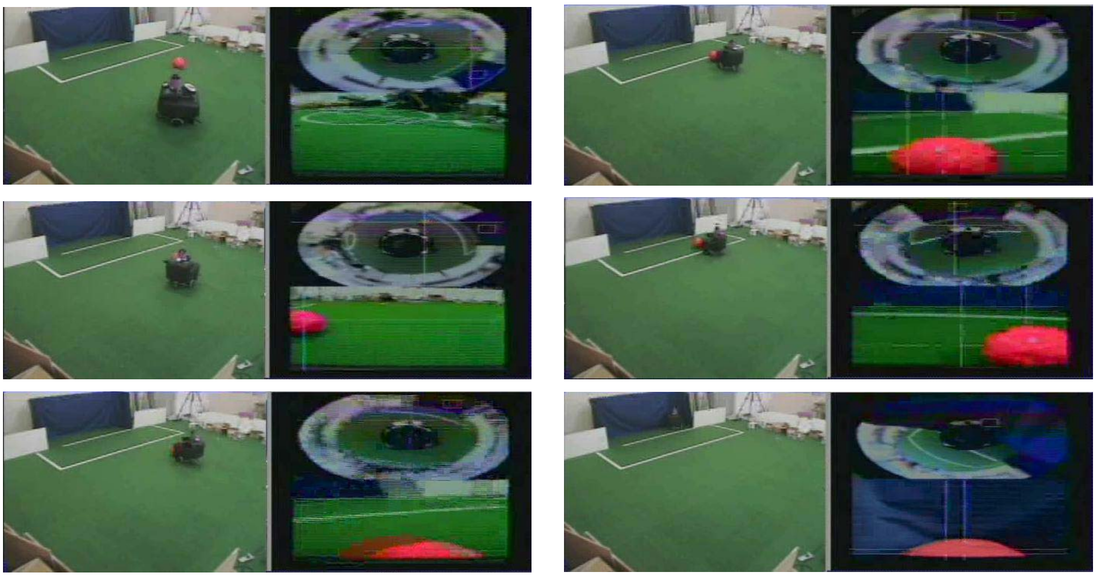
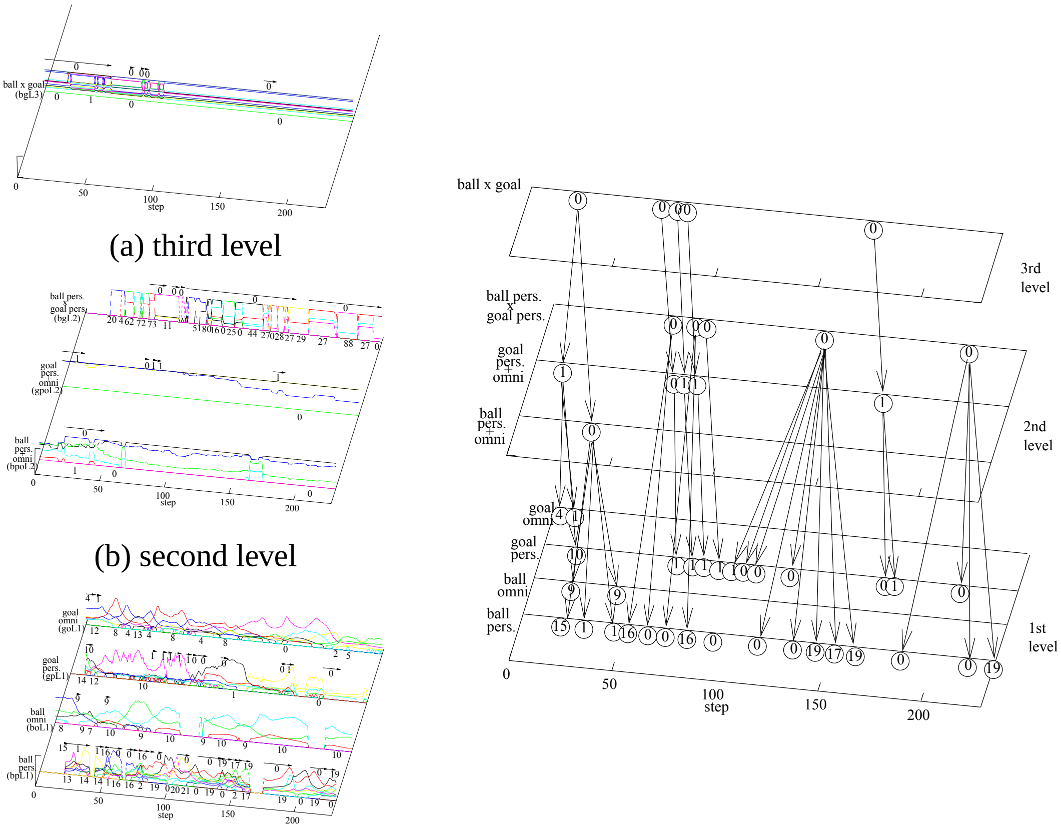


Figure 13: A sequence of a shooting behavior and its camera images

activation. The goal state activation g is a normalized state value V, and $g = 1$ when the situation is the goal state. When the module receives the behavior activation b from the higher level modules, it executes the optimal policy for its own goal, The right side of Figure 14 shows the sequence of the behavior activation of learning modules and the commands to the lower layer modules. The down arrows indicate that the higher learning modules fire the behavior activations of the lower learning modules.

3. Vision-based reinforcement learning for humanoid behavior generation with rhythmic walking parameters The research community for biped walking has been growing and various approaches have been introduced. Among them, there are two major trends in biped walking. One is a model based approach with ZMP (zero moment point) principle or the inverted pendulum model. The other one is inspired by the findings in neurophysiology that most animals generate their walking motions based on the central pattern generator (hereafter, CPG) or neural oscillator. This sort of approach does not need model parameters that are as precise as ZMP or the inverted pendulum, which might show the robustness against the environmental changes. In order to increase the adaptability, the external information such as vision is used, but often 3-D accurate reconstruction seems necessary.

This section presents a method for generating vision-based humanoid behaviors by reinforcement learning with rhythmic walking parameters [7]. A rhythmic motion controller such as CPG or neural oscillator stabilizes the walking. The learning process consists of building an action space with two parameters (a forward step width and a turning angle) so that infeasible combinations are inhibited, and reinforcement learning with the constructed action space and the state space consisting of visual features and posture parameters to find a feasible action. The method is applied to a situation from the Humanoid RoboCupSoccer league in RoboCup, that is, to approach the ball and to shoot it into the goal. Instructions by human are given to start up the learning process, and the rest is



(a) third level

(b) second level

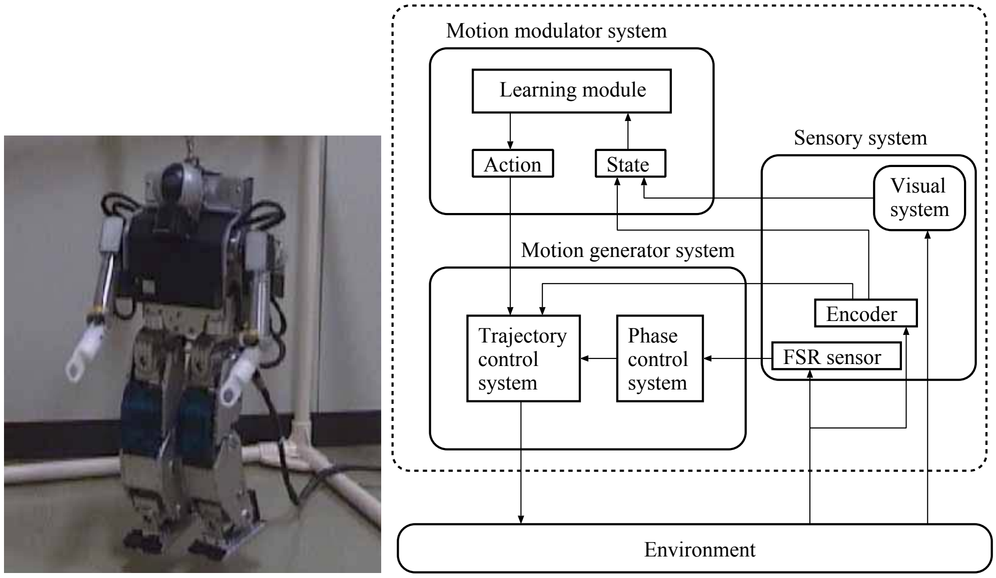
(c) first level

Figure 14: A sequence of the goal state activation and behavior activation of learning modules (left) and a sequence of the behavior activation of learning modules and the commands to the lower layer modules (right)

solely self-learning in real situations.

Figure 15 (a) and (b) show a humanoid platform Fujitsu HOAP-1 (the height is around 50cm) and an overview of the proposed system, respectively. We applied a CPG-based rhythmic walking developed by Prof. Tsuchiya's group (Kyoto University) in the RoboBrain project, which consists of trajectory control and phase shift control. The latter is triggered by the signals from FSR sensors attached on the soles.

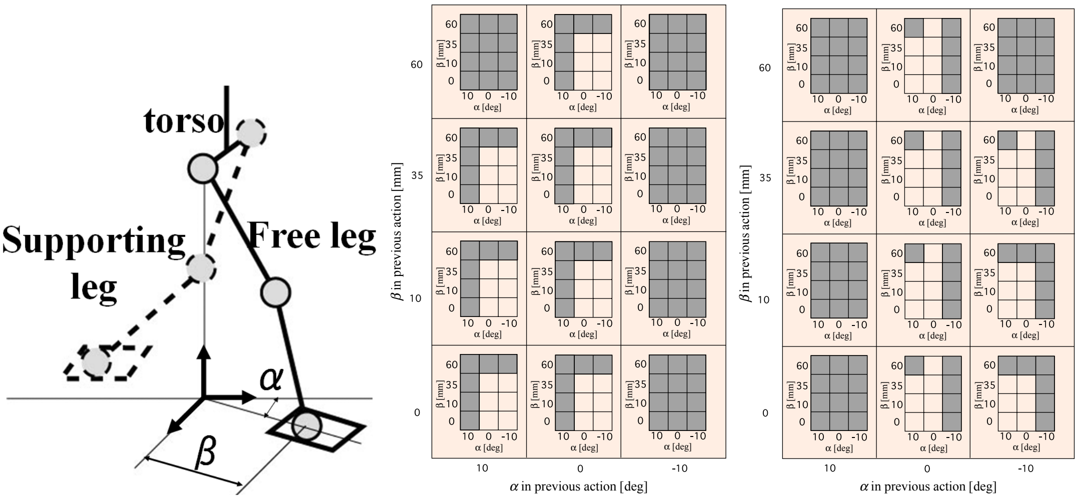
A reinforcement learning module above the motion generation module learns to determine the rhythmic walking parameters: turning angle α and step width β (see Figure 16(a)). These parameters depends on the posture before action execution, that is, feasible and infeasible ones. The latter means tumbling over that should be avoided in cases of real robot learning to minimize the damages to the robot hardware. Then, the feasible walking parameters are found by a trial and error technique. The turning angle α and step width β are quantized into three (-10, 0, and 10 degrees) and four (0, 10, 25, and 60mm), totally 12 selections. Depending on the walking parameters selected just before the selection, feasible combination of walking parameters are found. Figure 16 (b) and (c) show the results, in which the vertical and horizontal axes indicate the step width and turning angle of previously selected parameters, respectively. White boxes indicate feasible parameters



(a) a humanoid platform Fujitsu HOAP-1

(b) an overview of the proposed system

Figure 15: A humanoid and the proposed system



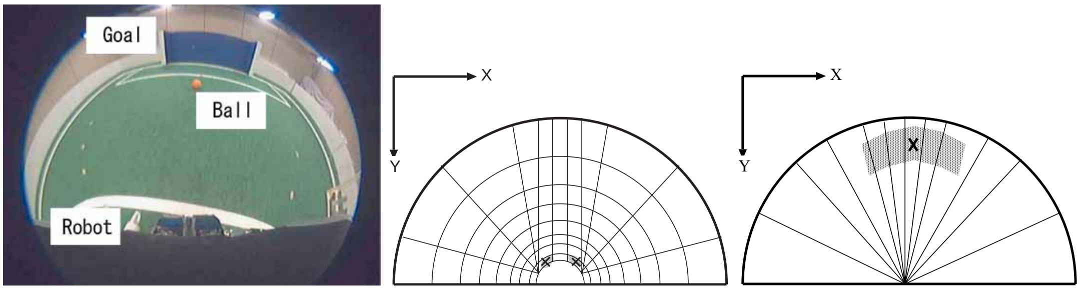
(a) rhythmic walking parameters (turning angle α and step width β)

(b) left leg

(c) right leg

Figure 16: rhythmic walking parameters and feasible waling parameters

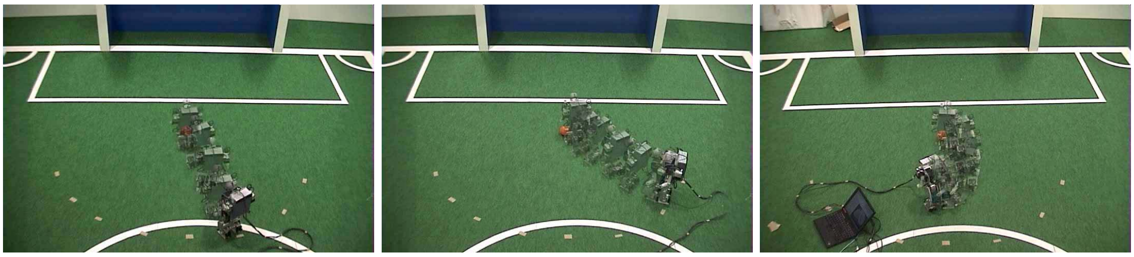
and gray boxes infeasible ones. Although the result should be symmetric, asymmetry appears due to the difference of physical properties between left and right parts of the platform.



(a) an on-board image (b) a ball state space (c) a goal state space

Figure 17: An on-board image and the state space

As an example task, a sort of situation from RoboCupSoccer Humanoid league is supposed. That is, to approach to the ball in front of the goal and to kick the ball into the goal. In addition to the visual information, the image of the ball and the goal, posture parameters (here, the walking parameters just before the action selection correspond to them) are prepared to construct the state space. Figure 17 (a) shows an on-board image from the camera within a fish-eye lens to capture the range of the field as wide as possible. Color information (the red ball and the blue goal) helps the object detection. Figure 17 (b) and (c) show the ball and the goal state space, respectively, where space quantization in terms of direction (and distance) are carried out. Grey regions with cross marks indicate the state where the reward 1 is give. Otherwise, zero rewards are given everywhere.



(a) a slightly oblique forward motion (b) an approach from side (c) a case where neither ball nor goal is observed from the initial position

Figure 18: Experimental results

One of the most serious issues in applying the reinforcement learning method to real robot tasks is how to accelerate the learning process. To do that, we give the instructions to start up the learning: during the first 50 episodes (about a half hour), the human instructor avoids useless exploration by directly specifying the action command to the learner about 10 times per episode. After that, the learner experienced about 1500 episodes. Owing to the initial instructions, learning converged in 15 hours, and the robot learned to get the right position from any initial positions inside the half field. Figure 18 shows three examples. (a) indicates a slightly oblique forward motion, and (b) an approach from side from where the ball and the goal image are far away, and therefore distant from the rewards. In both cases, we can see that the robot adjusted its step width as it approaches to the ball. (c) shows a case in which the robot can observe neither ball nor goal from the initial position. Then, it turned, found the ball and the goal, and approached to the ball.

(2) Results and their importance

1. Observation strategy learning for decision making of small quadruped based on information theory Our aim was to propose and validate methods to realize efficient observation for a mobile walking robot equipped with a limited view angles camera. We have proposed a observation strategy that select an observation target by the information gain, a observation strategy that select the attention window and the target, and an observation strategy during walking. For each proposed method we have validated that it can realize an efficient observation strategy with the experiments using a legged robot.

2. Multi-layered learning systems for vision-based behavior acquisition of a real mobile robot We proposed a mechanism which constructs learning modules at higher layers using a number of groups of modules at lower layers. We applied the method to a simple soccer situation in the context of RoboCup[4], showed the experimental results. We also proposed methods to merge state spaces at higher level while the layers at the lower level assigned to the subspaces, and applied to the real robot.

We proposed a simple mechanism of self-organization of hierarchical structure. However, we still need a mechanism which enables the system to select layers to be combined, to judge which approach is suitable, in order to develop various kinds of purposive behaviors. Further, the current method has focused on the state space hierarchy, but the idea of hierarchy construction seems applicable to the action space hierarchy, too. These are our future work.

3. Vision-based reinforcement learning for humanoid behavior generation with rhythmic walking parameters Vision-based humanoid behavior was generated by reinforcement learning with rhythmic walking parameters. Since the humanoid generally has many DoFs, it is very hard to control all of them. Instead of using these DoFs in the action space, we adopted rhythmic walking parameters, which drastically reduces the search space and, therefore, real robot learning was possible in a reasonable time. As a method of humanoid behavior generation adapting to external situations, the proposed architecture would be useful to be applied to various kinds of tasks. State space construction by learning is one of the future issues.

References

[1] Noriaki MITSUNAGA and Minoru ASADA: Visual attention control for a legged mobile robot based on information criterion, Proc. of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 244–249, 2002.

[2] Noriaki MITSUNAGA and Minoru ASADA: Sensor Space Segmentation for Visual Attention Control of a Mobile Robot based on Information Criterion, Proc. of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1714-1719, 2001.

[3] Noriaki MITSUNAGA and Minoru ASADA: Observation strategy for decision making based on information criterion, Proc. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1038-1043, 2000.

[4] M. Asada and H. Kitano and I. Noda and M. Veloso: RoboCup: Today and tomorrow – What we have learned, Artificial Intelligence, Vol.110, pp.193–214, 1999.

[5] Y. Takahashi and M. Asada: Vision-Guided Behavior Acquisition of a Mobile Robot by Multi-Layered Reinforcement Learning, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol.1, pp.395–402, 2000.

[6] Y. Takahashi and M. Asada: Multi-Controller Fusion in Multi-Layered Reinforcement Learning, Proc. of International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp.7–12, 2001.

[7] M. Ogino, Y. Katoh, M. Aono, M. Asada and K. Hosoda: Reinforcement learning of humanoid rhythmic walking parameters based on visual information, Proc. of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003 (to appear).