Offline Learning of Top-down Object based Attention Control

Ali Borji, Majid Nili Ahmadabadi, Babak Nadjar Araabi

Offline Learning of Top-down Object based Attention Control

Ali Borji[1], Majid Nili Ahmadabadi[1,2], Babak Nadjar Araabi[1,2]

[1]School of Cognitive Sciences, Institute for Studies in Theoretical Physics and Mathematics,
Tehran, IRAN
[2]Dept. of Electrical and Computer Eng, University of Tehran, Tehran, IRAN
{borji, mnili, araabi}@ipm.ir

Abstract. Like humans and primates, artificial creatures like robots are limited in terms of allocation of their resources to huge sensory and perceptual information. Serial processing mechanisms are believed to have the major role on such limitation. Thus attention control is regarded as the same solution as humans in this regard but of course with different attention control mechanisms than those of parallel brain. In this paper, an algorithm is proposed for offline learning of top-down object based visual attention control by biasing the basic saliency based model of visual attention. Each feature channel and resolution of the basic saliency map is associated with a weight and a processing cost. Then a global optimization algorithm is used to find a set of parameters for detecting specific objects. Proposed method is evaluated over synthetic search arrays in pop-out and conjunction search tasks and also for traffic sign recognition on cluttered scenes.

Keywords: Top-down attention, object based attention, saliency

1 Introduction

Both machine and biological vision systems have to process enormous amount of visual information they receive at any given time. Therefore attentional selection provides an efficient solution to this problem by proposing a small set of scene regions worthy further analysis to higher-level cognitive processes like scene interpretation, object recognition, decision making, etc. In this regard visual attention acts as a front end to a more complex vision system.

A model for bottom-up visual attention known as saliency-based model is proposed in [1] by Itti et al., which is an extension and implementation of an earlier model by Koch & Ullman [2]. Bottom-up model in its original form is solely data-driven and simply selects some spatial locations without using any feedback mechanism or top-down gains. In [3], this model is biased toward specific objects by strengthening the saliency of a target object over a distracting background. To this end, they have maximized the signal-to-noise ratio (SNR) between the target and background. In [4], authors have developed a method to account spatial relationships among objects in a Bayesian framework. Their model modulates the basic saliency model by a spatial saliency map that is learnt from probabilities of objects to appear in a scene. That way, for example when looking for a person in a beach scene, areas around the sea are more probable to contain a person (and thus are more salient) than the sky regions. However it is very hard to find an exact formulation for the visual search and object detection, since many parameters are involved. Therefore using an offline approach which could implicitly account such hidden parameters seems to be an appropriate approach. In this work, we use a global optimization algorithm for tuning parameters of the saliency model for object detection. Since main goal of attention is to reduce computational complexity, we have also put costs on features and resolutions of the images in order to find solutions with both low computation and high discrimination power.

2 Proposed Method

To furnish the saliency model for our purpose, i.e. biasing it for object detection, a dyadic Gaussian pyramid (with six scales) is built in each feature channel of the image by successively filtering the image with a Gaussian low-pass filter and then subsampling it. Then a surround inhibition operation is applied over each scale in the pyramid to enhance those parts of the image which are different from their surroundings. To implement surround inhibition, saliency model reduces finer and coarser scales from each other. Because we would like to also weight different resolutions in the map, we dissociate scales and then apply the surround inhibition over each scale separately. To accomplish surround inhibition, a non-linear filter which compares the similarity of a pixel with the average of its surrounding window and then inhibits the center pixel by an exponential weight was designed. Equation 1 shows this filter.

$$\mathrm{I}'(\mathrm{x,y}) = \mathrm{e}^{\gamma \mathrm{m}} - 1\ ,\ \ \mathrm{m} = \frac{|\mathrm{I(x,y)} - \mathrm{Avg(Surround(I(x,y)))}\,|}{\mathrm{I(x,y)}} \qquad \textbf{(1)}$$

where surround is a spatial n×n mask around pixel I(x,y) excluding the point itself. $I'(x,y)$ is the new value of the pixel. Output image was later normalized to its initial range. Figure 1 demonstrates surround inhibition operation over some test images.

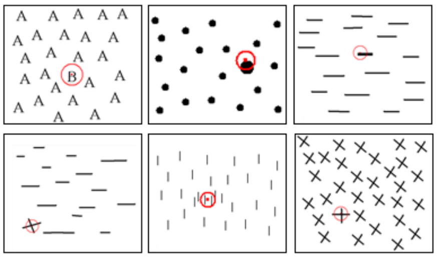

Fig. 1. Surround inhibition using equation one. Window sizes from top-left to bottom-right are: 15×15, 7×7, 15×15, 5×5, 9×9, 11×11. Red circles illustrate the salient locations (γ=.04).

The implementation of the Gaussian pyramid is illustrated in figure 2. Surround modulated images in each scale are upsampled to a predetermined scale and after normalization and weighting are summed to form feature maps.

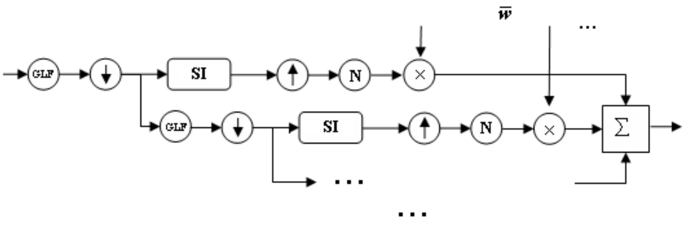

Fig. 2. Gaussian pyramid with surround inhibition. w shows weights of different scales of the pyramid. SI is the surround inhibition operation as in equation 1.

The input image is decomposed into three feature channels: Intensity (I), color (C) and orientation (O). Color channels are calculated as follows. If r, g and b are the red, green and blue dimensions in RGB color space normalized by the image intensity I, then R = r - (g+b)/2, G = g - (r + b)/2, B = b - (r +g)/2, and Y = r + g - 2 (|r - g| + b) (negative values are set to zero). Local orientations, O_θ are obtained by applying Gabor filters to the images in the intensity pyramid I.

$$\mathrm{F_{I,S}} = \mathrm{N}\ (\ \mathrm{SI}\ (\mathrm{I_s})\ ),\ \mathrm{F_{RG,S}} = \mathrm{N}\ (\ \mathrm{SI}\ (\mathrm{R_s} - \mathrm{G_s})\ ), \\ \mathrm{F_{BY,S}} = \mathrm{N}\ (\ \mathrm{SI}\ (\mathrm{B_s} - \mathrm{Y_s})\ ),\ \mathrm{F_{\theta,S}} = \mathrm{N}\ \left(\ \mathrm{SI}\ \left(\mathrm{O_{\theta,s}}\right)\ \right) \tag{2}$$

$F_{x,y}$ in above equation is the map at the scale s of channel x. SI is the surround inhibition operation. N(.) is the iterative normalization operator as defined in [5]. These feature maps are summed over scales and sums are normalized again:

$$\overline{\mathrm{F_l}} = \mathrm{N}(\ \textstyle\sum_{\mathrm{s}} \omega_{\mathrm{S}}.\mathrm{F_{l,S}}\ ), \text{with } \mathrm{l} \in \mathrm{L_I} \cup \mathrm{L_C}\ \cup\ \mathrm{L_O} \\ \mathrm{L_I} = \{\mathrm{I}\}, \mathrm{L_C} = \{\mathrm{RG}, \mathrm{BY}\}, \mathrm{L_O} = \{0^0, 45^0, 90^0, 135^0\} \tag{3}$$

In each feature channel, feature dimensions contribute to the conspicuity maps by weighting and normalizing once again, as shown in equation 4.

$$\mathrm{C_p} = \mathrm{N}\left(\sum_{\mathrm{l} \in \mathrm{L_p}} \omega_{\mathrm{p}}.\overline{\mathrm{F_l}}\right), \qquad \mathrm{p} \in \{\mathrm{I}, \mathrm{C}, \mathrm{O}\} \tag{4}$$

All conspicuity maps are weighted and combined at this stage into one saliency map:

$$\mathrm{SM} = \sum_{\mathrm{k}} \omega_{\mathrm{k}}.\mathrm{C_k}\,, \qquad \mathrm{k} \in \{\mathrm{I}, \mathrm{C}, \mathrm{O}\} \tag{5}$$

The locations in the saliency map compete for the highest saliency value by means of a winner-take-all (WTA) network of integrate-and-fire neurons. The winning location (x_n, y_n) of this process is attended to. Then this point is inhibited for a while (a phenomenon called inhibition of return) to let the attention shift to other locations. After a time constant it will have the chance to become active again and catch the attention focus. The whole process is summarized in figure 3.

In order to reduce the computational complexity of the model, in addition to a weight, a cost is also associated to each feature channel and image resolution. Weights and costs are represented by vectors $\overline{\omega} = (\omega_1, \omega_2, \ldots \omega_n)$ and $\overline{\mathrm{C}}\ (c_1, c_2, \ldots c_n)$ respectively.

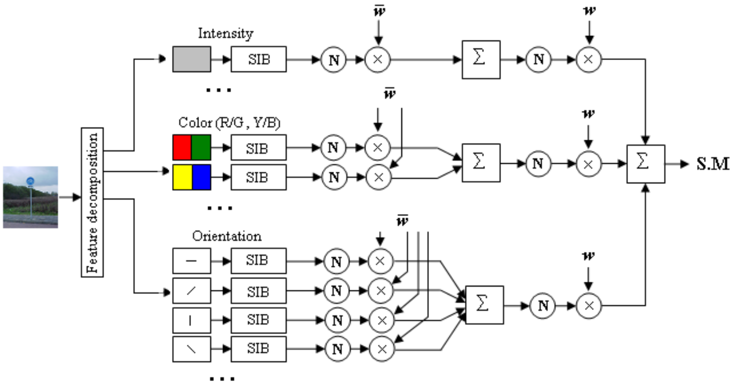

Fig. 3. Revised saliency model of visual attention. SIB is the Gaussian pyramid in figure 2.

Goal is to find a set of weights which determines the contributions of features (intensity, color, orientation and image resolutions) to detect a specific object in a set of images while taking into account their costs. Therefore the optimum weight vector must satisfy two conditions: 1) it must enable the saliency model to detect an object of interest (maximum detection rate), 2) its associated feature vector must have the minimum cost. Let $\overline{\text{Img}} = \{\text{Img}_1, \text{Img}_2, \dots \text{Img}_m\}$ be m training images with target object locations tagged. We use comprehensive learning particle swarm optimization (CLPSO) [6] to search for weight vectors satisfying above properties. An example fitness function satisfying the above two conditions is shown in equation 6.

$$\text{Fitness}(\overline{\omega}) = \frac{1}{\text{mpn}}\left(\sum_{i=1}^{m}\sum_{j=1}^{p}\text{dist}_j(\text{ Saliency}(\text{Img}_i, \overline{\omega}), t_i\text{ })\right)\sum_{k=1}^{n}u(\omega_k)c_k \qquad \textbf{(6)}$$

In above formula, n is the dimensionality of cost or weight vector, *dist*$_j$ () is the Euclidian distance between j-th salient point generated by the saliency model and the target location t_i in i-th image. u(x) is the step function and is one when a feature channel or resolution is used. Saliency is a function which takes an image and a weight vector as input and outputs a vector of p salient points.

3 Experiments and Results

We evaluated our biased saliency method for traffic sign detection over a dataset of 30 signs. They were bike, pedestrian and crossing signs. We also applied it for saliency detection over synthetic search arrays (pop out and conjunction search tasks). Twenty samples of each signs were used to derive a weight vector. Figure four illustrates achieved weight vectors after CLPSO convergence for a sample run. Table 1 shows the average detection rates using two first detection hits over remaining 10 test images. A detection hit was counted if a salient point was generated in a vicinity of 30 pixels around the target point (center of object). Feature costs were defined based on relative computational cost of model features. Figure 5 shows some example natural scenes with detected traffic signs. Figure 6 shows the average spatial map for each traffic sign. This map could be used as starting point for searching a specifc sign to speed up the detection process.

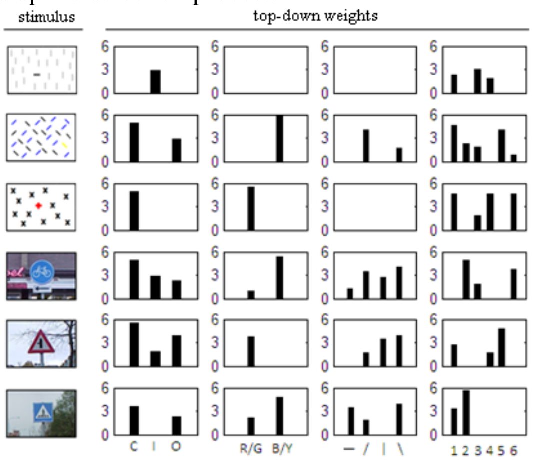

Fig. 4. Best weights for saliency model after CLPSO convergence.

Table 1. Average detection rates of biased and basic saleincy model and average cost of the final best weight vector of CLPSO over five runs. Costs = [3 1 4, 3 3, 4 4 4 4, 6 5 4 3 2 1].

Stimuli	**Detection rate (%)**		**Bottom-up model**	**Average cost**
	1st hit	1st & 2st hit	(1st & 2st hit) (%)	
Synthetic 1	100	100	100	4.6
Synthetic 2	100	100	100	14.5
Synthetic 3	100	100	100	8.2
Bike	76.67	86.67	76.67	20.5
Crossing	83.33	90	83.33	28.3
Pedestrian	86.67	90	90	19.8

4 Conclusions and future works

Our proposed model in this paper is constructed over the basic saliency based model of visual attention and biases it for detecting objects of interest. Weights and costs were associated with features and resolutions of the model and then an optimization algorithm was used to select features (sensors) which had lowest cost and high detection rate. Proposed method was used to detect traffic signs in cluttered road scenes. In order to reduce the cost, algorithm bypassed those channels which were not important in detecting the objects. Since the parameters of the saliency model are continuous-valued, online learning of those parameters degrades performance of any algorithm for learning top-down attention control based on the saliency model. Therefore an offline approach seems reasonable. Learned parameters then could be used online when a complex task demanding perceptual and physical actions has to be done. Therefore, future works in this regard should focus on how to use this knowledge in interactive tasks.

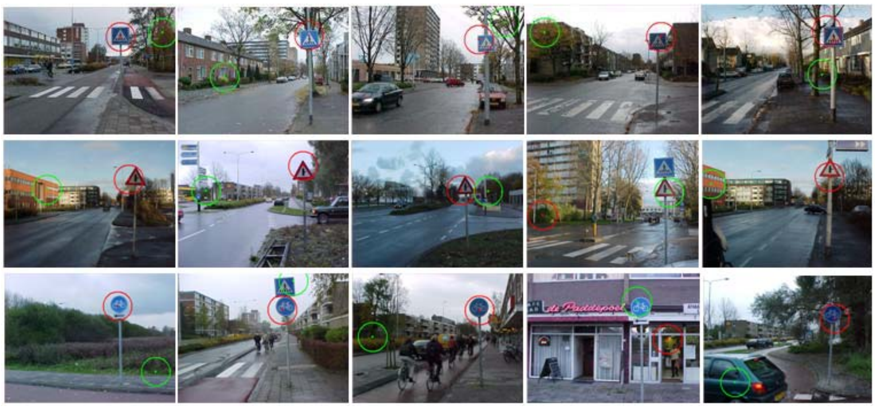

Fig. 5. Sign detection over 5 sample test images. Top row: pedestrian, middle: crossing, bottom: bike. Red circle is the first attended location and yellow one is the second.

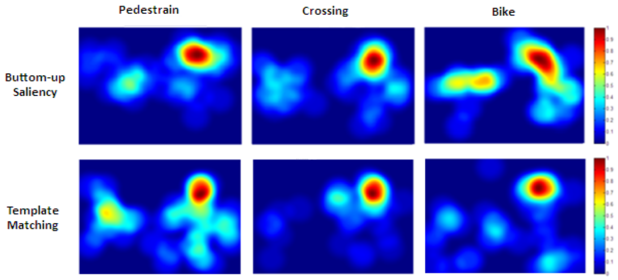

Fig 6. Spatial map showing the possible locations of traffic signs in the images of the dataset.

References

1. L. Itti, C. Koch, E. Niebur, A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11) 1254-1259, 1998.
2. C. Koch and S. Ullman, Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry, Human Neurobiology, 4:219–227, 1985.
3. V. Navalpakkam, L. Itti, An Integrated Model of Top-down and Bottom-up Attention for Optimal Object Detection, In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2049-2056, 2006.
4. Torralba, A., Oliva, A., Castelhano, M., & Henderson, J.M. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. Psychological Review, 113, 766-786. 2006.
5. L. Itti, C. Koch, Feature combination strategies for saliency-based visual attention systems, J. Electron. Imaging 10 (1) 161–169, 2001.
6. J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE trans. evolutionary computation, 9(3) 2006.