

# A Framework for Attention and Object Categorization Using a Stereo Head Robot\*

LUIZ M. G. GONÇALVES<sup>1,2</sup>, ANTONIO A. F. OLIVEIRA<sup>2</sup>, AND RODERIC A. GRUPEN<sup>1</sup>

<sup>1</sup>Laboratory for Perceptual Robotics - Dept of Computer Science  
University of Massachusetts (UMASS), Amherst MA 01003 USA  
(lmarcos, grupen)@cs.umass.edu

<sup>2</sup>Laboratório de Computação Gráfica - COPPE Sistemas  
Universidade Federal do Rio de Janeiro (UFRJ), CP 68511, Rio de Janeiro, RJ 21945-970  
(lmarcos, oliveira)@lcg.ufrj.br

**Abstract.** This work describes a framework for dealing with attention and categorization using a robot platform consisting of an articulated stereo-head with four degrees of freedom (pan, tilt, left verge, and right verge). As a practical result of this development, the system can select a region of interest, perform shifts of attention involving saccadic movements, perform an efficient feature extraction and identification/recognition, incrementally construct a world map, and keep this map consistent with a current perception of the world. Another important result for the attentional mechanism is that the system is capable to visit all regions of its restricted world, selecting one region at a time according to a salience map. For identification, the system starts without any knowledge of the environment and increases its knowledge base (associative memory) as necessary to deal with a current set of objects.

## 1 Introduction

This work presents a robotic system which is able to perform tasks involving attention and pattern categorization. We use visual information acquired in real-time by a stereo head hardware platform to provide on-line feedback about eventual changes in the environment. This feedback is in form of actions, resulting in a behaviorally active system that runs according to its perceptual state. Note that attention and identification and/or recognition are essential tasks in active vision systems. Also, we can get another motivation by looking towards finding an useful robotic system able to foveate (verge) the eyes onto a region of interest, to keep attention on the same region if more information is necessary to perform any given task (for example, allowing an arm to touch or to grasp an object), and to choose another region if the current one is no more of interest, by shifting its focus of attention. To validate such a “useful” system, a task involving all the above aspects must be defined. In our case, possible tasks are object recognition and identification for inspection (monitoring or surveillance), spatial orientation, and eventually navigation (path-planning). A map of the environment must be incrementally constructed and dynamically changed. Besides the pattern representation, this map also contains information about position and orientation. Once this kind of map is constructed, a robot agent is allowed to perform more specific tasks. Moreover, by adopting an active behavioral strategy we provide a dynamic way to interact with dynamic environments.

Basically, we use a bottom up driven salience map to direct attention. As result of this, a region of interest is selected and saccadic movements are computed for the eyes, eventually involving neck movements (pan and/or tilt), to put a region of interest in the fovea. Then a feature extraction is performed providing changes in the perceptual state. An associative memory maps these features into a pattern address, allowing the system to recognize/identify a possible instance of a representation or to discover new categories (unknown objects). Finally, an efficient mapping completes the architecture of such a system.

This research does not intend to suggest or describe biological models, to explain biological behaviors, or to explain the functionality of biological systems. The main purpose of this work is to give an active vision behavior to a robot. However, most parts of the computational architecture are inspired by the biological system, with some modification. So, some terminology resembling the parts of a biological organism are broadly used in the text.

## 2 Related work

Identification/recognition, feature mapping, and attention has been widely studied in the last two decades. Ultimately, most researchers have tried to reproduce or to imitate the human biological system and behavior. In this direction, the work of Kosslyn [6] suggests a descriptive model to explain how identification and recognition happen. The model suggests that features extracted directly from the eye images and also brain mental images formation (or image completion) are used. Besides missing explanations and practi-

---

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

cal difficulties, it is relatively easy to develop a visual system following that description. Ballard [8] also shows some work providing good models for recognition and identification. That work presents a set of operators based on Gaussian partial derivatives for feature extraction. The operators are suggested to be similar to biological models. We use such operators with some modification in our work for the same purpose.

On joining attention and identification topics, Rybak et al [10], using a simple model with monocular stationary images, treat perception and cognition as behavioral processes. 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 verifications of expected image fragments at the new 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 problem with this architecture is its incompleteness (no stereo features are used to help recognition but only a planar sequential perception). Also, it uses stationary image frames, not considering temporal aspects like motion.

A good approach towards providing a computational model to explain the neuro-physiology of attention can be found in the works of Van der Laar ([12, 13]. In those works, yet using stationary images, a multi-feature extraction is performed computing several feature maps. An attentional neural network receives a task dependent input, and gathers information from the feature maps to the salience map. Then, the place where to put attention is simply given by taking the most salient position in that salience map. In a similar approach, Koch et al [4] propose a model for attention that uses linear filters for orientation and spatial frequency extraction. These filters tuned for 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. A noise model together with a decision strategy are assumed in order to relate the model output to psychophysical data. The salience map is calculated and a statistical model determines the next attention-window.

For attention, we adopt an approach resembling the ones described above, considering an adapted set of Gaussian partial derivatives (order 0, 1, and 2 in two directions each). As we have sequences of image pairs, motion (actually, two directional gradients of consecutive frames differences) and stereo are also taken into account to determine the next region in which to pay attention (generation of a salience map). The main difference of our approach is that

we realize all of these in an active vision framework accomplishing real-time processing, instead of stationary images. Note that we need to effectively promote saccadic movements, eventually involving also pan and tilt besides the vergence motors, to get the attention window in the fovea (center of the current image frame).

An interesting approach involving visual and oculomotor mapping is the work of Ferrell [1]. She uses registered, multi-modal, topographically organized maps of the sensory-motor space to orient a robot (COG) 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 our approach we also use topographically organized multi-scale maps for feature extraction. The system selects one level at a time for the high level processes (identification and mapping). This selective aspect provides a substantial reduction in the amount of processing necessary for the feature extraction.

### 3 Stereo Head and Image Processing Devices

The robot used for this work consists of a Stereo Head platform shown in Figure 1. It has two cameras mounted on the top of a TRC Bisight head with four mechanical degrees of freedom which are shown in Figure 2. Some software restrictions to the movements can also be seen in that Figure (dotted lines). These restrictions are applied to avoid hard limit collisions. Motion commands can be sent via a PMAC/Delta TAU interface to control directly the pan (horizontal or lateral rotation), tilt (head inclination), left camera vergence, and right camera vergence. The cameras also have a zoom and focus controllable (independent of each other) via the PMAC controller, which are not used in this work.

The images acquired from each camera serve as input for the Datacube which consists of several image processing devices integrated using a pipeline array processing architecture. This architecture allows to real-time process the acquired images up to 30 frames per second. The image processor uses the concept of surfaces, pipes, and PATs. Each surface is an array of integer values (an image) that can be stored in one of 6 memories (storing devices, up to 4 Mb each one) for each one of two stereo boards. A pipe consists of taking one or more source surfaces, perform one or more image processing operations, and store the resulting surface or surfaces in a specified storing device. One

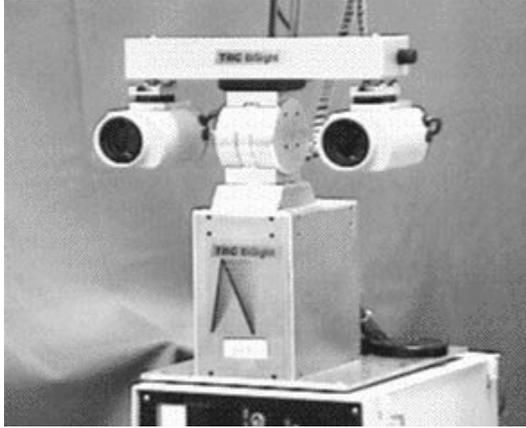


Figure 1: Stereo Head platform, consisting on two cameras mounted on a head, with 4 mechanical degrees of freedom.

can have up to 4 pipes running in parallel at a time. A PAT is necessary if one needs to execute pipes which depends upon the execution of other pipes (for example, if it uses data generated by other pipes). By using a PAT, the execution of a pipe can be deferred to be performed cyclically on every occurrence of an event which is generated by another pipe. Or a pipe can be executed as a one-shot pipe, on the generation of a triggering event created either by a pipe or by any host procedure, for example to solicit data from the IP device. In this sense, a control-loop architecture would include a combination of continuous and one-shot running pipes, to transform the input data and provide sufficient abstraction (or data reduction). The last operation on a processing cycle would eventually be the transference of abstracted data from the Datacube architecture to the host computer application. The last will decide in a final instance which high-level actions to perform (eventually a head movement). Some examples of common mathematical operations that can be computed inside the image processor are: correlations (or convolutions), Sobel gradients, Hough transforms, statistical operations, dyadic operations (sum, subtraction, or multiplication of two images) and monadic operations (thresholding and transformations using look-up tables), and feature extraction. The neighborhood multiply and accumulate device (convolutor) allows a pre-definition of a set of kernel masks with diameters up to 8x8 pixels. Then, one of these kernels can be selected at a very fast speed to be applied on a given surface.

#### 4 Controllers and the Architecture

We have developed a control architecture for a multi-modal sensory system which is described in [2, 3]. Based on that architecture, a "Controller Oriented" approach is used in this work for the implementation of the active vision system. A controller operates in a loop, transforming the

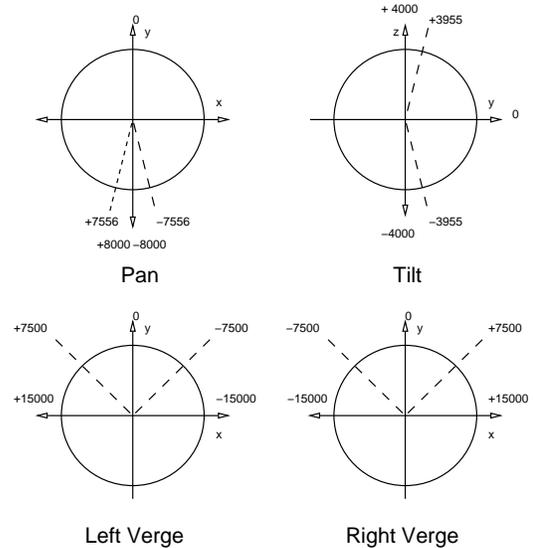


Figure 2: Degrees of freedom of the Stereo Head and motion range for each motor in number of counts. Dotted lines are software restrictions to avoid hard limit collisions.

input into output to satisfy a control strategy (policy). In general, the input is information regarding the current perceptual state like sensory information and robot pose. The transformation may be a physical action, such as a movement generated by the robot actuators, or other operations not involving physical movements, such as computations. The output is in the form of a report, updating the robot positioning or the perceptual state. Once a controller finds an equilibrium condition satisfying the control strategy it sets flags, asserting a predicate state. Those flags define a state vector which is shared between various controllers. Depending on the values of this state vector and on the policy adopted (task being performed) a controller will run, changing the predicate state again. In this way, a policy or behavioral program is established with a set of controllers and a set of flags representing the predicate state.

In a complex Markovian decision process (MDP), a policy for a given task consists of a sequential activation of one or more controllers in a loop, in general following restrictions, to try to reach the task goal. Each time one or more currently running controllers converge, another group is defined (by that policy), until the task is completed. A control-loop is established on top of a finite state machine defining the MDP, in general using reinforcement learning [11] or other approach. In this work, since we have a problem with a simple solution, we adopt a simple strategy carrying out the behavioral program shown in Figure 3. Each controller runs automatically when the previous controller converges.

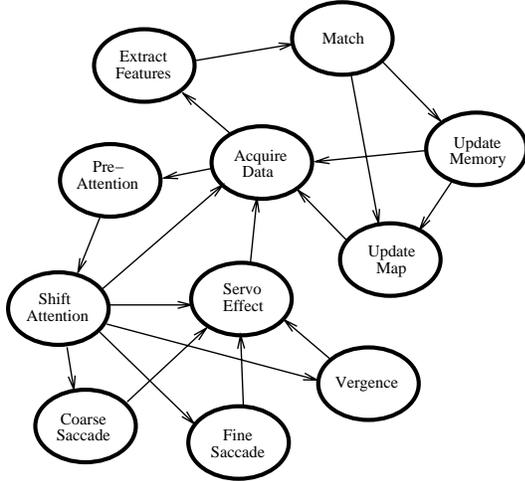


Figure 3: Behavioral program developed for attention and categorization. The arrows indicate transitions between states where a controller has not yet reached its convergence criteria. Circles represents reference states due to convergence events in the activated controllers.

## 5 A Multi-log-retina Representation (Visual Buffer)

A multi-logarithm representation is used in this work to encode the visual input data. A log-image is necessary to accomplish data reduction sufficient to obtain real-time processing and a multi-image is necessary for multiple feature extraction. These features will be used for both identification and attentional behaviors. A biologically inspired approach for generation of a multi-scale image could use Gaussian derivative filters with different scales (different standard deviation  $\sigma$ ) computing the derivatives directly from the original images. Then, sampling the resulting images at different resolutions only inside the area of scope of each level. Alternatively, a filter with a constant standard deviation could be used in a cascade process computing the next level from the previous one (as in [15]). It is shown in [16] that a Gaussian filter with a standard deviation of  $\frac{\pi}{\sqrt{2}}$  is ideal to generate the pyramid. We argue that the same result obtained by using the above approaches and with the same computational complexity is achieved by using the approach adopted in this work, described in the following. The data-reduction allows the host computer to perform other operations necessary to complement the image processing in real-time. In the experiments realized in this work, the Datacube Image Processing device can generate the 8 multi-scale images at a rate of 15 frames per second. This achieves a reasonable performance for real-time processing necessary in an active vision system.

The resulting representation for the multi-log-retina can be seen in Figure 4. Each one of the 8 images has 4 levels of resolution sampled by a factor 2. Six of these are

modified Gaussian partial derivatives (3 derivatives tuned in two directions each) and the remaining 2 are the first partial derivatives of frame differences in 2 directions, representing motion. So, we have 8 image derivatives overall, being 2 of order 0, two of order 1, and 2 of order 2 for intensity images, plus 2 of order 1 for the motion images. This data transformation is done in two phases for both Gaussian and motion parts. The first phase is a multi-scale image (like a pyramid) generation and the second phase is the derivatives computation.

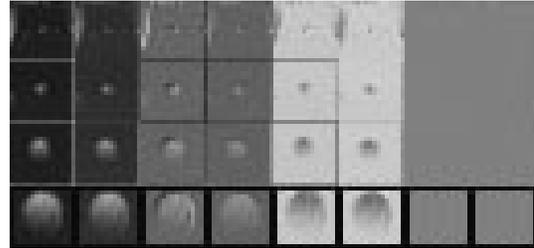


Figure 4: Multi-logarithm feature vector generated for identification and attentional behaviors.

### 5.1 Multi-scale Image Generation

The size of the original images captured by the two stereo cameras (further simply denominated eyes) is  $512 \times 480$  pixels. For the Gaussian part of the retina, the original images are directly used as input for the multi-scale image generation. For the motion part, the difference images of two consecutive frames (one previously stored and the one being acquired) are computed first. Each level is generated (for both motion and Gaussian parts) by applying a mean filter in the neighborhood of each pixel of those images and sampling it by a certain resolution factor, which is a function of the current level being generated. The diameter of that neighborhood as well as the region of scope of the input images which will be affected by the filtering also depend on the current level. For the initial level (coarse resolution level) the filter diameter is 8, it is applied in the whole image and sampled on every 8 pixels interval. For the last level (fine resolution), the filter diameter is 1 and the region of the image which it is to be applied is the region composed by the central  $64 \times 60$  pixels. So, for the last level, a simple surface transfer is performed without any filtering, with a sampling resolution equals to 1. At this point, each level has a size of  $64 \times 60$  pixels, resulting in two multi-scale images for each eye: one is an intensity-image and the other one is a difference-image.

## 5.2 Computing the Derivatives

As a second phase, partial derivatives are computed at each level of the multi-scale images to establish the desired features. For the Gaussian part, the multi-scale intensity image generated in the previous phase is convoluted with an adapted Gaussian (partial derivative) kernels in two directions each. The kernel masks are computed by means of Equations 1, 2, and 3. Also, a reduction to  $16 \times 15$  (sampling at every 4) pixels is performed in this phase reducing even more the volume of data. The Motion part is also computed in the same way. The first derivatives in two directions are computed generating the final motion multi-scale images (2 for each eye). Equation 4 is used to create the kernel mask. Note that the same Gaussian derivative is applied to the motion images. This helps reducing the amount of noise. An ideal approach could use those derivatives of difference images (our representation of motion) to actually compute the motion field for the whole images, using relaxation or other iterative approaches. The motion field computation is not necessary for attentional purposes, and for identification purposes such an approach is expensive.

$$\left. \begin{aligned} G_x^{(0)}(x, y) &= ke^{ax^2} \\ G_y^{(0)}(x, y) &= ke^{ay^2} \end{aligned} \right\} \quad (1)$$

$$\left. \begin{aligned} G_x^{(1)}(x, y) &= 2ake^{a(x^2+y^2)}x \\ G_y^{(1)}(x, y) &= 2ake^{a(x^2+y^2)}y \end{aligned} \right\} \quad (2)$$

$$\left. \begin{aligned} G_x^{(2)}(x, y) &= 2ake^{a(x^2+y^2)}(2a(x^2+x)+1) \\ G_y^{(2)}(x, y) &= 2ake^{a(x^2+y^2)}(2a(y^2+y)+1) \end{aligned} \right\} \quad (3)$$

$$\left. \begin{aligned} M_x^{(1)}(x, y) &= 2ake^{a(x^2+y^2)}x \\ M_y^{(1)}(x, y) &= 2ake^{a(x^2+y^2)}y \end{aligned} \right\} \quad (4)$$

$$\forall(x, y) \in [(-d, +d), (-d, +d)]$$

$$\text{where } a = \frac{-1}{2\sigma^2}, k = \frac{1}{\sigma\sqrt{2\pi}}, d = 3, \text{ and } \sigma = 1.7$$

## 5.3 Computing Stereo Disparity

Stereo disparity is computed in the host computer memory, after the multi-scale Gaussian image generation. A simple approach is used to compute stereo, by maximizing correlation measures using the second order Gaussian derivative images. One alternative approach could use the spatial frequency information to compute stereo directly from the input images inside the Datacube architecture. Such an approach using a phase shift model is presented in [15], using a simulation platform. Figure 5 shows the schema used for the stereo computation. Since the images are in a multi-scale representation, the results from one level (scale) are

used to predict the disparity on the next level. For the initial level, as the vergence movement has some constraints (see section 6.3), the disparity is inside a threshold. This cascade process gives a substantial reduction in the amount of computation necessary to find the best match for a point. Note that another constraint that can be used is given by the relative symmetry of the images with respect to the cyclopean axes. The cyclopean axis is the line defined by the central point in between the eyes and the horopter (point in which eye axes cross). Also, another detail to observe is that disparity is computed only for the  $x$  direction because we have a controlled geometry system (no  $y$  disparity).

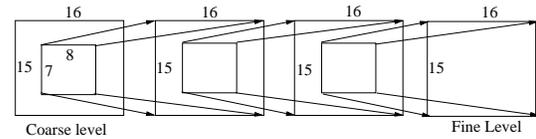


Figure 5: Stereo computation process in cascade. Each level predicts disparities for the next level.

## 6 Attentional Behavior Control

Basically, the attentional behavior expected for the system is to choose the most salient region in the world and to put its focus of attention in that region. This involves computing a salience map which is stimuli biased plus map-potential biased. Then, taking the winner position of this salience map and generating a saccadic movement (verge and eventually pan and/or tilt) to get the region in the center of the fovea. This salience map is computed by using the abstracted data provided by the Datacube (perceptual cues or stimuli biased) and also by looking for information contained in a world map currently being constructed (map-potential biased). Note that one could consider only the perceptual cues inside the field of view to define the next region of interest. But, in this case, there is no guarantee that the system will attend to all locations in the world. So, attended locations have to be potentiated in an internal map telling the system that they have been previously visited. This map also encodes other type of previous information, once a fast check is necessary to detect changes eventually occurred in the environment.

### 6.1 Defining a Target (Pre-attention)

The generation of the stimuli plus map biased salience map is done in a pre-attentional phase. This map also has a pyramidal representation like the images of the the multi-log-retina. Starting from the coarse level, an activation value is calculated for each position based on some normalized weight function of the perceptual cues plus the corresponding normalized activation in the world map. That function

is task dependent and can be learned using a neural network (see [13]) or reinforcement learning (see [3]) approach. The world map activation, initially high for all regions, is set to zero for a region if that region has received attention. Then, every time the pre-attentional procedure runs, that world map activation value increases slowly until it becomes high again. This simple behavior makes the system change its attention window from one region to another, covering the whole world and eventually returning to a region which has been visited previously, detecting possible changes that might occur in the environment. The result is an inspection task, where the robot has to keep a current representation of the world consistent with the reality. Note that the system will not visit the same region twice in sequence. Also, note that one region might be visited more than once, before the system completes all regions, depending on the function used to potentiate the world map activation values.

$$M_{ij} = w_M \sqrt{(m_{x,ij}^{(1)})^2 + (m_{y,ij}^{(1)})^2} \quad (5)$$

$$I_{ij}^{(0)} = w_{G^{(0)}} \sqrt{(g_{x,ij}^{(0)})^2 + (g_{y,ij}^{(0)})^2} \quad (6)$$

$$I_{ij}^{(1)} = w_{G^{(1)}} \sqrt{(g_{x,ij}^{(1)})^2 + (g_{y,ij}^{(1)})^2} \quad (7)$$

$$I_{ij}^{(2)} = w_{G^{(2)}} \sqrt{(g_{x,ij}^{(2)})^2 + (g_{y,ij}^{(2)})^2} \quad (8)$$

$$S_{ij} = P_{ij} + D_{ij} + M_{ij} + I_{ij}^{(0)} + I_{ij}^{(1)} + I_{ij}^{(2)} \quad (9)$$

Features considered for attention are stereo disparity ( $D_{ij}$ ), and the intensities of motion and of the Gaussian derivatives given by Equations 5, 6, 7, and 8, respectively. After the attentional features have been computed, Equation 9 is used for computing the salience map using those features. That equation is a simple summation of the above activation values, since the weights  $w_M$ ,  $w_{G^{(0)}}$ ,  $w_{G^{(1)}}$ , and  $w_{G^{(2)}}$  have been previously applied in the intensities computation. The factor  $P_{ij}$  is defined as proximity, another activation value which is proportional to the distance between the position in the salience map and the fovea. Regions close to the fovea have a higher activation.

## 6.2 Shifting Attention (Coarse Saccade Generation)

Shifting attention involves taking the most active region over all levels in the pre-attentional (salience) maps and effectively moving the eyes to the corresponding location. A coarse saccade movement is computed for both eyes to shift attention to the winner region. Since we have one salience map for each eye, we apply here the concept of dominant eye as the one whose salience map contains the most active region. For the dominant eye, the target position is simply determined by the displacement from the current position to the winner one. The target for the non dominant

eye is computed from the dominant eye target by adding the stereo disparity. Since the targets are defined for each eye, a model of the goal is acquired for the dominant eye to eventually help performing fine saccadic corrections. Features from a windowed region around the dominant target are taken for all levels of resolution. To complete the coarse saccadic generation, the displacements to be applied to the degrees of freedom of the Stereo Head are computed from the displacement determined for the eyes. Note that the displacement is in eye centered coordinates (a multi-scale image coordinate frame) and must be transformed into pan, tilt, and verges to be sent to the servo controllers. Some constraints are put in those degrees of freedom. The cyclopean angle must not be greater than a threshold. Actually, we experimented empirically that inside a range of 15 degrees the features provided by both eyes still give good results for object categorization. When the opposite happens (cyclopean angle greater than 15 degrees), a pan (like a neck movement) is necessary to get the target inside that range. Also, another constraint is imposed such that the verge axes must not be very opened (no more than parallel) and not very closed (less than 45 degrees) in relation to each other. A correction for that is applied to the non dominant eye. The tilt is directly computed from the dominant eye vertical displacement. As the Stereo Head has a controlled geometry, both eyes have the same tilt. Since the new position is determined for all degrees of freedom, the servo-effector operates, effectively moving the hardware platform to attend the new location.

## 6.3 Adjusting Attention

Due to errors, after a coarse saccade the eyes may not be at the specified target location (in general very close to it). Therefore, fine saccades are generated in an iterative approach to maximize correlation between the acquired target model and the dominant eye image center. This iterative process goes from the coarse level to the fine level. It will converge when the level which has determined the shift of attention reaches the maximum correlation value in a position at a threshold distance from the image center. At the same time that each fine saccade is generated, the vergence mechanism runs for the non dominant eye. Displacements are calculated for the non dominant verge axis to bring it to a position in which the image centers have the maximum correlation value. A threshold is used to avoid situations in which there is no match inside the field of view. In this way, the eyes verge at the same time or shortly after the fine saccade process convergence.

## 7 Identification

Since both image centers (or one in case of occlusions) are focused on a region of interest an efficient object catego-

rization can take place. Identification is done by using an associative memory implemented using a Back-propagation neural network (BP), shown in Figure 6. The associative memory matches the abstracted features computed from the Datacube output into an address of a long term memory which has stored all kind of information that can be retrieved by those features. Notice that information from an arbitrary resolution level can be used for the match. In a general situation, the level of resolution from which information is to be taken depends on the task (top-down and/or bottom-up attention), on the time available, and on the image characteristics. It is a function of the attentional mechanism to define what level shall be used.

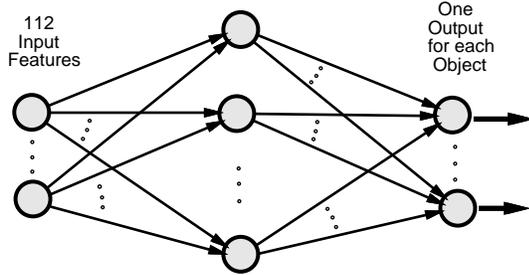


Figure 6: Backpropagation neural network used as associative memory. The output layer increases dynamically.

The BP network used has one input node for each abstracted feature. The number of nodes in the output layer changes dynamically. A new node is created for each new representation detected in the world. A weight function of the minimum and the maximum error given by the training procedure is used as threshold to decide if a representation is a new one. The hidden layer has number of nodes determined empirically. Actually, making that number equal to 1.5 times the number of nodes in the output layer gives good results. Equation 10 gives the best match. Equation 11 is used for the training.

$$o_i = (1 + e^{-\sum_{i=0}^A \omega_{ij} x_i})^{-1} \quad (10)$$

$$\Delta \omega_{ij}(t+1) = \epsilon \delta_j o_i + \alpha \Delta \omega_{ij}(t), \quad (11)$$

where  $o_i$  is as defined above, and

$$\delta_j = \begin{cases} o_j(1-o_j)(y_j - o_j), & \forall j \in \text{output} \\ o_j(1-o_j) \sum_{k=1}^B \delta_k \omega_{jk}, & \forall j \in \text{other} \end{cases}$$

Note that other classifier could be used here. For example, the ones used in [7, 5, 14]. We argue that the BPNN approach used here gives good results on identification and also returns what the activation for a given index in the output layer is (a normalized value in between 0 and 1). This activation value will be used to determine if a representation

is a new one, or to guide attention (top-down attentional tasks can use it to keep the attention in a given region).

## 7.1 Feature Extraction

Some experiments done using the multi-log-retina representation directly as features for the associative memory match gives good results. In those experiments, 8 feature vectors ( $2G_0 + 2G_1 + 2G_2 + 1Motion + 1Stereo$ ) composed of  $16 \times 15$  pixels are used, for each eye. A total of 3840 input features require a lot of computation. Therefore, some abstraction must be used to reduce even more the input data. There are several approaches that would result in a good abstraction (see for example [9]). The approach currently used considers information extracted by sampling the  $16 \times 15$  feature images in only 4 positions. For the Gaussian features, both directions are considered and for motion, its intensity previously computed in the pre-attentional phase is used. For one level, currently considered, this gives a total of 112 input features, being  $4Stereo + 4Motion + 24\mu + 24\sigma$  for each eye. This is reasonable for computational purposes. Also, instead of the feature image values, a local mean and a local variance applied to the neighborhood of the Gaussian features are used (Equations 12, 13) for feature sampling. A normalization of contrast is included by considering the normalized mean and variance. Also, as the neighborhood is taken into account, the responses spread according to the amount of local energy present, representing more than a local characteristics. Moreover, the resulting feature representation carries some invariance with respect to scale, rotation and shift. Rotations up to 30 degrees are well supported in the experiments.

$$\mu_{i,j} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \frac{G_{i+m,j+n}^{(i)}}{G^{(i)Max}} \quad (12)$$

$$\sigma_{i,j}^2 = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \left( \frac{G_{i+m,j+n}^{(i)}}{G^{(i)Max}} - \mu_{ij} \right)^2 \quad (13)$$

## 7.2 Mapping Objects and Updating Memory

Once a representation is classified as identified or new, the world maps are updated with the current information extracted from that region. The features used for attention which are sufficient to detect any change are stored and the world map activation is set to zero to allow a shift of attention towards other region. If the representation is a new one, the associative memory is retrained. This involves evocating a supervised learning module which inserts the new set of features in the long term memory, updates the associative memory creating the necessary nodes in the intermediate and output layers, and retrain the network.

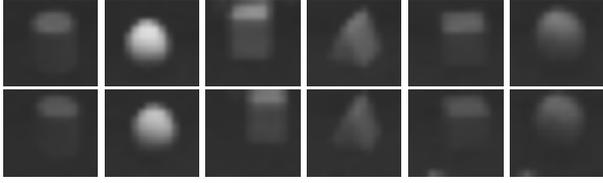


Figure 7: Pairs of images showing only the new objects detected in the environment. This figure also shows that both eyes are verged in the same objects. Top row is for left camera and bottom row is for right camera. From left to right: a red cylinder, a white golf ball, a natural wood cube, a red triangular prism, a blue cube, and a light green (dirty) tennis ball

## 8 Experiments and Results

Several experiments were done on tasks involving attention and identification behaviors. In the final tests both behaviors were integrated in a unique task. Basically, several instances of several types of objects are posted on a table. The robot, constrained to the region containing the table, has to learn characteristics of all objects inserting a representation for each one in the associative memory and updating its internal map. Figure 7 shows images of both multi-log-retinas (for the left and right cameras) recorded during one of the experiments. Although we have posted more than one instance for each object type and the robot has visited all objects, that figure shows only the new types of object detected in the environment. Also, it validates the vergence mechanism as for each pair both eyes are verged on the same objects.

In the expected attentional behavior, the stereo head has to move from one object to another, covering all objects on the table. Three modalities of attentional tasks were tested. In all three experiments, the robot visited all objects, discovering new representations, identifying existent ones, and mapping all objects. In the first experiment, we indicate sequentially the objects to the robot, by touching each object with a finger and coming to stand-still. This motion cue produces a high activation in the attentional process making the robot to put its fovea close to the object position. Then, without the initial motion cue, the intensity based cues wins in the attentional mechanism, putting the object completely in the fovea. In the second experiment, there is no initial motion cue, and the robot has to figure out for itself the regions where it has to go. So, the intensity based cues and the map interest value of each region are used for that. Figure 7 shows the different types of objects detected in the table for one experiment of this second type. The attentional mechanism works well putting all the objects in the fovea for this task. On the third experiment, after all objects on the table are detected and mapped, we either move an object from

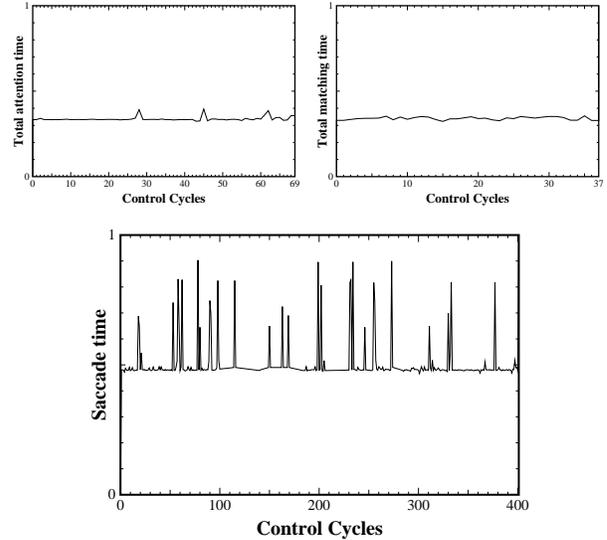


Figure 8: Total time required for attentional shifts, feature matching, and saccade generation. Time includes the data acquisition (with multi-log-image generation).

one position to another or else remove objects from the table top. As expected, the robot ratifies the world map for the changed regions. This is done using a combination of motion and intensity cues for movements that occur inside the view field or by using the inspection behavior that the system reaches after all regions in the restricted world are visited (see subsection 6.1).

Some experiments towards defining other meaningful results were also done. In those experiments we recorded some data while the system was operating. Figure 8 shows some of the experimental data obtained along several control cycles. The total times required for attentional shifts, saccade generation, and feature matching are shown in that figure. Table 1 shows separately the required time for each one of the processes involved in attention and identification tasks. In the second column are the minimum times to realize the tasks or phases. In the third column the maximum time. In the most right column are the average times over several hundreds of control cycles for each one of the respective processes. As we can see, the times above shown that a frame rate of up to four frames per second can be achieved by the system. But, this can be improved, as explained in the following. The times involving computations in the host computer can be considerably improved. The computer used in the experiments reported here is a Sun Sparc 10 (some 40 MHz processor). Currently, we have a dedicated Sun Ultra Sparc board in the Datacube cage that can operate closely to 300 MHz. Also, as the last board shares the same bus as the Datacube boards, the time for

Phase or process	Min(sec)	Max(sec)	$\mu$ (sec)
Computing retina	0.145	0.189	0.166
Transfer to host	0.017	0.059	0.020
Total acquiring	0.162	0.255	0.186
Pre-attention	0.139	0.205	0.149
Saliency map	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 matching	0.323	0.353	0.333

Table 1: Time required for each sub-process or phase. Computing the retina is done inside the Datacube. The pre-attention phase is done in the host computer using data from Datacube. Features for matching are computed also in the host computer. Both pre-attention and features computations includes an image display in the host computer. The saccade includes sending data to the PMAC interface and waiting for its completion.

transferring data from the Datacube will speed up. A bus-to-bus adaptor is necessary for reading data in the Sparc 10. Saccade is another component that can be improved by dealing with the gains in the PD controllers of the stereo head PMAC interface. An optimization can take place, making saccade as fast as a human being. A problem that might occur while increasing the gain, is that the head may present some instability, shaking.

## 9 Conclusion and Discussion

Despite of using only visual information in this work, the developed system is more general than that. Other sensory maps (haptic, auditive) can be easily incorporated to this platform. In this way, one could carry out a better saliency map and a better discriminative set of features for the associative memory match. By using a “controller oriented” approach with a supervisory loop controlling all resources, other processes can be developed independently and incorporated to the architecture.

From the experiments, we argue that it is practically impossible to deal with attention without dealing with object categorization. Moreover, not only all the sensory systems must act as a unity, but all the brain functions must be integrated together in order to start any development. In this sense, we have developed the basic architecture integrating a vision attentional mechanism and a neural network classifier (associative memory).

In this work we have used the same basic feature vector for both attention and identification purposes. By using intensity of the gradient of image differences as one of the

cues for attention, static regions with high intensity values subject to artificial lightening will eventually have different motion values. As artificial lightening is a discrete variable of time obeying to a cycle (sometimes equals to 60 Hz), intensity on the surface of a static object may vary over time. This is a good feature, since the system will shift attention to regions of high intensity values which are intuitively attractive. By also taking pondered local intensity features as attentional cues, attention is focused on positions that locally represents a region, more than a patch. In the hypothesis that this region is part of only one object, segregation can be done by fitting the best local average intensity and using that as a spread function for selecting a region that could eventually be an object.

Finally, one could ask why attention and identification are so important. Note that object categorization is necessary in almost all tasks that one can imagine. The “what” is the first important question involving any task. In its turn, the ability to change the focus of attention is the basis for cognition. Those two tasks are integrated to each other in a such a way that a behaviorally active system needs both sub-systems working to perform other tasks.

## 10 Future Works

Besides the simple and direct approach used in this work for directing attention, we will further realize studies towards finding a weight function that varies in time according to the task. In a first phase, the system can use motion cues to get a target and then use intensity cues (texture or contrast) to focus attention in the right place. Also, covert attention shift (changing region of interest without any physical movement) will also be included, as this feature is very useful for some tasks.

Focus can also be used together with the correlation approach used in this work to “tune” the system into an object and to help in the stereo vergence mechanism. In the occurrence of occlusions, only focus might be used. In this case, the dominant eye will give the approximate focus to a function that computes the angle necessary for vergence. Focus involves statistical measures in the image (changes in the histogram will determine the best focus), what can be done in the Datacube architecture.

Another future experiment that can be done is to consider bottom-up attention only on the coarse level (at least for motion). It seems that it happens in biological systems, since the magna-cellular pathway plays the major role in attention. In this case, an approaching object will receive attention in a certain distance, allowing the stereo Head to take any decision. Other levels will guide top-down attention. If an object needs specific attention in a certain region, the top-down mechanism will set the corresponding region in the superior level which has resolution necessary

to cover the detail. The system can select the level which completely involves the object by tuning its perceptual cues to a model acquired on-line and also by looking for continuity in disparity, intensity and/or motion, defining the attention window size. In case of an object covers more than one attention window even for most lower level, attention has to be shifted in the image and information has to be acquired in a manner that old information is kept in memory and new information completes the set of features available for recognition and identification.

Finally, the use of reinforcement learning [11] can play an important role towards finding a more reasonable weight function for attention. Given a set of tasks that the system is able to perform, the weights for the salience maps can be determined by using such approach. The system gets rewards if identification/detection of new objects and mapping occurs. The result would be a much more reasonable function for directing attention, perhaps closer to a biological model.

## References

- [1] Ferrell, C. 1998. Orientation behavior using registered topographic maps. Tr, Massachusetts Institute of technology, Cambridge, MA.
- [2] Gonçalves, L. M. G.; Oliveira, A. A. F.; and Grupen, R. A. 1998. A control architecture for multi-modal sensory integration. *XI International Conference on Computer Graphics and Image Processing (SIBGRAP'98)* 418–425.
- [3] Gonçalves, L. M. G.; Oliveira, A. A. F.; and Grupen, R. A. 1999. Multi-modal stereognosis. *To appear in III International Conference on Autonomous Agents (Agents '99)*.
- [4] Itti, L.; Braun, J.; Lee, D. K.; and Koch, C. 1997. A model of early visual processing. In *NIPS Int. Conference*.
- [5] Kohonen, T. 1990. The self organizing map. *Journal of Electrical and Electronics Engineers* 78:1464–1480.
- [6] Kosslyn, S. 1994. *Image and Brain. The Resolution of the Imagery Debate*. Cambridge, MA: The MIT Press.
- [7] Piater, J. H., and Grupen, R. A. 1999. A framework for learning visual discrimination. *To appear in Proceedings of the FLAIRS International Conference on Artificial Intelligence*.
- [8] Rao, R. P. N., and Ballard, D. 1995. An active vision architecture based on iconic representations. *Artificial Intelligence Journal* 78:461–505.
- [9] Ravela, S., and Manmatha, R. 1997. Retrieving images by similarity of visual appearance. *Workshop on Content Based Access of Image Databases (with CVPR)* 2:311–347.
- [10] Rybak, I. A.; Gusakova, V. I.; Golovan, A. V.; Podladchikova L, N.; and Shevtsova, N. A. 1998. A model of attention-guided visual perception and recognition. *Vision Research*.
- [11] Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: an Introduction*. Cambridge, MA: The MIT Press.
- [12] Van de Laar, P.; Heskes, T.; and Gielen, S. 1995. A neural model of visual attention. *Neural Networks: Artificial Intelligence and Industrial Applications, eds. Kappen, B. and Gielen, S.* 111–114.
- [13] Van de Laar, P.; Heskes, T.; and Gielen, S. 1997. Task-dependent learning of attention. *Neural Networks* 10(6):981–992.
- [14] Viola, P. A. 1996. Complex feature recognition: A bayesian approach for learning to recognize objects. AI Memo 1591, Massachusetts Institute of Technology.
- [15] Westelius, C.-J. 1995. *Focus of Attention and Gaze Control for Robot Vision*. Ph.D. Dissertation, Linköping University, Sweden, S-581 83 Linköping, Sweden. Dissertation No 379, ISBN 91-7871-530-X.
- [16] Westin, C.-F. 1994. *A Tensor Framework for Multidimensional Signal Processing*. Ph.D. Dissertation, Linköping University, Sweden, S-581 83 Linköping, Sweden. Dissertation No 348, ISBN 91-7871-421-4.