Towards a Framework for Robot Cognition *

LUIZ M. G. GONÇALVES[1 2], DAVID S. WHEELER[1], ANTONIO A. F. OLIVEIRA[2], AND RODERIC A. GRUPEN[1]

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

Abstract. This work describes a framework for control of attention and for pattern 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 work, the system can select a region of interest, perform attention shifts involving saccadic movements, perform efficient feature extraction and recognition, incrementally construct a world map, and keep the map 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 a salience map.

Keywords. Cognition, attention, categorization, learning

1 Introduction

This work presents a robotic system that performs tasks involving attention and pattern categorization. We use a real-time stereo vision platform to provide abstracted information about the environment. Based on this information, the system can select actions resulting in a behaviorally active system controlled by its perceptual state. Our goal is to develop a robotic system able to foveate (verge) the eyes on a region of interest, to maintain attention on the region as needed, and to shift attention when the current region is no longer of interest. Possible tasks include object recognition and identification for inspection, spatial orientation, or navigation. An environmental map containing pattern representation, position, and orientation information is incrementally constructed and dynamically updated. Using this map, the robot can perform more specific tasks. Moreover, by adopting an active behavioral strategy we provide dynamic interaction with changing environments.

This research is not intended to suggest or describe biological models or to explain biological systems. However, most parts of the computational architecture are inspired by biological systems, with some modification. Therefore, some biological terminology is used to refer to parts of the robot hardware.

Briefly, a bottom-up salience map directs attention and 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). An efficient mapping algorithm completes the system architecture.

*This work is supported by FAPERJ/Brazil and NSF under IRI-9503687, IRI-9704530, and CDA-9703217

2 Related work

Attention, feature mapping, and pattern categorization has been widely studied in the last two decades. Van der Laar et al. [12] provides a good approach to a computational model explaining the neuro-physiology of attention. An attentional neural network gathers information from feature maps to produce a salience map. The region of interest is determined by taking the most salient position in the salience map. Itti et al. [4] propose a model for attention that uses linear filters tuned for various orientations and spatial periods to compute a phase-independent linear response to visual stimuli. On joining attention and identification topics, Rybak et al. [10], also using a simple model with monocular stationary images, treat perception and cognition as behavioral processes. Kosslyn [6] 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. Rao and Ballard [8] presents a model using Gaussian operators for feature extraction. The operators are suggested to be similar to biological models.

Except for [6], which is a descriptive model and not a working system, all the above approaches fail in at least one of the following aspects: (i) it is not provided a reasonably complete model for cognition, including at least attention and categorization; (ii) it is not considered a reasonable set of features for attention and/or categorization; (iii) it is considered only stationary image frames, not including temporal aspects like motion or functional and behavioral aspects, nor does the approach provide real-time feedback to environmental stimuli.

For attention, we use features from an image filtered with Gaussian partial derivatives combined with motion and stereo to generate a salience map and determine the next region of interest. We also convert saccadic vergence movements to pan and tilt motion as necessary to foveate on an attention window. For pattern categorization, appearance-based features derived using the Gaussian operators plus stereo and motion patterns are used as input to an associative memory which remembers addresses into a pattern storage memory. The main difference between our work and the above is real time application in an active vision framework. Moreover, we have implemented a more complete cognitive model involving the control of attention and pattern categorization.

3 Stereo Head and Image Processing Devices

Our robot, 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. Motion is controlled via a PMAC/Delta TAU interface.

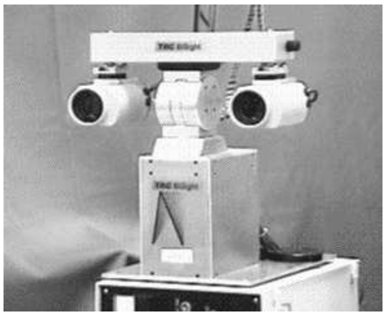

Figure 1: Stereo Head platform with 4 mechanical degrees of freedom.

Images from each camera are input to a Datacube pipelined array (image) processor (IP). Images are stored as integer arrays in any of six surfaces (memories) for each of the two stereo boards. Pipes take data from one or more surfaces, perform image processing operations, and store the result in another surface. Up to four pipes can run concurrently, and pipes can be chained together with minimal host computer intervention. The Datacube is used to reduce and abstract image data, which is then used by the host computer to decide which high-level actions to perform.

4 Controllers and the Architecture

We have developed a control architecture for a multi-modal sensory system described in [1, 2]. Based on that architecture, a "Controller Oriented" approach is used for implementation of the active vision system. A controller operates in a loop, transforming input into output to satisfy a control strategy or policy. In general, input is information regarding the current perceptual state. States are transformed using attentional control. The output is an update to the robot pose or the perceptual state. Once a controller finds an equilibrium condition satisfying the control strategy it asserts one or more boolean variables which form a control state that is shared among the working set of controllers. Depending on the values of this state vector and on a task dependent policy, another subset of controllers will run, changing the boolean state vector again. In this way, a policy or behavioral program is established by a set of controllers and a control-based state [3].In a complex Markovian Decision Process (MDP) [7], a policy for a given task consists of activation of one or more controllers, subject to constraints on allowable combinations, to reach the goal. A finite state supervisor can be derived using reinforcement learning [11] or some other approach, to solve such problems. In this work, we adopt a hand-coded solution, a simple strategy, resulting in the behavioral program shown in Figure 2.

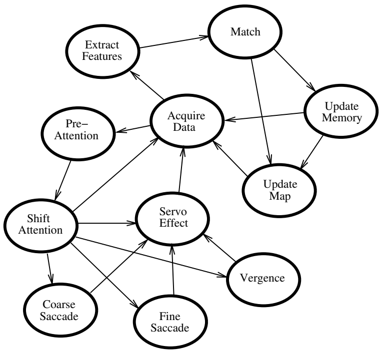

Figure 2: Behavioral program developed for attention and categorization. Arrows indicate transitions between states where a controller has not yet reached its convergence criteria. Circles represents reference states due to convergence events in the activated controllers.

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

A multi-scale representation is used to encode image data. Logarithmic data reduction permits the host computer to perform high-level processing in real time. Multiple images support feature extraction for identification and attentional behaviors. A biologically inspired approach for generation of multi-scale images could use Gaussian filters with different scales (σ) to compute derivatives directly from the original images, and could sample the resulting images at different resolutions within the area of interest. Alternatively, a filter with a constant σ could be used in a cascade process to compute the next level from the previous one as in [14]. We argue that the approach adopted in this work, described below, achieves the same result as the above approaches with the same computational complexity. In the experiments realized in this work, the Datacube IP device generates 64 images (32 per eye) at 15 frames per second, achieving reasonable performance for the real-time processing necessary in an active vision system.

Figure 3 shows the multi-log-retina representation of a sphere. Each of the eight image columns has four levels of resolution differing by a factor 2. Six columns are modified Gaussian partial derivatives (order 0, 1, and 2 in two directions each) and the remaining two columns are the first partial derivatives of the frame differences in two directions, representing motion. This transformation is performed in two phases: multi-scale image generation and derivative computation.

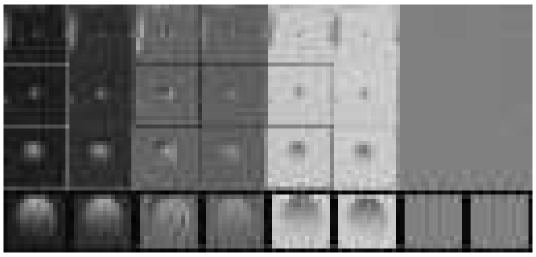

Figure 3: Multi-logarithm feature vector.

5.1 Multi-scale Image Generation

The original 512×480 pixel stereo images are used directly as input for multi-scale intensity image generation, and the difference between two consecutive image frames is computed for the motion images. 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 pixel intervals. For the finest resolution level, the central 64×60 pixel region of the image is simply taken. The result is multi-scale intensity and difference images for each eye.

5.2 Computing the Derivatives

For the Gaussian images, the multi-scale intensity images generated above are convoluted with Gaussian derivative kernels, given by Equations 1, 2, and 3, in two directions each, resulting in Equation 4. Four-pixel sampling is also performed, reducing the images to 16×15 pixels. The motion images are computed using Equation 5. The same first Gaussian derivative is applied to the motion images to help reduce 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, the motion field computation is unnecessary for attentional purposes, and expensive for identification purposes.

$$G^{(0)} = ke^{ar^2} \tag{1}$$

$$G^{(1)} = 2arke^{ar^2} \tag{2}$$

$$G^{(2)} = 2ake^{ar^2}(2ar^2 + 1) \tag{3}$$

where $r^2 = x^2 + y^2$, $a = \frac{-1}{2\sigma^2}$, $k = \frac{1}{\sigma\sqrt{2\pi}}$, $\sigma = 1.7$.

$$g_d^{(i)} = G_d^{(i)} * I_t \tag{4}$$

$$m_d = G_d^{(1)} * [I_{t+1} - I_t] \tag{5}$$

5.3 Computing Stereo Disparity

Stereo disparity is computed in the host computer after the multi-scale Gaussian image generation. One could use spatial frequency information to compute the disparity 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 use a simple cascade approach (see Figure 4) to compute disparity by maximizing correlation measures using the second order Gaussian derivative images. The cascade process substantially reduces the computations necessary to find the best match for a point. Since the images are a multi-scale representation, the results from one level are used to predict the disparity on the next level. For the initial level, the range of disparity is limited by vergence movement constraints (see section 6.3), and by the relative symmetry of the images with respect to the cyclopean axis (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.

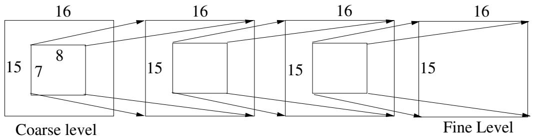

Figure 4: Stereo computation process in cascade. Each level predicts disparities for the next level.

6 Attentional Behavior Control

The desired attentional behavior for our system is to focus attention on the most salient region in 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 perceptual cues and stimuli in the abstracted data from the Datacube, and by interrogating the world map currently being constructed. Note that by considering only perceptual cues in the field of view, there is no guarantee that the system will attend to all locations in the world. Therefore, previously visited regions are given a low potential in an internal map to prevent an immediate revisitation. This map also encodes other information, allowing a fast check to detect changes in the environment.

6.1 Defining a Target (Pre-attention)

Salience maps are generated in a pre-attentional phase. Like the multi-log-retina images, salience maps have a cascade structure. Starting at the coarse level, an activation value is calculated for each position based on a normalized weight function of the perceptual cues and the normalized world map activation. The weight function is task dependent and can be learned using neural networks [12] or reinforcement learning [2]. The world map activation for a region is set to zero when a region is visited, and is increased each time the pre-attentional procedure runs. This technique makes the system change its attention window from region to region, covering the whole world but eventually returning to previously visited regions to detect possible changes. The result is an inspection or surveillance behavior, in which the robot maintains a representation of the world consistent with reality. Note that a region is never visited twice in a row, but (depending on the function used to update the world map activation values) a region may be revisited before all other regions are visited.

$$M_{ij} = w_M \left[(m^{(1)}_{x,ij})^2 + (m^{(1)}_{y,ij})^2 \right] \quad (6)$$

$$I^{(0)}_{ij} = w_{G^{(0)}} \left[(g^{(0)}_{x,ij})^2 + (g^{(0)}_{y,ij})^2 \right] \quad (7)$$

$$I^{(1)}_{ij} = w_{G^{(1)}} \left[(g^{(1)}_{x,ij})^2 + (g^{(1)}_{y,ij})^2 \right] \quad (8)$$

$$I^{(2)}_{ij} = w_{G^{(2)}} \left[(g^{(2)}_{x,ij})^2 + (g^{(2)}_{y,ij})^2 \right] \quad (9)$$

$$S_{ij} = P_{ij} + D_{ij} + M_{ij} + I^{(0)}_{ij} + I^{(1)}_{ij} + I^{(2)}_{ij} \quad (10)$$

The attentional features are stereo disparity (D_{ij}), and the magnitude of motion and Gaussian derivatives given by Equations 6 (motion), 7 (intensity), 8 (edge features), and 9 (Laplacian), respectively. After attentional features are computed, Equation 10 computes the salience map by a simple summation (the weights w_M, $w_{G^{(0)}}$, $w_{G^{(1)}}$, and $w_{G^{(2)}}$ were previously applied in the magnitude computations). The P_{ij} factor represents proximity (the distance between the position in the salience map and the fovea) and gives regions close to the fovea a higher activation.

6.2 Shifting Attention (Coarse Saccade Generation)

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. Each eye has a salience map; the dominant eye is the eye whose salience map contains the most active region. 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.

6.3 Adjusting Attention (Fine Saccade and Vergence)

After a coarse saccade, 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 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.

7 Identification

Once both image centers (one in the case of occlusions) are focused on the region of interest, object categorization takes place. Identification is done using an associative memory implemented by a back-propagation neural network (BP) (see Figure 5). The associative memory maps features abstracted by the Datacube to an address in a long term memory which stores various information. Note that information in the long term memory 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.

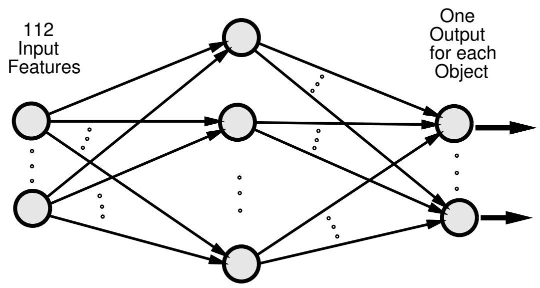

Figure 5: Backpropagation neural network used as associative memory. The output layer increases dynamically.

The BP network 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. 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. Equation 11 gives the best match. Equation 12 is used for the training.

$$o_i = (1 + e^{-\sum_{i=0}^{A} \omega_{ij} x_i})^{-1} \quad (11)$$

$$\Delta\omega_{ij}(t+1) = \epsilon\delta_j o_i + \alpha\Delta\omega_{ij}(t), \tag{12}$$

where o_i is as defined above, and

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

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). The activation 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.

7.1 Feature Extraction

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, giving a total of 3840 features. Therefore, abstraction is now used to further reduce the input data. Several approaches provide good abstraction (for example, see [9]). Our current approach uses only four samples from the 16×15 images. Both directions are used for Gaussian features, and the previously computed magnitude is used for motion. Instead of feature image values, locally normalized mean and variance in the vicinity of the Gaussian features are used (Equations 13 and 14). 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.

$$\mu_{i,j} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{G^{(i)}_{i+m,j+n}}{G^{(i)}{}_{Max}} \tag{13}$$

$$\sigma^2_{i,j} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(\frac{G^{(i)}_{i+m,j+n}}{G^{(i)}{}_{Max}} - \mu ij\right)^2 \tag{14}$$

7.2 Mapping Objects and Updating Memory

Once a representation is classified as new or identified, the world map is updated. Attentional features sufficient to detect changes are stored and the world map activation is set to zero to allow a shift of attention to another region. If the representation is new, supervised learning is invoked to insert the new feature set into long term memory, create new nodes in the hidden and output layers, and retrain the associative memory network.

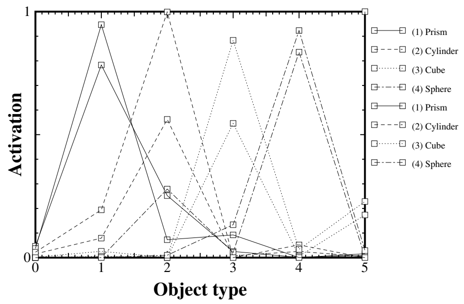

Figure 6: Activations for objects in the BP network. Figure shows only trials with lowest and highest activations for each winner object indicated on the upper right side.

8 Experiments and Results

Experiments involving attention, identification, and combined tasks were performed. Basically, many instances of objects of various types are placed on a table. On identification behavior, the robot learns the characteristics of all objects, inserts a representation for each in the associative memory, and updates the internal maps. Figure 6 shows simultaneous activations experimented in the BP network, using several instances of four types of objects placed on the table in controled poses. For each instance, upper line is highest activation (object on learned pose). The next line is lowest activation, yet allowing identification (poses were degraded with rotations up to 30 degrees when viewed from the stereo head).

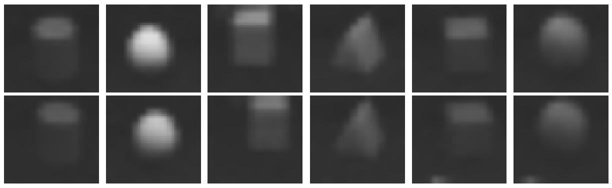

Figure 7: Pairs of images from finest resolution level of retina showing only new types of objects detected in the environment. Left to right: a red cylinder, a white golf ball, a natural wood cube, a red triangular prism, a blue cube, and a light green (dirty) tennis ball.

In attentional behavior, three experiments were tested. In the first experiment, we indicate the objects by touching them. The robot uses this motion cue followed by intensity cues to foveate on the objects. In the second experiment, there is no motion cue, so the robot relies solely on intensity cues. The attentional mechanism works well, foveating on each object in both the above tasks. In the third experiment, after all objects are mapped, we either move or remove an object. The robot updates the world map for the changed

Phase or process	Min(sec)	Max(sec)	μ(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
Salience 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: Processing time required in each phase.

regions using motion and intensity cues for movements in the field of view or by inspection behavior after all regions have been visited (see subsection 6.1). In all three experiments, the robot visits and maps all objects, discovering new representations and identifying existing ones. Figure 7 shows both cameras verged on the different types of objects detected in one of the experiments.

During some experiments, we also collected system performance data. Table 1 shows the time required for each of the processes involved in the inspection 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. Host computer computation times shown are for a 40 MHz Sun Sparc 10; that host has since been replaced with a 300 MHz Sun Ultra Sparc board that shares the Datacube bus for improved data transfer rates. Saccade speed can also be improved by adjusting PD controller gains in the stereo head PMAC, making saccade as fast as a human being.

9 Conclusion, Discussion and Future Work

Although this work uses only visual information, our system is more generally applicable, and could easily incorporate haptic, 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. A “controller oriented” approach to resource allocation would allow other process to be integrated into this architecture. One might ask why attention and identification are so important. Object categorization is necessary in almost all tasks that one can imagine; “what” is an important question in many tasks. The ability to focus attention is the basis for cognition. The two tasks 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 weight function. Given a set of tasks to be performed, the system can be rewarded for detection and identification of new objects important to the task. The result would be a more versatile function for directing attention, perhaps closer to a biological model.

References

[1] Gonçalves, L. M. G.; Oliveira, A. A. F.; and Grupen, R. A. 1998. A control architecture for multi-modal sensory integration. *Proceedings of the XI International Conference on Computer Graphics and Image Processing (SIBGRAPI’98)* 418–425.

[2] Gonçalves, L. M. G.; Oliveira, A. A. F.; and Grupen, R. A. 1999. Multi-modal stereognosis. *Proceedings of the III International Conference on Autonomous Agents (Agents ’99)*. 1999.

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

[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] Papoulis, A. 1991. *Probability, Random Variables, and Stochastic Processes*. MacGraw-Hill.

[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. 1997. Task-dependent learning of attention. *Neural Networks* 10(6):981–992.

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

[14] 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.