Towards a Real-time Framework for Visual Monitoring Tasks *

Luiz-Marcos Garcia
Laboratoire d'Analyse et d'Architecture des Systemes
7, Avenue du Colonel Roche
31077 Toulouse Cedex 4 France
lmgarcia@laas.fr

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

Abstract

This work describes an active vision framework that is able to perform visual monitoring tasks involving attention control and pattern categorization behaviors. We use an articulated stereo vision platform and image processing device, which provides abstracted information about the environment. As a practical result of this work, the system can select a region of interest in its environment, perform attention shifts involving saccadic movements, and perform an efficient feature extraction and recognition. Also, attentional maps of the scene are incrementally constructed, and the maps are kept consistent with a current perception of the world. Another important result for the attentional mechanism is that the system is capable of analyzing all regions of its world, selected according to salience maps.

1. Introduction

This work presents a robotic system that is able to perform visual monitoring tasks involving attention control and pattern categorization behaviors. We use a robot platform consisting of an articulated stereo-head with four degrees of freedom (pan, tilt, left verge, and right verge). A dedicated image processing device provides data reduction and feature abstraction. Based on this information the system selects the right actions as feedback to environmental stimuli. The result is a behaviorally active system controlled by its perceptual state. Our goal is to develop a robotic system able to foveate (verge) the eyes onto a region 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. Environmental (attentional) maps containing pattern representation, position and orientation information are incrementally constructed and dynamically updated. Moreover, by adopting an active behavioral strategy we provide a dynamic interaction with changing environments. Possible tasks include object recognition and identification for visual surveillance, monitoring, inspection, spatial orientation, and navigation. Here we will describe experiments involving a visual monitoring task. At first glance, the system has no knowledge about the environment (attentional maps are empty) nor about the objects (the system memory has no pattern representations). The objective is to learn pattern representations for all different types of objects detected, and also to fill the attentional maps with information about the environment (object pattern, position, and orientation). Briefly, bottom-up salience maps are used to direct attention. As a result, the system selects a region of interest. Saccadic movements are computed and executed for the eyes (possibly including pan and/or tilt movements) to foveate on the selected region. Feature extraction is then performed, producing changes in the perceptual state. An associative memory maps the features into a pattern address, allowing the system to recognize and identify an existing representation or to discover new categories (unknown objects). To complete the system architecture, information about the (new or known) representations detected in the environment is inserted in the attentional maps. After all regions are visited the system assumes a search posture, not ramdomly but internally guided, to detect any change that can eventually occur in the environment. So, the heart of this system is an attentional mechanism, which allows the system to change its focus of attention from one region to another, visiting all regions of the environment, and eventually returning to regions previously visited.

2. Background and related work

In a general context, attention can be defined as the ability of selecting a topic of interest (a target) 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 the most challenging problem). In this work context, the interest topics

*This work was done at UMASS/USA and was supported by NSF under IRI-9503687, IRI-9704530, and CDA-9703217, and by CNPp/Brasil

are real objects (or sometimes regions of interest) in a restricted robot environment. Attentional control and pattern categorization will be employed just to provide interaction between the robot and the environment. We consider mainly using bottom-up and overt attention, in a monitoring task. The robot has to learn how to construct attentional maps of its environment. Also, the robot has to learn the characteristics of all present objects, dealing with new or already known categories. After such a world representation is constructed, the robotic agent can perform other more specific tasks.

Ultimately, most researchers have tried to reproduce or to imitate the biological system and behavior in order to develop models for robot cognition. In this way, several approahes have been proposed recently for attention and categorization purposes based on multi-feature extraction. Following this methodology, a good method towards providing a computational model to explain the neuro-physiology of attention can be found in the work of Van der Laar et al. [12]. A multi-feature extraction is performed computing several feature maps from monocular, stationary images. An attentional neural network receives a task dependent input and gathers information from the feature maps to construct a salience map. Then, the place where to put attention is simply given by the most salient position in the salience map.

In a similar approach, in the work described in [4], Itti et al. propose a model for attention that uses linear filters for extraction of information related to orientation and spatial frequency. These filters, tuned to various orientations and spatial periods, are used to compute a phase-independent linear response to visual stimuli. Then, these linear filters interact through non-linear excitatory and inhibitory pooling and a noise model together with a sthocastic decision strategy are assumed in order to relate the model output to psychophysical data. In other words, a salience map is calculated and a statistical model determines the next attention-window.

The work described in [10] joins attention and identification topics in a unique system. Rybak et al. treat perception and cognition as behavioral processes, besides using a simple model with monocular, stationary images. These include both the sequential image scanning by an attention-window and the parallel processing of image fragments within the attention window. Pattern recognition is encoded in memory as a sequence of eye movements with detection of expected image fragments at the locations. The two well known pathways "what" and "where" are encoded using neural-network implementations. An object is sequentially recognized if the patterns of the eye movements and each corresponding invariant image fragment pattern are the same as a given stored representation of those.

The work of Kosslyn [6], based on results of neuro-psychology and neuro-biology, suggests a good descriptive model to explain how identification and recognition happen. The model suggests that features extracted from visual images are combined with mental image completion. Besides missing explanations that would allow one to perform a computational implementation of such a system, and practical difficulties, it is intuitively atractive to develop a visual system following that description.

In the work described in [7], Rao and Ballard provide a good feature based model for recognition and identification. The approach provides a set of operators based on Gaussian partial derivatives for feature extraction. These operators are suggested to be similar to biological models.

An interesting approach involving visual and oculo-motor mapping can be found in the work of Ferrell [1]. She uses registered, multi-modal, topographically organized maps of the sensory-motor space to orient a robot (the MIT COG project) towards environmental stimuli. A learning algorithm to train the humanoid robot is presented. In order to learn the connections between the oculo-motor and visual maps, these are initially connected to each other through largely overlapping receptive fields. Then the algorithm tries to adjust the connections by decreasing their neighborhood. The connections are updated according to a learning function which uses the error distance between a motion map site and a given target site.

In relation to the work discussed above, we provide a more complete *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 and texture based features, temporal properties like motion, and stereo disparity. Also, as the system is currently implementd in a robot platform which operates in a real environment, we include in the feature space the current functional state of the robot. Most of the above work considers using only stationary, monocular image frames [12, 4, 10], not including temporal aspects like motion and stereo or functional and behavioral aspects; nor does the approach provides real-time feedback to environmental stimuli. So, the main difference between our work and most of the above is real time application in an active vision framework. The challenging problem in attentional control solved here resides mainly in the construction of a mechanism to change attention from one place to another. To solve this problem we use features from an image filtered with adapted Gaussian partial derivatives combined with motion and stereo to generate a salience map and determine the next region of interest. We convert saccadic vergence movements to pan and tilt motion as necessary to foveate on an attention window. For pattern categorization, appearance-based (semi-invariant) features abstracted from Gaussian operated im-

ages plus stereo and motion patterns are used as input to an associative memory which remembers addresses into a pattern storage memory.

3. Robot platform and feature abstraction

The robot used in this work, shown in Figure 1, consists of two video cameras mounted on a TRC Bisight head providing four mechanical degrees of freedom: pan, tilt, and left and right vergence. Zoom and focus, not used in this work, are also controllable. 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.

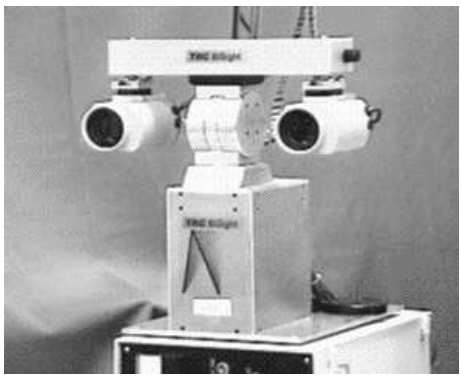

Figure 1. UMASS Stereo Head platform with 4 DOFs.

A $multi\text{-}resolution$ (MR) representation (as in Figure 2) accomplishes data reduction. A $multi\text{-}feature$ (MF) representation provides abstraction to support attention and identification behaviors. So, the result is a $multi\text{-}resolution\text{-}multi\text{-}feature$ (MRMF) representation, as in Figure 3. The first six column images are adapted Gaussian partial derivatives of gray-level intensity images and the last two are derivatives of frames difference representing motion. Considering the total number of pixels, original data is decreased by a factor 32. In the current implementation, the Datacube IP device generates 64 images (32 per eye) at 15 frames per second. This achieves reasonable performance for the real-time processing necessary in an active vision system. This transformation is performed in two phases: multi-resolution images generation and derivatives computation, described next.

3.1. Multi-resolution image generation

Two multi-resolution images are generated for each eye (camera). One of them will be the basis for "$intensity$" based features and the other one for "motion" features. For the MR $intensity$ images, the original 512×480 pixel stereo images are directly used as input. For the MR motion images, the difference between two consecutive image frames is first computed for each eye. Then, successive resolution levels for both the $intensity$ and $motion$ images are computed by applying a mean filter in the neighborhood of each pixel and sampling with a level dependent factor. The size of the neighborhood and the affected region of the input image are also level dependent. For the coarsest resolution level, an 8×8 pixel filter is applied to the whole image and sampled at 8×8 pixel intervals. The filter diameter is half-decreased and the pixel interval is two times increased until the finest resolution level. In this level, the central 64×60 pixel region of the image is simply taken. The result is two multi-resolution (MR $intensity$ and MR $difference$) images for each eye.

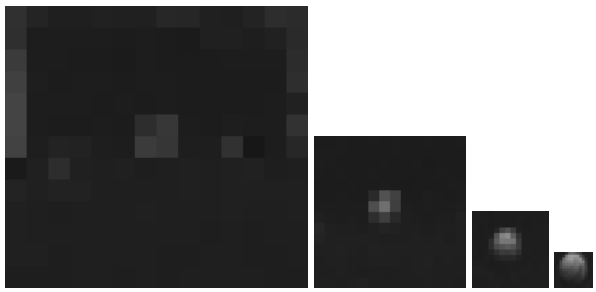

Figure 2. Multi-resolution image of a sphere.

3.2. Computing derivatives of intensity and motion

For the $intensity$ based MF images generation (we will refer to this simply as Gaussian images), the MR $intensity$ images generated above are convoluted by using Equation 1 with the adapted Gaussian derivative kernels given by Equations 3, 4, and 5, in two orthogonal directions each. Four-pixel sampling is also performed, reducing each image to 16×15 pixels. The motion images are computed using Equation 2. Note that the same first Gaussian derivative is applied to the motion images. This helps reducing noise. An ideal approach could use the derivatives of difference images to actually compute the motion field for the whole images, using relaxation or other iterative approaches. However, in our approach the motion field computation is unnecessary for attentional purposes, and expensive for identification purposes.

$$G_{d=x,y}^{(k=0,1,2)} = g_d^{(k)} * I_t \tag{1}$$

$$M_{d=x,y} = g_d^{(1)} * [I_t - I_{t-1}] \tag{2}$$

$$\left.\begin{array}{l} g_x^{(0)}(x,y) = \lambda e^{ax^2} \\ g_y^{(0)}(x,y) = \lambda e^{ay^2} \end{array}\right\} \tag{3}$$

$$\left.\begin{array}{l} g_x^{(1)}(x,y) = 2a\lambda e^{a(x^2+y^2)}x \\ g_y^{(1)}(x,y) = 2a\lambda e^{a(x^2+y^2)}y \end{array}\right\} \tag{4}$$

$$\left.\begin{array}{l} g_x^{(2)}(x,y) = 2a\lambda e^{a(x^2+y^2)}(2ax^2+1) \\ g_y^{(2)}(x,y) = 2a\lambda e^{a(x^2+y^2)}(2ay^2+1) \end{array}\right\} \tag{5}$$

$\forall(x,y) \in ([-s,+s],[-s,+s])$; where $a = \frac{-1}{2\sigma^2}$, $\lambda = \frac{1}{\sigma\sqrt{2\pi}}$, $\sigma = 1.7$, and $s = 3$.

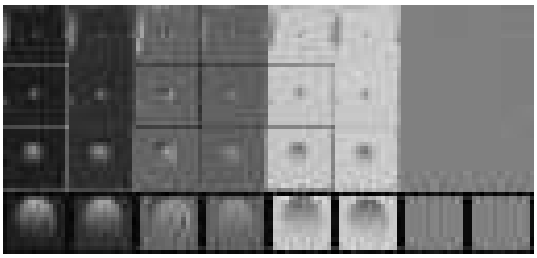

Figure 3. MRMF matrices.

4. Attentional behavior control

The desired attentional behavior for our system is to focus attention on the currently most salient region of its environment. This involves computing a salience map for each eye, taking the winning region, and generating saccadic eye (and possibly pan and/or tilt) movements to foveate on that region. Salience maps are computed by using the abstracted data from the Datacube (MRMF), by interrogating the attentional maps currently being constructed, and also by considering other attentional features that are task dependent. Note that by considering only the perceptual cues in the field of view (attentional features extracted from the MRMF), there is no guarantee that the system will attend to all locations in the world. Therefore, we use other task dependent attentional features to make the robot accomplish the desired behavior (monitiring task). As an example, previously visited regions are given a low potential interest in the internal attentional maps after an attentional visit to prevent an immediate revisitation. This attentional map also encodes other information, allowing a fast check to detect changes in the environment.

4.1. Defining a target (pre-attention)

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 calculated from the MRMF representations. The map M_a is calculated as the square of magnitude of $motion$ MR images (last two columns of Figure 3) as:

$$M_a = (M_x)^2 + (M_y)^2. \tag{6}$$

Each map $I_a^{(k)}$, made for Gaussian kernel response $k = 0, 1, 2$, is also the square of magnitude of the Gaussian MR images (first three pairs in Figure 3), calculated as:

$$I_a^{(0)} = (G_x^{(0)})^2 + (G_y^{(0)})^2 \tag{7}$$

$$I_a^{(1)} = (G_x^{(1)})^2 + (G_y^{(1)})^2 \tag{8}$$

$$I_a^{(2)} = (G_x^{(2)})^2 + (G_y^{(2)})^2 \tag{9}$$

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 the corresponding region receives attention and increases slowly over time until the region becomes again with a high "$interest$". One could use spatial frequency information to compute the disparity map D_a directly from the input images in the Datacube device. Such an approach using a phase shift model is presented in [14] on a simulation platform. However,we compute stereo disparity map D_a by using a simple cascade correlation approach (see Figure 4) on the second order Gaussian $intensity$ $I_a^{(2)}$ computed above. Since the images are an MR representation, the results from one level are used to predict the disparity for the next level. For the initial level, the range of disparity is limited by vergence movement constraints. It is also constrained by the relative symmetry of the images with respect to the cyclopean axis of the system (the line defined by the central point between the eyes and the horopter, point at which the eye axes cross). Also, disparity is computed only for the x direction since our system produces no y disparity. The cascade process substantially reduces the computations necessary to find the best match for a point. Stereo process is computed for both eyes, and it requires less than 0.1 sec.

Since all attentional feature maps are computed, starting at the coarsest resolution level, an activation value for each position in the salience maps (S) is calculated as a simple

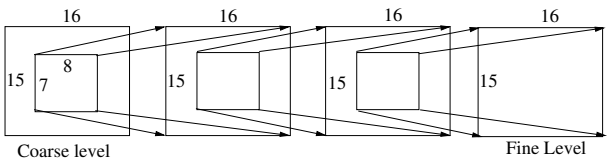

Figure 4. Stereo computation process in cascade.

weighted summation of all attentional features above as:

$$S = w_{D_a} D_a + w_{M_a} M_a + w_{I_a^0} I_a^{(0)} + w_{I_a^1} I_a^{(1)} + w_{I_a^2} I_a^{(2)} + w_E E_a + w_P P_a + w_T T_a . \quad (10)$$

The weights ($w_{()}$) of this function are task dependent and can be learned using neural networks [12] or reinforcement learning [2]. 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 or surveillance behavior, in which the robot maintains a representation of the world (the attentional maps) consistent with reality. Note that a region is never visited twice in a row, but, depending on the weight function used to update the world map activation values, a region may be revisited before all other regions are visited. 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.

4.2. Shifting attention (coarse saccade)

Shifting the attention involves taking the most active region over all levels in the salience maps and computing coarse saccade movements to foveate each eye on the target. Features near the target in the dominant eye are used to build a model to aid in later fine saccadic corrections. For the dominant eye, the target position is determined by the displacement from the current position to the winning one. The target for the non-dominant eye is computed by adding the stereo disparity to the dominant eye target position. The displacements for the four degrees of freedom of the stereo head are computed from the eye displacements according to several constraints. If the cyclopean angle exceeds 15 degrees, a pan movement is initiated to reduce the angle. If the eye axes diverge, or if they converge at more than 45 degrees, a correction is applied to the non-dominant eye. Finally, tilt motion is computed directly from the dominant eye; the stereo head geometry ensures both eyes have the same tilt. To complete this first phase of the attentional process, the PMAC interface effectively produces physical movements running its PD servo controllers concurrently to put the attentional target region in the fovea.

4.3. Adjusting fine saccade and vergence

After a coarse saccade, due to several fonts of error, the target region may not be in the fovea, but in general very close to it. To correct that, fine saccades are performed at increasing levels of resolution to maximize the correlation between the target model and the dominant eye image center. This process converges when the resolution level which determined the attention shift reaches a maximum correlation at (or very near) the image center. Simultaneously, the vergence algorithm calculates displacements for the non-dominant eye to maximize the correlation at the center of the two eye images. A threshold is used to avoid situations where there is no match in the field of view due to occlusions. As a result of both iterative processes, the system will have both cameras foveated in the target region.

5. Categorization behavior

Since attention is in focus, identification behavior computes other feature set from the Datacube output (MRMF). Experiments using the multi-log-retina representation directly as features for associative memory lookup give good results, but are computationally expensive. In those, eight feature vectors ($2G_0 + 2G_1 + 2G_2 + 1Motion + 1Stereo$) composed of 16×15 pixels each are used for each eye, gives a total of 3840 features. Therefore, some abstraction is used here to further reduce the input data. Several approaches provide good abstraction (for example, see [8]). Our current approach uses four locally normalized mean $\mu_{i,j}$ and variance $\sigma^2_{i,j}$ (Equations 11 and 12), in the vicinity of only four positions - $(i,j) = (3,3), (3,8), (8,3), (8,8)$ - for each one of the 16×15 images of a given level:

$$\mu_{i,j} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{G^{(i)}_{i+m,j+n}}{G^{(i)_{Max}}} \quad (11)$$

$$\sigma^2_{i,j} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \left(\frac{G^{(i)}_{i+m,j+n}}{G^{(i)_{Max}}} - \mu ij \right)^2 \quad (12)$$

The above local statistical moments are calculated for each one of the six Gaussian images in Figure 3 and for the magnitudes of motion and stereo disparity computed in the attentional phase. For a given level, this gives a total of 112 input features ($4Stereo + 4Motion + 24Mean + 24Variance$) for each eye. Also, as a result of the averaging and variance, feature matching is more tolerant of scaling, rotation, and shift. Experiments indicate rotations up to 30 degrees are acceptable. Identification is done using

an associative memory implemented by a multi-layer perceptron with a back-propagation training algorithm (BPNN) [9]. The BPNN maps the features to an address in a long term memory (LTM), which stores various other information. It is important to note that other classifiers could be used, such as those in [5, 13]. We argue that the BP network approach used here 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 is used to determine if a representation is new, and can also be used in top-down attentional tasks to keep attention in a given region.

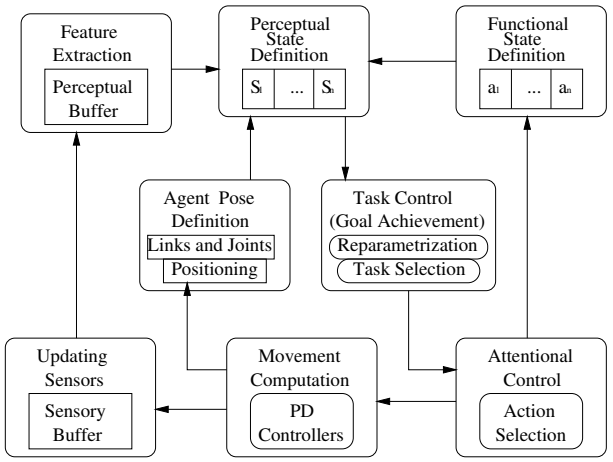

Figure 5. System Control Architecture

Once a representation is classified as new or identified, the current position in the attentional maps is updated. The attentional features, which are 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 (creating new nodes in the hidden and output layers) and retraining the BPNN. Note that information in the LTM can be retrieved using features from any resolution level. In general, the resolution level depends on the task, available time, and image characteristics, and is determined by the attentional mechanism.

6. Active vision architecture and controllers

Figure 5 shows the main aspects of a general software architecture used to implement the robotic system (this architecture is better described in [3]). Briefly, information from the sensory buffers are abstracted to construct a perceptual buffer (feature maps). The robot uses this set of

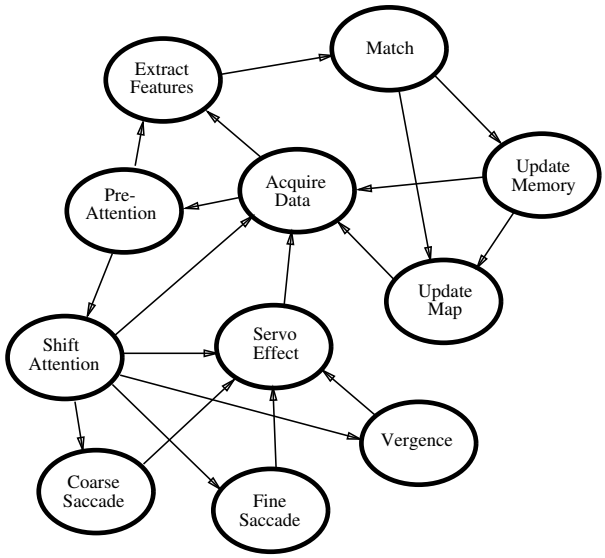

Figure 6. Behavioral program for a monitoring task.

features plus its pose and functional state to define its perceptual state. The task controller evaluates the current task goal achievement (monitoring task). Based on the results of this evaluation, it re-parameterizes the current task (allowing the creature to choose the right set of actions) or eventually selects a new task (if the goal is achieved - this is not the case here, since we have only a monitoring task). After this task selection/reparametrization, the attentional mechanism operates based on the current task parameters. Based on a task dependent policy, it changes attentional focus (topic of interest) and calculates the movements to be applied to the degrees of freedom of the stereo-head to attend the new position. These parameters are sent trough the servo controllers, which effectively moves the stereo-head to its new pose. This brings the stereo-head to a new pose and also puts a new set of information in the sensory buffers, re-starting the cycle (feature extraction). A "Controller Oriented" approach gives a better high-level explanation on how the active vision architecture works in the monitoring task. As in this work we adopt a hand-coded solution (a simple strategy), the result is the behavioral program shown in Figure 6. Basically, on the upper-right side of the figure we have the categorization behavior, and in the lower-left side the attentional behavior.

7. Experiments and results

Experiments involving attention and categorization behavior in a monitoring (or surveillance) task were performed. Basically, many instances of objects of various

types 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 prioritized search (monitoring) 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 7 shows a sequence ilustrating this type of experiment. The stereo head goes after the motion cues, detecting all objects on the table.

Figure 7. Robot following motion appeals.

In the second experiment type, there is no motion cues, so the robot relies solely on intensity cues. Figure 8 ilustrates one of the experiments of his type. Here the attentional mechanism makes the robot visit all regions, using mainly mapping term T_{ij} and interest rate E_{ij}.

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

Figure 8. Attention without motion cues.

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 eventually returns its attentional focus to that region by using T_{ij} and E_{ij} terms. Then it detects the eventual change by comparing current attentional features with stored ones. Figure 9 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 behavior, 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.

Figure 10 shows BPNN confidence (simultaneous activations in its last layer), using several instances of four types of objects, placed on the table with controlled poses. For each instance, upper line is highest activation (object on learned pose). 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 the rotations up to 30 degrees were well suported by the system. Figure 11 shows only new object types discovered (new representations) in one of the experiments. The figure also ilustrates the vergence mechanism working, since same object is in focus for each image pair.

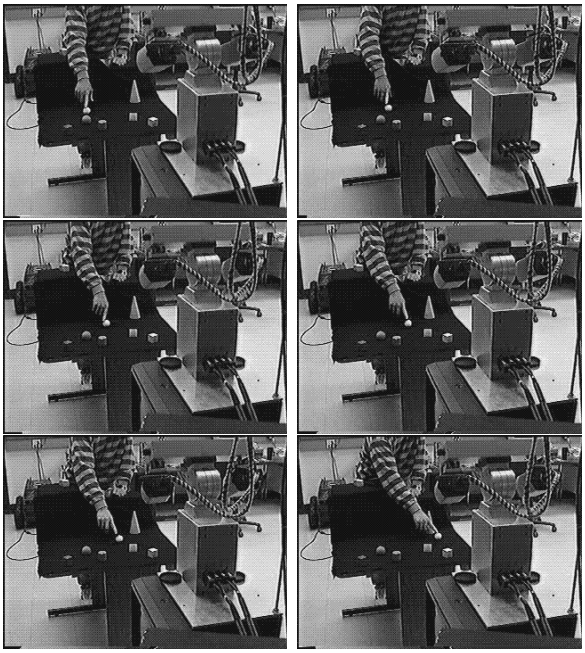

Figure 9. Robot updating attentional maps (tracking of a “golf ball”).

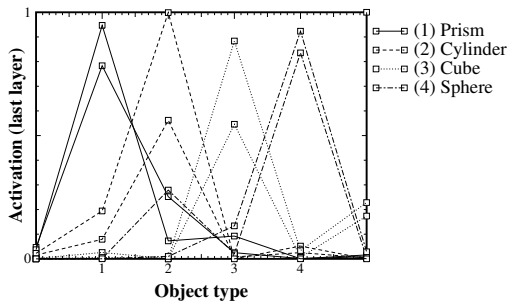

Figure 10. Activations for objects in BPNN.

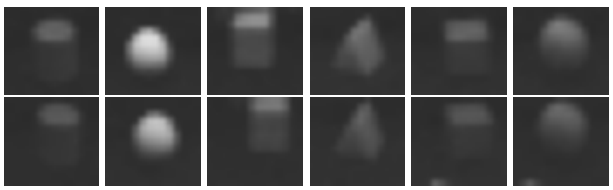

Figure 11. Different types of objects detected.

Several experiments were done to test the associative memory (BPNN) performance. Figure 12 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 [9]. In practice, this issue does not compromise 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 very small, not interfering in system performance.

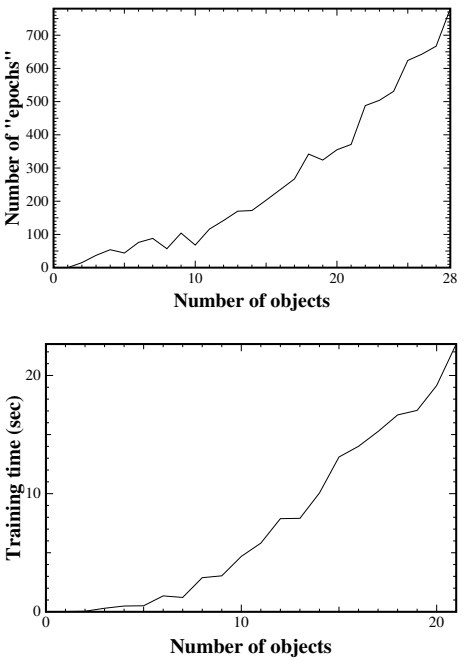

Figure 12. Training performance.

During some experiments, we also collected system performance data. Table 1 shows the time required for each of the processes involved in the monitoring task. The first column identifies the process, and the remaining columns show the minimum, maximum, and average times required, sampled over several hundred control cycles. The host computer used (a some 40 MHz Sun Sparc 10) has since been replaced with a some 300 MHz Sun Ultra 10 board that shares the Datacube bus for improved data transfer rates.

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
Salience maps	0.067	0.134	0.075
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
Memory match	0.012	0.028	0.019
Total categorization	0.323	0.353	0.333

Table 1. Processing time required in each phase.

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 interface, making saccade as fast as a human being.

8. Conclusion, discussion and future work

The immediate application that comes to mind to test our architecture is surveillance or monitoring tasks. In such tasks, a map of the environment is incrementally constructed and dynamically updated. Questions like "*who or what are in the lab?*" or "*is/was there an object at that position?*" can be addressed by simple analysis of this dynamic attentional map. Other applications like automatic interpretation of agent behaviors, study of dynamical issues of environments, building security and policing can use such an approach.

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 here 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 basis not only for this simple monitoring task, but also for 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 simple approach used for directing attention can be improved with a weight function that varies according to the task. The use of reinforcement learning [11] 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. The result would be a more versatile function for directing attention, perhaps closer to a biological model. Also, 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 saliency maps are calculated. Then, by using identification/recognition behavior, the most salient regions from the saliency maps (candidate objects) can be evaluated as being the searched one. In the same way, control tasks involving covert attention (changing the interest region out of fovea region, without any physical movements) can also use the same architecture.

References

[1] C. Ferrell. Orientation behavior using registered topographic maps. Technical report, MIT, Cambridge, MA, 1998.

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

[3] L. M. G. Gonçalves, F. W. d. Silva, R. C. Farias, and R. A. Grupen. Towards an architecture for artificial animated creatures. In *Proc. of VI CGS/IEEE Computer Animation Conference*, May, 3-5 2000.

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

[5] T. Kohonen. *Self Organizing Maps*. Springer Verlag, 1997.

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

[7] 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.

[8] S. RAVELA and R. MANMATHA. Retrieving images by similarity of visual appearance. *Workshop on Content Based Access of Image Databases (with CVPR'97)*, 2:311–347, 1997.

[9] 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 Press, 1993.

[10] 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.

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

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

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

[14] C. J. WESTELIUS. *Focus of Attention and Gaze Control for Robot Vision*. PhD thesis, Linköping University, Sweden, S–581 83 Linköping, Sweden, 1995. Dissertation No 379, ISBN 91–7871–530–X.