State Space Construction for Behavior Acquisition in Multi Agent Environments with Vision and Action

Eiji Uchibe Minoru Asada Koh Hosoda
Dept. of Adaptive Machine Systems
Osaka University
Suita, Osaka 565, Japan

Abstract

This paper proposes a method which estimates the relationships between learner's behaviors and other agents' ones in the environment through interactions (observation and action) using the method of system identification. In order to identify the model of each agent, Akaike's Information Criterion is applied to the results of Canonical Variate Analysis for the relationship between the observed data in terms of action and future observation. Next, reinforcement learning based on the estimated state vectors is performed to obtain the optimal behavior. The proposed method is applied to a soccer playing situation, where a rolling ball and other moving agents are well modeled and the learner's behaviors are successfully acquired by the method. Computer simulations and real experiments are shown and a discussion is given.

1 Introduction

Building a robot that learns to accomplish a task through visual information has been acknowledged as one of the major challenges facing vision, robotics, and AI. In such an agent, vision and action are tightly coupled and inseparable [2]. For instance, we, human beings, cannot see anything without the eye movements, which may suggest that actions significantly affect the vision processes and vice versa. There have been several approaches which attempt to build an autonomous agent based on tight coupling of vision (and/or other sensors) and actions [15, 13, 14]. They consider that vision is not an isolated process but a component of the complicated system (physical agent) which interacts with its environment [3, 16, 8]. This is a quite different view from the conventional CV approaches that have not been paying attention to physical bodies. A typical example is the problem of segmentation which has been one of the most difficult problems in computer vision because of the historical lack of the criterion: how significant and useful the segmentation results are. These issues would be difficult to be evaluated without any purposes. That is, instinctively *task oriented*. However, the problem is not the straightforward design issue for the special purposes, but the approach based on physical agents capable of sensing and acting. That is, segmentation and its organization corresponds to the problem of building the agent's internal representation through the interactions between the agent and its environment.

The internal representation can be regarded as state vectors from a viewpoint of control theory because they include the necessary and sufficient information to accomplish a given task, and also as state space representation in robot learning for the same reason as in the control theory, especially, in reinforcement learning which has recently been receiving increased attention as a method with little or no a priori knowledge and higher capability of reactive and adaptive behaviors [7].

There have been a few work on the reinforcement learning with vision and action. To the best of our knowledge, Whitehead and Ballard proposed an active vision system [19] in which only a computer simulation has been done. Asada et al. [6] applied the vision-based reinforcement learning to the real robot task. In these methods, the environment does not include independently moving agents, therefore, the complexity of the environment is not so high as one including other agents. In case of multi-robot environment, the internal representation would become more complex to accomplish the given tasks [4]. The main reason is that the other robot has perception (sensation) different from the learning robot's. This means that the learning robot would not be able to discriminate different situations which the other robot can do, and vice versa. Therefore, the learner cannot predict the other robot behaviors correctly even if its policy is fixed unless explicit communication is available. It is important for the learner to understand the strategies of the other robots and to predict their movements in advance to learn the behaviors successfully.

There are several approaches to multiagent learning problem (ex., [11], [17]) which utilize the current sensor outputs as states, and therefore they can not cope with the changes of the current situation. Further, they need well-defined attributes (state vectors) in order for the learning to converge correctly. However, it is generally difficult to find such attributes in advance. Therefore, the modeling architecture (state vector estimation) is required to enable the reinforcement learning applicable.

In this paper, we propose a method which finds the relationships between the behaviors of the learner and the other agents through interactions (observation and action) using the method of system identification. In order to construct the local predictive model of other agents, we apply Akaike's Information Criterion(AIC) [1] to the results of Canonical Variate Analysis(CVA) [10], which is widely used in the field of system identification. The local predictive model is based on the observation and action of the learner (observer). We apply the proposed method to a simple soccer-like game. The task of the robot is to shoot a ball which is passed back from the other robot. Because the environment consists of the stationary agents (the goal), a passive agent (the ball) and an active agent (the opponent), the learner has to construct the appropriate models for all of these agents. After the learning robot identifies the model, the reinforcement learning is applied in order to acquire purposive behaviors. The proposed method can cope with a moving ball because the state vector is estimated appropriately to predict its motion in image. Simulation results and real experiments are shown and a discussion is given.

2 Construct the internal model from observation and action

2.1 Local predictive model of other agents

In order to make the learning successful, it is necessary for the learning agent to estimate appropriate state vectors. However, the agent can not obtain the complete information to estimate them because of the partial observation due to the limitation of its sensing capability. Then, what the learning agent can do is to collect all the observed data with the motor commands taken during the observation and to find the relationship between the observed agents and the learner's behaviors in order to take an adequate behavior although it might not be guaranteed as optimal. In the following, we consider to utilize a method of system identification, regarding the previous observed data and the motor commands as the input, and future observation as the output of the system respectively.

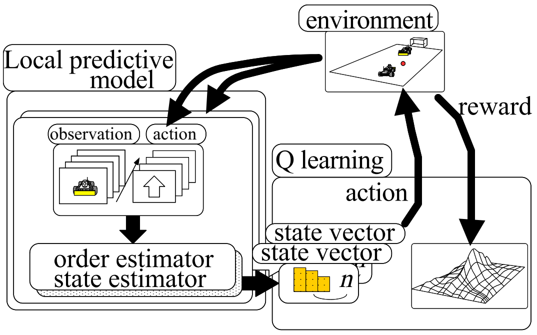


Figure 1: An overview of the learning architecture

Figure 1 shows an overview of the proposed learning architecture consisting of an estimator of the local predictive models and a reinforcement learning method. At first, the learning agent collects the sequence of sensor outputs and motor commands to construct the local predictive models, which are described in section 2.2. By approximating the relationships between the learner's action and the resultant observation, the local predictive model gives the learning agent not only the successive state of the agent but also the priority of the state vectors, which means that the validity of the state vector with respect to the prediction.

2.2 Canonical Variate Analysis(CVA)

A number of algorithms to identify multi-input multi-output (MIMO) combined deterministic-stochastic systems have been proposed. Among them, Larimore's Canonical Variate Analysis (CVA) [10] is one representative, which uses canonical correlation analysis to construct a state estimator.

Let $\boldsymbol{u}(t) \in \Re^m$ and $\boldsymbol{y}(t) \in \Re^q$ be the input and output generated by the unknown system

$$\begin{aligned} \boldsymbol{x}(t+1) &= \boldsymbol{Ax}(t) + \boldsymbol{Bu}(t) + \boldsymbol{w}(t), \\ \boldsymbol{y}(t) &= \boldsymbol{Cx}(t) + \boldsymbol{Du}(t) + \boldsymbol{v}(t), \end{aligned} \qquad (1)$$

with

$$E\left\{ \begin{bmatrix} \boldsymbol{w}(t) \\ \boldsymbol{v}(t) \end{bmatrix} \begin{bmatrix} \boldsymbol{w}^T(\tau) & \boldsymbol{v}^T(\tau) \end{bmatrix} \right\} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & \boldsymbol{R} \end{bmatrix} \delta_{t\tau},$$

and $\boldsymbol{A}, \boldsymbol{Q} \in \Re^{n\times n}$, $\boldsymbol{B} \in \Re^{n\times m}$, $\boldsymbol{C} \in \Re^{q\times n}$, $\boldsymbol{D} \in \Re^{q\times m}$, $\boldsymbol{S} \in \Re^{n\times q}$, $\boldsymbol{R} \in \Re^{q\times q}$. $E\{\cdot\}$ denotes the expected value operator and $\delta_{t\tau}$ the Kronecker delta. $\boldsymbol{v}(t) \in \Re^q$ and $\boldsymbol{w}(t) \in \Re^n$ are unobserved, Gaussian-distributed, zero-mean, white noise vector sequences. CVA uses a new vector $\boldsymbol{\mu}$ which is a linear combination of the previous input-output sequences since it is difficult to determine the dimension of $\boldsymbol{x}$. Eq.(1) is transformed as follows:

$$\begin{bmatrix} \boldsymbol{\mu}(t+1) \\ \boldsymbol{y}(t) \end{bmatrix} = \boldsymbol{\Theta} \begin{bmatrix} \boldsymbol{\mu}(t) \\ \boldsymbol{u}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{T}^{-1}\boldsymbol{w}(t) \\ \boldsymbol{v}(t), \end{bmatrix}, \quad (2)$$

where

$$\hat{\boldsymbol{\Theta}} = \begin{bmatrix} \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} & \boldsymbol{T}^{-1}\boldsymbol{B} \\ \boldsymbol{C}\boldsymbol{T} & \boldsymbol{D} \end{bmatrix}, \quad (3)$$

and $\boldsymbol{x}(t) = \boldsymbol{T}\boldsymbol{\mu}(t)$. We follow the simple explanation of the CVA method.

1. For $\{\boldsymbol{u}(t), \boldsymbol{y}(t)\}$, $t = 1, \cdots N$, construct new vectors

$$\boldsymbol{p}(t) = \begin{bmatrix} \boldsymbol{u}(t-1) \\ \vdots \\ \boldsymbol{u}(t-l) \\ \boldsymbol{y}(t-1) \\ \vdots \\ \boldsymbol{y}(t-l) \end{bmatrix}, \quad \boldsymbol{f}(t) = \begin{bmatrix} \boldsymbol{y}(t) \\ \boldsymbol{y}(t+1) \\ \vdots \\ \boldsymbol{y}(t+k-1) \end{bmatrix},$$

2. Compute estimated covariance matrices $\hat{\boldsymbol{\Sigma}}_{pp}$, $\hat{\boldsymbol{\Sigma}}_{pf}$ and $\hat{\boldsymbol{\Sigma}}_{ff}$, where $\hat{\boldsymbol{\Sigma}}_{pp}$ and $\hat{\boldsymbol{\Sigma}}_{ff}$ are regular matrices.

3. Compute singular value decomposition

$$\hat{\boldsymbol{\Sigma}}_{pp}^{-1/2}\hat{\boldsymbol{\Sigma}}_{pf}\hat{\boldsymbol{\Sigma}}_{ff}^{-1/2} = \boldsymbol{U}_{aux}\boldsymbol{S}_{aux}\boldsymbol{V}_{aux}^{T}, \quad (4)$$
$$\boldsymbol{U}_{aux}\boldsymbol{U}_{aux}^{T} = \boldsymbol{I}_{l(m+q)}, \; \boldsymbol{V}_{aux}\boldsymbol{V}_{aux}^{T} = \boldsymbol{I}_{kq},$$

and $\boldsymbol{U}$ is defined as:

$$\boldsymbol{U} := \boldsymbol{U}_{aux}^{T}\hat{\boldsymbol{\Sigma}}_{pp}^{-1/2}.$$

4. The n dimensional new vector $\boldsymbol{\mu}(t)$ is defined as:

$$\boldsymbol{\mu}(t) = [\boldsymbol{I}_n \;\; 0]\boldsymbol{U}\boldsymbol{p}(t), \quad (5)$$

5. Estimate the parameter matrix $\boldsymbol{\Theta}$ applying least square method to Eq (2).

Strictly speaking, all the agents do in fact interact with each other, therefore the learning agent should construct the local predictive model taking these interactions into account. However, it is intractable to collect the adequate input-output sequences and estimate the proper model because the dimension of state vector increases drastically. Therefore, the learning (observing) agent applies the CVA method to each (observed) agent separately.

2.3 Determine the dimension of other agent

It is important to decide the dimension n of the state vector $\boldsymbol{x}$ and lag operator l that tells how long the historical information is related in determining the size of the state vector when we apply CVA to the classification of agents. Although the estimation is improved if l is larger and larger, much more historical information is necessary. However, it is desirable that l is as small as possible with respect to the memory size. For n, complex behaviors of other agents can be captured by choosing the order n high enough.

In order to determine n, we apply Akaike's Information Criterion (AIC) which is widely used in the field of time series analysis. AIC is a method for balancing precision and computation (the number of parameters). Let the prediction error be $\boldsymbol{\varepsilon}$ and covariance matrix of error be

$$\hat{\boldsymbol{R}} = \frac{1}{N-k-l+1}\sum_{t=l+1}^{N-k+1}\boldsymbol{\varepsilon}(t)\boldsymbol{\varepsilon}^{T}(t).$$

Then $AIC(n)$ is calculated by

$$AIC(n) = (N-k-l+1)\log|\hat{\boldsymbol{R}}| + 2\lambda(n), \quad (6)$$

where λ is the number of the parameters. The optimal dimension n^* is defined as

$$n^* = \arg\min AIC(n).$$

Since the reinforcement learning algorithm is applied to the result of the estimated state vector to cope with the non-linearity and the error of modeling, the learning agent does not have to construct the *strict* local predict model. However, the parameter l is not under the influence of the $AIC(n)$. Therefore, we utilize $\log|\hat{\boldsymbol{R}}|$ to determine l.

1. Memorize the q dimensional vector $\boldsymbol{y}(t)$ about the agent and m dimensional vector $\boldsymbol{u}(t)$ as a motor command.

2. From $l = 1\cdots$, identify the obtained data.

   (a) If $\log|\hat{\boldsymbol{R}}| < 0$, stop the procedure and determine n based on $AIC(n)$,

   (b) else, increment l until the condition (a) is satisfied or $AIC(n)$ does not decrease.

3 Reinforcement Learning

After estimating the state space model given by Eq. 2, the agent begins to learn behaviors using a reinforcement learning method. Q learning [18] is a form

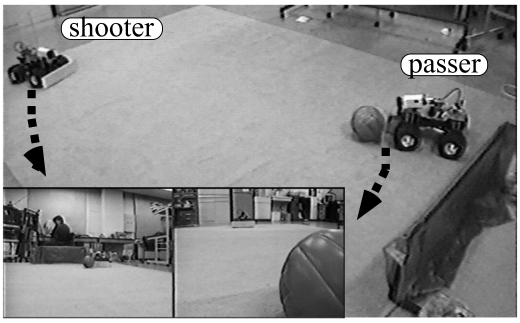
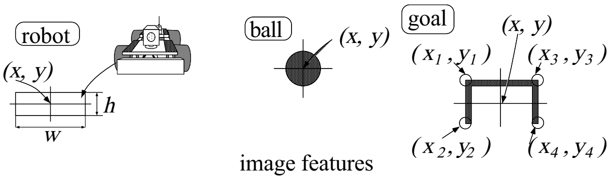


Figure 2: The environment and our mobile robot



robot	ball	goal
area	area	area
center position	center position	center position
height	radius	4 corners
width		

Figure 3: Image features of the ball, goal, and agent

of reinforcement learning based on stochastic dynamic programming. It provides robots with the capability of learning to act optimally in a Markovian environment. In the previous section, appropriate dimension n of the state vector $\boldsymbol{\mu}(t)$ is determined, and the successive state is predicted. Therefore, we can regard an environment as Markovian.

4 Task and Assumptions

We apply the proposed method to a simple soccer-like game including two agents (Figure 2). Each agent has a single color TV camera and does not know the location, the size and the weight of the ball, the other agent, any camera parameters such as focal length and tilt angle, or kinematics/dynamics of itself. They move around using a 4-wheel steering system. As motor commands, each agent has 7 actions such as go straight, turn right, turn left, stop, and go backward. Then, the input $\boldsymbol{u}$ is defined as the 2 dimensional vector as

$$\boldsymbol{u}^T = [v \;\; \phi] \,, \quad v, \phi \in \{-1, 0, 1\},$$

where v and ϕ are the velocity of motor and the angle of steering respectively and both of which are quantized.

The output (observed) vectors are shown in Figure 3. As a result, the dimension of the observed vector about the ball, the goal, and the other robot are 4, 11, and 5 respectively.

5 Experimental Results

5.1 Simulation Results

Table1 shows the result of identification. In order to predict the successive situation, $l = 1$ is sufficient for the goal, while the ball needs 2 steps. The motion of the random walk agent can not be correctly predicted as a matter of course while the move-to-the-ball agent can be identified by the same dimension of the random agent, but the prediction error is much smaller than that of random walk.

Table 1: The estimated dimension (computer simulation)

agent	l	n	$\log\lvert\boldsymbol{R}\rvert$	AIC
goal	1	2	−0.001	121
ball	2	4	0.232	138
random walk	3	6	1.22	232
move to the ball	3	6	−0.463	79

Table 2 shows the success rates of shooting and passing behaviors compared with the results in our previous work [6] in which only the current sensor information is used as a state vector. We assign a reward value 1 when the robot achieved the task, or 0 otherwise. If the learning agent uses the only current information about the ball and the goal, the leaning agent can not acquire the optimal behavior when the ball is rolling. In other words, the action value function does not become to be stable because the state and action spaces are not consistent with each other.

Table 2: Comparison between the proposed method and using current information

state vector	success of shooting (%)	success of passing (%)
current position	10.2	9.8
using CVA	78.5	53.2

5.2 Real Experiments

We have constructed the radio control system of the robot, following the remote-brain project by Ina-

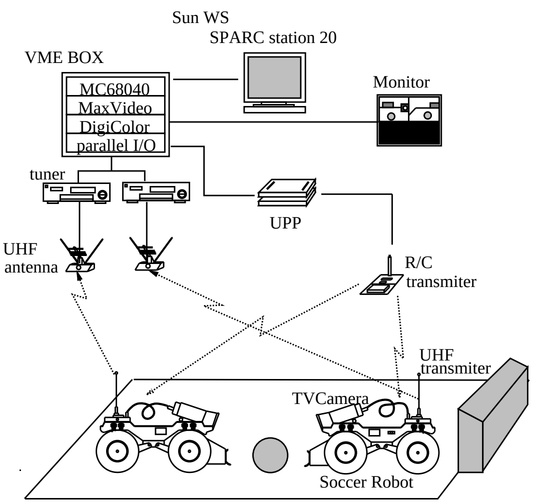


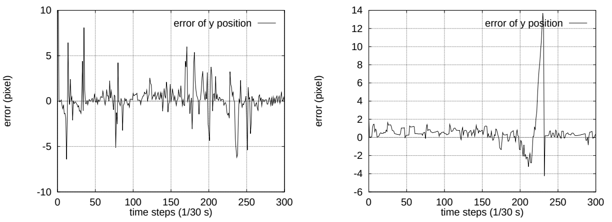
Figure 4: A configuration of the real system.

Table 3: The estimated dimension (real environment)

from the shooter				
	l	n	$\log\|\boldsymbol{R}\|$	AIC
ball	4	4	1.88	284
goal	1	3	-1.73	-817
passer	5	4	3.43	329

from the passer				
	l	n	$\log\|\boldsymbol{R}\|$	AIC
ball	4	4	1.36	173
shooter	5	4	2.17	284

ba et al. [9]. Figure 4 shows a configuration of the real mobile robot system. The image taken by a TV camera mounted on the robot is transmitted to a UHF receiver and processed by Datacube MaxVideo 200, a real-time pipeline video image processor. In order to simplify and speed up the image processing time, we painted the ball, the goal, and the opponent red, blue, and yellow, respectively. The input NTSC color video signal is first converted into HSV color components in order to make the extraction of the objects easy. The image processing and the vehicle control system are operated by VxWorks OS on MC68040 CPU which are connected with host Sun workstations via Ether net. The tilt angle is about -26 [deg] so that robot can see the environment effectively. The horizontal and vertical visual angle are about 67 [deg] and 60 [deg], respectively.



(a) y position of the ball

(b) y position of the left-upper of the goal

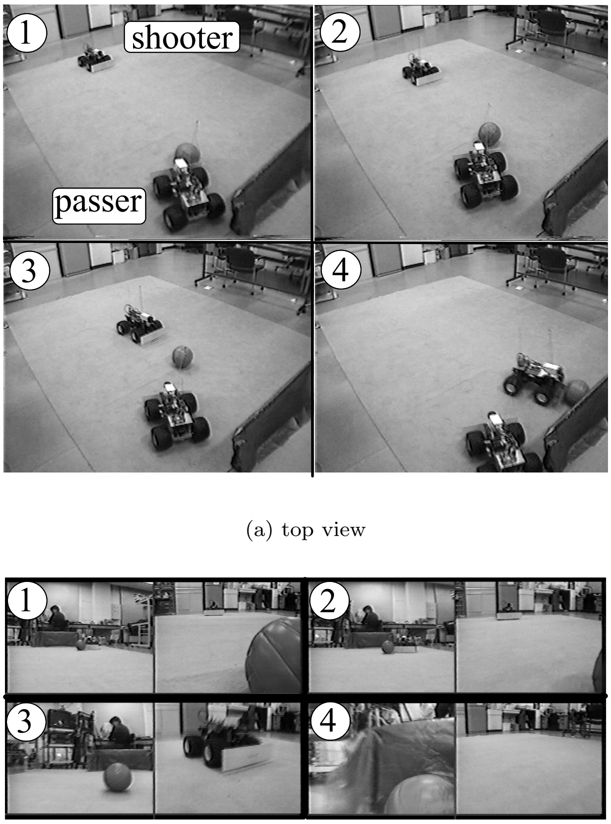
Figure 5: Prediction errors in the real environment

The task of the passer is to pass a ball to the shooter while the task of the shooter is to shoot a ball into the goal. Table 3 and Figure 5 show the experimental results. The value of l for the ball and the agent are bigger than that of computer simulation, because of the noise of the image processing and the dynamics of the environment due to such as the eccentricity of the centroid of the ball. Even though the local predictive model of the same ball for each agent is similar ($n = 4$, and slight difference in $\log|\boldsymbol{R}|$ and AIC) from Table3, the estimated state vectors are different from each other because there are differences in several factors such as tilt angle, the velocity of the motor and the angle of steering. We checked what happened if we replace the local predictive models between the passer and the shooter. Eventually, the large prediction errors of both side were observed. Therefore the local predictive models can not be replaced between physical agents. Figure 6 shows a sequence of images where the shooter shoot a ball which is kicked by the passer.

6 Conclusion

This paper presents a method of behavior acquisition in order to apply reinforcement learning to the environment including other agents. Our method takes account of the trade-off among the precision of prediction, the dimension of state vector and the length of steps to predict. Spatial quantization of the image into objects has been easily solved by painting objects in single color different from each other. Rather, the organization of the image features and their temporal segmentation for the purpose of task accomplishment have been done simultaneously by the method. The method is applied to a soccer playing game in which one robot can shoot a rolling ball passed by other robot.

In the proposed method, we need a quantization procedure of the estimated state vectors. Several segmentation methods such as Parti game algorithm[12] and Asada's method [5] might be promising.



(a) top view

(b) obtained images (left:shooter, right:passer)

Figure 6: Acquired behavior

References

[1] H. Akaike. A new look on the statistical model identification. *IEEE Trans. AC-19*, pp. 716–723, 1974.

[2] Y. Aloimonos. Introduction: Active vision revisited. In Y. Aloimonos ed., *Active Perception*, chapter 0, pp. 1–18. Lawrence Erlbaum Associate, Publishers, 1993.

[3] Y. Aloimonos. Reply: What i have learned. *CVGIP: Image Understanding*, 60:1:74–85, 1994.

[4] M. Asada. An agent and an environment: A view of "having bodies" – a case study on behavior learning for vision-based mobile robot –. In *Proc. of 1996 IROS Workshop on Towards Real Autonomy*, pp. 19–24, 1996.

[5] M. Asada, S. Noda, and K. Hosoda. Action-based sensor space categorization for robot learning. In *Proc. of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[6] M. Asada, S. Noda, S. Tawaratumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.

[7] J. H. Connel and S. Mahadevan. *Robot Learning.* Kluwer Academic Publishers, 1993.

[8] S. Edelman. Reply: Representation without reconstruction. *CVGIP: Image Understanding*, 60:1:92–94, 1994.

[9] M. Inaba. Remote-brained robotics : Interfacing AI with real world behaviors. In *Preprints of ISRR'93*, Pitsuburg, 1993.

[10] W. E. Larimore. Canonical variate analysis in identification, filtering, and adaptive control. In *Proc. 29th IEEE Conference on Decision and Control*, pp. 596–604, Honolulu, Hawaii, December 1990.

[11] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. of the 11th International Conference on Machine Learning*, pp. 157–163, 1994.

[12] A. W. Moore and C. G. Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning*, 21:199–233, 1995.

[13] T. Nakamura and M. Asada. Motion sketch: Acquisition of visual motion guided behaviors. In *14th International Joint Conference on Artificial Intelligence*, pp. 126–132. Morgan Kaufmann, 1995.

[14] T. Nakamura and M. Asada. Stereo sketch: Stereo vision-based target reaching behavior acquisition with occlusion detection and avoidance. In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 1314–1319, 1996.

[15] G. Sandini. Vision during action. In Y. Aloimonos ed., *Active Perception*, chapter 4, pp. 151–190. Lawrence Erlbaum Associate, Publishers, 1993.

[16] G. Sandini and E. Grosso. Reply: Why purposive vision. *CVGIP: Image Understanding*, 60:1:109–112, 1994.

[17] E. Uchibe, M. Asada, and K. Hosoda. Behavior coordination for a mobile robot using modular reinforcement learning. In *Proc. of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1329–1336, 1996.

[18] C. J. C. H. Watkins and P. Dayan. Technical note: *Q*-learning. *Machine Learning*, pp. 279–292, 1992.

[19] S. D. Whitehead and D. H. Ballard. Active perception and reinforcement learning. In *Proc. of Workshop on Machine Learning-1990*, pp. 179–188. Morgan Kaufmann, 1990.