Learning Policies for Attentional Control *

LUIZ M. G. GONÇALVES[1,2], GILSON A. GIRALDI[2], ANTONIO A. F. OLIVEIRA[2], AND ROD A. GRUPEN[1]

[1]Laboratory for Perceptual Robotics - Dept of Computer Science
University of Massachusetts (UMASS), Amherst MA 01003 USA
Phone (413) 545-3143 and FAX (413) 545-1249
(lmarcos,grupen)@cs.umass.edu
[2]Laboratório de Computação Gráfica - COPPE Sistemas
Universidade Federal do Rio de Janeiro (UFRJ), CP 68511, Rio de Janeiro, RJ 21945-970
(lmarcos,giraldi,oliveira)@lcg.ufrj.br

Abstract. In this work we propose two behavioraly active policies for attentional control. These policies must act based on a multi-modal sensory feedback. Two approaches are used to derive the policies: the first one follows a simple straightforward strategy and the second one uses Q-learning to learn a policy based on the perceptual state of the system. As practical result of both algorithms, a robotic agent is capable to select a region of interest and perform shifts of attention focusing on the selected region. Then, a multi-feature extraction can take place allowing the system to identify or recognize a pattern representing that region of interest. Also, the policies have the desired property that all objects in the environment are visited at least once, although some of them can be visited more. In this way a robotic agent can relate sensed information to actions, abstracting and providing a feedback (categorization and mapping) for environmental stimuli.

Keywords. Attentional control, Pattern categorization, Q-learning.

1 Introduction

In this work we propose behavioraly active policies for attentional control. More especifically, two policies were developed: one follows a straightforward strategy and another one was carried out by using Q-learning. Both policies allows a system perform in such a way that vision and touch sensory information are integrated in a behaviorally cooperative active system. The result of that integration can be used by a robotic agent to perform real-time tasks. The relative importance of touch and vision is assumed to be context dependent. They work in parallel, providing (ambiguous or complementary) information to a decision system, which is responsible to give adaptive responses (actions) to environmental stimuli.

We are interested in a vision-touch system which is able to foveate (verge) the eyes onto an object, keep attention on the same object until more information necessary to perform a given task is obtained, move subsequently the arms to reach and grasp that object, and shift its focus of attention to another object once the current one is no more of interest. To validate such a system, we tested it to execute a task of inspection which involves the actions mentioned above. To perform that task a robot agent must learn how to construct an incremental map of its environment, dealing with new or already known objects and how to classify (categorize) an object that has just been detected. Q-learning will be employed to enable that system to take adequate actions in function of its perceptual state. In particular, both policies obtained have the desired property that all objects in the world are visited at least once, although some of them can be visited more.

*This work is supported by FAPERJ and CNPQ/Brazil, and by NSF under NSF CDA-9703217 and KCS-9704530

2 Related Work

Q-learning [5, 6, 7] has been widely used to generate behavioral policies applied on the control of multi-decision tasks. It is basically a dynamic programming solution for control problems that can be modeled as Markovian Decision Processes (MDP). A Markovian process is a stochastic process where only the present state and not the past, influences the future (see [1] for more details). A Q-table is one whose rows are relative to states, the columns to actions and where a general element $Q(s,a)$, called a Q-value, is an evaluation of the utility of an agent take action a when it is in state s. After the execution of action a, $Q(s,a)$ is usually updated by means of the following expression:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a))$$

In the above expression, r is the instantaneous reward action a receives, s' is the state resulting of a and the set A consists of all actions a' that are possible in s'. The constants α and γ are respectively the learning rate and the learning discount factor usually made 0.5 and 0.1 respectively. From that expression one can conclude that a transition in the state space is fully characterized by a vector

(s, a, r, s'). Also, suppose that for all pairs (s, a), $Q(s, a)$ is updated infinitely. Then, it can be proven that $\forall\ (s, a)$, $Q(s, a)$ will converge with probability 1 to the best possible reward for the pair (s, a)[6].

In recent works reinforcement learning has been applied to the case where the state is characterized by a representation of behavioral alternatives obtained with combination of native control primitives and sensorimotor resources. This structure has been successfully applied to tasks of foraging in simulated biological systems [4], to tasks of reaching, grasping, and manipulating objects [2], and for making a multi-legged robot to learn walking gaits [3]. These researches have demonstrated that reinforcement learning and in particular, Q-learning techniques are specially appropriate in the case where an agent must learn a specific task. We applied this approach to obtain policies for coordinating the behavior of a hand/eye integrated system. In our case, the task can be object recognition and identification for inspection (or surveillance), spatial orientation and eventually navigation or path-planning. We adopt an active behavioral strategy which provides a way to interact with dynamic environments. A map of the scene is incrementally constructed and dynamically updated. It contains the position and orientation of all recognized objects and once it has been obtained, the robot agent is ready to perform the specific tasks indicated earlier.

Figure 1 shows the functional aspects of an integrated eye-arm system architecture (see [8] for a better functional description). We call $perceptual\ buffer$ to the union of the visual and the haptic ones. Briefly, input from each sensory system is organized into spatially-indexed areas of the perceptual buffer which are called Regions of Interest (ROIs). Topological/spatial information stored in the ROIs is then mapped to features like intensity, texture, shape, size and weight. These features are used as a pattern activation code in a central $associative\ memory$ which is a BP neural-network [5] with a winner-take-all mechanism. That network matches the pattern to an address in the long term memory where information about the objects already recognized and the environment is stored.

The $attention\ window$ is a selective structure that allows the access to the information provided by a single ROI. A $pre\text{-}attentive$ mechanism operates at low level, computing activation values for the ROIs which will be used by a winner-take-all mechanism to decide where to focus the attention. These activation values are normalized functions (assuming values in $[0.0, 1.0]$) which depends on the retinal size, the average intensity, the visual motion, the tactile response information, and the mapping status. This last is a flag which is set, for a ROI, when that ROI becomes the focus of attention. Finally, the $attention\text{-}shift$ mechanism is responsible for selecting a new region and effectively move the attention window to it.

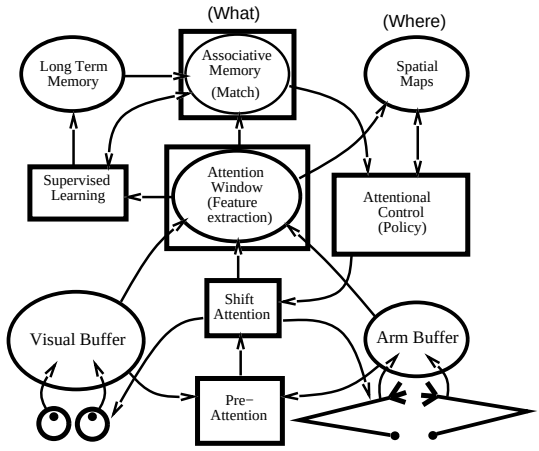

Figure 1: System Control Architecture

3 The Simulation Environment

The simulated robot "Roger-the-Crab" presented in Figure 2 has five controllers (one for the neck, two for the eyes and two for the arms) integrated in a single platform. Roger's world can be specified and dynamically changed. For each pixel in Roger's retinas an intensity value is calculated in function of the radiance of the world patch corresponding to it. To compute that value a Phong illumination model is used. Following, a Laplacian of Gaussian filter, operating at three levels of resolution, is applied to the retinas and the result is transfered to the visual buffer.

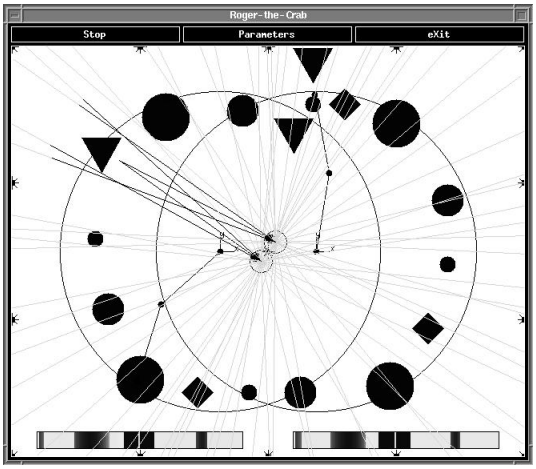

Figure 2: Roger the Crab interface. The thin lines separate regions of interest segmented from visual input data.

Also, a compensation for the gravity and other ambient effects and basic robot kinematic and dynamic equations are determined to be used by the servo controllers of an arm or an eye. Independent controllers run concurrently for each eye and each arm. Coordinating these controllers is basically a question of determining which set of them will be run at a given time. This is decided by applying a straightforward or a Q-learning policy, resulting in a model for attentional control described in the next section.

In the remainder of this section we will describe how Roger transforms the filtered data stored in the visual buffer into the features it uses to compare objects.

3.1 Feature Extraction

Intensity and texture are computed directly from the result of the LoG filtering applied to Roger's retinas. Stereo disparity is obtained from the correlation between pairs of pixels, one in the filter response of each retina. Since we have three kernel diameters for the LoG filter, a 3D vector is generated for each one of intensity, texture, and stereo. For each L∘G resolution, $k = 1, 2, 3$, the number of pixels (M) where stereo-disparity, texture and intensity are computed is independent of the size of the ROI.

Each intensity value $I_i^{(k)}, k = 1, 2, 3$ (see Equation 1 below) is calculated as an average in the vicinity of the corresponding L∘G response $G_{i+m}^{(k)}, m = 1, \ldots, M$ normalized by a maximum value $G_{Max}^{(k)}$ in all these responses. Each component of the texture vector $T_i^{(k)}$ (see Equation 2) is the variance in the vicinity of an L∘G response also normalized by $G_{Max}^{(k)}$. In a similar way, $S_i^{(k)}$ (see Equation 3) is the variance in the vicinity of the stereo measurements $s_i^{(k)}$ made for resolution k, normalized by $s_{Max}^{(k)}$, the maximum value of those measurements in all resolutions. Although the vector $S = (S_i^{(1)}, S_i^{(2)}, S_i^{(3)})$ is not the best characterization of an object shape, it suffices to differentiate objects among a reasonable set of them having close shape characteristics. The size D (Equation 4) of an object is also extracted from stereo measurements. It is normalized between zero and a maximum length which is arbitrated in function of lenses and cameras geometry. Analogously, the weight W (equation 5) is extracted from the arms sensors. It is normalized by the maximum weight W_{Max} that the arms can lift.

$$I_i^{(k)} = \frac{1}{M} \sum_{m=1}^{M} \frac{G_{i+m}^{(k)}}{G_{Max}^{(k)}} \tag{1}$$

$$T_i^{(k)} = \frac{1}{M} \sum_{m=1}^{M} \left(\frac{G_{i+m}^{(k)}}{G_{Max}^{(k)}} - I_i^{(k)} \right)^2 \tag{2}$$

Object	Intensity	Size	Weight
01	99	30	15
	0.83	0.45	0.72
02	79	30	10
	0.65	0.43	0.51
03	69	30	10
	0.56	0.44	0.47
04	59	30	5
	0.47	0.44	0.24

Table 1: World and perceived feature values.

$$S_i^{(k)} = \frac{1}{M} \sum_{m=1}^{M} \left(\frac{s_{i+m}^{(k)}}{s_{Max}^{(k)}} - \frac{1}{N} \sum_{n=1}^{N} \frac{s_{i+n}^{(k)}}{s_{Max}^{(k)}} \right)^2 \tag{3}$$

$$D = \frac{d}{D_{Max}} \tag{4}$$

$$W = \frac{w}{W_{Max}} \tag{5}$$

For some of the objects represented in Figure 2, Table 1 shows the values (odd rows) of three object properties and those of the corresponding features (even rows), computed by Roger. These last ones will be the input to the matching process in the associative memory. In the situation displayed in figure 2, the right arm and the controllers of the neck and the eyes have converged and the information about the object in focus (a triangle) is going to be extracted from the ROI containing it and subjected to a matching process in the associative memory.

4 Control Policies for Atention

In this section we will discuss the two methods applied for attentional control. The first method uses a simple straightforward strategy. The second method uses Q-learning to develop the control policy.

4.1 A Straightforward Algorithm

In the algorithm described below, we assume that the $pre\text{-}attention$ mechanism runs immediately after each movement and that feature extraction and the matching process are automatically performed each time the controllers converge.

$Step\ 0$: Initialize the back-propagation network (associative memory) synaptic weights, and start the concurrent controllers of arms, neck, and eyes. Also, initialize the associative memory.

$Step\ 1$: Re-direct the attention (or keep the present attention window).

Step 2: If an object is identified after that shift of attention, *update* the spatial map (set to 1 the "mapping status" of the ROI corresponding to the object) and return to *Step* 1. Otherwise, apply *visual-improvement*.

Step 3: If an object is identified only with *attention-shift* and *visual-improvement*, *update* the spatial map and return to *Step* 1. Otherwise, apply a *haptic-improvement*.

Step 4: If the object is identified after that *haptic-improvement*, *update* the spatial map and return back to *Step* 1. Otherwise call a supervised learning module which stores a new set of features in the long-term memory and re-trains the BP network. After that, *update* the spatial map and return to *Step* 1.

We must observe that a *haptic-improvement* after the shift of attention can be more effective than a visual one to improve the features quality and produce a positive identification.

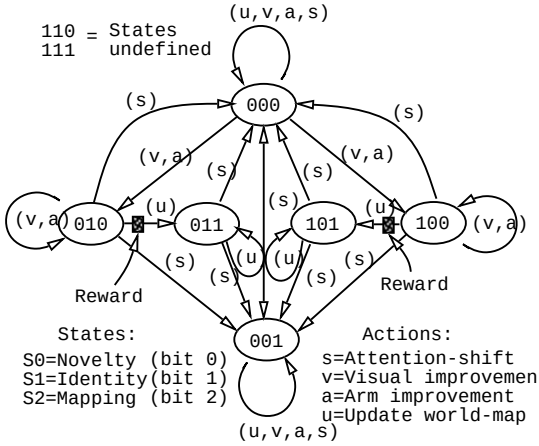

Figure 3: Finite State Machine with the state-space and set of actions. Arrows indicate transitions between states with the letters in between parenthesis showing the action which is being performed. Circles represents reference states due to convergence events in the activated actions. The result is a behavioral policy for attentional control.

4.2 A Control Strategy using Q-learning

Figure 3 shows a finite state machine (FSM) representing the state space where the Q-learning process is performed. Each state is characterized by a sequence of bits. Each bit is relative to a property of the current ROI like *novelty*, *identity*, and *mapping*. *Novelty* and *identity* are set to zero just after a shift of attention, previous to a match. *Identity* is set to 1 if a match is detected in the associative memory and *novelty* is set to 1 if no match occurs after all possible trials (So, we have a new object). Initially we do not know whether an object is a new one or has already a representation in the associative memory. Then, *identity* and *novelty* are set to 0 at start. After a matching process is concluded we have three possible cases:

a) The object is identified as one which has already a representation in the long-term memory. In this case *identity*=1 and *novelty*=0.

b) A new object has been discovered. In this case *identity*=0 and *novelty*=1.

c) It is not possible to conclude whether the object is new or has already a representation. In this case both variables remain 0 and a *visual* or *haptic-improvement* is then performed to get more (or better) features of the object. As *identity* and *novelty* cannot be simultaneously 1, the number of the FSM states reduces to 6 as shown in Figure 3. The last state (*mapping*) can be retrieved from the pre-attentional maps.

The set of actions which determines a transference from a state to another refers to the physical movements of the arms and eyes, and to a *map-updating* operation. This last one is performed after an object is classified as already identified or new. The physical actions are: a complete *attention-shift*, a *visual-improvement* (a fine vergence adjustment or a search for individual characteristics), and a *haptic-improvement* (that is, an arm move to get a better tactile input or to evaluate object weight). We have not attributed rewards to states but only to transitions between them. This avoids a policy which makes the system remains forever on a state offering a reward. As shown in 3 rewards are given only to the *map-updating* actions determining the transition from state 100 to 101 and from 010 to 011. In practice, in the case considered here a reward will be given in the three following circumstances (the first one is a subsequence of the other two):

a) When the robot finds un-mapped positions in the pre-attentional maps;

b) When objects without representation in the associative memory (new objects) are detected.

c) When a positive identification of an object occurs.

5 Experimental Results

In the experimental tests, both approaches, Q-learning and the straightforward strategy discussed in subsections 4.2 and 4.1 respectively, have been applied to a task of inspection and the results have been acceptable. The Q-learning approach has performed slightly better (this will be discussed below). If an object is not identified only by visual improvement, the arm trial is accomplished by moving one of the arms and say measuring the object weight. After all trials, in case of no identification, the supervised learning procedure dynamically updates the BP network. After all regions of interest are visited, the eyes remain in a vigilant state. We have defined another activation value (*interest*) that is set to zero when a ROI is first settled in the pre-

attentional maps. This value increases with time and is only reset when attention is focused on the ROI. Choosing the ROI with the largest interest to be the focus of attention we force the system to eventually focus any ROI. This allows the detection of environment changes and puts the robot into a behavioral active state. Without this mechanism, the eyes would remain at rest until a change occurs in the visual-field of the robot.

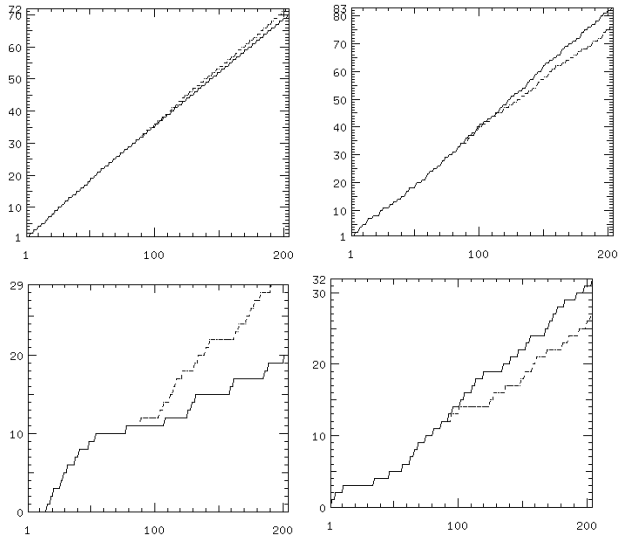

Figure 4: Partial evaluation. Horizontal axes show number of control cycles. Vertical axes as folows: (upper left) number of attention shifts realized, upper line (dotted) is Q-learning and lower line is simple approach; (upper right) number of arm/eye improvements, upper line is simple approach and lower line (dotted) is Q-learning; (lower left) number of positive identifications, upper line (dotted) is Q-learning and lower line is simple approach; (lower right) number of new objects detected, upper line is simple approach and lower line (dotted) is Q-learning.

Performance is something quite difficult to measure in this kind of task because we do not have ways to define an ideal situation to compare with the algorithms. So, some evaluation of the algorithms applied to a same environment (with the same light definitions, and the same objects) are shown in figure 4. The Q-learning approach has performed more attentional shifts, less eye/arm improvements, more positive identifications, and a few less new-object detections than the simple approach. Figure 5 shows another evaluation for the number of map updates realized by using both approaches. As in the Q-learning approach rewards are given for map update, it was expected that the Q-learning approach performs slightly better in this graph. Some analyses can also be made using Table 2. This result was obtained also using the same environment for both

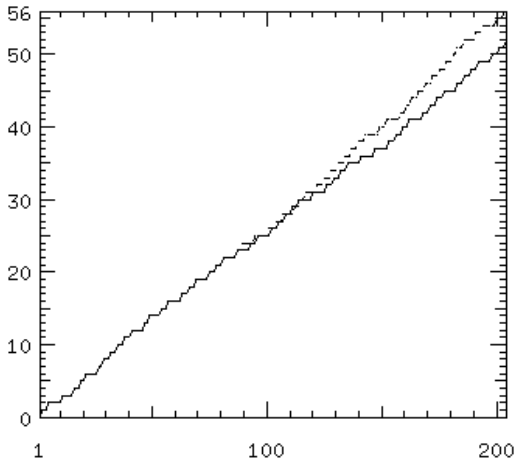

Figure 5: Overall evaluation. Horizontal axis shows number of control cycles. Vertical axis shows number of map updates. Upper (dotted) line is Q-learning and lower (solid) line is simple approach.

Appr	Att Shift	Eye/Arm Improv	Posit Ident	New Obj	Map Update
(1)	72	76	29	27	56
(2)	70	82	20	32	52

Table 2: Data obtained after 204 control-loops using the Q-learning(1) and simple(2) approaches: number of attention shifts performed, number of eye/arm improvements, number of objects positively identified, number of new objects detected, and number of updates realized (objects settled) in the pre-attentional maps.

methods. This particular experiment was interrupted after 204 control loops, for comparison effect. Similar results were obtained from other experiments in which the system runs until no more new representations were detected. In these, all regions of interest in the environment were visited (looked at by Roger). By the results all we can see is that both methods worked well in the inspection task involving attention and categorization. At this point, we can not say for sure that the Q-learning approach works better. In order to afirm that, other tasks with a more complex state space and set of actions have to be tested.

Figure 6 shows the convergence of the Q-learning process from which the control policy is derived. For this simple task, it takes no long to complete the trainning. We can not tell how long it takes to train in number of seconds because in the experiments done the system uses a simulated

clock. Each of the control cycles involves one action or the transference from one state to another in the finite state machine described in section 4.2.

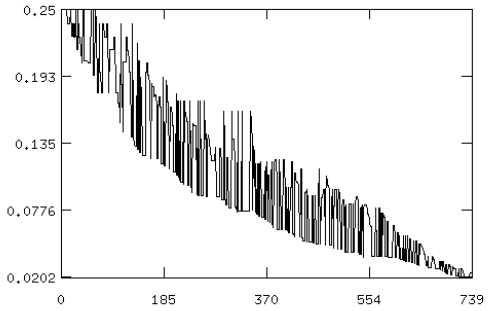

Figure 6: Q-learning convergence (number of trials versus temporal-differential error).

6 Discussion, Conclusion and Future Work

We have built an useful attentional mechanism which was successfully used by a multi-modal sensory system in the execution of inspection tasks. Besides using only visual and haptic information in this work, similar strategies can be applied to a more general system involving other kind of sensory information (for example, adding auditory information). Also, other tasks involving recognition and identification of objects can use a similar Q-learning approach, with few changes in the state space. Immediate extensions of this work are to increase the state space and/or the set of actions. We can also define other hierarchical tasks using a sub-set of the same state-space and actions. Then, it would be possible to derive various policies, each one appropriate for a given task.

The multi-modal system that has been implemented in a simulated environment is currently also implemented on a vision hardware platform composed of a stereo-head and Datacube image processing devices [9]. The next step is to test this learning architecture on the real environment and to measure its performance in a variety of tasks. Also, it will involve two robotic arms with hands.

We have chosen a task of inspection to test the approaches because the bottom-up aspects of attention can be fully explored in this task. Furthermore, questions relative to the current environment state can be addressed by a simple analysis of dynamic maps. As more advanced general tasks, robots can use extensions of the basic procedures developed here to learn how to navigate between different rooms in a building and even to learn facts, history, and other useful information.

We believe that the top-down aspects of attention can also be explored by using the same architecture, but deriving other policies. Thus, a final possibility for future works is to derive through reinforcement learning, policies to control the attention considering these aspects.

References

[1] A. PAPOULIS. Probability, Random Variables, and Stochastic Processes, MacGRAW-HILL, 1991.

[2] J. COELHO and R. GRUPEN. A Control Basis for Learning Multifingered Grasps, Journal of Robotic Systems 14(7):545-557, 1997.

[3] M. HUBER and R. GRUPEN. A Feedback Control Structure for On-line Learning Tasks. Journal of Robotics and Autonomous Systems 22(3-4):303-315, Dec. 1997.

[4] E. ARAUJO and R. GRUPEN. Learning Control Composition in a Complex Environment. Proceedings of Int. Conf. on Simulation of Adaptive Behavior (SAB'96). Cape Cod, MA, September, 1996.

[5] D. H. BALLARD. An Introduction to Natural Computation. The MIT Press, Cambridge, MA, 1997.

[6] C. WATKINS. Learning from Delayed Rewards. PhD thesis, King's College, Cambridge, UK, 1989.

[7] R. S. SUTTON and A. G. BARTO. Reinforcement Learning: an Introduction. The MIT Press, Cambridge, MA, 1998.

[8] L. M. G. GONÇALVES, R. A. GRUPEN, and A. A. F. OLIVEIRA. A Control Architecture for Multimodal Sensory Integration. Proc. of XI Int. Conference on Computer Graphics and Image Processing (SIBGRAPI'98). Rio de Janeiro, Brazil, October, 1998.

[9] L. M. G. GONÇALVES, D. WHEELER, R. A. GRUPEN, and A. A. F. OLIVEIRA. Towards a Framework for Robot Cognition. Submitted to the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation.