Neural Mechanisms for Learning of Attention Control and Pattern Categorization as Basis for Robot Cognition *

Luiz M. G. Gonçalves
Laboratoire d'Analyse et d'Architecture des Systemes
7, Avenue du Colonel Roche 31077 Toulouse France
lmgarcia@laas.fr

Cosimo Distante
Innovation Engin. Dept. University of Lecce
Via Monteroni 73100 Lecce Italy
cosimo.distante@unile.it

Antonio A. F. Oliveira
Universidade Federal do Rio de Janeiro
CP 68511, 21945-970, Rio de Janeiro, Brasil
oliveira@lcg.ufrj.br

David Wheeler
Roderic A. Grupen
University of Massachusetts (UMASS/LPR)
140 Governor's Drive, Amherst MA 01003 USA
{dwheeler,grupen}@cs.umass.edu

Abstract

We present mechanisms for attention control and pattern categorization as the basis for robot cognition. For attention, we gather information from attentional feature maps extracted from sensory data constructing salience maps to decide where to foveate. For identification, multi-feature maps are used as input to an associative memory, allowing the system to classify a pattern representing a region of interest. As a practical result, our robotic platforms are able to select regions of interest and perform shifts of attention focusing on the selected regions, and to construct and maintain attentional maps of the environment in an efficient manner.

1 Introduction

We present behaviorally active neural mechanisms for attention control and for pattern categorization developed as support to robot cognition. The proposed mechanisms support complex, behaviorally cooperative, active sensory systems as well as different types of tasks including bottom-up or top-down attention. Our goal is to develop an active system able to foveate (verge) its eyes onto a region of interest, keep attention on the same region as necessary, move subsequently the arms to reach and grasp an object, and shift its focus of attention to another region. Also, our robotic agents must learn how to construct incremental attentional maps of their environment, dealing with new or already known objects, and how to classify (categorize) a pattern that has just been detected. Image processing techniques provide data reduction and feature abstraction from the visual input data. Based on the features of this "perceptual buffer" and on the current robot pose and functional state, the system is able to define its perceptual state. So, the robot can make control decisions based on the information contained in the perceptual state, selecting the right actions in response to environmental stimuli. This approach is reactive, choosing actions based on the current perception of the world rather than by using a geometric model as in traditional planning techniques. By reducing and abstracting data, our system performs fewer computations and substantially improves its performance.

Several new approaches have been suggested using multi-feature extraction as basis for cognitive processes. Van der Laar [10], Itti [2], and Milanesi [5] use a transfer function to gather information from the feature maps to construct a salience map that governs attention. Rybak et al. [8] treat perception and cognition as behavioral processes. Rao and Ballard [6] provide a set of operators based on Gaussian partial derivatives for feature extraction, which are motivated by biological models. Similar feature models are used by Viola [11] in his belief network approach. In a purely descriptive work, Kosslyn [4] suggests that features extracted from visual images are combined with mental

*This work was partially supported by NSF under RI-9704530 and CDA-9703217, by DARPA under AFOSR F49620-97-1-0485, by CNPq and FAPERJ/Brazil, and by LAAS/CNRS

image completion for recognition purposes. Most of the above work consider using only stationary, monocular image frames [10, 2, 8, 11], not including temporal aspects like motion or functional and behavioral aspects; nor do these approaches provide real-time feedback to environmental stimuli. We adopt a behaviorally active strategy, providing a more complete and *working* model for robot cognition, including at least attention and pattern categorization behaviors. We consider an improved (practical) set of features for both behaviors, extracted from real-time sequences of stereo images. This includes static spatial properties (like intensity, texture), other temporal properties (like motion), and also stereo disparity features.

2 Environments and data abstraction

We used two platforms in the experiments to validate the mechanisms.

The Simulator "Roger-the-Crab", (figure 1) has five controllers (one for the neck, two for the eyes and two for each arm) integrated in a single platform. For each pixel in Roger's unidimensional retinas (Figure 1) an intensity value is calculated in function of the radiance of the world patch corresponding to it. Also, a compensation for gravity and the 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.

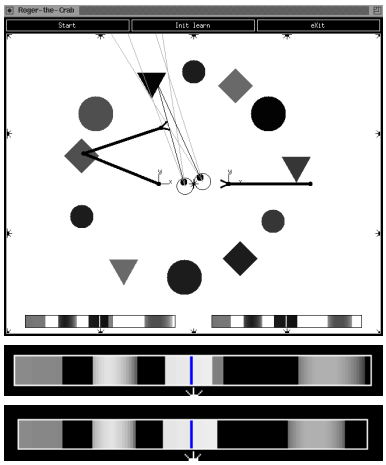

Figure 1: *Roger-the-Crab* and its 1D retinas.

The torso robot platform ("Magilla"), consists of two Whole Arm Manipulators (WAMs), two multi-fingered hands, and a BiSight stereo head on the top of the torso, which provides the visual information for *Magilla* (Figure 2). This vision platform consists of two video cameras mounted on a TRC BiSight head providing four mechanical degrees of freedom: pan, tilt, and, independent, left and right vergence (see Figure 3). Motion is controlled via a "PMAC/Delta TAU" interface. Images from each camera input into a "Datacube" pipelined array (image) processor (IP), able to perform image processing operations in real-time (up to 30 frames per second). We use this IP mainly to reduce and abstract data. Actually, the work reported in this paper was done entirely on *Roger-the-Crab* and on the stereo head (without using the arms and hands).

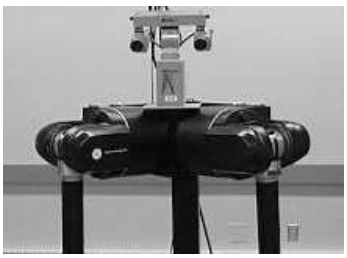

Figure 2: Current configuration of UMASS Torso.

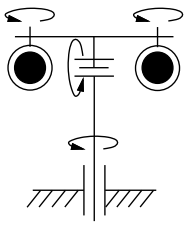

Figure 3: Mechanical DOFs in the TRC BiSight.

We get **data reduction** using a *multi-resolution* (MR) representation. In simulation it is composed of three resolution levels and in the stereo-head of four resolution levels. We provide **abstraction** to support attention and categorization behaviors by using a *multi-feature* (MF) representation. So, the combined result is a *multi-resolution-multi-feature* (MRMF) representation, as in Figure 4 (for the stereo-head). The first six column images are Gaussian partial derivatives of gray-level intensity images and the last two are derivatives of frames difference representing motion. The Datacube can compute such a representation at a rate of 15 frames per second.

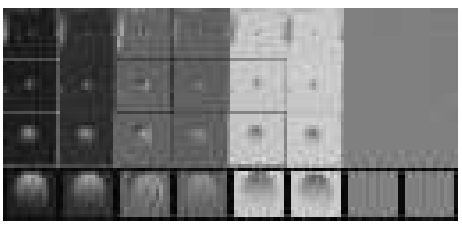

Figure 4: MRMF matrices.

3 Control of attentional behavior

Changing the attentional focus involves computing a salience map for each eye, taking the winning region,

and generating eye movements to foveate on that region. Like the MRMF representations, each salience map has an MR structure. For its generation, we consider the following attentional feature maps: stereo disparity D_a, Gaussian magnitudes $I_a^{(0)}$, $I_a^{(1)}$, $I_a^{(2)}$, motion magnitudes M_a, proximity P_a, mapping T_a, and interest E_a. The map M_a is calculated as the square of magnitude of motion MR (Figure 4, for the stereo-head). Each map $I_a^{(k)}$ is also the square of magnitude of each Gaussian MR. The values in map T_a tells whether a region has been previously visited. In the map P_a, the value of each position is inversely proportional to its distance to the fovea. The value of each position in map E_a is set to zero when a region receives attention and increases slowly over time. Stereo disparity map D_a is computed by using a simple cascade correlation approach over the second order Gaussian *magnitude* $I_a^{(2)}$ computed above. An activation value for each position in the salience maps (S) is calculated as a simple weighted summation of the above features:

$$S = w_{D_a} D_a + w_{M_a} M_a + w_{I_a^0} I_a^{(0)} + w_{I_a^1} I_a^{(1)} + w_{I_a^2} I_a^{(2)} + w_E E_a + w_P P_a + w_T T_a. \quad (1)$$

The weights ($w_{()}$) are task dependent and can be learned using a neural network approach [10] or reinforcement learning [9]. Here we determined them experimentally. This simple function makes the system change its attention window from region to region, covering the whole scene, but eventually returning to previously visited regions to detect changes. The result is a monitoring behavior, in which the robot maintains a representation of the world (the attentional maps) consistent with reality. As we have a salience map for each eye, the winning region also determines the "dominant" eye, or the eye whose salience map contains the most active region.

Shifting attention involves taking the most active region over all levels in the salience maps and computing a coarse saccade movement to foveate each eye on the target. The displacements for the degrees of freedom of the robot platforms are computed from the eye displacements according to several constraints, to produce the motion parameters. After a coarse saccade, due to several types of error, the target may not be in the fovea. To correct, fine saccades are iteratively performed at increasing levels of resolution to maximize the correlation between an acquired target model and the dominant eye image center. Simultaneously, the vergence algorithm, also iteratively, calculates displacements for the non-dominant eye to maximize correlation at the center of the two eye images. As a result of both iterative processes, the system will have both cameras foveated in the target.

4 Categorization Behavior

After attention is in focus, the system computes other feature set from the MRMF for categorization behavior. We experimented with two types of classifiers: a multi-layer perceptron trained with a backpropagation algorithm [7] and a self organizing map (SOM) [3]. We developed self-growing models, that means, we set a threshold to tell whether a representation is a new one; if this threshold is below a certain value, determined empirically, a supervised learning module is automatically invoked, inserting the new feature set into memory, and updating (creating new nodes or neurons) and retraining the BPNN or SOM.

4.1 Feature extraction

The feature vectors that we consider for categorization purposes are: *intensity*, *texture*, *shape*, *motion*, *size*, and *weight* (last two used in simulation). Each *intensity* vector $I_c^{(k)}$ is calculated as an average in the vicinity of the corresponding Gaussian response $G^{(k)}$. Each component of the texture vector $T_c^{(k)}$ is the variance in the vicinity of a Gaussian response. In a similar way, each shape vector $S_c^{(k)}$ is the variance in the vicinity of the stereo disparity magnitude ($D_a^{(k)}$) computed in the attentional phase. Motion $M_c^{(k)}$ is an average in the vicinity of the motion magnitude ($M_a^{(k)}$). The size (used in simulation) is extracted from stereo measurements and the weight extracted from the arms sensors. All the above features are normalized by maximum values. As a result of the above averaging and variance, feature matching is more tolerant of scaling, rotation, and shift. Experiments indicate rotations up to 30 degrees are acceptable.

4.2 Backpropagation Classifier

In this first implementation of the associative memory, we used a multi-layer perceptron trained with a back-propagation algorithm (BPNN) [7]. It has one input node for each abstracted feature. The number of nodes in the output layer changes dynamically by using the self-growing mechanism; a new node is created for each new representation detected. A weighted function of the minimum and maximum error obtaining during the training phase is used as a threshold to decide if a representation is new. The number of hidden nodes is determined empirically. Actually, 1.5 times the number of output nodes gives good results.

4.3 The Self Organizing Map

In other implementation of the associative memory, we developed a network based on the self-organizing

map introduced by Kohonen [3]. This network embeds a competition paradigm for data clustering by imposing neighborhood constraints on the output units, such that a certain topological property in the input data is reflected in the output's unit weights. We considered the Euclidean distance as the measure of similarity (dissimilarity) and the winning neuron is the one with the largest activation (the lowest distance). The Kohonen network moves the neurons towards the input probability distribution. Input neurons are represented by the abstracted features, which are mapped onto a lower dimensional space represented by the output neurons. In a first stage (off-line) the net is trained with only a few objects by presenting each abstracted feature and selecting the winner neuron. This is made to roughly approximate the input probability distribution of the feature vectors. Then the winner's neighborhood is adapted in order to move each neuron (weighted with a Gaussian function) towards the input vector. During the on-line stage, the quantization error measured on each input vector controls the growing process. The network does not need any further training since its recognition codes (*codebooks*) are organized in the Euclidean space. A new neuron can be allocated between the winner and the second closest neuron. The allocation process of new codebooks (the self-growing property) is controlled by a threshold value, empirically found.

5 Demonstrations and Results

We performed several demonstrations involving attention and categorization behaviors for the construction of attentional maps in a monitoring task. In the final experiments, basically, we placed several instances of objects of various types on a table (or in *Roger*'s environment). We expect the robots to focus their attention on all objects, learning their characteristics and constructing their attentional maps.

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

Figure 5: Partial evaluation.

5.1 Attentional Behavior

Table 5 shows a partial evaluation of two approaches that we developed to define a policy for control of attention in a monitoring task for the simulator *Roger*: a simple *straight-forward*, handy-coded strategy (1) and a *Q-learning* (2) approach [1]. In these experiments, all regions of interest in the environment were visited (looked at) by *Roger* (Figure 6 illustrates this). By the results we could see that both methods worked well in simulation for the monitoring task.

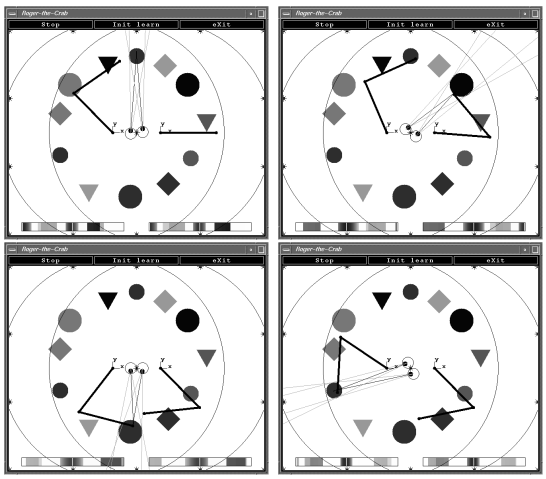

Figure 6: *Roger* constructing its attentional maps.

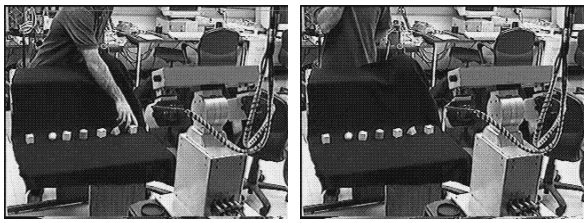

Figure 7: Robot following motion cues.

In the stereo-head robot, we conducted three types of attentional tasks, using a simple policy. In the first demonstration, we indicate the objects to the robot by touching or pointing to them in a sequence. The robot uses mainly this motion cue to foveate on the objects. Figure 7 illustrates this case (the stereo head goes after the motion cues). In the second demonstration, there are no motion cues, so the robot relies solely on intensity cues. Figure 8 illustrates this situation; the attentional mechanism causes the robot to visit all regions using mainly mapping T_a and interest E_a attentional terms. In the third demonstration, after all objects are inserted in the attentional maps, we either move or remove an object. Figure 9 illustrates this case showing a sequence where the robot tracks an object by virtue of differences observed in the working attentional maps.

5.2 Categorization Behavior

In categorization behavior, the system, without any prior knowledge, learns the characteristics of the objects, inserting a representation for each new object in the associative memories. Figure 10 shows confidence for the BPNN, simultaneous activations in its last layer using several instances of four types of objects. Variations in the object poses were included with rotations

Figure 8: Attention without motion cues.

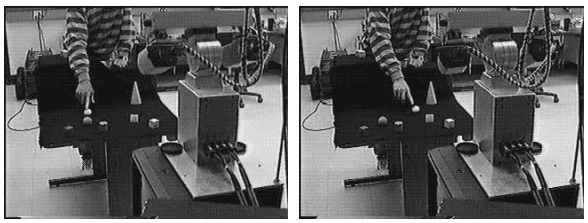

Figure 9: Updating attentional maps (tracking).

up to 30 degrees relative to the cameras. For each instance, the upper line is the highest activation (with object on learned pose). The next line is the lowest activation (object pose degraded). It yet allows categorization since the activation is still over the threshold. We could verify that these rotations were well supported by the system.

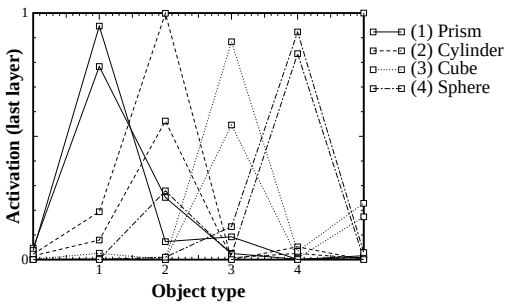

Figure 10: Activations for objects in BPNN.

We conducted several experiments to test the BPNN performance. Figure 11 shows a graph with the time in seconds as the number of objects increases, for the BPNN training procedure: an apparently soft parabola (this is a characteristic of the BP model). In practice, this issue does not compromise system performance as a model of short-term, working memory of ten objects (less than 3 sec) seems quite acceptable.

Figure 12 shows only new object types discovered (for the stereo head) in one of the experiments. It

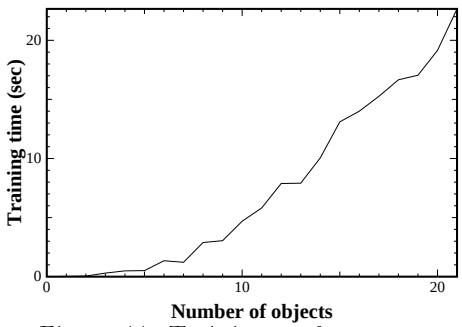

Figure 11: Training performance.

also validates the vergence mechanism, since the same object is in the fovea for each pair of images.

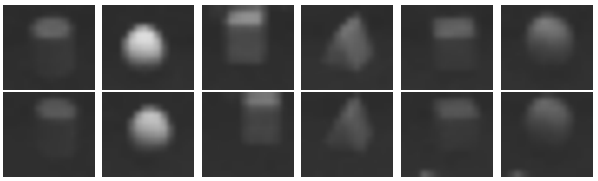

Figure 12: Different types of objects detected.

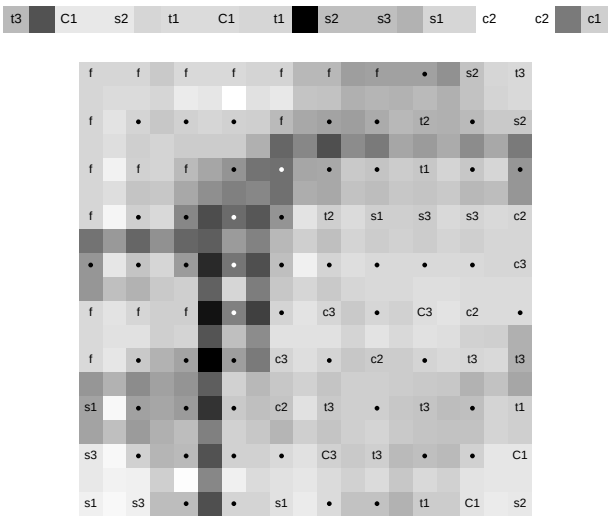

Figure 13: Initial (top) and final (bottom) 1D SOM maps (due to space reasons, the final map is shown in 2D). Object types: circles(ci), squares(si), and triangles(ti), and class f for "background" regions. Objects have different sizes (small letters are small objects).

The SOM worked well classifying well-comported data, with objects on controlled positioning. Besides these successful experiments, we conducted some other experiments towards testing SOM confidence, using raw scattered and not well comported data, obtained from *Roger*'s retinas. The initial map with few data is shown in top of figure 13, where is clear the ordering mechanism of the classified objects. We used a 1D map for the growing process. Bottom of Figure 13 shows the

Phase or process	Min(sec)	Max(sec)	μ(sec)
Computing MRMF	0.145	0.189	0.166
Pre-attention	0.139	0.205	0.149
Total attention	0.324	0.395	0.334
Total saccade	0.466	0.903	0.485
Features for match	0.135	0.158	0.150
Total matching	0.323	0.353	0.333

Table 1: Processing time required in each phase.

final map with 18 neurons. As we can see, class "f" was broken into several subclasses because of using a 1D map and of the quality of the data (class "f" has a very sparse distribution). A threshold is not a good compromise to allocate new neurons. Besides these small problems, on the overall evaluation, the SOM has attended our expectations.

During experiments in the stereo head, we collected system performance data. Table 1 shows the minimum, maximum, and average times required for each processes involved in the attention and categorization behaviors. With the host computer used in these experiments, a Sun Sparc 4 with some 40 MHz processor, the system was able to operate at roughly 3 frames per second. We predict that with the currently updated hardware (a Sun Ultra 10 with some 300 MHz processor), a frame-rate of 10-15 (about 70 to 100 milliseconds per object) can be achieved for both behaviors. Saccade speed can also be improved by adjusting the PD controller gains in the stereo head PMAC interface, making saccade as fast as a human being.

6 Conclusion and Future Work

We have built useful mechanisms for attentional control and pattern categorization, successfully used by a multi-modal sensory system in the execution of monitoring 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, to provide a more discriminative feature set. We believe that these two abilities (attention and categorization) are a basis not only for this simple monitoring task, but also for other more complex tasks involved in robot cognition. We found out that the SOM works faster than the BPNN approach, and that it can accomplish a better performance even using the original features abstracted for categorization, without any sampling for data reduction. Self organizing maps are useful because they have a relatively low training time against the resolution of the map. The SOM network used here can also be improved by considering a 2D growing map (we have used a 1D map).

The approach used for attention can be improved with a task dependent weight function. Reinforcement learning [9] can play an important role in the weights definition. Given a set of tasks, the system is rewarded for detection of objects important to each task. By using the model proposed in this work, top-down tasks (search for an object) can also be formulated. Giving an object index to search, the weights associated with its model are retrieved and the saliency maps are calculated. Then, by using categorization behavior, the most salient regions from the salience maps (candidate objects) can be evaluated as being the searched one. In the same way, control tasks involving covert attention can also use the same architecture. Finally, another possibility of future work is to introduce, by software, a moving fovea representation (currently, our fovea is defined in the image center).

References

[1] L. M. G. Gonçalves and A. A. F. Oliveira. A reinforcement learnig approach for attentional control based on a multi-modal sensory feedback. *III Workshop on Cybernetic Vision*, February 1999.

[2] L. Itti, J. Braun, D. K. Lee, and C. Koch. A model of early visual processing. In *Advances in Neural Information Processing Systems*, pages 173–179, Cambridge, MA, 1998. The MIT Press.

[3] T. Kohonen. *Self Organizing Maps*. Springer Verlag, (1997).

[4] S. M. Kosslyn. *Image and Brain. The Resolution of the Imagery Debate*. MIT Press, Cambridge, MA, 1994.

[5] R. Milanese, S. Gil, and T. Pun. Attentive mechanisms for dynamic and static scene analysis. *Optical Engineering*, 34(8), 1995.

[6] R. P. N. Rao and D. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence Magazine*, 78(1-2):461–505, October 1995.

[7] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Proc. of the International Conference on Neural Networks (ICNN'93)*, pages 123–134. IEEE Computer Society Press, 1993.

[8] I. A. Rybak, V. I. Gusakova, A. V. Golovan, L. N. Podladchikova, and N. A. Shevtsova. A model of attention-guided visual perception and recognition. *Vision Research*, 38(2):387–400, 1998.

[9] R. S. Sutton and A. G. Barto. *Reinforcement Learning: an Introduction*. The MIT Press, Cambridge, MA, 1998.

[10] P. Van de Laar, T. Heskes, and C. Gielen. Task-dependent learning of attention. *Neural Networks*, 10(6):981–992, August 1997.

[11] P. A. Viola. Complex feature recognition: A bayesian approach for learning to recognize objects. AI Memo 1591, Massachusetts Institute of Technology, November 1996.