Using visual attention in a Nao humanoid to face the RoboCup any-ball challenge

Juan F. García #1, Francisco J. Rodríguez #2 Francisco Martín *3 Vicente Matellán #4

#Grupo de Robótica - Escuela de Ingenierías Industrial e Informática
Universidad de León. 24071 León (Spain)
1jfgars@unileon.es
2frodl@unileon.es
4vicente.matellan@unileon.es

*Grupo de Robótica - Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad Rey Juan Carlos. Fuenlabrada, Madrid (Spain)
2fmartin@gsyc.es

***Abstract*— Visual attention is a natural tool which allows animals to locate relevant objects or areas in a given scene, discarding the rest of elements present and thus reducing the amount of information to deal with. In this paper we present the design an implementation of a visual attention mechanism based on a saliency map and its implementation in the Nao humanoid. This control mechanism is applied to solve one of the challenges proposed in the RoboCup competition named ”any-ball”. The results obtained are analysed and future works derived from that analysis are presented.**

I. Introduction

Vision and control systems in Robotics are usually implemented in an impulse-analysis-response fashion. Given a visual impulse, the analysis subsystem generates a “world model” which is then used by the response module to generate an action. In this case, vision is just a step previous to planning. However, attention can be used to further relate these two systems: control system can establish the kind of objects that should be looked for (top-down, control modulates attention) and attended locations restrict what can be done in that moment (bottom-up, attention modulates control) [1].

The latest attention models are mostly bioinspired and try to reproduce the way primates' and humans' attention works [2], [3], [4]. Color contrast, intensity difference, orientation and motion are just some of the key elements considered by these models.

In this paper we present a bioinspired attention model mainly based on Itti et al. research [3], [5], [6].

But our model does not use orientation maps to build up the final saliency map, since information they provide is not necessary for this environment.

We will demonstrate the use of this attention system in one of the proposed challenges proposed in the RoboCup Standard Platform League (SPL): the “any ball” challenge. In this league all teams use the same hardware platform, the Nao robot (see figure1).

Nao robot has two 30 fps video cameras located in the forehead and in the mouth, each one with a maximum resolution of 640x480, but they can neither be used simultaneously nor are capable of stereo vision since their field of view is not overlapped. All control is made on-board using a x86 AMD Geode chip at 500 MHz, 256 MB of SDRAM memory and a standard 1 Gb in flash memory that can be upgraded. Given that all teams use this same platform in the RoboCup SPL competition, software optimisation becomes critical. More than 75% of processor time is used for visual related tasks, which means every little improvement in this area will greatly benefit the whole system.

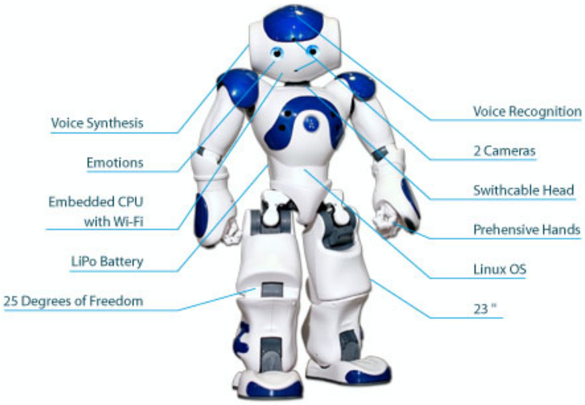

Fig. 1. Nao robot (figure copyrighted by Aldebaran Robotics)

The rest of the paper is organised as follows. In the second section, some of the most notable attention models are enumerated. In the third section, the attention model is explained, both the principles and the software structure are detailed. In the forth section, the attention algorithms used are described. In the fifth section, experiments used for the model validation are summarised. Finally, in the last section, the results obtained are discussed and also the future works envisioned are enumerated.

II. Attention models

Animals, and humans specifically, can change their focus of attention either by moving their fixation point across the visual scene or by focusing on a given area of the current visual field. The former is known as "overt attention" and the latter, which is the one we mainly describe in this article, as "covert attention" [7]. Covert changes are much faster (up to five times) than overt ones, which makes this early attention an important tool to decide whether it is suitable or not to change the current fixation point (move our eyes or even the head).

Several attention models have been proposed over the years, mainly from a psychological and neurological point of view [8], [9]. Natural attention is the starting point of all of them. Since a detailed analysis transcends the scope of this paper, a list of those more related to this work is given:

- Classic Attention model by Koch and Ullman [10]. Several feature maps are extracted from the input image and then used to build a saliency map. A WTS (*winner-takes-all*) process will then select the more relevant areas in this map and direct attention to them. It is the base of most of the other models explained in here.
- Wolfe's Guided Search model [11]. Based on Koch and Ullman model, it starts with the computation of basic features, such as color and orientation, which are then used to build the so called feature maps. These maps are finally merged in an activation map which will be used for guiding the attention to the most relevant areas (those with higher values in the map).
- Saliency Map models. Itti et al. [3], [5], [6], [12] developed a model closely related to Koch and Ullman studies. This model builds up a saliency map to guide attention using color, intensity, orientation and movement maps which are extracted from the input images.

All these models are often called "Feature-Based Attention Models". Their main objective is identifying the more conspicuous areas in the current scene. There are several other approaches created to model attention, such as "Connectionist Attention Models" [13], [14], [15], which are oriented to create a reference frame for specific objects or some of their environmental interaction features (for instance, specific movement patterns).

The model described in this paper is an adaptation to the Robocup SPL environment of Itti's proposal. While being conceptually simple, it offers great results while not consuming a high amount of resources. To further prioritise performance, several of its elements have been simplified: some of its maps are dispensed.

The model will be further reviewed in the next section. It has been proved to obtain excellent results even with high sensor noise or when working with high informative content images. It was even capable of finding object such as traffic better than software specifically programmed for that task [12].

III. MODEL

Our model is based on Itti et al. saliency map attention model [3], [5], [6], [12]. At any given time, the maximum registered in the saliency defines the most important region from an attentive point of view.

To build up the saliency map in our model, two maps are used: an intensity map and a color map. The other two maps of the original model (orientation and movement) are dispensed since we do not find them necessary for our environment.

The maps assign high values to those areas which stand out in the magnitude they meassure: intensity map will assign high values to those areas the intensity (light) of which changes a lot in relation to their surround, while color maps will do the same for the ones with a high color contrast. The maps are obtained using the original camera image.

To avoid revisiting regions which have been recently analysed, an inhibition mask can be applied to the last visited locations, both locally for the image and globally for the camera angle: after checking a given area, it is masked so it can not be revisited as soon as the analysis process finishes.

A. Multiscale pyramid and maps

Most attention models use a multiscale pyramid to represent the visual information they use [16], [17], [6], [12], [3], [5]. Visual information can be arranged in a multiscale pyramid [18], [19], [20], [21] with several levels, each of them with an image (corresponding to the current visual visual) at a different size and resolution, see Fig. 2. The higher we are in the pyramid, the lower the image resolution is.

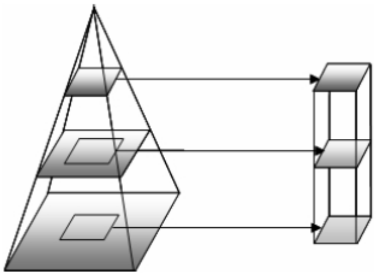

Fig. 2. Multiscale pyramid

The input of the model are 640x480 pixels static RGB color images (Fig. 3 shows the input image which will be used as an example during the explanation of the map construction process). These images are used to build multiscale pyramids [18] for every map used in the model. Each pyramid has 9 levels and a resolution reduction factor of $1 : 2^n$ for each of them. Level 0 means then no reduction (1:1, original image), while maximum reduction happens at level 8 (1:256). The specific image resolution for each level is then the following:

level 0: 640x480
level 1: 320x240
level 2: 160x120
level 3: 80x60
level 4: 40x30
level 5: 20x15
level 6: 10x8
level 7: 5x4
level 8: 3x2

Intensity maps

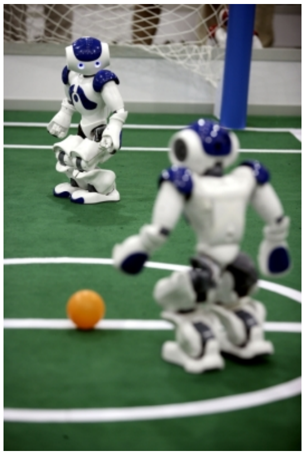

Fig. 3. Original input image

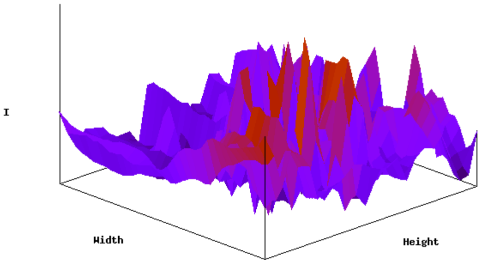

Fig. 4. Intensity map

The first step of the model consists of creating a nine level intensity pyramid which represents the “intensity” (luminosity) of each image pixel. Using the original image, a intensity matrix M_I is obtained by combination of the R, G and B channels value:

$$m_I(i,j) = \frac{m_R(i,j)+m_G(i,j)+m_B(i,j)}{3}$$

The intensity pyramid is then created using M_I, with $M_I(n)$ being the intensity matrix corresponding to the nth level of the pyramid. Using the pyramid, six intensity maps are obtained by across-scale difference, $\ominus$, which is obtained by interpolation of the maps to the finer scale and point-by-point subtraction:

$$\begin{aligned} M_{I(2,5)} &= |M_{I(2)} \ominus M_{I(5)}| \\ M_{I(2,6)} &= |M_{I(2)} \ominus M_{I(6)}| \\ M_{I(3,6)} &= |M_{I(3)} \ominus M_{I(6)}| \\ M_{I(3,7)} &= |M_{I(3)} \ominus M_{I(7)}| \\ M_{I(4,7)} &= |M_{I(4)} \ominus M_{I(7)}| \\ M_{I(4,8)} &= |M_{I(4)} \ominus M_{I(8)}| \end{aligned}$$

This across-scale difference between maps allows for detecting locations at center (areas at scale 2,3,4) which stand out from their surround (scale 5,6,7,8), the same way it happens in human retina [6]. Using several scales for center and surround, instead of just one for each of them, yields truly multiscale feature extraction [6]. The finnest scale is $n = 2$ and not $n = 0$ to reduce noise, excessive detail, and the amount of pixels to be computed (160x120 at scale 2 instead of 640x480 at scale 0), improving both performance and robustness.

Finally, the intensity map I, representing those conspicuous locations from an intensity point of view, is generated combining all the previous maps through across-scale addition, $\oplus$, which consists of reduction of each map to scale $n = 4$ ($40x30$ resolution) and point-by-point addition:

$$I = \oplus M_{I(m,n)}$$

Figure 4 shows the intensity map I for the image at Fig. 3.

Color maps

Four pyramids representing “color” of each image pixel are created using the normalised R, G and B channels and a yellow channel Y (obtained using the three previous ones): RGB color space channels include intensity information, thus, in order to make the result independent to environmental light, they have to be normalised by intensity. To do so, we applied the same formulae used in [3]

The four color pyramids are used to generate a set of 12 color maps, six for difference between red and green components, $M_{RG(m,n)}$, and six for blue and yellow difference, $M_{BY(m,n)}$, in a similar fashion to the intensity maps.

$$\begin{aligned} M_{RG(2,5)} &= |(M_{R(2)} - M_{G(2)}) \ominus (M_{R(5)} - M_{G(5)})| \\ M_{RG(2,6)} &= |(M_{R(2)} - M_{G(2)}) \ominus (M_{R(6)} - M_{G(6)})| \\ M_{RG(3,6)} &= |(M_{R(3)} - M_{G(3)}) \ominus (M_{R(6)} - M_{G(6)})| \\ M_{RG(3,7)} &= |(M_{R(3)} - M_{G(3)}) \ominus (M_{R(7)} - M_{G(7)})| \\ M_{RG(4,7)} &= |(M_{R(4)} - M_{G(4)}) \ominus (M_{R(7)} - M_{G(7)})| \\ M_{RG(4,8)} &= |(M_{R(4)} - M_{G(4)}) \ominus (M_{R(8)} - M_{G(8)})| \end{aligned}$$

$M_{BY(m,n)}$ are obtained in a similar way to $M_{RG(m,n)}$ but using the Blue and Yellow components instead.

Finally, a color map C, representing those conspicuous locations from a color contrast point of view, is generated combining all the previous maps:

$$C = \oplus[RG_{I(m,n)} + BY_{I(m,n)}]$$

Figure 5 shows the color map C for the image at Fig. 3.

Orientation Maps

Te original model builds up a set of orientation maps which are merged in a final orientation map O which represents the location of those elements which stand out from an orientation point of view in comparison to the rest of the objects present in the image.

Such maps have not yet been implemented in the current version, mainly due to the fact that they are not so important for a controlled environment like ours (Robocup SPL) in which colour and intensity are already very conspicuous by themselves.

Normalisation

Before obtaining the final saliency maps, all maps have to be normalised.

The Color and Intensity maps obtained are normalised to the same static range $[0..M]$ in order to compare them. Modality

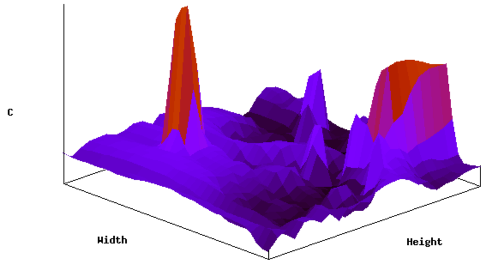

Fig. 5. Color map

dependant differences would also have to be removed. However, since we do not compute orientation maps, this step is not necessary: a 5% intensity difference between two pixels can not be a priori compared to a 0.2 rad orientation difference, but color and intensity differences can be compared without further modification.

A mechanism to promote maps with a small number of strong peaks of activity (conspicuous locations) is also applied. It consists of finding the map's global maximum (M) and computing the average of all its other local maxima (m), globally multiplying the map by $(M-m)^2$. The biggest advantage of this method is its simplicity and speed, while the major drawback is that if a map has two important locations it will only promote the most conspicuous one, hiding the other (humans would probably attend to both of them instad).

In [6], a more complex and efficient method for normalisation based on DoG (Difference of Gaussians) filters is proposed, but it has not yet been implemented.

Saliency map

Once the color and intensity maps have being obtained and normalised, they are combined in the final saliency map S which will guide attention to the most relevant location in the field of view:

$$S = \frac{I+C}{2}$$

Fig. 6 shows the 3D (left) and 2D (right) saliency map S for the image at Fig. 3.

The saliency map S is then applied to the original image as obtained by the robot camera, promoting the most relevant locations and hiding the rest. In Fig. 7 this process is illustrated: left image is the original coloured image. Central image shows the results of applying the saliency map in Fig. 6 to the original image (the darker the area, the less salient it is). Right image shows the regions with higher saliency across the whole map (green rectangles). Please note that the system proposed only tell us "where" to look at (area) and not "what" (object) to look for; the fact that the ball and the keep are in those areas is a consequence of being the most notorious regions of the image from a color and intensity point of view.

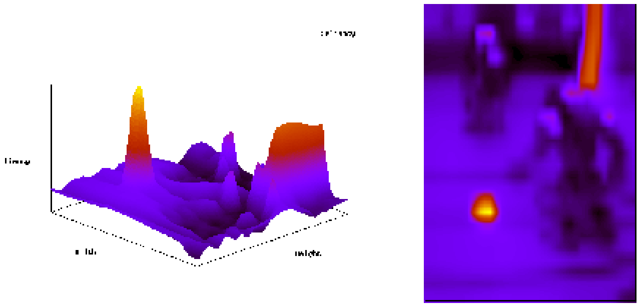

Fig. 6. Saliency map

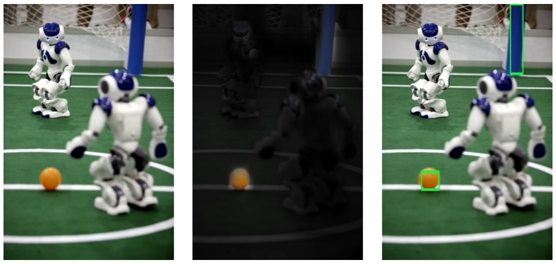

Fig. 7. Saliency map applied to the original image

IV. Experiments

To test the effectiveness of our approach, the "any ball" Robocup challenge has been chosen. For this challenge, the robot is placed in the game field along with a couple of random coloured and multi-sized balls. The robot has then a couple of minutes to score the biggest amount of goals possible. Classic color filter algorithms used for image segmentation are not useful in this scenario, since not only ball color is unknown, but they can also have the same color as the ground (green).

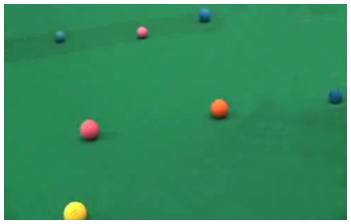

Fig. 8. Any ball challenge input image

In order to solve this challenge both a simulation and a real scenario have been used.

Simulation is so called because only the attention model is implemented, supposing that the robot would be able to kick the ball after knowing its location. In this case, just finding the location of any ball is enough to consider the test a success.

The term "real scenario" means that ball approaching and kicking are also implemented.

A. Simulation

To simulate this scenario for our system we have given the robot some pictures of the game field containing a random number of different color balls (see Fig. 8).

The model proposed always finds the most salient region in the image, and as long as that region is not dealt with (or inhibited), it will not find any other region. This means that regions chosen as most salient which do not contain any ball must be masked (inhibited), so that others containing a ball can be chosen as focus.

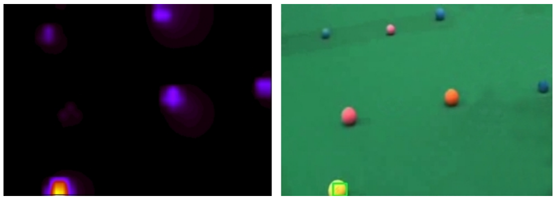

Fig. 9. Most salient region of the input image

Once a region containing a ball is chosen (see Fig. 9), robot should approach to it and try to score a goal by kicking it. This part of the experiment has not been implemented yet, but it can be assumed that the ball will end up further from the robot than it was when chosen as focus. To simulate that, once a region containing a ball is chosen by the model, it is assumed that the robot could kick it and that specific ball is removed from the next input image for the robot.

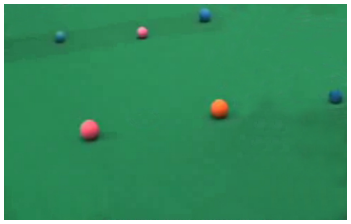

Fig. 10. Any ball challenge second input image

With the originally most salient ball no longer present in the field of view (see Fig. 10), the saliency map changes and a new most salient region is chosen (see Fig. 11). The previously explained process is now repeated: if the new region contains a ball, it is chosen as focus and kicked, otherwise, it is inhibited and the second most salient region is checked, repeating the process until finding a region containing a ball or not finding any at all.

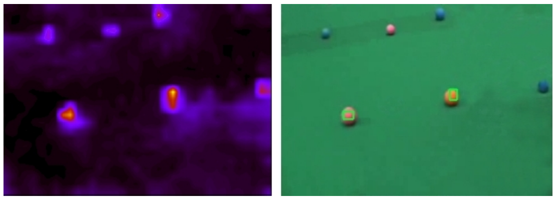

Fig. 11. Most salient regions of the second input image

The results obtained are very promising, with a 100% success rate for the images used. Even the regions containing small balls with almost the same color of the ground are chosen in the last iterations of the algorithm (see Fig. 12). It can be easily understood that the color map (top left image at Fig. 13) gives no useful information in this case, since the whole field of view is almost of the same color (except for the lines). However, the intensity map (top right image at Fig. 13) shows strong peaks at those areas containing either shades, which should be minimal except for the one belonging to the ball (due to it being the only object in the field apart from the robot), or different light reflection patterns, as it happens with the region containing the ball since the ball is made of a different material from the ground's. The final saliency map obtained once again chooses the region containing the ball as the most salient one (see bottom left and bottom right images at Fig. 13).

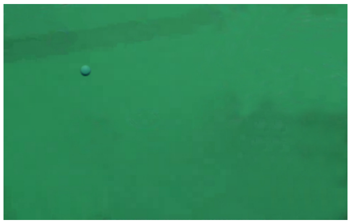

Fig. 12. Input image containing a ball of the same color of the ground

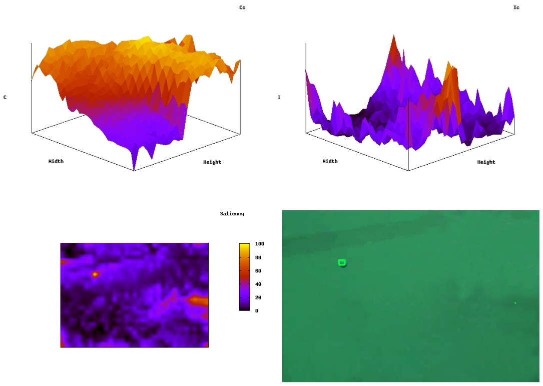

Fig. 13. Color, intensity and saliency map and most salient regions of an input image containing a ball of the same color of the ground

B. Real scenario

In this case, the whole process involving scoring a goal is implemented: the attention algorithm is used to locate a target ball to which the robot approaches and kicks in order to score.

To put the whole system to the test, three different colored objects have been positioned in the field at random locations. As it may be expected from the results commented in the previous section, the robot successfully chooses every target, one after another, gets close to them and kicks them. A video showing this behaviour along with the attention model input, the real images, and its output, the target locations, can be watched at `http://robotica.unileon.es/~jfgars/pubs`.

The major problem of the approach can also be seen in the same video: high computation time needed for the attention algorithm to work makes it unable to operate at real time. Specifically, frame rate drops down to 5 when computing saliency maps.

V. Discussion and further work

In this paper we have presented an attention control model based on a saliency map which mainly differs from the original by Itti [6] in two aspects: the saliency map is obtained using only intensity and color information, dispensing orientation and movement data. These modifications improve the model performance and allow for a better adaptation to the Robocup SPL environment.

The model has proved to be useful for the "any ball" challenge, with better results than classic filter and segmentation algorithms, which do not provide results robust enough when trying to identify balls of similar color to the field.

The main drawback of our proposal is the time it consumes, which makes the model not usable for real time game play. However, the system remains suitable for competition when combined with classic color filter algorithms, applying the saliency calculation only to certain images or situations (finding areas in the field containing interesting objects, for instance a ball in the proposed challenge) and using the classic color filter approach for the rest of the tasks (object recognition and subsequent tracking).

There are mainly two topics which would need to be addressed in the near future: a more effective normalisation operator and time consumption optimisation.

Orientation and movements maps are not computed in the model since their usefulness is arguable for our environment, in which color and intensity give enough saliency information to properly detect regions containing all interesting elements (ball and keeps).

The simple normalisation operator used tends to promote only one activity peak in the intermediate maps, which makes the most conspicuous area hide the rest even if there is a second one very close to it (and thus also very important from a saliency point of view). This leads to occasional problems. For instance, when both the ball and the yellow keep are visible, specially with partial ball occlusions, the yellow net may hide the ball in the final saliency map. In the "any ball" challenge experiment here explained, it can be seen that, for the same reason, some of the regions containing balls are not found until second iteration (compare Fig. 9 to Fig. 11) when the previously most salient region (the one containing the yellow ball) has been removed. Itti et al. already solved this issue by using DoG filters instead [6], which makes the system work better in these cases.

As previously stated, time consumed by the maps generation algorithm is too high. One of the main advantages of attention is the great reduction in the amount of information to process, specially since processing a stream of video in limited hardware as a robot is a high time-consuming task. However, the whole process is taking around 200 ms, which is an excessive amount of time to make it worthwhile in this sense. An optimisation of the code could make the system much more suitable for full time use.

References

[1] P. Bachiller, P. Bustos, and L. J. Manso, "Attentional selection for action in mobile robots," *Advances in Robotics, Automation and Control*, 2008.

[2] S. Frintrop, G. Backer, and E. Rome, "Goal-directed search with a top-down modulated computational attention system," *Lecture Notes in Computer Science*, vol. 3663, pp. 117–124, 2005.

[3] L. Itti and C. Koch, "A saliency-based research mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, pp. 1489–1506, 2000.

[4] A. Torralba, A. Oliva, M. S. Castellanos, and J. M. Henderson, "Contextual guidance of eyes movements and attention in real-world scenes: the role of the global features in object research," *Psychological Review*, vol. 113, pp. 766–786, 2006.

[5] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, pp. 194–203, 2001.

[6] L. Itti, C. Koch, and E. Niebur, "Attentive mechanisms for dynamic and static scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254–1259, 1998.

[7] R. Wright and L. Ward, *Orienting of Attention*. Oxford University Press, 2008.

[8] B. Julesz, "Early vision and focal attention," *Review of Modern physics*, vol. 66(3), pp. 735–772, 1991.

[9] D. LaBerge, R. Carlson, J. K. Williams, and B. G. Bunney, "Shifting attention in visual space: Tests of moving-spotlight models versus an activity-distribution model," *J. Experimental Psychology: Human Perception and Performance*, vol. 23, pp. 1380–1392, 1997.

[10] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, pp. 219–227, 1985.

[11] J. M. Wolfe, *Guided Search 4.0: Current Progress with a model of visual search*. Brigham and Womens Hospital and Harvard Medical School, 2007.

[12] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254–1259, 1998.

[13] G. Deco, *A Neurodynamical Model of Visual Attention: Feedback Enhancement of Spatial Resolution in a Hierarchical System*, G. Baratoff and H. Neumann, Eds., 2000.

[14] J. M. Gryn, R. P. Wildes, and J. K. Tsotsos, "Detecting motion patterns via direction maps with application to surveillance," *Computer Vision and Image Understanding*, vol. 113, pp. 291–307, 2009.

[15] B. Olshausen, C. Anderson, and C. Essen, "A multiscale dynamic routing circuit for forming size- and position-invariant object representations," *J. Computational Neuroscience*, vol. 2, pp. 45–62, 1995.

[16] J. Tsotsos, S. Culhane, W. Winky, Y. L. and N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning model," *Artificial Intelligence*, vol. 78, pp. 507–545, 1995.

[17] H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit, and C. Anderson, "Overcomplete steerable pyramid filters and rotation invariance," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 222–228, 1994.

[18] C. Anderson and D. V. Essen, "Shifter circuits: a computational strategy for dynamic aspects of visual processing," in *Proc. Nat. Acad. Sci. USA*, 1987.

[19] *P. Burt, Attention mechanisms for vision in a dynamic world, Proceedings Ninth International Conference on Pattern Recognition, Beijing, China*, 1988.

[20] *J.K. Tsotsos, A complexity level analysis of vision, in: Proceedings International Conference on Computer Vision: Human and Machine Vision Workship, London, England*, 1987.

[21] L. Uhr, "Layered recognition cone networks that preprocess, classify and describe," *IEEE Transactions on Computing*, vol. 21, pp. 758–768, 1972.