Video Coding Based on Pre-Attentive Processing

Çağatay Dikici[ab] and H. Işıl Bozma[a]

[a]Intelligent Systems Laboratory,
Electric and Electronics Engineering Department
Boğaziçi University, 34342, Bebek, Istanbul, Turkey
[b]INSA de Lyon, LIRIS, UMR 5205 CNRS, France

ABSTRACT

Attentive robots have visual systems with fovea-periphery distinction and saccadic motion capability. Previous work has shown that spatial and temporal redundancy thus present can be exploited in video coding / streaming algorithms and hence considerable bandwidth efficiency can be achieved. In this paper, we present a complete framework for real-time video coding with integrated pre-attentive processing and show that areas of greatest interest can be ensured of being processed in greater detail. The first step is pre-attention where the goal is to fixate on the most interesting parts of the incoming scene using a measure of saliency. The construction of the pre-attention function can vary depending on the set of visual primitives used. Here, we use Cartesian and Non-Cartesian filters and build a pre-attention function for a specific problem − namely video coding in applications such as robot-human tracking or video-conferencing. Using the most salient and distinguishing filter responses as the input, system parameters of a neural network are trained using resilient back-propagation algorithm with supervised learning. These parameters are then used in the construction of the pre-attentive function. Comparative results indicate that even with a very limited amount of learning, performance robustness can be achieved.

Keywords: Attentive Vision, biologically motivated vision, foveation, real-time video processing

1. INTRODUCTION

Attentive robots explore their surroundings in a loop of pre-attention and attention.[1] The aim of the pre-attention stage is to determine the next region of attention. This is achieved through the fovea-periphery mechanism.[2] The distribution of receptor cells on the retina is Gaussian-like with a small variance, resulting in a loss of resolution as we move away from the optical axis of the eye.[3] The fovea is the small region of highest acuity around the optical axis and the rest of the retina is called periphery. The robot foveates on the most salient region in its visual field. The saliency is measured by a pre-attention function as dictated by the current task. The location of the next fovea is then determined by the most salient regions in in its periphery. Following, saccades - very rapid jumps of optical axis - are used to bring this region into to fovea. In attentive processing, complex processing is applied only on the fovea. Hence, the generated video has varying resolution and redundancy. Previous work has shown that video streaming methods that exploit these properties and which can be naturally integrated to these robots can provide considerable bandwidth efficiency. In this paper, we show that furthermore, using pre-attentive processing, the robot can be ensured of keeping the most interesting objects in its fovea.

2. PRE-ATTENTION CRITERIA

The pre-attention measure is a function a that is dependent on the task and should therefore be learned using the set of visual primitives available. Once the task is selected, the construction of a consists of the following stages:

Corresponding author currently at [b] INSA de Lyon, Laboratoire d'InfoRmatique, en Images et Systèmes d'information (LIRIS), Bât. St-Exupéry, 69621 Villeurbanne Cedex, France. Email: cagatay.dikici@liris.cnrs.fr.

- **Selection of visual primitives:** Suppose there are M different primitives, and let the m^{th} visual primitive be denoted by Ω_m. The value of each visual primitive is obtained via an operator $f_m : I_c \rightarrow \Omega_m$ acting on each candidate fovea I_c.

- **Learning:** First, a sample set of foveal images containing both positive and negative examples is selected. For example, in face tracking, these are foveal images containing faces or no faces. These filters are then applied to this sample set. Based on statistical properties and the ability to differentiate positive foveas from negative, few of these filters are selected. Let us denote this as $M_t << M$. The pre-attention criteria is then defined as a function of the responses of these filters. We use two different approaches in its construction.

  ◇ **Biological filters:** Biological filters[4, 5] are used as visual primitives and the attention criteria is constructed as a function of the most salient few using either their weighted linear combination or neural-net based learning.

  ◇ **Haar filters:** Haar filters are used as visual primitives and attention criteria is constructed based on cascaded adaboost learning .[6]

- **Foveation:** The next fovea can be determined based on different strategies. In simple voting, the candidate fovea that maximizes our attention measure is then designated to be the next fovea as

$$I_f^{t+1} = arg \max_{\forall I_c \in C(I_f^t)} a(I_c) \tag{1}$$

  An alternative strategy is to allow multiple foveae. Instead of choosing the maximum response fovea, those foveae whose responses exceed a given threshold can all be selected as the next fovea set and the camera is made to look all them in a rapid sequential manner repeatedly. The temporal redundancy allows the current fovea to be retained for a certain period of time and pre-attentive processing is applied every 3 frames or so. The details of this processing are presented in .[7]

3. SAMPLE FOVEAS

The pre-attention criteria is constructed based on a sample foveal set – containing both target and non-target objects, faces in our case. VirtualDub has been modified so that the user can manually mark the desired foveal region of each frame in a selected video sequence and the filter responses are automatically generated and stored. The user can specify the following parameters:

- **Frame index:** The frame number in the video sequence.

- **Foveal size:** The target fovea can be selected in any length and width.

- **Foveal center:** The coordinates of the foveal center are specified as a function of the position of the left-bottom corner of the foveal region.

- **Task keyword:** A task keyword is used index all the files containing the calculated filter responses. For example, this phrase may be “face”,“non-face”. For visuality and later use, the candidate foveae that are generated during the learning phase are also stored as bitmap images.

- **Scaling factor:** In order to search the desired object in multiple scales, a gaussian pyramid of the frame is constructed. The scaling factor is the decimation factor of the Gaussian pyramids.

Figure 1 shows positive and negative sample foveas respectively.

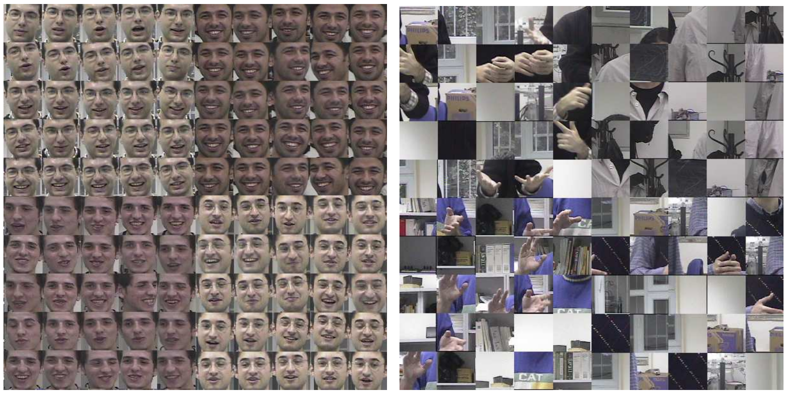

Figure 1. Left: Positive training samples including 100 human face fovea candidates; Right: Negative training samples including 100 non-face fovea candidates.

4. BIOLOGICALLY MOTIVATED FILTERS & NEURAL NET BASED LEARNING

APES uses a biologically motivated set of visual primitives in its attention and cognition stages.[4, 5] These filters consist of Cartesian and Non-Cartesian filters as described originally in.[8–11] They mimick the response characteristics of area V4 cells of macaque monkeys – which have been determined to selective to both Cartesian (planar texture surfaces) and Non-Cartesian (textured spheres and saddles) stimuli.[8] In order to be consistent and integrated with the rest of the system, we also use this set as our basis. This set consists of fifty filters – six Cartesian orientations, concentric and radial filters and two hyperbolic filters with five different frequencies as shown in Figure 2. The mathematical formulas of these filters are presented in Appendix A and the interested reader is referred to[4, 5] for further details.

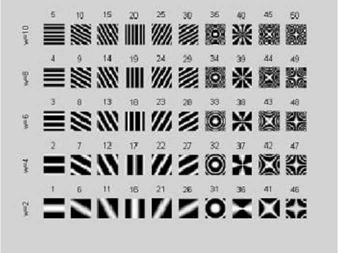

Figure 2. Cartesian and Non-Cartesian filters where the $[-1, +1]$ range is mapped to grayscale.

4.1. Filter Responses

Although foveal size may vary, they are resized to 40×40 before applying the biological filters with the aim of introducing normalization for the learning stage. In order to determine the most salient filters, first all the filter responses are computed for all the sample set and stored using the "Training Software". Average and standard deviation of each filter for both positive and negative samples are then calculated. The filters having well-separated responses for positive and negative samples and small variations are designated as salient filters. In our experiments, these turn out to be filters as shown in Table 1.

4.2. Choosing the Filter Subset

The filter responses o_c^t are calculated and stored by using the training samples as shown in Figures 1. For each filter, the mean and standard deviation of its responses for both positive and negative sample sets are computed respectively. Following, based on visual inspection, those filters having well separated mean values are selected. Let us denote this number by $M_t < M$. In our case, $M_t = 7$ * Figure 3 identifies these filters. Table 1 presents

*Admittedly, a more rigorous approach can be utilized to determine M_t.

the mean and standard deviation of the responses of these filters. The filter response set is then constructed as consisting of these filters $\Omega' = [\Omega'_1\ \Omega'_7\ \Omega'_9\ \Omega'_{12}\ \Omega'_{23}\ \Omega'_{27}\ \Omega'_{46}]$. Here Ω'_i is the normalized amount of deviation that the i^{th} filter response from the mean face response and is computed as $\Omega'_i = \frac{\Omega_i - \mu_{\Omega_i(face)}}{\sigma_{\Omega_i(face)}}$.

Table 1. Mean and standard deviation of seven filter responses calculated from face and non-face training samples

Filter index	μ Face resp.	σ Face resp.	μ Non-Face resp.	σ Non-Face resp.
1	1317.94	297.85	859.38	677.00
7	489.88	129.42	274.82	200.80
9	160.65	35.76	98.76	66.38
12	433.42	118.83	260.04	181.17
23	268.93	63.17	154.44	93.23
27	508.37	118.84	291.55	196.26
46	2272.64	630.94	1450.64	943.98

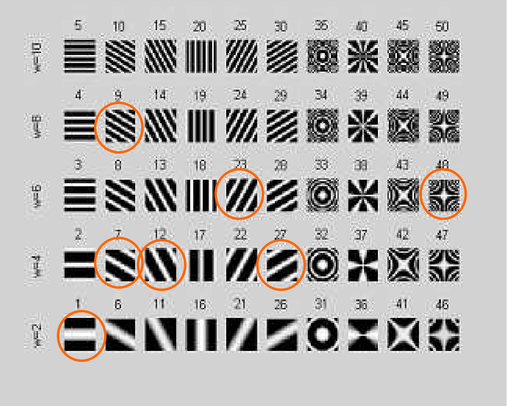

Figure 3. Seven selected visual primitives within 50 Cartesian and Non-Cartesian filter set

4.3. Construction of Pre-Attention Function

The pre-attention function a is constructed using a neural network.[12–14] In particular, a simple feed-forward neural network having M_t inputs as shown in Figure 4 is constructed. As discussed earlier, the network has $M_t = 7$ inputs. The outputs of the first layer are fully connected to an intermediate layer consisting of H hidden units with hyperbolic tangent transfer function. In our case, the hidden layer consists of $H = 15$ perceptrons. Hence, the output of each neuron within the hidden layer is: $a_i = \tanh(\sum_{j=0:M_t-1} Wij \times fr_i + b_i)$. The output consists of one perceptron whose output is as follows:

$$a = \tanh(\sum_{j=0:H-1} W'_j * a_j + b_o) \tag{2}$$

Note that this value ranges in $[-1, +1]$.

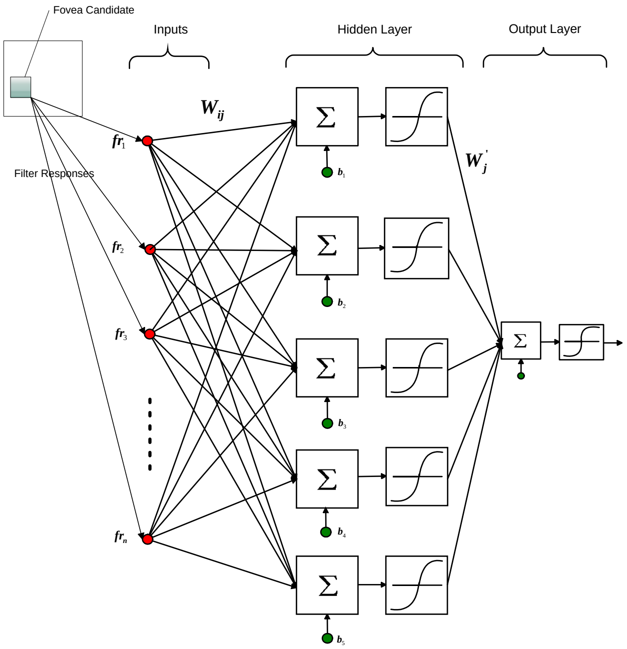

Figure 4. Neural network structure of the Cartesian and Non-Cartesian visual primitives

In our case, the input weights, biases, and layer weights of the system:

$$b = \begin{bmatrix} 21.85 \\ -1.81 \\ -5.77 \\ -13.53 \\ 1.49 \\ -2.91 \\ 1.36 \\ 1.16 \\ 1.82 \\ -0.17 \\ 2.53 \\ 7.43 \\ 5.63 \\ 0.46 \\ 5.46 \end{bmatrix} \qquad b_o = -0.034 \quad W^{'} = \begin{bmatrix} 0.45 \\ -12.97 \\ -13.25 \\ -5.08 \\ 5.87 \\ -6.73 \\ 4.36 \\ -9.33 \\ 6.83 \\ 8.39 \\ -13.58 \\ 6.21 \\ 6.93 \end{bmatrix} \tag{3}$$

System parameters are trained using resilient back-propagation algorithm with supervised learning, which means the weights are adjusted such that given the desired output of the inputs, the system minimizes the output error. The network converges to 10^{-3} error rate after training for 702 epochs. W corresponds to the input weight matrix of the neural net. W_{ij} indicates the weight of j^{th} input and the i^{th} perceptron of the hidden layer.

$$W = \begin{bmatrix} -4.77 & 13.78 & -27.19 & 25.32 & -4.85 & -0.30 & -65.66 \\ -2.15 & 0.53 & 0.21 & -0.12 & 1.60 & 0.62 & -0.05 \\ 1.28 & 0.82 & -0.26 & 0.96 & 0.35 & 0.67 & -0.83 \\ 4.78 & 30.71 & 0.31 & -5.69 & -2.33 & -3.07 & 14.91 \\ -1.21 & -0.84 & -0.33 & 4.35 & -0.14 & -0.76 & 0.53 \\ -0.77 & 0.57 & 1.69 & 0.80 & -0.47 & 0.97 & 0.80 \\ 0.36 & -0.29 & -0.26 & -0.76 & 0.91 & -0.59 & -0.65 \\ 1.96 & -2.32 & -0.97 & -0.67 & 0.16 & 3.92 & -6.50 \\ -0.48 & -0.09 & -0.50 & -1.21 & 0.98 & 1.25 & -1.11 \\ -0.07 & 0.82 & 0.22 & -0.11 & -0.74 & 0.57 & -0.29 \\ -0.03 & -0.69 & -0.06 & -0.61 & 0.26 & -0.79 & 0.18 \\ -4.04 & -0.68 & -5.02 & 4.33 & -0.82 & 1.78 & -11.72 \\ -5.89 & 1.91 & 0.48 & -5.30 & 0.45 & -5.10 & 0.77 \\ 0.45 & -1.25 & -16.15 & 7.43 & 0.22 & 4.83 & -0.14 \\ 0.78 & -0.36 & 0.99 & -1.26 & -2.15 & 2.03 & -2.18 \end{bmatrix} \tag{4}$$

5. HAAR FILTERS & CASCADED ADABOOST

The pre-attention criterion a is also constructed using Cascaded Adaboost[15, 17] where Haar filters are used as visual primitives. The approach is based on the creation of several weak hypotheses followed by their integration in order to end up with a final hypothesis – a process known as boosting. This is achieved by also extracting several features from integral of the image itself. These features are used during learning. For each feature, a boosted classifier is trained with a target hitrate and false alarm rate. Afterwards, these boosted classifiers are cascaded within the network such that the strong features are located at early stages of the classifier. In this study, we use the cascade of the classifiers reported in[17] using about 10000 sample face images. The i^{th} classifier is defined as:

$$h_i(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_{it} h_{it}(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_{it} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Here, $h_{it}(x)$ is the t^{th} weak learner of the i^{th} classifier and α_{it} is the associated weight. Hence a tree based decision structure is constructed. If the classifier rejects the input in early stages, then the system automatically rejects the candidate without applying the rest of the cascades.

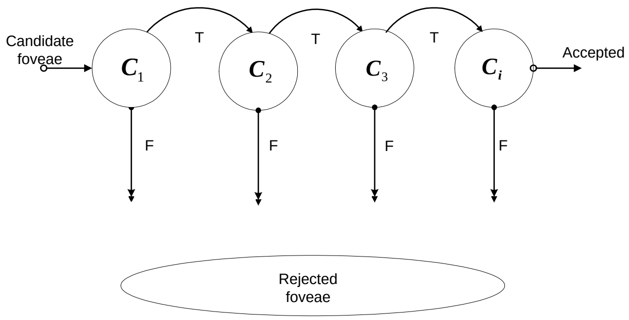

Figure 5. Schematic description of the detection of an Adaboost Cascade

6. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed approach, an extensive statistical comparative study has been conducted. The application is human face tracking. The goal is to ensure that the fovea always remains on the person's face. In this study, APES Video Database[18] is used. The database consists of videos of a person – talking and making gestures at three different distances (long, intermediate, short distances). Each video is recorded in 384 × 288 resolution, 20 second long, $25frames/sec$ RGB video in uncompressed AVI format. Furthermore, three different noisy versions of the input videos are created by using an additive gaussian noise to all RGB channel with $N(0, \sigma)$ zero mean, variances 5, 10, and 20 respectively.

Table 2. Comparison of the proposed methods with respect to the cascaded adaboost method in case of additive gaussian noise with 0 mean and variances $5, 10, 20$.

	Without Noise		$\mu = 0, \sigma = 5$		$\mu = 0, \sigma = 10$		$\mu = 0, \sigma = 20$	
	Detection	NFA	Detection	NFA	Detection	NFA	Detection	NFA
Simple voting	78%	19/500	77%	20/500	77%	21	78%	26/500
Multiple foveae	79%	20/500	78%	23/500	77%	23	76%	24/500
Adaboost	92%	3/500	90%	8/500	89%	9/500	88%	9/500

Three different approaches are used in this task: Simple voting, multiple foveae and adaboost algorithm. Performance is evaluated in terms of two measures: detection performance and normalized false alarm. The detection performance is the percentage of the frames that the face candidate is successfully detected. Normalized false alarm (NFA) is defined as the number of foveae that does not contain a human face within its region normalized by the number of frames per video – recalling that each video consists of 500 frames. The outputs of the experiments can be seen in Table 2. The performance of the adaboost is superior to the proposed methods with respect to detection rate and false alarms. Let us note that nevertheless, the performance of the proposed methods are quite good – considering that the pre-attention function is based on learning from a very limited (based on about 150 samples) as compared to that of Adaboost (based on about 10000 samples). Increasing the learning set would certainly increase performance. Furthermore, the proposed method seems to be more robust against noise. Performance remains roughly about the same, while that of the adaboost deteriorates to some extent. In the voting method, we choose the most distinctive fovea candidate as the next fovea thus only strongest one is selected, while in the multiple foveae approach, the robot fixates on a sequence of foveae. Finally if we compare the computational complexity of adaboost and the proposed method, while the former needs 13 × 14 multiplication and 13 × 14 addition for 24 × 24 window, the latter has 15 × 6 multiplication and 16 addition for 40 × 40 window. Hence the proposed method is considerably less complex than the adaboost method.

7. CONCLUSION

In this paper, we propose a complete framework for real-time video coding with integrated pre-attentive processing. A set of Cartesian and non-Cartesian filters are used to construct a pre-attention function using neural networks. Two strategies are used. In simple voting, the maximizing candidate fovea is selected to be the next fovea. In multiple foveae, all foveae exceeding a given threshold are selected. High resolution spatial processing is then applied only around foveal regions and thus considerable bandwidth efficiency can be achieved as was shown in our earlier work. Here, we show that using a properly constructed pre-attention function, areas of greater interest can be ensured of being covered by the fovea. In particular, the problem of human face tracking is considered and comparative performance results are presented. These results indicate that even with learning based on a fairly limited set such as 200 frames, acceptable performance can be achieved. Interestingly, the system seems to be more robust against additive Gaussian noise where the performance statistics remains roughly the same. Finally, it has considerably less computational complexity. Hence, the approach seems to be promising for real-time applications such robot video-streaming and video-conferencing.

APPENDIX A. VISUAL PRIMITIVES

In this section, the construction of visual primitives is reviewed briefly. The reader is referred to[4] for all the details.

A.1. Cartesian Filters

The Cartesian filters $f_{cw} : SO(1) \times SO(1) \to [-1, 1]$ can be formulated in Equation 6. Please note that $SO(1) =^{\triangle} [-\frac{\pi}{2}, \frac{\pi}{2}]$.

$$f_{cw}(x, y) = \cos(\omega \times (\alpha \times x + \beta \times y)) \tag{6}$$

In Equation 6, $\alpha = \sin((c-1) \times \Pi/\Lambda)$, and $\beta = \cos((c-1) \times \Pi/\Lambda)$. The parameters c, and w are orientation, and frequency of the sinusoid respectively. By choosing $c = 1, ..., \Lambda$, and $w \in \{k \times \delta w \ |k = 1, ..., K\}$, the Cartesian filters look similar to those in.[8] In this work, we choose $\Lambda = 6$, $\delta w = 2$, and $K = 5$.

A.2. Non-Cartesian Filters

Non-Cartesian filters[8] are also a function of sinusoids, but the arguments of the sinusoid have nonlinear component. The Non-Cartesian filters can be grouped as concentric, polar, and hyperbolic filters. The concentric filters $f_{7w} : SO(1) \times SO(1) \to [-1, 1]$ can be modelled as $f_{7w}(x, y) = \cos(\omega \times (x^2 + y^2))$. By varying $w \in \{k \times \delta w \ |k = 1, ..., K\}$, we generate circular filters with different frequencies.

The polar filters $f_{8w} : SO(1) \times SO(1) \to [-1, 1]$ is defined as $f_{8w}(x, y) = \cos(\omega \times \arctan(y/x))$. Again by varying the frequency parameter $w \in \{k \times \delta w \ |k = 1, ..., K\}$, a set of circular filters are modeled.

In the final step, we generate two different forms of hyperbolic filters. First of the hyperbolic filters $f_{9w}(x, y) = \cos(\omega \times \arctan(y/x))$ is $f_{9w}(x, y) = \cos(\omega \times (y^2 + x^2))$. The second set of hyperbolic filters $f_{10w}(x, y) = \cos(\omega \times \arctan(y/x))$ is obtained by rotating the $f_{9w}(x, y)$ function around the origin by θ degrees, which yields $f_{10w}(x, y) = f_{9w}(\cos(\theta) \times x + \sin(\theta) \times y, -\sin(\theta) \times x + \cos(\theta) \times y)$. By choosing $\theta = \pi/4$ and $w \in \{k \times \delta w \ |k = 1, ..., K\}$, we obtain another set of hyperbolic filters.

A.3. Filter Responses

The response of each filter is computed based on $2 - D$ convolution of the candidate fovea I_c^t and the filter kernel $f_{cw} \in F$. It has been determined experimentally that the best choice of Ω_m for Cartesian filters $f_{cw} \in F$, $c = 1, .., 6$ can be calculated by discarding the mean intensity level of the filter outputs.[4] Here, we use the standard deviation of the convolution and define $\Omega_m = stdev(f_{cw} \star I_c^t)$. Similarly, the experimental results indicate that for non-Cartesian filters, the best choice of Ω_m is found as the greatest magnitude of the response and thus $\Omega_m = max(f_{cw \star I_c^t})$, where $m = 31, ..., 50$.

ACKNOWLEDGMENTS

This work has been supported by Bogazici University Scientific Research Projects Grant #BAP02S106. We gratefully acknowledge Doğan Can for his meticulous collection of the APES video database and helping with the experiments.

REFERENCES

1. Ballard, D. H. and C. M. Brown, "Principles of Animate Vision", *CVIP:Image Understanding*, Vol. 56, July 1992.
2. Kowler, E., (editor), "Eye Movements and Their Role in Visual and Cognitive Processes", *Elsevier*, 1990.
3. Gouras, P. and C. H. Bailey, "The Retina and Phototransduction", *Principles of Neural Science Elsevier*, 1986.
4. Bozma, H. I., G. Cakiroglu and C. Soyer, "Biologically Inspired Cartesian and Non-Cartesian Filters for Attentional Sequences", *Pattern Recognition Letters (SCI)*, Vol. 24/9-10, pp. 1261-1274, June 2003.
5. Bozma, H. I. and G. Cakiroglu, "Dynamic Integration for Scene Recognition Using Complex Attentional Sequences", *Proceedings of Intelligent Autonomous Systems*, Amsterdam, 2004.
6. Papageorgiou, C., M. Oren, and T. Poggio, "A general framework for Object Detection", *In International Conference on Computer Vision*, 1998.
7. Dikici, C., "Fovea Based Coding For Video Streaming", M.S. Thesis, *Dep. of EE Bogazici University*, Jun. 2004.
8. Gallant, J. L., H. C. Nothdurft and D. C. Van Essen,"Two-Dimensional and Three Dimensional Texture Processing in Visual Cortex of the Macaque Monkey", *In Early Vision and Beyond*, T.V. Papathomas, C. Chubb, A. Gorea and E. Kowler, (editors), MIT Press, pp. 89-98, 1995.
9. He, Z. J. and K. Nakayama, "Attention to Surfaces: Beyond a Cartesian Understanding of Focal Attention", *In Early Vision and Beyond*, T.V. Papathomas, C. Chubb, A. Gorea and E. Kowler, (editors), pp. 69-77, 1995.

10. Sagi, D., “The Psychophysics of Texture Segmentation”, *In Early Vision and Beyond*, T.V. Papathomas, C. Chubb, A. Gorea and E. Kowler, (editors), pp. 69-77, 1995.
11. Connor, C. E., J. L. Gallant, J. W. Lewis, S. Rakshit and D. C. Van Essen, “Neural Responses to Polar, Hyperbolic and Cartesian Gratings in Area V4 of the Macaque Monkey”, *Journal of Neurophysiology*, Vol. 76, No. 4, pp. 2718-2739, 1996.
12. Fausett, L., “Fundamentals of Neural Networks”, *Prentice-Hall*, 1994.
13. Haykin, S., “Neural Networks: A Comprehensive Foundation”, *Prentice-Hall*, 1994.
14. Lippman, R. P., “An Introduction to Computing with Neural Nets”, *IEEE ASSP Magazine*, pp. 4-22, 1987.
15. Jones, M. J. and P. Viola, “Rapid Object Detection Using a Boosted Cascade of Simple Features”, *IEEE CVPR*, 2001.
16. Mohan, A., C. Papageorgiou, and T. Poggio, “Example-based object detection in images by components”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 4, pp. 349 -361, April 2001.
17. Lienhart, R. and J. Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection”, *ICIP*, 2002.
18. Dikici, C. and H. I. Bozma, “Fovea Based Coding For Video Streaming”, *ICIAR*, 2004.