Target Reaching Behavior Learning with Occlusion Detection and Avoidance for A Stereo Vision-Based Mobile Robot

Minoru Asada and Takayuki Nakamura
Dept. of Mech. Eng. for Computer-Controlled Machinery
Osaka University, 2-1, Yamadaoka, Suita, Osaka 565, Japan
asada@robotics.ccm.eng.osaka-u.ac.jp
Tel. +81-6-879-7347, Fax. +81-6-879-7348

Abstract We have proposed *motion sketch* [Nakamura and Asada, 1995] as a representation of interaction between a one-eyed learning agent (a mobile robot) and its environment. In this paper, we extend the basic idea in *motion sketch*, tight coupling of perception and action, to *stereo sketch* by which a stereo-vision based mobile robot learns to reach a target simultaneously detecting and avoiding occlusions. First, an input scene is segmented into homogeneous regions by the enhanced ISODATA algorithm with MDL principle in terms of image coordinates and disparity information obtained from the fast stereo matcher based on the coarse-to-fine control method. Then, the segmented regions including the target area and their occlusion status identified during the stereo and motion disparity estimation construct a state space for a reinforcement learning method to obtain target reaching behavior. As a result of learning, the robot can avoid obstacles without describing them explicitly. We give the computer simulation results and real robot implementation to show the validity of *stereo sketch*.

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 Robotics and AI. That is, the autonomous agents have to learn. Recent research in artificial intelligence has developed computational approaches of agent's involvements in their environments [Agre, 1995]. 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.

We have proposed *motion sketch* as a representation of interactions between a one-eyed learning agent (a mobile robot) and its environment [Nakamura and Asada, 1995]. *Motion sketch* has an important role of connecting motor behaviors which consist of uninterpreted motor command sequences and predefined visual behaviors. Through the learning process, motor sequences which realize desired behaviors such as obstacle avoidance and target reaching are obtained via *motion sketch* step by step. In this paper, we extend the *motion sketch* to *stereo sketch* by which a stereo-vision based mobile robot learns to reach a target by detecting and avoiding occlusions. As a result of learning, the robot realizes obstacle avoidance without describing them explicitly. The role of the *stereo sketch* is the same as the *motion sketch*, that is, to connect visual behaviors and motor behaviors.

In computer vision area, integration of stereo and motion has been done by several researchers [Waxman and Duncan, 1986; N. M. Nasrabadi and Liu, 1989; E. Grosso and Tistarelli, 1989]. Since their main purpose is to reconstruct the precise 3-D geometry of the objects or the environment, these approaches need very precise stereo camera calibration processes that are tedious and often difficult to implement. However, in robotics the extraction of the information necessary to derive desired behaviors from the sensory data in real-time seems much more important than time-consuming reconstruction of the precise 3-D geometry.

Recently, Huber and Kortenkamp [Huber and Kortenkamp, 1995] used a real-time stereo vision system [Nishihara, 1984] to pursue moving agents while still performing obstacle avoidance. Since their stereo matching is based on edges extracted by a Laplacian-Gaussian filter, they have the following drawbacks: The system likely looses the target because tracking module is easily attracted by higher texture areas than the current target one, and therefore they cannot cope with any occlusions of the target area by other objects. Since the system try to keep the target in 3-D space at a fixed distance, they cannot cope with changes in scale of the target image (a group of edges) due to motions of the target and/or the robot, nor reach the target.

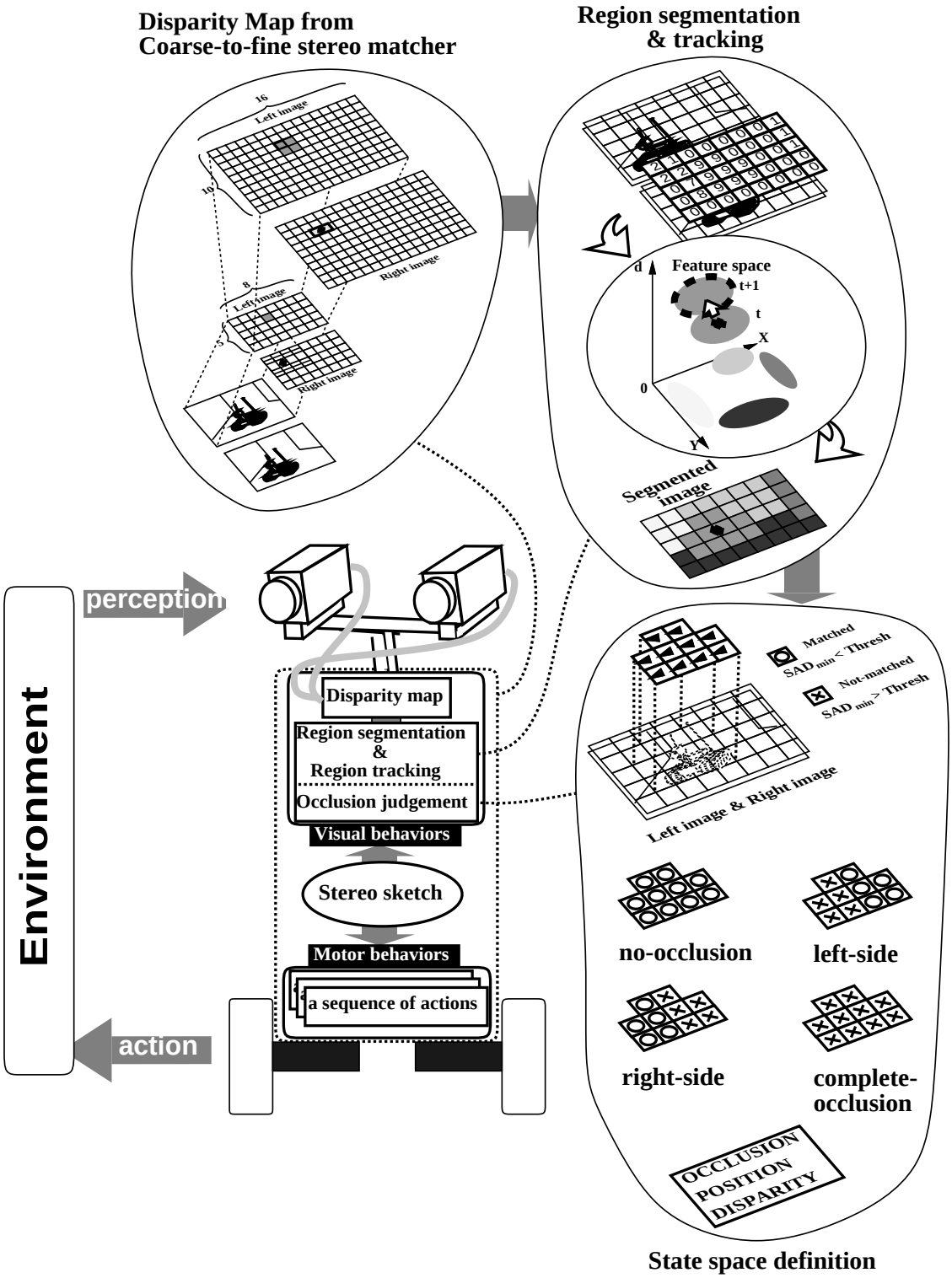


Figure 1: Stereo sketch

In this paper, we propose a stereo vision-based behavior learning method. As a real-time stereo matching method, we use a block correlation of image intensity [Inoue *et al.*, 1992], by which the capabilities of stereo and motion disparity estimation are much improved than only edge-based approach. By adopting the coarse-to-fine control of stereo [Marr and Poggio, 1979; Grimson, 1982] and motion disparity estimation techniques, we can cope with changes in scale due to target and/or the robot motions. Further, we apply a reinforcement learning method to obtain a reaching behavior for the environmental adaptation using a well-defined state space consisting of occlusion status identified during the stereo and motion disparity estimation.

The remainder of this article is structured as follows: In the next section, we describe basic ideas of *stereo sketch*. Then, we give explanations of visual behaviors, the method of learning, and state space definition. Finally, we give computer simulations, real robot implementation results, and concluding remarks.

2 Stereo Sketch

The interaction between an 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 have proposed in [Nakamura and Asada, 1995] is one of such formalizations of interactions by which a one-eyed vision-based learning agent with real-time visual tracking routines behaves adequately against its environment to accomplish a variety of tasks.

In the motion sketch, first the robot obtained the sensorimotor apparatus using optical flows on the floor caused by the robot motions. Then, it learned to detect/avoid obstacles, to reach a target, and to coordinate the learned behaviors step by step.

The motion sketch has the following limitations:

- Tracking performance severely depends on the constancy of image intensity inside tracking windows. Therefore, sometimes it lost the target area.
- The flow difference between the current scene and the template obtained by assuming no obstacles was used to detect obstacles. Since the stationary obstacles showed only small flow differences, they were difficult to detect.

To cope with these limitations, we add one more camera on the robot and realize a real-time stereo-vision system with the same tracking routines. The stereo disparity information not only improves the tracking performance but also provides useful information about occlusion and disocclusion. Motor behaviors are coordinated so as to reach the target area via reinforcement learning in which the state space is defined in terms of image locations of occluded and disoccluded areas. Supported by these visual behaviors and learned motor behaviors, the robot can reach the target area without explicitly describing obstacles.

Figure 1 shows a basic idea of *stereo sketch*. The basic components of the stereo sketch are disparity cues and the motor behaviors.

Both *motion sketch* and *stereo sketch* represent tight couplings between an agent that can perform an appropriate action sequence so as to accomplish the given tasks and its environment. The *stereo sketch* has a more abstract form of state space such as occlusion and disocclusion supported by a variety of powerful visual behaviors while *motion sketch* represents a more direct coupling between perception and action. However, the basic assumptions in these sketches are the same. They do not need *a priori* knowledge about the environment or kinematics/dynamics of the robot itself, any calibrations, nor any 3-D quantitative reconstruction so as to accomplish the given task. The cues of stereo and motion disparities do not seem dependent on scene components nor limited to the specified situations or the task. Figure 2 shows an example task we deal with in this paper. The robot tries to reach the target object while avoiding occlusions. The behavior acquisition scheme consists of the following procedures:

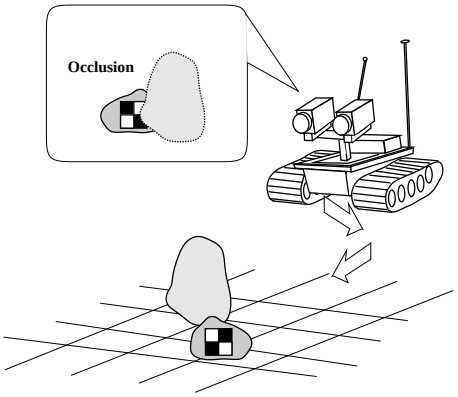


Figure 2: An example task

Visual Behaviors

1. Scene decomposition by using the enhanced ISODATA algorithm with MDL principle in terms of image location and disparity information obtained by coarse-to-fine stereo matching procedure.

2. Specify one target area among the segmented regions. Human operator can specify or the system can pick up, for example, the region closest to the robot.

3. Track the target and its neighbor regions between consecutive frames.

4. If a part of the target area is occluded (mismatch of the target area in part and correct match of its neighbor), note the location of the occluded area relative to the target area.

5. If a part of the target area is disoccluded (correct match of a new region around the target area with the same disparity as the target area), note the location of the disoccluded area.

Learning Phase

1. Construct a state space consisting of combinations of the location and occlusion status of the target area between the left and right images.

2. Obtain the data with random motions and find hidden states by the statistics of the state transitions.

3. Add the hidden states as new states and apply reinforcement learning given the goal state (reaching the target area).

3 Visual Behaviors

As visual functions, we have prepared the following routines in the previous work [Nakamura and Asada, 1995] using a real-time visual tracker by a simple block correlation based on SAD (Summation of Absolute Difference) [Inoue *et al.*, 1992]:

1. robust target tracking coping with partial occlusion and small deformation of image pattern using multiple searching blocks covered over the same target,

2. multi-resolution based matching to cope with scale changes due to robot and/or target motions, and

3. global search when a local search fails.

In addition to these visual routines, we add a stereo matching routine based on coarse-to-fine control and a region segmentation routine by using the enhanced ISODATA algorithm with MDL principle in terms of image location and disparity information obtained by coarse-to-fine stereo matching procedure. In this section, we show only the results of visual behaviors and we describe the details in Appendices.

Figure 3 shows the stereo matching result by the method where three pairs of the left and right images are stacked in three rows, respectively. Due to the hardware limitation, we are currently using the middle (coarse image: 128×120 pixels) and large (fine image: 256×240 pixels) scaled image pairs. The final matching result is shown at the right-bottom as a disparity map.

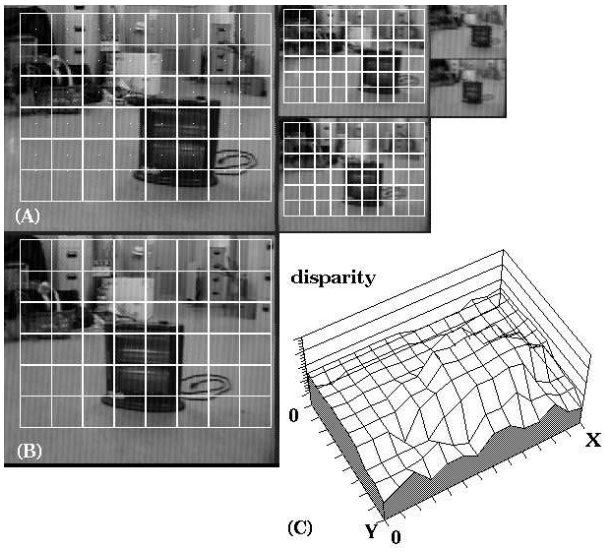


Figure 3: A coarse-to-fine stereo matching

Figure 4 shows an example of region tracking during the target reaching without obstacles. The method can cope with changes of target size in image, and successfully track the target.

4 Behavior Learning

As a method of learning for behavior acquisition, we use Q-learning [Watkins, 1989], a most widely used reinforcement learning method. To apply Q-learning to our task, we need to define a state space which consists of descriptions of target and its surroundings obtained by the visual behaviors described above. In this section, we first give basics of reinforcement learning, and then explain how to construct a state space for the learning.

4.1 Basics of Reinforcement Learning

Reinforcement learning agents improve their performance on tasks using reward and punishment received from their environment. They are distinguished from supervised learning agents in that they have no "teacher" that tells the agent the correct response to a situation when an agent responds poorly. An agent's only feedback indicating its performance on the task at hand is a scalar reward value. One step Q-learning [Watkins, 1989] has attracted much attention

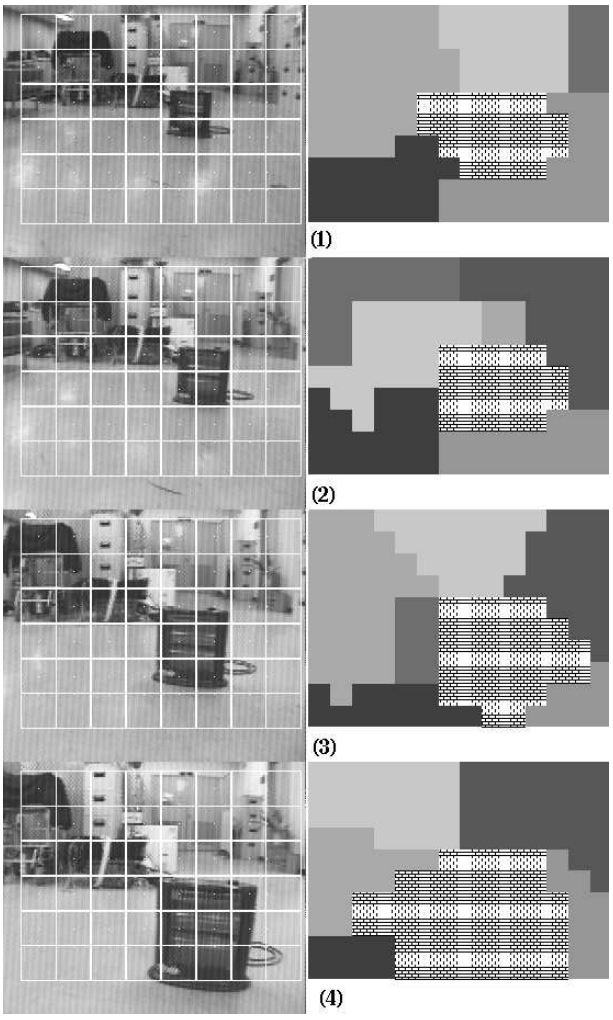


Figure 4: Region tracking

as an implementation of reinforcement learning because it is derived from dynamic programming [Bellman, 1957]. The following is a simple version of the 1-step Q-learning algorithm we used here.

Initialization: $Q \leftarrow$ a set of initial values for the action-value function (e.g., all zeros).

Repeat forever:

1. $s \in \mathbf{S} \leftarrow$ the current state
2. Select an action $a \in \mathbf{A}$ that is usually consistent with the policy f but occasionally an alternate.
3. Execute action a, and let s' and r be the next state and the reward received, respectively.
4. Update $Q(s,a)$:

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

5. Update the policy f:

$$f(s) \leftarrow a \quad \text{such} \quad \text{that} \quad Q(s,a) = \max_{b \in \boldsymbol{A}} Q(s,b) \quad (2)$$

4.2 State Space Construction

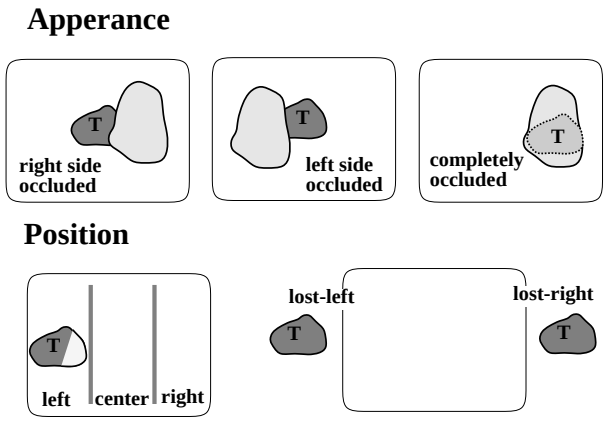


Figure 5: State space construction

State space should include necessary and sufficient information to achieve the given goal while it should be compact because the unbiased (without no *a priori* knowledge about agent and its environment) Q-learning can be expected to take execution time exponential in the size of the state space [Whitehead, 1991].

Focusing on target reaching behavior, we construct a state space in terms of occlusion status of the target area and its neighbor which is obtained by visual behaviors such as stereo and motion disparity estimation, and region clustering and tracking. Occlusion status is defined as combinations of target states in the left and right images. The target state is a triplet of appearance (the left side is occluded, the right side is occluded, completely occluded, no occlusion, or disappearance), location (left, center, or right), and disparity (far, middle, or near). In case of disappearance, we prepare two situations (lost-into-the-right or lost-into-the-left) (see Figure 5).

One of features of the state space is that it does not include explicit description of obstacles. Instead, the occlusion status of the target area tells indirectly the status of obstacles. Table 1 shows an example state which might tell that an obstacle is located in the right side and closer than the target.

image	appearance	position	disparity
left	*right side is occluded*	*center*	*any*
right	*completely occluded*	*center*	*any*

Table 1: An example state

Although the state space seems complete, it suffers from "perceptual aliasing problem" [Whitehead and Ballard, 1990] due to the limitation of the perceptual capacity. It is generally defined as "a problem caused by multiple projections of different actual situations into one observed state." The multiple projections make it very difficult for a robot to take an optimal action. To find such states (called "hidden states"), we estimate the state transition probabilities by using the MLE (Maximum Likelihood Estimation). If the state transition probability density function has multiple peaks, we trace back the history of state transitions until this hidden state can be discriminated [McCallum, 1995]. After finding hidden states and adding them to the state space, we apply Q-learning to our task.

5 Experimental Results

The experiment consists of two parts: first, learning the optimal policy f through the computer simulation, then applying the learned policy to a real situation.

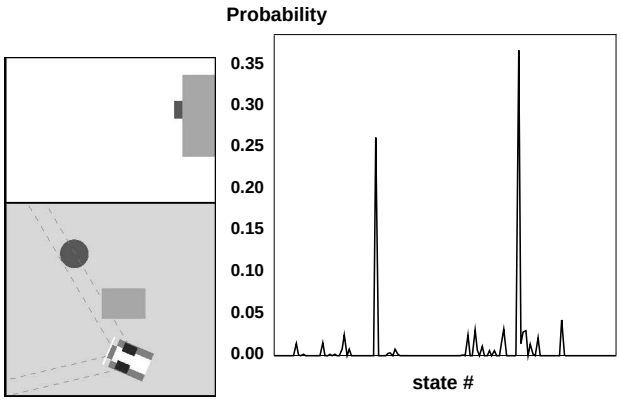
5.1 Simulations

5.1.1 Action space definition

In applying Q-learning to our task, we have to define action space. Our robot can select an action to be taken in the current state of the environment. The robot moves around using a PWS (Power Wheeled Steering) system with two independent motors. Since we can send the motor control command to each of the two motors separately, we construct the action set in terms of two motor commands ω_l and ω_r, each of which has 3 sub-actions, forward, stop, and back. All together, we have 9 actions.

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 called this the **"state-action deviation problem"** in [Asada *et al.*, 1995b]. To avoid this problem, we reconstruct the action space as follows. Each action defined above 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 defined as an action.

5.1.2 Hidden states, goal and reward



(a) A hidden state

(b) A state transition probability

Figure 6: An example of hidden states

Figure 6 shows an example of hidden state and its state transition probability density function estimated by MLE. In the right image, the target is observed (left side is occluded) while the target is not observed in the left image. Since the disparity information is not available, the target can be exist from near to far. This means that in the left image, the target is completely occluded or disappeared. Two peaks in Figure 6 (b) correspond to these two situations. These hidden states are found and added into the state space.

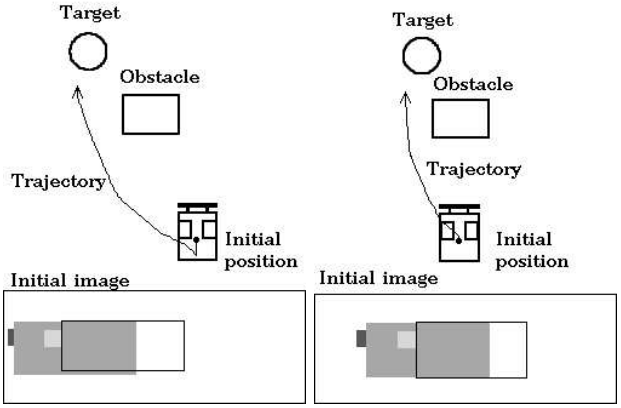
The goal state is shown in Table 2. We give a reward 1 when the robot achieves the goal state, otherwise 0. When the robot makes a collision with an obstacle, we do not give any negative reward but reset the robot position because the negative rewards make many local maxima of Q-values. Reset and 0 reward indirectly suggest the negative situations although the convergence time might spend a lot.

image	appearance	position	disparity
left	*no occlusion*	*right*	*near*
right	*no occlusion*	*left*	*near*

Table 2: A goal state

Figure 7 shows an example of target reaching behavior obtained by the learning method. In Figure 7 (b), the robot directly moved toward the target avoiding an obstacle while in figure 7 (a) it moved back to observe better, then moved toward the target. Note that during the learning process, the robot obtained

a desirable behavior which includes not only "observation for actions" but also "actions for observation."



(a) Example 1 (b) Example 2

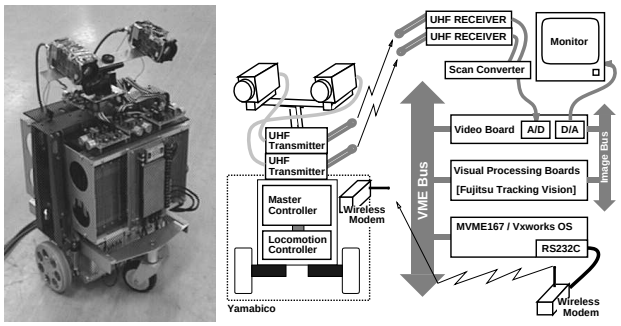
Figure 7: Reaching behavior obtained by the learning method

5.2 Real Robot Experiments

Figure 8 shows our real robot system. The stereo cameras are set on a mobile platform (Yamabico) controlled by MVME167/VxWorks OS through RS232C. The base line is about 17cm and both lines of sight are almost parallel. The tilt angle is about 10 degrees. The camera height is about 60cm. These parameters are very rough and not strictly adjusted. The visual angles of both cameras are the same and about 60 degrees. The maximum vehicle speed is about 60cm/s.

One visual tracking board has a capability of video rate tracking of about 140 windows. We are now using four boards, but the procedures of coarse-to-fine stereo matching, region clustering with MDL, and tracking seem too much for one CPU board, and now it takes about 160ms for one cycle.

Figure 9 shows sequences of the left and right images taken by the robot during the process in which it reaches the target, avoiding an obstacle. **Table** 3 shows the result of state discrimination for the scene shown in Figure 9. In **Table** 3, discriminated state steps, target states in both of the left and right images (Occlusion Status OL: left-side-part occluded, OR: right-side-part occluded, CO: completely occluded, NO: no occlusion, Location L: left position, C: center position, R: right position, and Disparity L: large, M: medium, S: small), control commands to the right and left motors (Forward (F), Stop (S), Backward (B)) are shown. The marks "*" indicate steps



(a) A robot (b) A system architecture

Figure 8: Our robot and system

Table 3: **State-Action data in a real environment**

state step	state left	state right	action L	action R
1*(1-11)	(NO,M,C)	(NO,M,C)	F	F
2*(12-14)	(OL,M,C)	(NO,M,C)	S	B
3(15-16)	(OL,M,C)	(NO,M,C)	F	B
4(17)	(OL,M,C)	(NO,M,L)	F	B
5*(18)	(OL,M,L)	(NO,M,L)	F	F
6*(19-21)	(NO,M,L)	(NO,M,L)	S	F
7*(22-28)	(NO,M,C)	(NO,M,L)	F	F
8(29-38)	(NO,N,C)	(NO,N,L)	F	F
9(39)	(NO,N,C)	(NO,N,C)	S	F
10*(40-44)	(NO,N,C)	(NO,N,L)	F	F

shown in Figure 9. In this table, the numbers in () shows the sampling steps.

At the second state step (2) in **Table** 3, the robot took a backward action. This means an "action for observation" described above because a backward action is not directly effective for the target reaching task but useful for expanding the field of view.

6 Discussion and Future Works

The stereo sketch consists of sophisticated visual behaviors and behavior learning phase. As we have shown in the experimental results, the learning method could obtain the desirable behaviors of target reaching and obstacle avoiding without predefining them. The conventional approach based on geometry reconstruction needs accurate calibration for sensors and actuators and planning in order to realize the same performance. While, behavior-based approach by [Huber and Kortenkamp, 1995] has carefully designed the switching conditions between avoidance and following behaviors, which are often difficult to be

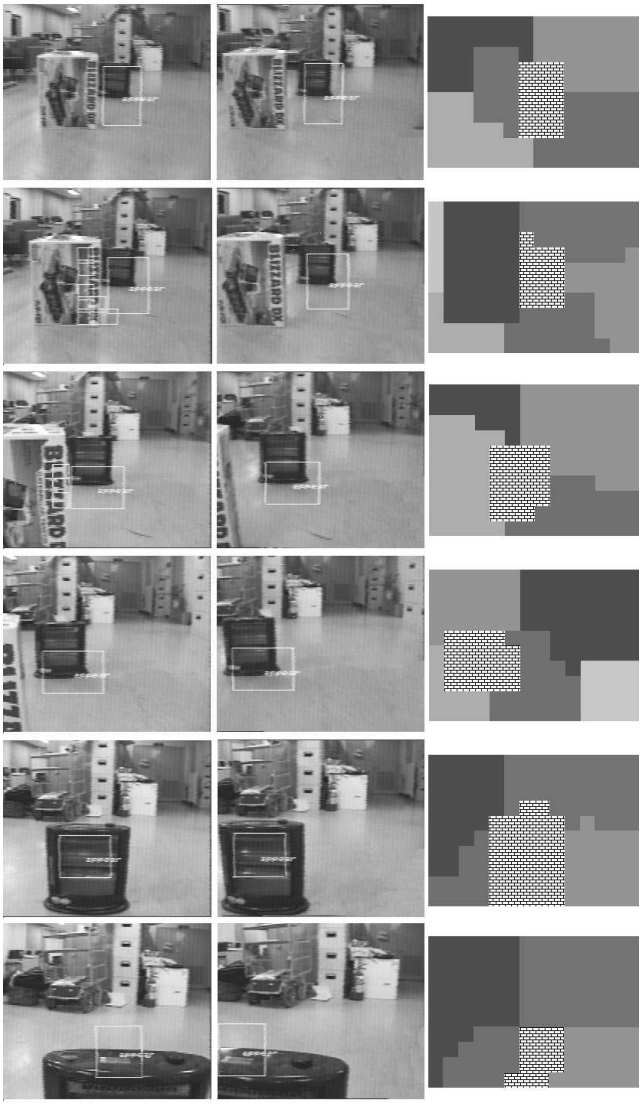


Figure 9: Reaching behavior by the real robot

determined. Our method does not care about these issues, but the following issues should be considered:

- The visual behaviors have to be obtained in advance and be expected to work without any errors due to several causes such as video noise while motor behaviors are obtained in the learning process. Although it seems difficult to learn both behaviors simultaneously, we need some methods for visual behavior learning related to motor behaviors.
- State space construction is much more important issue in robot learning. The main reason why our method works successfully is that we have prepared the sophisticated visual behaviors and carefully designed state space which could drastically reduce the learning time. Even though the programmer designed it, the state space in *stereo sketch* still suffers from hidden states. The state space designed by the programmer seems optimal, but actually not. The robot has to construct the state space through its experiences. We have done one work on this issue [Asada *et al.*, 1995a] with a different task, and are doing other applications for state space construction.

References

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

[Asada *et al.*, 1995a] M. Asada, S. Noda, and K. Hosoda. Non-physical intervention in robot learning based on lfe method. In *Proc. of Machine Learning Conferen Workshop on Learning from Examples vs. Programming by Demonstration*, pages 25–31, 1995.

[Asada *et al.*, 1995b] 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.

[Ball and Hall, 1965] G. H. Ball and D. J. Hall. "ISODATA, a novel method of data analysis and pattern classification". *Stanford Research Institute*, AD-699616, 1965.

[Bellman, 1957] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[Duda and Hart, 1973] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc, 1973.

[E. Grosso and Tistarelli, 1989] G. Sandini E. Grosso and M. Tistarelli. "3D object reconstruction using stereo and motion". *IEEE Trans. on SMC*, 19(6):1465–1476, 1989.

[Grimson, 1982] W. E. L. Grimson. "a computatinal theory of visual surface interpolation". In *Proc. of Royal Soc. London B298*, pages 395–427, 1982.

[Huber and Kortenkamp, 1995] E. Huber and D. Kortenkamp. Using stereo vision to pursue moving agent with a mobile robot. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 2340–2346, 1995.

[Inoue *et al.*, 1992] H. Inoue, T. Tachikawa, and M. Inaba. "Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation". In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 1621–1626, 1992.

[Marr and Poggio, 1979] D. Marr and T. Poggio. "a computatinal theory of human stereo vision". In *Proc. of Royal Soc. London B204*, pages 301–338, 1979.

[McCallum, 1995] R.A. McCallum. "instance-based utile distinctions for reinforcement learning with hidden state". In *Proc. of the 12th Int. Conf. on Machine Learning*, pages 387–395, 1995.

[N. M. Nasrabadi and Liu, 1989] S. P. Clifford N. M. Nasrabadi and Y. Liu. "Integration of stereo vision and optical flow by using an energy-minimization approach". *Journal of Opt. Soc. Am. A*, 6(6):900–905, 1989.

[Nakamura and Asada, 1995] T. Nakamura and M. Asada. Motion sketch: Acquisition of visual motion guided behaviors. In *Proc. of IJCAI-95*, pages 126–132, 1995.

[Nishihara, 1984] K. Nishihara. "practical real-time imaging stereo matcher". *Optical Engineering*, 23-5, 1984.

[Rissanen, 1989] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific, 1989.

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

[Waxman and Duncan, 1986] A. M. Waxman and J. H. Duncan. "Binocular image flows: Steps toward stereo-motion fusion". *IEEE Trans. on PAMI*, 8(6):715–729, 1986.

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

[Whitehead, 1991] S. D. Whitehead. "A complexity analysis of cooperative mechanisms in reinforcement learning". In *Proc. AAAI-91*, pages 607–613, 1991.

Appendix A: Stereo Matching

A coarse-to-fine stereo matching method [Marr and Poggio, 1979; Grimson, 1982] is implemented based on block correlation with SAD criterion by using the real-time visual tracking routines between the left and right images. In the first matching stage, each of a coarse image pair is tessellated into 8 × 5 grids, each grid consists of 16 × 16 pixels and the search area for each grid is 32 × 8 pixels to cope with rough stereo camera calibration. In the second matching stage, each of a fine image pair is tessellated into 14 × 10 grids, each grid consists of 16 × 16 pixels and the center of the search area (16× 8 pixels) for each grid is located at the position where the stereo correspondence in the coarse matching stage is found.

Appendix B: Region Segmentation and Tracking

In order to reach the target area, the robot always needs to identify the area to be tracked which might be partially or completely occluded by other objects. Then, we prepare region segmentation and tracking routines as higher visual behaviors.

Region segmentation

As a region clustering algorithm, we enhance the ISODATA algorithm [Ball and Hall, 1965] by applying the Minimum Description Length (MDL) principle [Rissanen, 1989] as a splitting criterion. The original ISODATA algorithm has a heuristic rule for region separation. We replace this rule with one that searches for the minimum value of description length grounded by information theory.

We assume that the distribution of the pixels in each segmented region can be represented by a mixture density model (a multivariate normal distribution: one kind of Gaussian Model) [Duda and Hart, 1973]. The model can be described by:

$$\boldsymbol{p}_i(\boldsymbol{x}|\boldsymbol{\mu}_i, \sum_i) = \frac{1}{(2\pi)^{d/2}|\sum_i|^{1/2}} \times \exp\left\{\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^t \sum_i^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_i)\right\},$$

where $\boldsymbol{x}$, $\boldsymbol{\mu}$, and $\sum$ denote a column vector, the mean vector, and the covariance matrix, respectively (i and d indicate the cluster number and the size of the vector, respectively). $omega_i$ and

In order to segment the input scene, we form a three-dimensional feature space where:

$$\boldsymbol{x} = (image\_location(x, y), disparity(d)).$$

Every cell in the input scene is grouped into homogeneous regions using our modified ISODATA clustering algorithm. ISODATA clustering is a iterative clustering algorithm whose heuristics were developed by Ball and Hall[Ball and Hall, 1965].

Its basic form consists of a $k-$means clustering procedure embedded in a iterative loop containing heuristics that determine splitting or merging clusters. As a distance measure, we use Mahalanobis distance metric.

Instead of heuristic rules for region clustering [Ball and Hall, 1965], we use MDL principle based on information theory. Roughly speaking, the total description length is a summation of the model description and its fitting error term. The former consists of bits to describe a Gaussian model itself and the mean vectors $\boldsymbol{\mu}$s, and the latter is related to the covariance matrices $\sum$s. If we have many clusters, the former is bigger than the latter, and vice versa. The MDL principle provides the optimal trade-off between them by minimizing the description length from a viewpoint of information theory. For the detailed procedures, please see [Rissanen, 1989].

Figure 10 shows an example of region clustering for the image shown in Figure 3 by our method. An object centered in the image was clustered into one region before applying MDL principle((A) 683.2 bits), but eventually decomposed into two ((B) 451.1 bits) because it is slanted a little bit in depth, therefore it cannot be described with a single disparity.

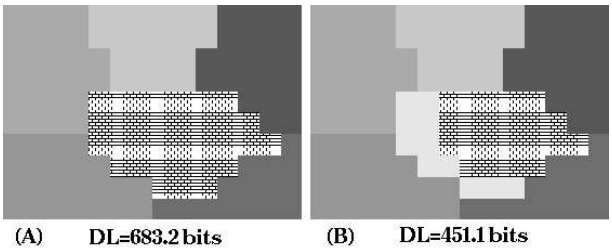


Figure 10: Region segmentation

Region tracking

As a result of region segmentation at time t_i, we have non-overlapping regions each of which consists of several blocks that are units for motion tracking routines between t_i and t_{i+1} by the same method of stereo matching with a rectangle search area (16×16 pixels). Each motion tracker decides if its matching is success or not and tells the motion vector when the matching is success. The total information of all trackers provides new locations of each cluster and occlusion status such as partial or complete occlusions at t_{i+1}. The occlusion status is used in the learning phase for behavior acquisition.

These clusters of which locations have been predicted by the tracking process are seed regions in region clustering at t_{i+1}. Adding the disparity information at t_{i+1}, re-clustering by the ISODATA algorithm with MDL principle is performed to obtain the refined parameters of image location and disparity in each cluster. If the target area is split into two clusters or merged with other cluster, the model of the target area is updated to describe these situations. If the target area is completely occluded, the location is updated by the linear prediction, but the image pattern and disparity information are not updated.