Direct Coupling of Multisensor Information and Actions for Mobile Robot Behavior Acquisition

Takayuki Nakamura, Jun Morimoto, and Minoru Asada
Dept. of Mech. Eng. for Computer-Controlled Machinery
Osaka University, Suita, Osaka 565, Japan
nakamura@robotics.ccm.eng.osaka-u.ac.jp

Abstract

Many conventional methods for multisensor fusion have been based on the predefined selection such as "which sensor is used for which situation," and mainly they have seemed to concentrated on the reconstruction of a 3-D geometric model of the environment supposing such a model would be generally useful. However, optimality of a priori selection by the programmer is not guaranteed, nor is such a geometric model easy to be applied.

This paper proposes a method which enables a mobile robot to acquire a purposive behavior for accomplishing a given task by directly coupling multisensor information and actions through interaction between the robot and its environment. We use reinforcement-learning scheme to formalize such a coupling process. First, we define states described by combinations of various kinds of data provided by different types of sensors and motor commands to the mobile robot. Then, we acquire pairs of robot actions and states suitable for achieving the given goal by using Q-learning algorithm. As a result of learning, the goal-directed behavior is obtained and information needed for the current subtask is automatically selected among multisensor information. The validity of the method is demonstrated by computer simulations and real robot experiments.

1 Introduction

When a mobile robot must operate in an unknown dynamic environment, it becomes necessary to consider to integrate or to fuse the data from different types of sensors so that useful information from the environment can be obtained. The main advantage of using multisensor systems is the increase in reliability and flexibility provided by the redundant and diverse sensor information [1]. However, it is difficult to deal with multisensor integration problem because different sensors provide competing and conflicting information. Therefore, the correlation between the observed multisensor data needs to be found. There have been some approaches to multisensor integration and fusion for mobile robot research.

Moravec [2], Elfes [3], and Wallner et al. [4] developed similar methods for fusing sensor data taken at different times, from different positions, and by different sensor system such as vision and sonar into so-called "certainty grids." Their method aims to obtain a precise map of the environment. Thus, many researchers have tried to construct a geometrical world model in order that information from different sensors may be transformed to a common, abstracted representation, although construction of such a geometrical model requires tedious camera calibration, time-consuming iterative procedures, and the results are often sensitive to noise and must be transformed into the information necessary for the task accomplishment.

Huber and Kortenkamp [5] proposed a robot control system that consists of a tracking system based on a stereo vision and obstacle avoidance system that use the VFH algorithm [6]. This means that the designers have to program some behaviors of the robot and decide the roles of sensors even though such a sensor selection is not always optimal for the robot to achieve the goal.

In this paper, we propose a method for directly coupling multisensor information and the robot's actions through the interaction between the robot and its environment. By our method, our mobile robot can acquire goal-directed behavior that consists of multiple subtasks. The remainder of this article is structured as follows: In the next section, we give a basic idea of our method. Then, we describe our robot and the inherent characteristic of sonar and visual sensor. Next, we explain the method for behavior acquisition and for constructing the state space required for behavior learning. Finally, we give the results of both simulation and real robot implementation and concluding remarks.

2 Our Approach

In many conventional methods for multisensor fusion in the navigation task, first, one investigates how each sensor in the multisensor system can be utilized to accomplish the given task. Then, one can break down the given task into the subtasks in such a way that each subtask can be accomplished by using the predetermined single sensor or a combination of multiple sensors. For example, sonar sensors are used to avoid obstacles and a visual sensor is used to recognize what the object is. However, from the viewpoint of the sensing cost, it is not clear that this combination of sonar and visual sensors is optimal for the robot to accomplish a given task.

Our approach aims to construct an autonomous agent in which both of functions "perception for action" and "action for perception" emerge simultaneously by means of the integration of perception and action. Through the learning process based on perception and action, our robot obtains the correlation between different types of sensors and acquires the goal-directed behavior that seems to consist of multiple behaviors (see **Fig.1**). Here, we formalize such a process by utilizing the reinforcement learning scheme [7].

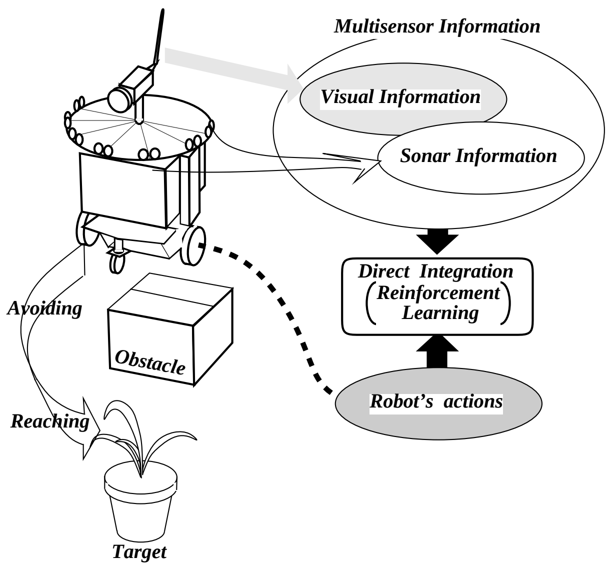


Figure 1: Overview of our method

The task for a mobile robot is to reach a target avoinding an obstacle as shown in **Fig.1**. To perform this task, the robot is expected to use visual and sonar sensors effectively and coordinate two behaviors (target reaching and obstacle avoidance behaviors) in a seamless manner. Here, as multisensor information, we deal with the visual and sonar information. First, the vector quantization technique known as a Kohonen network [8] is applied to reduce the dimensionality of the sonar data stored during the exploration in the environment. Then, sonar data can be described as some representative patterns, which are used as states for a reinforcement-learning method to obtain target reaching behavior. Next, the position and size of the target specified as visual information is also used as states for behavior learning. As a result of learning, the robot obtains the correlation between sonar and visual information and acquires target reaching behavior through the interaction with its environment.

We assume that the environment consists of a target and an obstacle, and that the robot does not know how large and where the target and the obstacle are, any camera parameters such as focal length and tilt angle, or its own kinematics/dynamics.

Our method has the following two advantages:

- We do not need to determine strategies for selecting appropriate sensors to use in response to the change of environment. Through our learning process, the strategy for sensor selection is automatically acquired.

- In order to integrate multiple sensors, we do not need any transformation between the information obtained from different sensors through some geometric world model as a common abstracted representation. Through the learning process, our mobile robot behave as if it had been given such an abstracted represetation.

3 Our Robot and Sensor Characterization

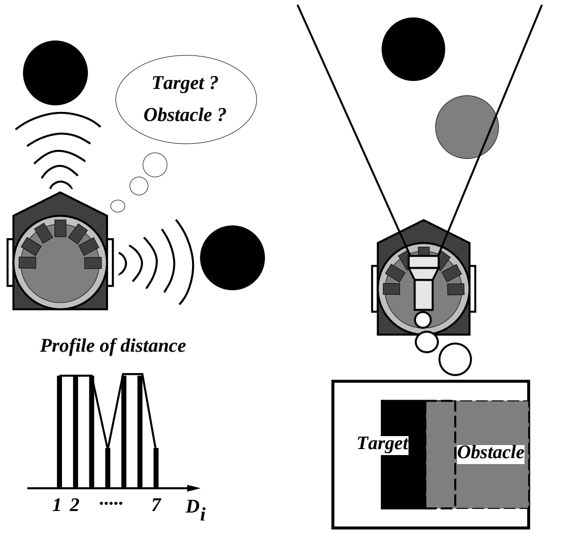
Our robot has a Power Wheeled Steering (hereafter PWS) system driven by two motors into each of which we can send a motor command, independently. In our experiment, we quantized each motor command $\omega_{l(r)}$ into three levels which correspond to forward, stop, and backward, respectively. Totally, the robot has 9 actions.

The robot is equipped with a ring of 12 ultrasonic ranging sensors (ranging from 0.0 to 250 cm) which have high accuracy for incident angle less than 20° from the surface normal. Here, the robot uses 7 sonar sensors on the front and the side, and does not use the 5 on the back. The robot is also equipped with a CCD camera. This camera looks forward and down. The tilt angle is about 10°. These sensors have their inherent characteristics as follows:

- **Sonar Sensor**
  Using 7 sonar sensors, our robot can sense its forward surrounding environment in robot centered polar coordinates as a profile of the distance D_i $(i = 1 \sim 7)$ as shown in **Fig.2(a)**. Each sonar sensor in the ring has a field view of roughly 30°. Sonar sensors cannot identify what the object is (the target or others).

- **Vision**
  Image processing procedures provide the position and the size of the target in the image as visual

information, even if the pattern of the target is deformed by occlusion (see **Fig.2(b)**). However, it cannot detect obstacles, because it is not given how an obstacle can be seen.



(a) Sonar Sensor (b) Vision

Figure 2: Characteristics of each sensor

4 Behavior Acquisition by Fusing Sonar and Vision

As a method of learning for behavior acquisition, we use one step Q-learning [7], 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 sonar and visual sensors described above. In this section, we explain the basics of behavior learning and 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 [7] has attracted much attention as an implementation of reinforcement learning because it is derived from dynamic programming [9]. The following is a simple version of the 1-step Q-learning algorithm we used here. If the process is Markov and enough exploration is done, the acquired policy will converge to the optimal one.

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 $Q(s,a)$:
$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in A} Q(s',a')).$$
5. Update the policy f:
$$f(s) \leftarrow a \quad \text{such} \quad \text{that} \quad Q(s,a) = \max_{b \in \boldsymbol{A}} Q(s,b)$$

4.2 Construction of State Space

4.2.1 A State Space for Vision

We define a state of the target in the image in terms of its position and size detected by the image processing. The state of the target, $\boldsymbol{S}$ in the image is quantized into 9 sub-states, combinations of three positions (left, center, and right) and three sizes (large (near), medium, and small (far)). We add two lost situations (target is lost into the left side or the right side) in the state space. Totally, we have 11 states in the state space for vision. (see **Fig.3**).

4.2.2 A State Space for Sonar

Here, we describe two-phase approach for constructing the state space for sonar. Based on the region perceivable by sonars, the state space for sonar is constructed by segmenting the space around the robot into relevant sensory regions (see **Fig.4**). Each sector indicates the space measurable by each sonar sensor. The area in front of the robot is divided into two regions as follows:

- $\forall D_i > D_T$ $\rightarrow$ **Far zone**
- $\exists D_i < D_T$ $\rightarrow$ **Near zone**,

where D_i denotes a reading provided by sonar sensor i. The threshold between these zones D_T is derived from the robot's velocity and the minimum range of sonar sensors.

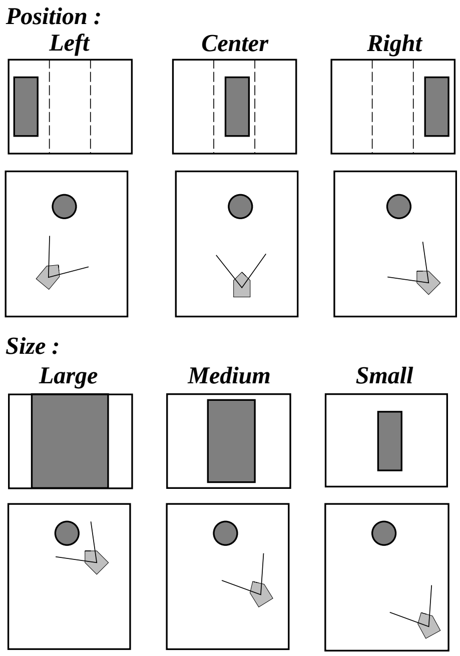


Figure 3: States discriminated by the visual sensor

(A) Far zone

The sensory profile cosists of $N(=7)$ range values each of which is a real number. Since the sensory profile at each moment has a large amount of the information about its surroundings, we have to compress the information. We use the vector quantization technique known as a Kohonen network [8], which can perform classification, data compression and so on. The characteristic of a Kohonen network is that the original topological structure of the input space is well preserved in the output space, that is, the output vector turns out to be similar if the input profile is similar and that the weighting vectors representing how strongly the input and output nodes are connected show the represetative patterns of input space.

In our implementation, the Kohonen network has 7 nodes input layer and 96 nodes output layer. The address of the winner unit in the output layer denotes the output vector of the network that corresponds to the state represented by sonar sensors, which is used for behavior learning (see **Fig.4**.).

Fig.5 shows 16 examples of the represetative patterns of the profile of distance detected by sonar sensors after training, where a reading by each sensor is denoted by the distance from the center of the robot (a solid square) to the small solid circle in each sector.

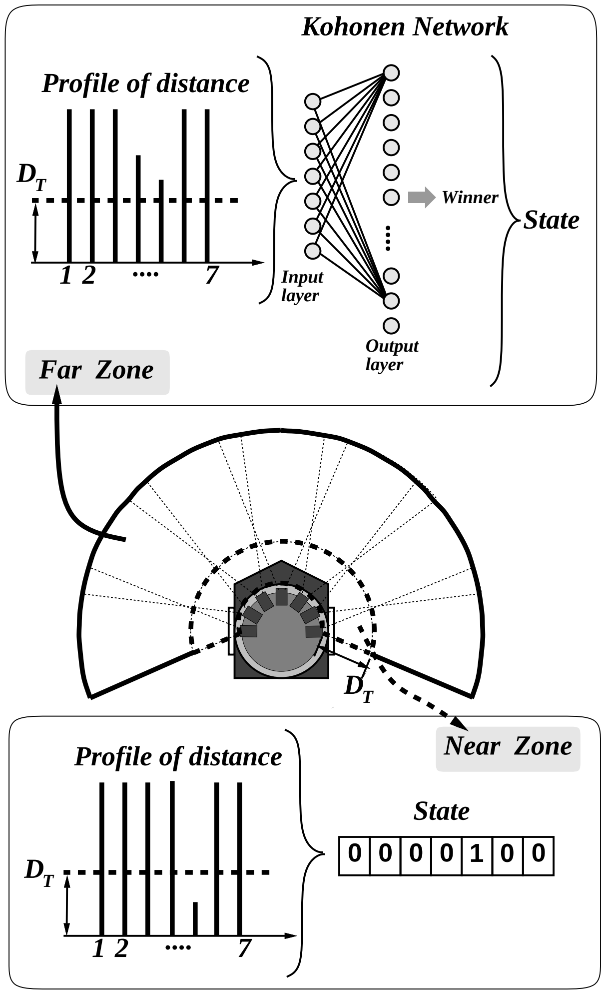


Figure 4: Sensory Region and Discrimanating method for Sonar sensors

(B) Near zone

In the near zone, we use only two range bins (these values range from 0 (empty) to 1 (occupied)) for each sensor (see **Fig.4**). The $N(=7)$ sonar range bins (7 bits of information) coming from 7 sonar sensors describe the state for sonar in the near zone. That is, there are 2^7 sonar states.

To summarize, totally, we have $11 \times (96+128) = 2464$ states in the set $\boldsymbol{S}$ generated by visual and sonar sensors.

5 Experimental Results

5.1 Simulation

We performed computer simulations with the following specifications. The field is a $300[cm] \times 300[cm]$square. The target is a cylinder with a diameter of 40 $[cm]$. The

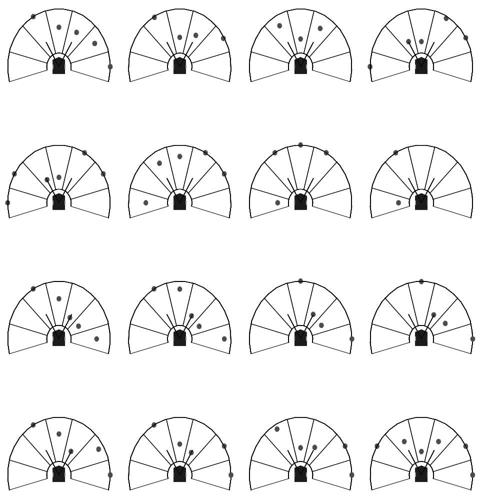


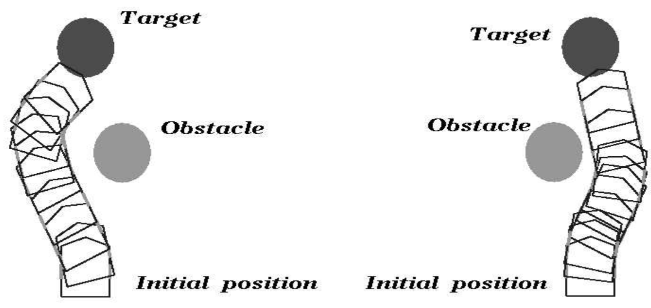
Figure 5: Examples of representative sonar patterns

robot is 40 [cm] wide and 40 [cm] long. The camera is mounted on the robot and looks toward the floor (10 degree tilt). Its visual angle is 60 degrees. These and other parameters such as friction between the floor and the tires are chosen to simulate the real world. The target is fixed in the environment. Various kinds of the configuration of the target and the obstacle are set in the learning process. The threshold between zones D_T is set to be 20 [cm]. We give a reward value 1 when the robot reached the target, or a reward value 0 otherwise. We choose an action among the $\boldsymbol{A}$ based on the bolztman distribution. The learning process ends if the success rate exceeds 95%. We define the success rate as (# of successes)/(# of trials) $\times$ 100(%). The learning has converged after about 24 hours running on Sun SPARCstation-20.

Fig.6 shows the target reaching behavior acquired by our method. As shown in **Fig.6**, the robot tried to avoid in the early part of the trial, then it ran to the target quickly. It seems that the robot reconstructs geometrical sturcture of its surrounding and plans a safe path using the reconstructed local map.

5.2 Real Robot Experiment

Fig.7 shows a configuration of the real mobile robot system. Our robot is a mobile platform (Yamabico) controlled by MVME167/VxWorks OS through RS232C. In this implementation, the vehicle speed is set to be about $5cm/s$. 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 Ethernet. The



(a) case 1 (b) case 2

Figure 6: Target reaching behavior in computer simulation

video signal from a color CCD camera mounted on the robot is sent to Datacube DIGICOLOR and MaxVideo 200 each of which is a real-time pipelined video image processor. In order to simplify and speed up the image processing time, we painted the target in yellow. In **Fig.7**, a picture of the real robot with a color CCD camera and sonar sensors is shown.

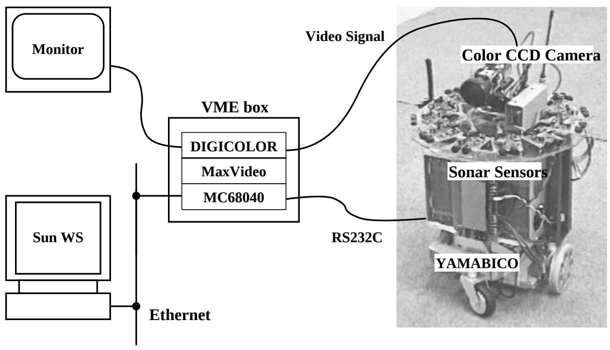


Figure 7: A configuration of the real mobile robot system

In the real robot experiment, we apply the learned policy in a computer simulation to a real situation. **Fig.8** shows a sequence of the target reaching behavior of the real robot in which the robot succeeded in reaching the target avoiding the obstacle. **Fig.9** shows 4 snapshots in **Fig.8**, where each figure $\#step1 \sim \#step4$ shows a bird's eye view of the environment, the image taken by a camera on the robot, and the result of the target detection at each time step (the position and size of the target are calculated in real-time (1/30 seconds)). At the bottom of each figure, the state that the robot discriminated by visual and sonar sensors and control commands to left and right motors are shown. The discriminated state consists of three substates: two for target position (Left (Lt), Center (C), Right (R)) and size (Large (Lg), Medium (M), Small (S)), and one for the state number discriminated by sonar. Action commands consist of a combination of two independent motor commands (Forward (Fw), Stop (St), Backward (Bw)).

These figures show the following situation: initially, the target image (yellow box) is specified. The robot starts to pursue the target. While the robot is pursuing the target, the target disappears gradually due to an obstacle. In this state, the robot avoids the obstacle. Then, the robot runs to target quickly. Finally, the robot succeeds in reaching the target.

Figure 8: Target reaching behavior in the real world

6 Discussion and Future Works

We have shown the validty of the proposed method for directly integrating multisensor information and robot actions with computer simulation results and preliminary real experiments. Since we deliberately construct the state space for sonar, the learning phase can enjoy the compact state space without describing all states of sonar sensors explicitly. Optimality of the state space designed by the programmer is not guaranteed to the robot. Therefore, the method of the adaptive construction of state space for behavior learning would be necessary.

Our robot sometimes fails, because we still have a gap between the computer simulation and the real system. For example, in real world, if a sonar reading contains a error due to specular reflection of ultrasonic wave, our robot could not discriminate the state correctly and would collide with the obstacle. The reason is that we have not made the real robot learn but only execute the optimal policy obtained by the computer simulation. To cope with this problem, we are planning to make the real robot begin to learn from the policy obtained by computer simulation.

Here, although we do not deal with the environment where the target and/or the obstacle are/is moving, we can cope with such environment by adding the state variable by which the moving target and/or obstacle can be described in the state space.

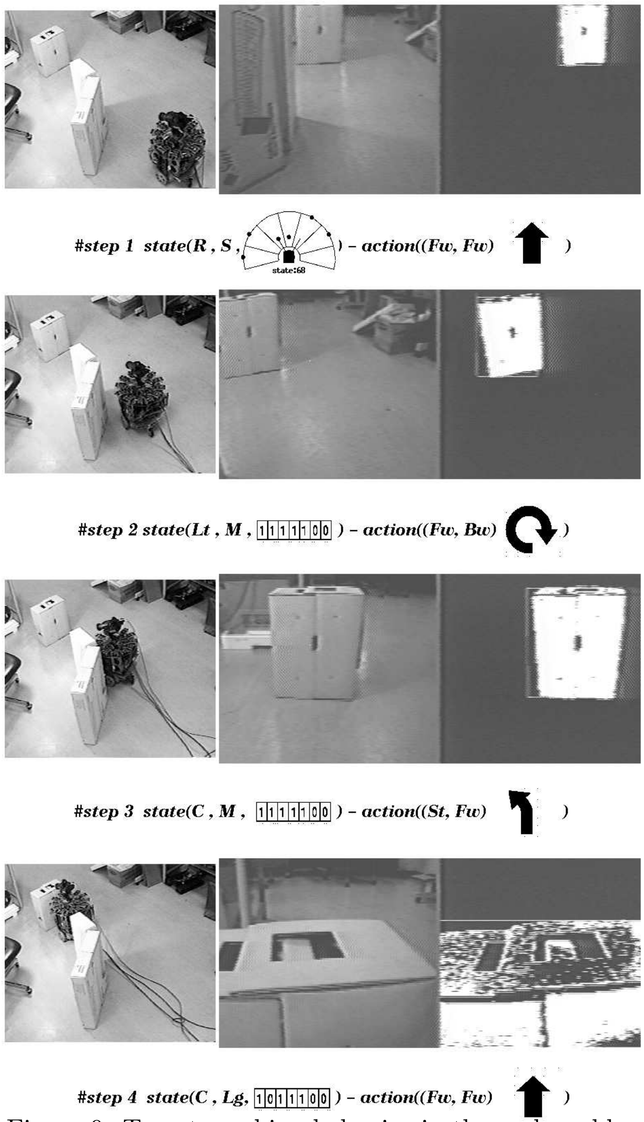


Figure 9: Target reaching behavior in the real world

References

[1] R. C. Luo and G. K. Kay. "Multisensor Integration and Fusion in Intelligent Systems". *IEEE Trans. on Systems, Man and Cybernetics*, 19(5):901–931, 1989.

[2] H. P. Moravec. "Sensor fusion in certainty grids for mobile robots". *AI Magazine*, 9(2):61–74, 1988.

[3] A. Elfes. "Using occupancy grids for mobile robot perception and navigation". *IEEE Computer Magazine*, 22(6):46–57, June 1989.

[4] F. Wallner R. Graf and R. Dillmann. "Real-time map refinement by fusing sonar and active stereo vision". In *Proc. of 1995 IEEE Int. Conf. on Robotics and Automation*, pages 2968–2973.

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

[6] J. Borenstein and Y. Koren. "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots". *IEEE Journal of Robotics and Automation*, 7(3):278–288, 1991.

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

[8] T. Kohonen. "Self organizing formation of topologically correct feature maps". *Biological Cybernetics*, 43:59–69, 1982.

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