Integrating Attention and Categorization Behaviors in Robotics *

Luiz M. Garcia
Laboratoire d'Analyse et d'Architecture des Systemes (LAAS/CNRS)
7, Avenue du Colonel Roche 31077 Toulouse Cedex 4
lmgarcia@laas.fr

Roderic A. Grupen
University of Massachusetts (UMASS/LPR)
140 Govenor's Drive, Comp Sci Building, Amherst MA 01003 USA
grupen@cs.umass.edu

Abstract. This work presents an active vision system for control tasks involving attention and pattern categorization based on visual sensory information. The system is currently implemented in a robot consisting of an articulated stereo-head with four degrees of freedom (pan, tilt, left and right verge). As a practical result, the robot is able to analyze all regions of its environment, selected according to salience maps. This involves performing attentional shifts and an efficient feature extraction and pattern recognition. Also the system incrementally construct an attentional map, and keep the map consistent with a current perception of the environment. Experimental result shows succesful applications in inspection, monitoring, or surveillance tasks.

1 Introduction

We introduce a methodology that integrates attention control and pattern categorization behaviors in robotics. Our goal is to develop a robotic system capable of interacting with its environment in a dynamic way. We use an articulated stereo head to maintain an internally consistent spatial model of the neighboring visual landscape. A dedicated image processing device provides data reduction and feature abstraction from the visual input data. This approach can be applied to a variety of visual monitoring tasks including attentional control and pattern categorization for constructing attentional maps of the environment. To perform this mainly bottom-up attentional task, our robot must be able to foveate (verge) its eyes to regions of interest, to maintain attention on the region as needed, and to shift its attentional focus when the current region is no more of interest. Attentional maps (containing pattern representation, position and orientation information) are maintained and are incrementally updated as needed. Initially, the system has no knowledge about the environment (attentional maps are empty), nor does it knows about the objects (the system memory has no pattern representations). The goal is to learn the characteristics of new objects detected in the environment. To do so, an active behavioral strategy engages in dynamic interaction with a changing environment and discovers perceptual events that are worthy of attention. Other

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

possible tasks that the system is currently able to perform include visual surveillance, inspection, spatial orientation, and navigation.

Briefly, salience maps are constructed bottom-up from attentional features maps and are used to direct attention to regions of interest. Saccadic movements are used (possibly including pan and/or tilt movements) to foveate on the selected region. A feature extraction for categorization is then performed, producing changes in the perceptual state. An associative memory maps the features into a pattern address, allowing recognition of existing perceptual categories or discovery of new categories. To complete the system architecture, information about the (new or known) representations detected in the environment are inserted in the attentional maps. After all regions are visited, the system initiates a guided search to detect any change that may occur in the environment. The heart of the system is the attentional mechanism which controls a change in the focus of attention. The mechanism of attention is designed to visit all regions in the visible environment and eventually returns to regions previously visited. We note that our approach is inherently reactive, choosing actions based on perceptions of the world at different temporal scales rather than by using a geometric model as in traditional planning techniques. Also, by reducing and abstracting data, the system performs fewer computations and substantially improves its performance. Finally, the attentional mechanism developed here can deal with dynamic environments and provides real-time feedback to stimuli of varying bandwidth. For these reasons, this system architecture is particularly relevant to autonomous robot systems.

2 Background and Related Work

In a general context, attention can be defined as the ability of selecting a topic of interest (a goal) for extraction of information useful for a given task. We can add to the definition the ability of changing the attentional focus from one topic to another if necessary (this is currently a more challenging problem). In this work context, the interest topics are real objects (or sometimes regions of interest) in a restricted robot environment. Attentional control will be employed just to provide interaction between the robot and the environment. We consider mainly using bottom-up and overt attention. Several approcahes based on multi-feature extraction have been proposed recently for attention and categorization purposes. In [1], an attentional neural network gathers information from feature maps to produce a salience map. The interest region is given by the most salient position in the map. Itti et al. [2] propose a model for attention that uses linear filters tuned to various orientations and spatial periods, computing a phase-independent linear response to visual stimuli. Rybak et al. [3] treat perception and cognition as behavioral processes in which images are scanned sequentially by an attention-window with multiple image fragments processed in parallel. The work of Kosslyn [4] suggests that features extracted from visual images are combined with mental image completion for recognition purposes. Although Kosslyn suggests few practical mechanisms, the overall approach is intuitively appealing. Rao and Ballard [5] provide a set of operators based on Gaussian partial derivatives for feature extraction, which are motivated by biological models.

In relation to the above works, we provide a more complete *working* model for cognition, including at least attention and categorization. We consider an improved (practical) set of features for both behaviors, extracted from real-time sequences of images. The system is currently implementd in a robot platform which operates in a real environment. So, the main difference between our work and the above is real time application in an active vision framework. Most of the above works consider using only stationary image frames [1, 2, 3] not including temporal aspects like motion or functional and behavioral aspects; nor does the approach provides real-time feedback to environmental stimuli. The challenging problem in attentional control solved here resides mainly in the construction of a mechanism to change attention from one place to another. For pattern categorization, appearance-based (semi-invariant) features abstracted from Gaussian operated images plus stereo and motion patterns are used as input to an associative memory which remembers addresses into a pattern storage memory.

3 Robot Environment and Data Abstraction

The robot used in this work consists of two video cameras mounted on a TRC Bisight head providing four mechanical degrees of freedom: pan, tilt, and left and right vergence. Motion is controlled via a PMAC/Delta TAU interface. Images from each camera are input to a Datacube pipelined array (image) processor (IP), which performs image processing operations in real-time (up to 30 frames per second). The Datacube is mainly used to reduce and abstract data. A *multi-resolution* (MR) representation (as in Figure 3) accomplishes data reduction. A *multi-feature* (MF) representation provides abstraction to support attention and identification behaviors. So, the result is a *multi-resolution-multi-feature* (MRMF) representation (Figure 3). Each pair of images of the first six column (MR) images are Gaussian partial derivatives (order 0, 1 and 2) of intensity images, in two orthogonal orientations for each pair. The last two MR images are the derivatives of difference images (also in two orthogonal directions) representing motion. Considering the total numbers of pixels, the amount of data is decrased by a factor 32. The datacube can compute such a representation at a rate of 15 frames per second.

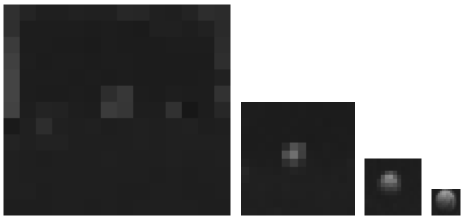

Figure 1: Multi-resolution image.

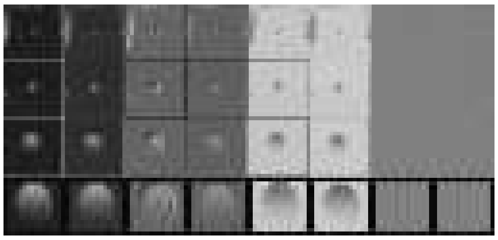

Figure 2: MRMF matrices.

4 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. The salience maps are computed from attentional feature maps extracted from the previous abstracted data (MRMF), by interrogating attentional maps currently being constructed, and also by considering other attentional features that are task dependent. Like the MRMF representations, each salience map has a pyramidal 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 subscript a denote "attentional phase"). These attentional feature maps are (bidimensional) matrices. The map M_a is calculated as the square of magnitude of the previous MRMF motion representation part (last two columns (MR) of Figure 3). Each map $I_a^{(k)}$ is the square of magnitude of Gaussian images (first three MR pairs pairs in Figure 3). The values in map T_a tells whether a region has been previously visited (received attention). 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 on the second order Gaussian *intensity* $I_a^{(2)}$ computed above. Since all attentional features are computed, starting at the coarsest level, an activation value for each position in the salience maps (S) is calculated as a simple weighted summation of all attentional features, above:

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

The weights ($w_{()}$) of this function are task dependent and can be learned using a neural network approach [1] or reinforcement learning [6]. In our monitoring task, they were determined 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 possible 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 this target. The displacements for the degrees of freedom of the robot platform are computed from the eye displacements according to several constraints, to produce the motion parameters. To complete this first phase of attentional process, the motion interface effectively produces physical movements, running its PD servo controllers concurrently, to put the attentional goal in the foveal region. After a coarse saccade, due to several types of error, the goal 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 the 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 goal.

5 Identification Behavior

Since attention is in focus, we compute a feature set from the the Datacube output (MRMF) representation for categorization behavior. Experiments using the MRMF representations directly as features for the associative memory also gave good results, but are computationally expensive. Some abstraction is used here to further reduce the input data. Our current approach uses locally normalized mean and variance in the vicinity of only four positions - (3,3), (3,8), (8,3), and (8,8) - for each one of the 16×15 images of a given level. We compute these local statistical moments for each one of the six Gaussian images in Figure 3 and for the magnitudes of motion and stereo disparity computed in the attentional process. For a given level, this gives a total of 112 input features.

Identification is done using an associative memory implemented by a multi-layer perceptron using a back-propagation training algorithm [7] (BPNN). The BPNN maps the features to an address in a long term memory (LTM) which stores various other information. We note that other approaches like Viola's belief networks [8] or Kohonen's self-organizing maps [9] could be used here. We argue that this BPNN approach gives good results on identification and also returns the activation for a given index in the output layer (a normalized value between 0 and 1). This value besides used here to determine if a representation is new, can also be used in top-down attentional tasks to keep attention in a given region. The BPNN has one input node for each abstracted feature. The number of nodes in the output layer changes dynamically by using a self-growing mechanism; a new node is created for each new representation detected. A weighted function of the minimum and maximum error during training is used as a threshold to decide if a representation is new. The number of hidden nodes is determined empirically: 1.5 times the number of output nodes gives good results.

Once a representation is classified as new or identified, the current position in the attentional maps is updated. Attentional features sufficient to detect on-line changes are stored and *mapping* (T_{ij}) and *interest* (E_{ij}) values are set to zero allowing a shift of attention to another region. If the representation is new, supervised learning is automatically invoked, inserting the new feature set into LTM and updating/retraining the BPNN.

Figure 3: Robot following motion appeals.

6 Experiments and Results

Experiments involving attention, identification, and combined tasks were performed. Basically, many instances and types of objects are placed on a table. We expect that the robot focus attention on all objects, learns their characteristics (inserting a representation for each in the BPNN), and incrementally updates the attentional maps. At the end a monitoring/inspection behavior is to be adopted.

In attentional behavior, three types of experiments were tested. In the first type, we indicate the objects to the robot by touching or pointing to them in a sequence. The robot uses mainly this motion cue (motion wins in the attentional process) to foveate on the objects. Figure 3 shows a sequence ilustrating one of the experiments. The stereo head goes after the motion cues, detecting all objects on the table.

In the second experiment type, there is no motion cues, so the robot relies solely on intensity cues. Figure 4 ilustrates one of the experiments. Here the attentional mechanism makes the robot visit all regions, using mainly mapping factor T_{ij} and interest rate E_{ij}.

Figure 4: Attention without motion cues.

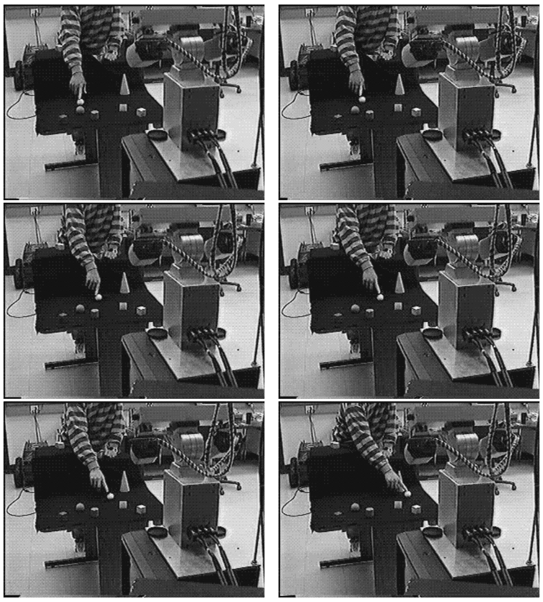

Figure 5: Robot updating maps (tracking).

In the third experiment type, after all objects are inserted in the attentional maps, we either move or remove an object. The robot updates the attentional maps for the changed regions using mainly motion and intensity cues for changes (or movements) inside the field of view. If a change occurs outside field of view the robot will eventually return its attentional focus to that region by using T_{ij} and E_{ij} values. Then it will detect the eventual change by comparing current attentional features with stored ones. Figure 5 shows a sequence where the robot follows/track an object, after the attentional maps construction. The maps are rectified in each instant, untill the ball stops in final positions.

In the experiments involving identification, the system, without any prior knowledge, learns in automated way the characteristics of the objects, inserting a representation for each new object in the associative memory. A threshold is used to determine whether a representation is a new one. The system is capable of detecting new objects and of recognizing instances of already known objects. Several experiemnts were done to test the associative memory (BPNN) performance. Figure 6 shows two graphs (one with number of *"epochs"* and another with time in seconds) for the BPNN training procedure. The graphs are apparently soft exponential functions. This is a characteristic of the BP model [7]. In practice, this issue does not compromise the system performance if we suppose that a robot will not work with an infinite set of objects. As the second graph shows, a reasonable set of objects (more than 20) can be learned in about 20 seconds of training time. This time can yet be improved by applying local training (it is not the case in this work). Also, in this work, to test the automatic supervised learning procedure, the network is reinitialized for each running experiment (it starts with zero objects). In a practical situation, the system would save the synaptic weights and network configuration (also the attentional cues) for future use. In this situation, the probability of finding a new object would be small, not interfering in system performance.

Figure 7 shows BPNN confidence (simultaneous activations in its last layer), using several instances of four types of objects, placed on the table in controlled poses. For

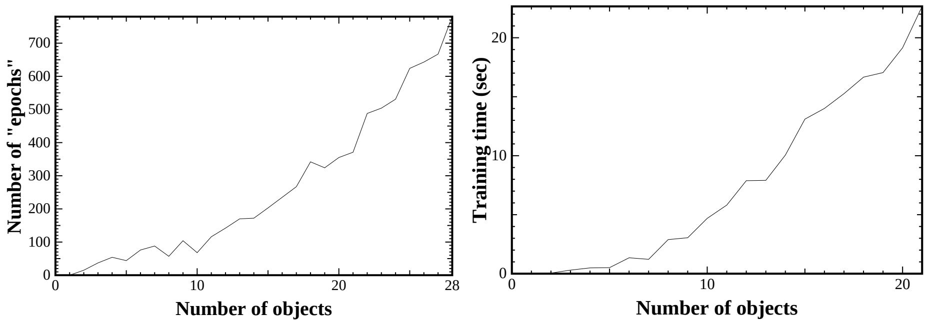

Figure 6: Training performance.

each instance, the upper line is the highest activation experimented with the objects on learned poses. The next line is lowest activation yet allowing identification (object poses were degraded with rotations up to 30 degrees when viewed from the stereo head). We could verify that these rotations were well suported by the system. Figure 8 shows only new object types discovered (new representations) in one of the experiments. The figure also ilustrates the vergence mechanism working, since the same object is in focus for each pair of images.

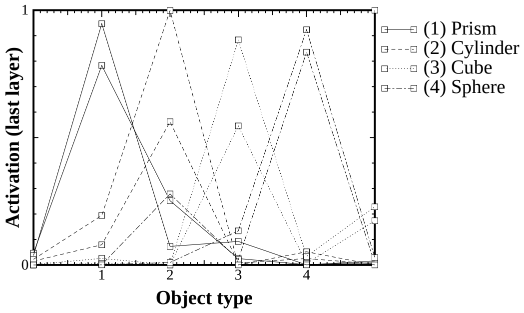

Figure 7: Simultaneous activations.

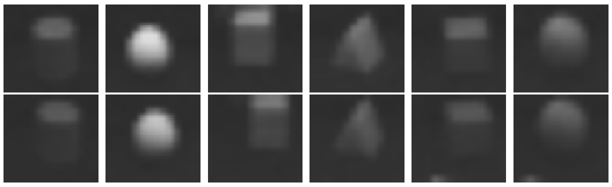

Figure 8: New types of objects detected.

During some experiments, we also collected system performance data. Table 1 shows the time required for the main phases involved in attention and categorization behaviors, sampled over several hundred control cycles. The host computer used (a 40 MHz Sun Sparc 10) has since been replaced with a 300 MHz Sun Ultra 10 board that shares the Datacube bus for improved data transfer rates. So, a frame-rate of 10-15 (about 70 to 100 miliseconds per object) can be achieved currently for both behaviors. Saccade speed can also be improved by adjusting PD controller gains in the stereo head PMAC, making saccade as fast as a human being.

7 Conclusion, Discussion and Future Work

Although this work uses mainly visual information, our system is more generally applicable, and can easily incorporate haptics, auditory, or other sensory maps to provide a more discriminative feature set. We have developed the basic architecture integrating an attentional mechanism and a neural network classifier. The "controller oriented" approach used to resource allocation would allow other process to be integrated into this architecture. We believe that the two abilities (attention and categorization) are a

Phase or process	Min(sec)	Max(sec)	μ(sec)
Total acquiring (MRMF)	0.162	0.255	0.186
Features for 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 Identif.	0.323	0.353	0.333

Table 1: Processing times required for each phase.

basis not only to the monitoring tasks described in the paper, but also to other more complex tasks involved in robot cognition. The two behaviors are interrelated, and a behaviorally active system needs both to perform other tasks.

The approach used for attention can be improved with a weight function that varies according to the task. Reinforcement learning [6] can play an important role in the weights definition. Given a set of tasks, the system can be rewarded for detection and identification 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. In this case, the attentional mechanism shall be done as a closed loop controller. Giving an object index to search, the weights associated with its model are retrieved and the salience maps are calculated. Then, by using the identification/recognition 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.

References

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

[2] L. Itti, J. Braun, D. K. Lee, and C. Koch. A model of early visual processing. In *Advances in Neural Information Processing Systems*, volume 10, pages 173–179, Cambridge, MA, USA, 1998. The MIT Press.

[3] 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*, 38:2387–2400, 1998.

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

[5] 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.

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

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

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