Vision-Based Robot Learning for Behavior Acquisition

Minoru Asada, Takayuki Nakamura, and Koh Hosoda
Dept. of Mechanical Eng. for Computer-Controlled Machinery,
Osaka University, Suita 565 JAPAN
E-mail: asada@robotics.ccm.eng.osaka-u.ac.jp

Abstract

We introduce our approach, a new direction of robotics research that makes a robot learn to behave adequately to accomplish a given task at hand through the interactions with its environment with less *a priori* knowledge about the environment or the robot itself. We briefly present three research topics of vision-based robot learning in each of which visual perception is tightly coupled with actuator effects so as to learn an adequate behavior. First, "motion sketch" for a one-eyed mobile robot to learn several behaviors such as obstacle avoidance and target pursuit is introduced. Next, a method of vision-based reinforcement learning by which a robot learns to shoot a ball into a goal is presented. Finally, we show a method of purposive visual control consisting of an on-line estimator and a feedback/feedforward controller for uncalibrated camera-manipulator systems. All topics include the real robot experiments.

1 Introduction

Realization of autonomous agents that organize their own internal structure in order to take actions towards achieving their goals is the ultimate goal of AI and Robotics. That is, the autonomous agents have to learn. Recent research in artificial intelligence has developed computational approaches of agent's involvements in their environments [1]. Our final goal, in designing and building an autonomous agent with vision-based learning capabilities, is to have it perform a variety of tasks adequately in a complex environment. In order to build such an agent, we have to make clear the interaction between the agent and its environment.

In physiological psychology, Held and Hein [2] have shown that self-produced movement with its concurrent visual feedback is necessary for the development of visually-guided behaviors. Their experimental results suggest that perception and behavior are tightly coupled in autonomous agents that perform tasks. In biology, Horridge [3] similarly has suggested that motion is essential for perception in living systems such as bees.

In computer vision area, so-called "purposive active vision paradigm" [4, 5, 6] has been considered as a representative form of this coupling since Aloimonos et al. [7] proposed it as a method that converts the ill-posed vision problems into the well-posed ones. However, many researchers have been using so-called active vision systems in order to reconstruct 3-D information such as depth and shape from a sequence of 2-D images given the motion information of the observer or capability of controlling the observer motion. Furthermore, though purposive vision does not consider vision in isolation but as a part of complex system that interacts with world in specific ways [4], very few have tried to investigate the relationship between motor commands and visual information [8].

In robot learning area, the researchers have tried to make agents learn a purposive behavior to achieve a given task through agent-environment interactions. However, almost of them have only shown computer simulations, and only a few real robot applications are reported which are simple and less dynamic [9, 10]. there are very few examples of use of visual information in robot learning, probably because of the cost of visual processing.

In this paper, we introduce our approach, a new direction of robotics research that makes a robot learn to behave adequately to accomplish a given task at hand through the interactions with its environment with less *a priori* knowledge about the environment or the robot itself. We briefly present three research topics of vision-based robot learning in each of which visual perception is tightly coupled with actuator effects so as to learn an adequate behavior.

The remainder of this article is structured as follows: First we introduce a method to represent an interaction between the agent and its environment which is called "motion sketch" for a one-eyed mobile robot to learn several behaviors such as obstacle avoidance and target pursuit. Next, a method of vision-based reinforcement learning by which a robot learns to shoot a ball into a goal is presented. Finally, we show a method of purposive visual control consisting of an on-line estimator and a feedback/feedforward controller for uncalibrated camera-manipulator systems. All topics include the real robot experiments.

2 Motion Sketch [11]

2.1 Basic Ideas of Motion Sketch

The interaction between the agent and its environment can be seen as a cyclical process in which the environment generates an input (perception) to the agent and the agent generates an output (action) to the environment. If such an interaction can be formalized, the agent would be expected to carry out actions that are appropriate to individual situations. "Motion sketch," we proposed here, is one of such formalizations of interactions by which a vision-based learning agent that has real-time visual tracking routines behaves adequately against its environment to accomplish a variety of tasks.

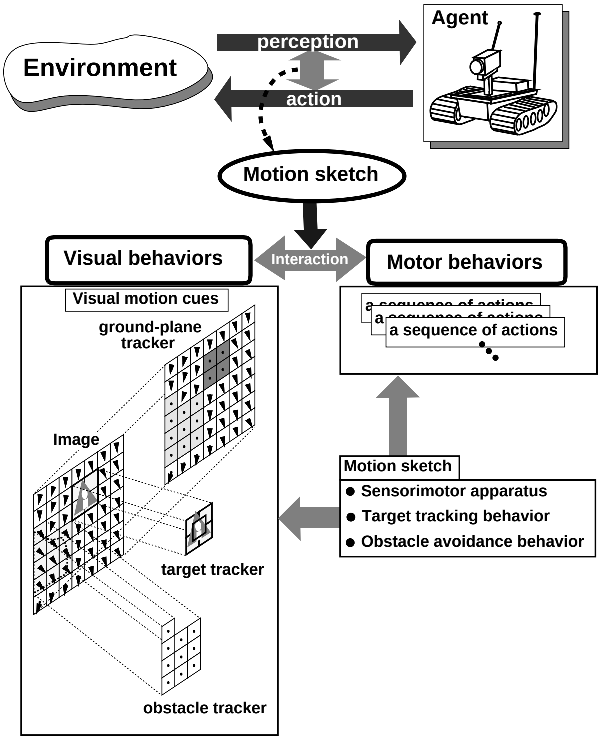


Figure 1: Motion sketch

Figure 1 shows a basic idea of the motion sketch. The basic components of the motion sketch are visual motion cues and the motor behaviors.

Visual motion cues are detected by several visual tracking routines of which behaviors (called visual behavior) are determined by individual tasks. The visual tracking routines are scattered over the whole image and an optical flow due to an instantaneous robot motion is detected. In this case, the tracking routines are fixed to the image points. The image area to be covered by these tracking routines are specified or automatically determined depending on the current tasks, and the cooperative behaviors between tracking routines are performed for the task accomplishment. For the target pursuit task, the multiple templates are initialized and every template looks for the target to realize stable tracking. In the task of obstacle detection and avoidance, the candidates for obstacles are first detected by comparing the optical flow with that of non-obstacle (ground plane) region, and then the detected region is tracked by multiple templates each of which tracks the inside of the moving obstacle region.

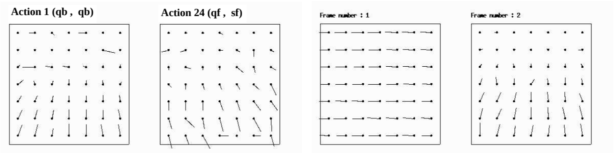
The motor behaviors are sets of motor commands obtained by Q-learning [12], a most widely used reinforcement learning method, based on the detected motion cues and given task. The sizes and positions of the target and the detected obstacle are used as components of a state vector in the learning process.

Visual and motor behaviors work in parallel in the image and compose a layered architecture. The visual behavior for monitoring robot motion (detecting the optical flow on the ground plane on which the robot lies) is the lowest and might be subsumed in part due to occlusion by other visual and motor behaviors for obstacle detection/avoidance and target pursuits which might occlude each other.

The motion sketch does not need any calibrations nor any 3-D reconstruction so as to accomplish the given task. The visual motion cues for representing the environment does not seem dependent on scene components nor limited to the specified situations and the task. Furthermore, the interaction is quickly obtained owing to the use of real-time visual tracking routines.

The behavior acquisition scheme consists of the following four stages: i) Obtaining the fundamental relationship between visual and robot motions by correlating motion commands and flow patterns on the floor with very few obstacles. ii) Learning target pursuit behavior by tracking a target. iii) Detection of obstacles and learning an avoidance behavior. iv) Coordination of the target pursuit and obstacle avoidance behaviors. At each stage, we obtain the interaction between the agent and its environment.

2.2 Obtaining sensorimotor apparatus



(a) examples of flow patterns

(b) obtained two principal flows

Figure 2: Acquisition of principal motion vectors

We place 49(7 × 7) visual tracking routines to detect changes in the whole image. Therefore, we obtain an optical flow composed of 49 flow vectors. In the environment without obstacles, the robot randomly selects a possible action among the action space, and executes it. While randomly wandering, the robot stores the flow patterns $\boldsymbol{p}_i$ due to its actions i. After the robot performed all possible actions, we obtain the averaged optical flows $\boldsymbol{p}_i$ removing the outliers due to noise or small obstacles based on the LMeS method. Figure 2 (a) shows examples of flows detected during random motions.

Using the averaged optical flows obtained above, we acquire principal motion patterns which characterize the space of actions. This is done by analyzing the space of averaged optical flow that robot is capable of producing. We want to find a basis for this space, i.e., a set of representative motion patterns from which all the motion patterns may be produced by their linear combinations. We can obtain representative motion patterns by using Principal Component Analysis that may be performed using a technique called Singular Value Decomposition(hereafter SVD). The first two principal components obtained in the real experiment are shown in Figure2 (b). Obviously, the left corresponds to a pure rotation and the right to a pure backward motion.

2.3 Behavior acquisition based on visual motion cues

Target tracking behavior acquisition
We use the visual tracking routines in order to pursue a target specified by a human operator and obtain the information about the target in the image such as its position and size which are used in the Q-learning algorithm [12] for acquisition of target pursuit behavior.

Obstacle avoidance behavior acquisition
We know the flow pattern $\boldsymbol{p}_i$ corresponding to the action i in the environment without any obstacles. The noise included in $\boldsymbol{p}_i$ is not so much, because this flow pattern is described as a linear combination of the two principal motion vectors. Therefore, it makes motion segmentation easy. Motion segmentation is done by comparing the flow pattern $\boldsymbol{p}_i$ with the flow pattern $\boldsymbol{p}_i^{obs}$ which is obtained in the environment with obstacles. The area in the $\boldsymbol{p}_i^{obs}$ is detected as the area for obstacle candidates if its components are different from that of $\boldsymbol{p}_i$. This information (position and size in the image) is used to obtain the obstacle tracking behavior. After obstacle detection, the visual tracking routines are set up at the positions where the obstacle candidates are detected and the regions are tracked until the region disappears from the image.

Learning to avoid obstacles consists of two stages. First, the obstacle tracking behavior is learned by the same manner as in learning the target pursuit behavior. Next, the obstacle avoidance behavior is generated by using the relation between the possible actions and the obstacle tracking behavior.

2.4 Experimental results

Figure 3 shows a configuration of the real mobile robot system. We have constructed the radio control system of the robot [13]. The image processing and the vehicle control system are operated by VxWorks OS on MVME167(MC68040 CPU) computer which are connected with host Sun workstations via Ether net. The image taken by a TV camera mounted on the robot is transmitted to a UHF receiver and subsampled by the scan-line converter (Sony Corp.). Then, the video signal is sent to a Fujitsu tracking module. The tracking module has a function of block correlation to track some pre-memorized patterns and can detect motion vectors in real time.

Figures 4 and 5 show sequences of images where the robot succeeded in target pursuit and avoiding a moving obstacle, respectively. The top shows the images taken and processed by the robot and the bottom images show how the robot behaves. In Figure 5, the rectangles indicate the obstacle candidate regions.

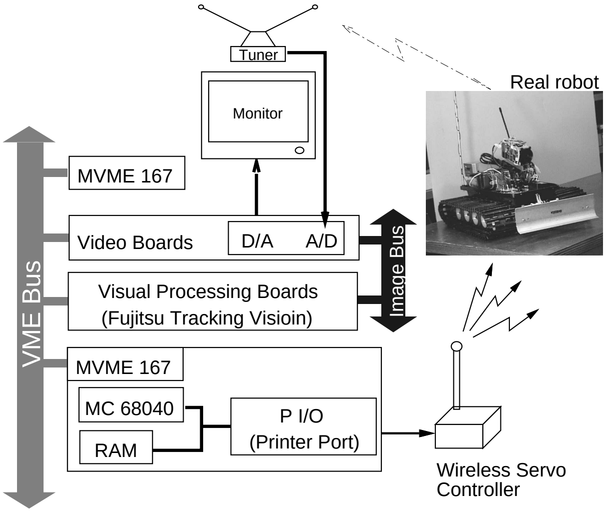


Figure 3: Configuration of the experimental system

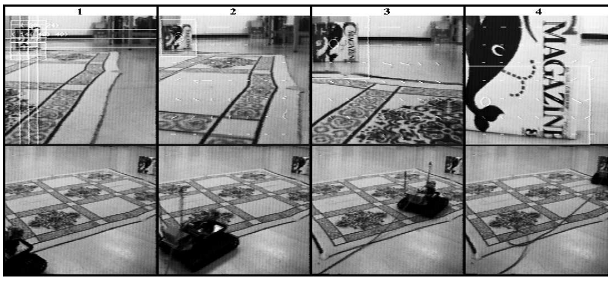


Figure 4: The robot succeeded in pursuing a target.

3 Vision-Based Reinforcement Learning for Behavior Acquisition [14]

Reinforcement learning has recently been receiving increased attention as a method for robot learning with little or no *a priori* knowledge and higher capability of reactive and adaptive behaviors [15]. In the reinforcement learning method, a robot and its 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.

Although the role of reinforcement learning is very important to realize autonomous systems, the prominence of that role is largely dependent on the extent to which the learning can be scaled to solve larger and more complex robot learning tasks. Many researchers in the field of machine learning have been concerned with the convergence time of the learning, and have developed methods to speed it up. However, almost

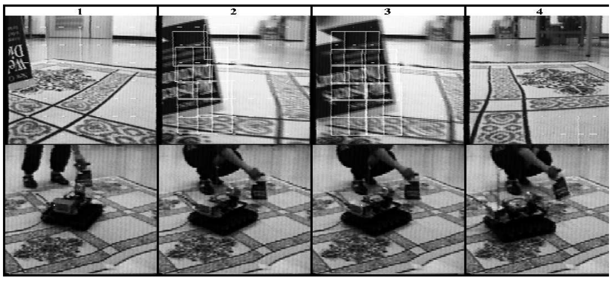


Figure 5: The robot succeeded in avoiding a moving obstacle.

all of them have only shown computer simulations in which they assume ideal sensors and actuators, where they can easily construct the state and action spaces consistent with each other.

Here, we present a method of vision-based reinforcement learning by which a robot learns to shoot a ball into a goal. The robot does not need to know any parameters of the 3-D environment or its kinematics/dynamics. The image captured from a single TV camera mounted on the robot is the only source of information on the changes in an environment. Image positions and sizes of the ball and the goal are used as a state vector. We discuss several issues from a viewpoint of robot learning: a) coping with a "state-action deviation" problem which occurs in constructing the state and action spaces in accordance with outputs from the physical sensors and actuators, and b) starting with easy missions (rather than task decomposition) for rapid task learning.

3.1 Task and assumptions

The task for a mobile robot is to shoot a ball into a goal. The problem we address here is how to develop a method which automatically acquires strategies for doing this. We assume that the environment consists of a ball and a goal; the mobile robot has a single TV camera; and that the robot does not know the location/size of the goal, the size/weight of the ball, any camera parameters such as the focal length and tilt angle, or the kinematics/dynamics of itself.

3.2 Construction of State and Action Spaces

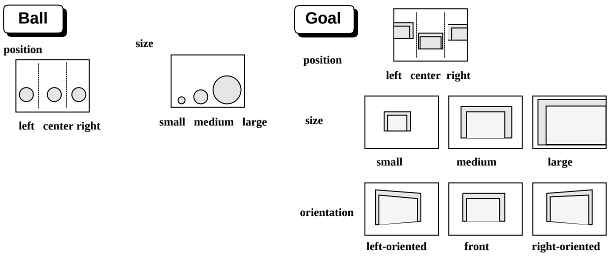


Figure 6: The ball sub-states and the goal sub-states

Figure 6 shows sub-states of ball and goal in which the position and the size of the ball or goal are naturally and coarsely classified into each state. Due to the peculiarity of visual information, that is, a small change near the observer results in a large change in the image and a large change far from the observer may result in a small change in the image, one action does not always correspond to one state transition. We call this the **"state-action deviation problem"**: Figure 7 indicates this problem, the area representing the state "the goal is far" is large, therefore the robot frequently returns to this state if the action is forward. This is highly undesirable because the variations in the state transitions is very large, consequently the learning does not converge correctly.

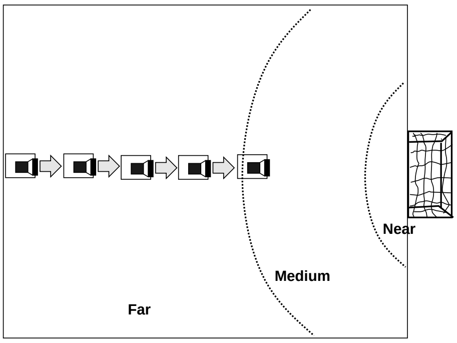


Figure 7: A state-action deviation problem

To avoid this problem, we reconstruct the action space as follows. Each action defined 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. In the above case, the robot takes a forward motion many times until the state "the goal is far" changes into the state "the goal is medium".

3.3 Learning from Easy Missions

In order to improve the learning rate, the whole task was separated into different parts in [10]. By contrast, we do not decompose the whole task into subtasks of finding, dribbling, and shooting a ball. Instead, we first used a monolithic approach. That is, we place the ball and the robot at arbitrary positions. In almost all the cases, the robot crossed over the field line without shooting the ball into the goal. This means that the learning did not converge after many trials. This is the famous *delayed reinforcement* problem due to no explicit teacher signal that 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 LEM).

3.4 Experimental results

We applied the LEM algorithm to the task in which the order of easy situations are $\mathbf{S_1}$ ("the goal is large"), $\mathbf{S_2}$ ("the goal is medium", and $\mathbf{S_3}$ ("the goal is small"). Figure 8 shows the changes of the summations of Q-values of the action-value function in the Q-learning method with and without LEM, and its

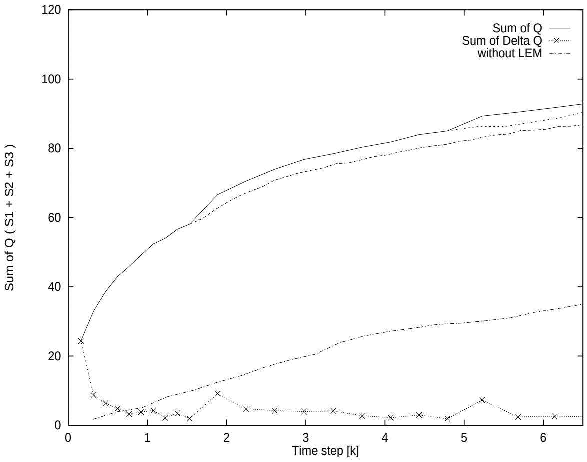


Figure 8: Change of the sum of Q-values with LEM in terms of goal size

temporal derivative ΔQ. The axis of time step is scaled by M (10^6), which corresponds to about 9 hours in the real world since one time step is 33ms. The solid and broken lines indicate the summations of the maximum value of Q in terms of action in all states with and without LEM, respectively. The Q-learning without LEM was implemented by setting initial positions of the robot at completely arbitrary ones. Evidently, the Q-learning with LEM is much better than that without LEM. The broken line with empty squares indicates the change of ΔQ. Two arrows indicate the time steps (around 1.5M and 4.7M) when a set of the initial states changed from $\mathbf{S_1}$ to $\mathbf{S_2}$ and from $\mathbf{S_2}$ to $\mathbf{S_3}$, respectively. Just after these steps, ΔQ drastically increased, which means the Q-values in the inexperienced states are updated. The coarsely and finely dotted lines expanding from the time steps indicated by the two arrows show the curves when the initial positions were not changed from $\mathbf{S_1}$ to $\mathbf{S_2}$, nor from $\mathbf{S_2}$ to $\mathbf{S_3}$, respectively. This simulates the LEM with partial knowledge. If we know only the easy situations ($\mathbf{S_1}$), and nothing more, the learning curve follows the finely dotted line in Figure 8. The summation of Q-values is slightly less than that of the LEM with more knowledge, but much better than that without LEM.

We used the same experimental set up as that described in the previous section. In Figure 9 (raster order), the images are taken every second. First, the robot lost the ball due to noise, and then it turned around to find the ball, and finally it succeeded in shooting.

4 Purposive Visual Control for uncalibrated camera-manipulator systems [16]

Recently, there have been several studies on visual servoing, using visual information in the dynamic feedback loop to increase robustness of the closed loop system (we can find a summary in [17]). In most of the previous works on visual servoing, they assumed that the system structure and parameters are known, or that the parameters can be identified in an off-line process or on-line parameter identification with restrictions and assumptions on the system.

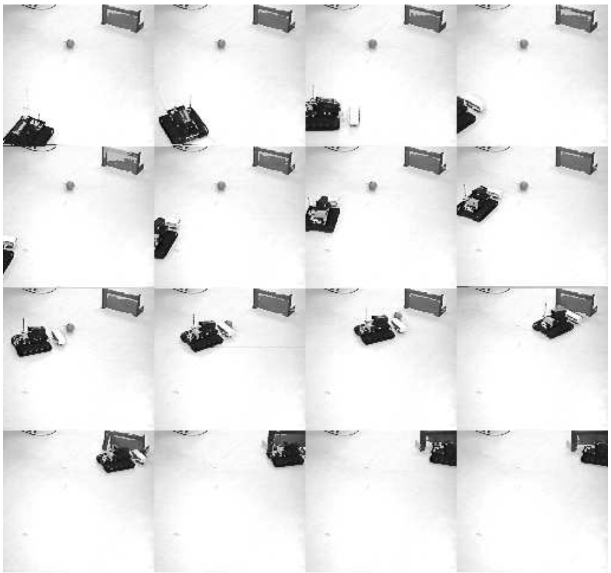


Figure 9: The robot succeeded in shooting a ball into the goal.

On the other hand, the previous works payed attention only to feedback servoing. They sensed positions of targets and made feedback inputs subtracting the sensed positions from the desired ones. Using these controllers, the manipulator does not work until error is observed, which can be considered as *reactive* movement. For intelligent control of camera-manipulator systems, not only the reactive but also *purposive* visual movement must be realized. At the level of control, we believe that feedforward terms should play a great part in realizing the purposive movement, but no one has mentioned to the effectiveness of feedforward terms to the best of our knowledge.

Here, we propose purposive visual control consisting of an on-line estimator and a feedback/feedforward controller for uncalibrated camera-manipulator systems. It has the following features:

1. The estimator does not need any a priori knowledge on the kinematic structure nor the system parameters. We can eliminate the tedious calibration process owing to this feature.

2. There are no restrictions on the camera-manipulator system: the number of cameras, kinds of images features, structure of the system (camera-in-manipulator or camera-and-manipulator) ,the number of inputs and outputs (SISO or MIMO). The proposed method is applicable to all cases. It is strongly related with the fact that the estimator does not need any a priori knowledge on the system.

3. The aim of the estimator is not to estimate the true parameters, but to ensure asymptotical convergence of the image features to the desired values under the proposed controller. Therefore, the estimated parameters do not necessarily converge to the true values. The existing methods such as [18, 19] tried to estimate the true parameters, and therefore they need restrictions and assumptions.

4. The proposed controller can realize *purposive* movement of the system utilizing its feedforward terms. The feedforward terms of the proposed scheme are based on *estimated* parameters intending to realize visual tasks on the image planes (mentioned in 3). In this sense, this feedforward terms help realizing purposive movement at the control level.

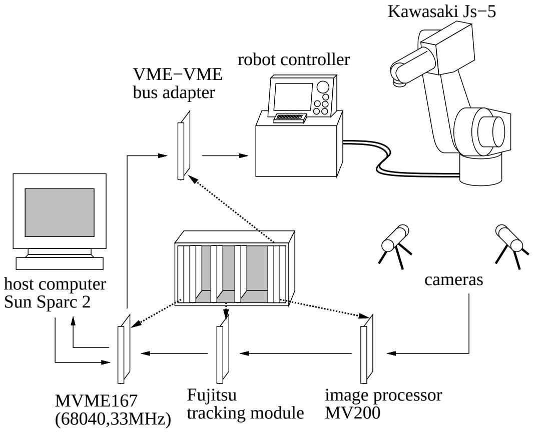


Figure 10: Experimental system

Figure 10 shows the experimental system we used. Figure 11 (a) shows an experimental set up with two cameras fixed, and (b) indicates the result of step response with and without on-line estimator, where vertical and horizontal axes indicate the error in pixels and time steps (second), respectively. Evidently, the performance without the estimator was much worse that with the estimator.

References

[1] Philip E. Agre. "Computational research on interaction and agency". *Artificial Intelligence*, 72:1–52, 1995.

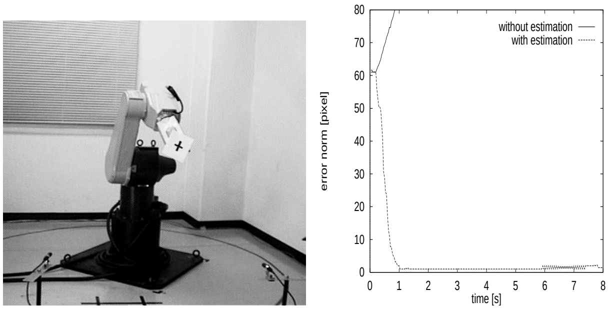
[2] R. Held and A. Hein. "Movement-produced stimulation in the development of visually guided behaviors". *Jounal of Comparative and Physiological Psycology*, 56:5:872–876, 1963.

[3] G. A. Horridge. "The evolution of visual processing and the construction of seeing systems". In *Proc. of Royal Soc. London B230*, pages 279–292, 1987.

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

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

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



(a) experiment with cameras fixed

(b) results

Figure 11: Visual servoing with tracking vision

[7] Y. Aloimonos, I. Weiss, and A. Bandyopadhyay. "Active vision". In *Proc. of first ICCV*, pages 35–54, 1987.

[8] G. Sandini. "Vision during action". In Y. Aloimonos, editor, *Active Perception*, chapter 4. Lawrence Erlbaum Associates, Publishers, 1993.

[9] P. Maes and R. A. Brooks. "Learning to coordinate behaviors". In *Proc. of AAAI-90*, pages 796–802, 1990.

[10] J. H. Connel and S. Mahadevan. "Rapid task learning for real robot". In J. H. Connel and S. Mahadevan, editors, *Robot Learning*, chapter 5. Kluwer Academic Publishers, 1993.

[11] T. Nakamura and M. Asada. Motion sketch: Acquisition of visual motion guided behaviors. In *Proc. of IJCAI-95*, pages –, 1995.

[12] C. J. C. H. Watkins. *Learning from delayed rewards"*. PhD thesis, King's College, University of Cambridge, May 1989.

[13] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. "Vision-based behavior acquisition for a shooting robot by using a reinforcement learning". In *Proc. of IAPR / IEEE Workshop on Visual Behaviors-1994*, pages 112–118, 1994.

[14] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. Vision-based reinforcement learning for purposive behavior acquisition. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 146–153, 1995.

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

[16] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true jacobian. In *Proc. of IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 1994 (IROS '94)*, pages 186–193, 1994.

[17] P. I. Corke. Visual control of robot manipulators – a review. In *Visual Servoing*, pages 1–31. World Scientific, 1993.

[18] B. Nelson, N. P. Papanikolopoulos, and P. K. Khosla. Visual servoing for robotic assembly. In *Visual Servoing*, pages 139–164. World Scientific, 1993.

[19] N. P. Papanikolopoulos, B. Nelson, and P. K. Khosla. Six degree-of-freedom hand/eye visual tracking with uncertain parameters. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 174–179, 1994.