Comparison of Smart Visual Attention Mechanisms for Humanoid Robots

Carlos E. Agüero¹, Francisco Martín¹*, Luis Rubio¹ and José María Cañas¹

¹ Robotics Lab at Universidad Rey Juan Carlos, Spain
* Corresponding author E-mail: fmartin@gsc.es

Received 9 May 2012; Accepted 19 Sep 2012

© 2012 Agüero et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract Cameras are one of the most relevant sensors in autonomous robots. One challenge with them is to manage the small field of view of regular cameras. A method of coping with this, similar to the attention systems in humans, is to use mobile cameras to cover all the robot surroundings and to perceive all the objects of interest to the robot tasks even if they do not lie in the same snapshot. A gaze control algorithm is then required that continuously selects where the camera should look. This paper presents three different covert attention mechanisms that have been designed and compared: one based on round-Robin sharing, another based on dynamic salience and one with fixed pattern camera movements. Several experiments have been performed with a humanoid robot in order to validate them and to give an objective comparison in the context of RoboCup, where the robots have several perceptive needs like localization and object tracking that must be satisfied and may not be fully compatible.

Keywords Active robot vision, Sensor attention, Humanoid robot

1. Introduction

The use of cameras in robots is continuously growing. They can potentially provide the robot with much information about its environment and in recent years they have become a cheap sensor. Most service robots, including humanoid prototypes, are equipped with vision as it is the most promising technology for human-robot interaction. However, dealing with the huge amount of data carried by video streams is not easy. Visual attention offers a solution to the processing bottleneck generated by such an overwhelming source of raw data and it is especially convenient in machines with limited computational resources.

The attention mechanism of the human vision system has been the source of inspiration for machine visual systems, in order to sample data non-uniformly and to utilize computational resources efficiently [1]. The performance of the artificial systems has always been compared to the performance of several animals, including humans, in simple visual search tasks. In recent years, biological models are moving into the real-time arena and offer an impressive flexibility to deal simultaneously with generic stimulus and with task specific constraints [2][3].

Machine attention systems have typically been divided into overt and covert ones. The covert attention mechanisms [4][5][6] search inside the image flow for relevant areas for the task at hand, leaving out the rest.
The search for autonomous vehicles in outdoor scenarios for military applications [4] and the search for traffic signals inside images from on-board car cameras are just two sample applications. Salience is an interesting concept.

Overt attention systems [7][8][9] use mobile cameras and cope with the problem of how to move them: looking for salient objects for the task at hand, tracking them, sampling the space around the robot, etc. The saccadic eye movements observed in primates and humans are their animal counterpart. They have been used, for instance, to generate a natural interaction with humans in social robots like Kismet [10]. This active perception system can guide the camera to better perceive the relevant objects in the surroundings. The use of camera motion to facilitate object recognition was pointed out by [1] and has been used, for instance, to discriminate between two shapes in an image [6].

Most successful systems define low level salient features like colour, luminance gradient or movement [8]. Those features drive the robot attention following an autonomous dynamics in a close loop with the images. In this way the system is mainly bottom-up guided by the low level visual clues. One active research area is the top-down modulation of these systems, i.e., how the current task of the robot or even the high levels of perception, like object recognition [11][12], can tune the attention system and maybe generate a new focus of attention.

In our scenario, visual representation of interesting objects in the robot’s surroundings improve the quality of humanoid behaviour as its control decisions may take more information into account. This poses a problem when there are several objects which are not covered completely by camera in same field of view.. Some works use omni directional vision and this has been successfully applied in problems like visual localization or soccer behaviours in the RoboCup middle size league. Other approaches use a regular camera and an overt attention mechanism [3][13], which allows for rapid sampling of several areas of interest. This is the only available approach in the RoboCup SPL humanoid league.

In this paper we report three different overt attention systems for a humanoid robot endowed with a camera on its head, which can be oriented at will independently from the robot base. These algorithms combine the several perceptive needs of the robot and control the camera movements in order to keep all of them satisfied, providing them with enough images along time to achieve its perceptive goal. For instance, to track several task-oriented objects or to visually localize the robot inside the environment they implement different time sharing policies for the camera control: round-Robin, a saliency-based one and a fixed movement pattern.

Following this introduction, the second section describes some related works. Our three attention mechanisms and where the robot control architecture has been developed are presented. Many experiments have been carried out on a real humanoid robot to validate and compare the results, and they are described in the fifth section. Finally some brief conclusions end the paper.

2. Related work

One of the concepts widely accepted in visual attention is the salience map. This is found in [5], as a covert visual attention mechanism, independent of the particular task to be performed. This bottom-up attention builds in each iteration the conspicuity map for each one of the visual features that attract attention (such as colour, movement or edge orientations). There are competition dynamics inside each map and they are merged into a single representative salience map that drives the focus of attention to the area with highest value.

Regarding overt visual attention mechanisms, Hulse [14] presented an active robotic vision system based on the biological phenomenon of inhibition of return, used to modulate the action selection process for saccadic camera movements. Arbel and Ferrie presented in [15] a gaze-planning strategy that moves the camera to another viewpoint around an object in order to recognize it. Recognition itself is based on the optical flow signatures that result from the camera motion.

Researchers within the RoboCup community typically maintain an object representation known as the world model containing the position of relevant stimuli: ball, goals, robots, etc. The world model is updated using the instantaneous output of the detection algorithms or by running an extra layer that implements some filtering. In this RoboCup scenario, policies to decide when and how to direct the gaze to a particular point can be divided into three groups. First, those that delegate to each behaviour the decision on the positioning of the head. Second, those which continuously move the robot’s camera in a fixed pattern to cover the entire search space of the robot. Its main drawback is that it does not allow tracking a detected stimulus. In addition, much time is wasted on exploring areas where we know that there are no stimuli. The third group includes those using a specific component responsible for making this decision based on the requirements of active behaviours. Here we can find attention mechanisms guided by utility functions based on the task the robot is currently doing [16], salience-based schemes which increase with time [17] or time-sharing mechanisms, among others.

In one SPL team [17] the behaviours define the importance of each stimulus. Depending on this
importance and the value of its current associated uncertainty, the active vision system decides which stimulus to focus on at any time. The behaviours themselves establish which stimuli should be observed. This approach does not tolerate observations with occlusions and partial observations. Also, in this work the search for new objects uses the same pattern fixed for head positions, independently of the type of the object to search.

In [18] and [19], the state of the robot is modelled using Monte Carlo Localization (MCL). The state includes both the robot’s position and the ball. The aim of the active vision system here is to minimize the entropy of the state of the robot. To accomplish this task, it divides the field into a grid and it calculates the utility of pointing the camera towards each cell, taking into account the cost of performing this action. In this way, they calculate the position where the camera focuses at all times. This approach emphasizes the idea that the active vision should be associated with a utility (self-localization and detection of the ball), and that the utility of turning our gaze towards one place or another is quantifiable depending on how it decreases the entropy of the system. These approach behaviours do not define the importance of the stimuli and they do not modulate the active vision system in any way.

3. Visual attention architecture

The visual attention system controls the camera’s position and orientation to perceive objects in the environment. This system connects the perception system and actuation to meet the perceptual needs which set the behaviours.

In this section we will present three different attention algorithms. Our system has been architecturally designed with the intention to develop different algorithms of attention and establish which one is active. The design must be clean, flexible and scalable. At the robot startup, we can select an attention algorithm.

The main elements involved in the process of visual attention, as shown in Figure 1, are behaviours, detectors and the attention controller. The generic attention system operation is described as follows:

- Behaviours establish which objects need to be perceived at any time and its importance, in a range [0,1] where 1 is the highest importance and 0 (or no specification) lack of interest. Behaviours are software components which run iteratively activating other components (auto-location, navigation, etc.) in cascade. These additional components may define other attentive requirements. Activation and deactivation of the software components in our architecture is silent, so each component must refresh their perceptual requirements each time they are executed, and the visual attention controller must be aware of when it should no longer perceive an object. The interface between the controller behaviour and attention is through the call setImportance (object, importance). For example, to perceive object A with 90% importance, a behaviour makes the call setImportance ("A", 0.9).

- Detectors are software components specialized on detecting perceptive objects using the information from the camera. They are part of the execution in cascade initiated by high-level behaviours. Once an object in the image has been detected, the position in the image is transformed into a 3D position relative to the robot, whose axis of reference has its origin on the floor, between the two legs of the robot. Calculating the object 3D position with a single camera is possible using extra information like the Z coordinate position, which is 0 for the object on the floor. The 3D position of the object is incorporated into a visual memory distributed among the detectors. Visual memory uses a set of extended Kalman filters, one for each object, to keep the estimation and the uncertainty on the position of each object. From the attentive system point of view, each detector defines how to track an object and where to look for it if this object has not been recently perceived. Each detector has a particular way to seek and track objects because: a) there are static and dynamic objects, b) each object have unique or multiple instances, c) some objects are best detected on the horizon, or in the floor, and others can be detected anywhere.

![Figure 1. Attention system architecture. The attention controller receives the importance values for each detector. It asks the detectors for 3D attention points and informs when an attention point is reached. The robot’s neck angles are calculated from each 3D attention point.](image-url)

- When behaviours set the attention controller to perceive object A, it requests (using getAPI() method) to the detector of object A a 3D attention point where to orientate the camera. If the object has been
perceived recently, it returns the 3D position of A stored in the visual memory, or a null value if it is not necessary to perceive it again (this is common for static objects when the robot has not moved). If the detector does not know the position of an object, it defines a list of 3D search points to visit. Each time the attention controller requests a point to look at, it returns the same point on the list. If the controller informs (using atAP() method) that the point has already been reached, it replies with the next point on the list, and so on.

- The controller of the attention is responsible for combining the perceptual requirements from behaviours and information of the detectors by sending joint action commands to the neck. This controller acts as a referee which determines the information of which object must be refreshed at any time. As introduced above, the interface with the detectors consists of two calls: getAP() and atAP(). The first requests from a detector a 3D point where to direct the camera, and the second informs that the attention point has been reached.

This description defines the attention system in general terms, but it leaves open many questions: how does the controller set the turns using the importance of each object? How does a detector define which point on the search list is better to return? Can detectors collaborate on finding objects? Our system allows us to implement different mechanisms of attention which define these topics. Each mechanism has its own different attention controller and defines the attentive operation of each detector. We designed three algorithms of visual attention: round-Robin, salience-based and using fixed patterns.

3.1 Round-Robin algorithm

This algorithm plans the attention giving each detector a time slot. The duration of this time slot is proportional to the importance defined by the behaviours. If the attention controller receives different importance values for the same detector, it chooses the maximum value received.

In the slot assigned to a specified detector, an attention point is asked for. The detector may return the 3D point corresponding to the known position of the object, or iterate between the known position of the objects in the case of more than one object of a type. If the object position is not known, the detector returns a 3D search point from the list of positions which may be the object, changing this attention point as soon as the attention controller notifies that the point has been reached and the detector has not detected the object in the image yet. In the case of static objects, if the robot has not moved and the objects have been detected in a slot, the detector can transfer the remainder of the slot to another one.

Figure 2 shows an example of this algorithm. Behaviours define the importance of each type of objects. In this Figure the behaviours A1, A2, A3 and A4 define different importances for the objects A, B and C. In the case of the object C, three values (0.75, 0.5 and 1.0) are defined, but only takes into account the largest (1.0). From the maximum values, A (0.5), B (1.0) and C (1.0), the length of the slot for each set is set. The total length of the slot, T_cycle, is set within the range [5-10] seconds. A detector is set to 20% of T_cycle, B and C detectors are set to 40%, respectively.

3.2 Salience-based algorithm

In contrast to the previous algorithm, this algorithm selects the active detector depending on the difference between the detection quality and the importance established for it by the behaviours. The quality of the information of a detector is the average of the quality of the detector objects that it has to detect. If the item is unique, the detector quality depends only on the uncertainty of the object.

The quality of an item is set to 1 when the object is seen in the image. This quality decreases over time depending on their characteristics (the quality value reduces quickly in the case of dynamic objects and slowly if they are static) and robot motion.

Figure 3 illustrates the mechanism of arbitration. Behaviours define the importance of perceiving objects A
and B. These values are used as reference (\(q_{\text{ref}}(A)\) and \(q_{\text{ref}}(B)\)). At any time, the difference between the reference value and the quality of each item defines the priority. The larger the difference below the reference value, the higher the priority for the controller attention. In Figure 3, we can observe how the quality of detector B decreases faster than that of detector A. This may be because the object that detects B is dynamic and the object that detects A is static.

Each detector sets a list of 3D search points and the latest known positions for the objects that it has to seek and track. For each 3D point, a value of salience, in the range [0,1] is associated. In visual attention literature, salience is defined as the desire to perceive an object. The longer this value the greater the need to direct the camera at that point.

As a detector does not perceive the object that it is looking for, the salience of all the points increases with time. Attention points’ salience covered by the camera’s field of view are set to 0 if the object is not present. When a detector finds an object, the salience of the searching points are set to 0, and the tracking point is set with a small (greater than 0) value. The salience of the attention points of all the detectors are updated following these rules, even those which are not active. This makes the search more efficient, because the attention points covered during the execution of other detectors are not revisited.

**Figure 3.** Evolution of the quality for two different detectors.

Figure 4 shows an example where three detectors are active (A, B and C). The attention points for each detector are represented with a different colour and a shape, and each point has an identifier that indicates which detector it belongs to and in brackets the value of salience. Detector A is responsible for searching an object that is unique, while detector C is responsible for detecting multiple objects in the same class. In the same figure, the field of view is covering several attention points belonging to different detectors. We can assume that, for example, detector B is the active one (the difference of its quality and the reference value is largest), and so it controls the position of the camera. Detector B has detected the desired object (represented by coloured circles) in the image, so it sets the salience to 0 for the other points of attention, and a value greater than 0 for the point where the object has been detected. Detector C finds one of the objects that it is being looked for, so it sets the saliency to a value greater than 0. It also maintains the salience of the other attention points where objects have been detected previously (C_AP2).

This allows the detector to visit alternatively the position of these objects, when detector C takes the control of the camera. Detector A has not found any object yet and sets to 0 the salience of the covered attention points in the field of view of the camera. The salience of its attention points establishes the search path for detector A.

**Figure 4.** Centinal view of the robot space. Attention points from different detectors are represented with different colours and shapes. The blue parallelogram represents the camera field of view. Circles represent the detection of the objects. Each attention point is labelled with an id and the salience value.

### 3.3 Fixed pattern algorithm

The two methods presented above have complex mechanisms to establish which one is active and the movement to look for the objects. This can adjust attention to the behaviour requirements accurately, but it has a drawback: the camera moves so fast that often the images are not useful while the camera changes attention point.

The last algorithm that we present is much simpler. There is a fixed pattern for the camera movement defined by a list of attention points shared by all the detectors. The attention system moves the camera slowly (~ pi / 3 per second) visiting the attention points, regardless of whether or not the objects of the active detectors are perceived. If there is a detector critical for the robot operation, the last known position of the object is pushed back in the shared list of attention points.
Figure 5. Cenital view of the robot space. The fixed pattern comprises 10 attention points (circles) and an attention point for a critical detector (box). The blue parallelogram represents the camera field of view when pointing the 6th attention point.

Figure 5 shows the shared set of attention points and the fixed path to visit them. In this example, there is a critical detector which sets an extra point to visit, that corresponds to the last known position of the object to critically detect.

4. Experiments

Several experiments have been carried out to validate the system described in this article. We have used the NAO robot as a test platform. The Nao robot v3.3 is a medium-sized humanoid robot with 58 cm of height, 21 degrees of freedom and a built-in x86 AMD Geode CPU at 500 MHz running GNU/Linux. The Nao features two CMOS 640x480 cameras, Ethernet, Wi-Fi, an inertial unit, force sensitive resistors, sonars, bumpers, four microphones, two hi-fi speakers and a complete set of LEDs.

The environment for the experiments is the scenario proposed by the RoboCup Standard Platform League2 (SPL) competition. RoboCup [20] uses soccer as a central topic of research, in which many fields of research can be tested, evaluated and compared. In the SPL league all teams use the same robot and changes in hardware are not allowed, so the effort focuses on software. Robot soccer with this humanoid is a complex and challenging scenario for robotics research as it is dynamic, with opponents, and allows the cooperation of several robots inside a team.

A fully featured soccer player has to get information from the environment, mainly using the camera. The images from a single onboard camera provide high volume data about the environment, and task relevant information must be extracted from them. Control decisions may be based exclusively on this information, but this suffers from some limitations. For instance, the camera scope is limited to a 60° field of view, and it is common to have occlusions of the objects and even false positives. Robots must identify and locate the ball, goals, lines and other robots. Having this information, the robot has to self-localize and plan the next action: move, kick, search for another object, etc. The robot must perform all these tasks very fast in order to be reactive enough to be competitive in a soccer match.

We have integrated the three proposed attention systems into our Nao robot architecture [21]. The attention module controls the head movement and continually shifts the focus of attention so the camera covers the perceptive requirements of the robot, looking at different areas of the scene, and providing new images to feed the tracking algorithms.

During the experiments, we collected a great amount of data used to analyse the visual attention system. The data collected during the execution of the experiments is stored in a log file for offline processing.

Meanwhile, we have adapted the ssl-vision [22] ground-truth system that captures the correct position of the dynamic elements of the field (robots and ball). This system is composed of two cameras mounted above the centre of each half field, and a computer running ssl-vision software. The ground-truth program broadcasts the position of the elements over the network. Figure 6 illustrates the ssl-vision architecture.

During the experiments, the robot is equipped with a visual pattern easily detected by the cameras, as we can see in Figure 7. The error of the ground-truth system is less than 3 cm in position and less than 5 degrees in orientation. Post-processing of robot log data and the ground truth data led us to accurately calculate the error and uncertainty in the robot’s perception module for each attention algorithm.

Figure 6. Environment overview for the experiments with ssl-vision installed.

---

1 www.aldebaran-robotics.com
2 www.tzi.de/spl
During the experiments we placed a robot on the field, so that it could perceive one of the goals and the ball. However, all objects could not be perceived simultaneously, so the attention algorithm must control the head.

The perception module tracks the ball and goal posts positions separately while the experiment is conducted. In addition, the self-localization algorithm is running, estimating the robot’s own position on the field.
The goal of the experiments is to quantify how each attention algorithm affects the accuracy obtained by the perception and the self-localization modules. In other words, we measure the quality of the attention algorithms indirectly by measuring the accuracy of the perception and auto-localization modules.

Figures 9, 10, and 11 show the data extracted from the experiments where data analysed from the static robot are located in the upper half of the images and the data extracted from the robot in motion are displayed in the lower half of the images. Table 1 summarizes the mean error and standard deviation obtained for all the elements analysed, divided by attention algorithm.

The data extracted from the experiments show that the fixed pattern attention strategy is the worst in terms of accuracy and uncertainty obtained. The continuous movement of the head is the cause of this result. Each instantaneous ball or post observation is done by a transformation from the 2D camera plane to the 3D real world space. To perform this transformation it is assumed that the objects are on the ground and the 3D camera position is known. The 3D camera position is algebraically calculated using the odometry values of each robot’s joint.

As usual, all measurements are noisy and therefore the estimate of the 3D camera position is not perfect. This fact adds some error to the object estimation proportional to the amount of movement and velocity of the joints.

In the round-Robin and salience-based methods, the head is still for a longer period of time, improving the estimation of the objects. In the salience-based method error and uncertainty are even lower than in the round-Robin variant because the head only moves when necessary, i.e., when the quality of the estimates decreases below a threshold.

The two groups of experiments obtained with the stationary robot and with the robot in motion show similar results. The most accurate attention algorithm is the salience-based, followed by the round-Robin approach, and then the fixed pattern technique. The standard deviation shown in Table 1 indicates that the estimated position for every object and the robot position tend to be spread out over a large area for the fixed pattern algorithm. The round-Robin approximation keeps a lower standard deviation value and the salience-based method has the lowest value of all the algorithms, demonstrating a better stability for the tracking and auto-localization algorithms.

The average execution time of the attention components was very similar among the three proposed approximations. The measured value on the Nao was 0.850 ms, proving that the three attention algorithms are suitable for real-time applications.

| Attention | Ball (still robot) | Left Post (still robot) | Right Post (still robot) | Auto-localization (still robot) | Ball (moving robot) | Left Post (moving robot) | Right Post (moving robot) | Auto-localization (moving robot) |
|-----------|--------------------|-------------------------|--------------------------|-------------------------------|----------------------|--------------------------|--------------------------|-------------------------------|
| Error mean (mm) | Std. Dev. (mm) | Error mean (mm) | Std. Dev. (mm) | Error mean (mm) | Std. Dev. (mm) | Error mean (mm) | Std. Dev. (mm) | Error mean (mm) | Std. Dev. (mm) |
| Fixed pattern | 166.00 | 126.22 | 117.94 | 117.94 | 120.37 | 54.49 | 53.78 | 53.78 | 53.78 |
| Round-Robin | 108.63 | 59.60 | 58.19 | 58.19 | 70.48 | 50.48 | 50.48 | 50.48 | 50.48 |
| Salience-based | 66.48 | 58.19 | 58.19 | 58.19 | 53.78 | 47.22 | 47.22 | 47.22 | 47.22 |

Table 1. Summary of accuracy and uncertainty results for each attention algorithm.

5. Conclusions and further work

In this work we have presented three different visual attention algorithms, valid for robots equipped with cameras and limited field of view, but with capabilities for controlling the gaze. The attention systems control the head movement and continually shift the focus of attention so the camera covers the perceptive requirements of the robot, by looking at different areas of the scene. The advantages of the attention system include the convenient combination of perception requirements, usually contradictory, and the delegation of gaze commands to each specialized object tracker. For instance, the combination mechanism solves the problem.
of moving the robot head to focus on the ball, which is not always compatible with the importance of looking at the goals. This organization allows the independent development of different behaviours, because their perceptive requirements can be met regardless of other behaviours.

The round-Robin and salience-based approaches allow a wider degree of tuning by changing the search points and priorities of each object to track.

Saliency-based algorithms are better suited to robot motion. The uncertainty of the objects grows at different rates, depending on whether objects are dynamic (fast growing) or static (growing slowly). If a robot moves, the uncertainty of the movement is added to the uncertainty of each object, making it grow faster. Algorithms based on salience determine the next object of attention depending on this uncertainty. If the robot is stopped and the newly detected objects are in their environment, it is only necessary to address the dynamic objects, optimizing the camera movement. Round-Robin and attention fixed pattern algorithms do not take into account the movement of the robot and do not take full advantage of the characteristics of objects.

An interesting discussion is related to the scalability of the proposed techniques. If the number of objects to track substantially grows, the round-Robin algorithm will cause excessive head movement. However, the salience-based approach would control the head in an efficient way taking advantage of the better object knowledge.

The RoboCup SPL environment has been selected as a validation scenario. The variety of objects to track and their different natures (stationary objects such as the goal posts and moving objects such as the ball or robots) represents a good benchmark for the proposed techniques.

![Figure 12. A relative comparison of all the attention algorithms.](image)

The description and validation of the attention algorithms has been carried out by measuring the quality of the tracking and auto-localization algorithms. Figure 12 shows the relative analysis of the data captured from the experiments. The bars on the graph prove the advantage of using a specific algorithm for object tracking and self-localization. The salience-based attention approach increments the accuracy and reduces the standard deviation of the estimations compared with the other alternatives.

We are extending this work in several directions. First of all, alternative indicators could be measured to understand the consequences of each algorithm. The time required to find an unknown object is an example.

Another future direction is the evaluation of combined approaches - one of the methods could be more suitable for some specific conditions. The capacity of dynamically changing the visual attention algorithm on demand could also be an interesting line to explore in the future.

6. Acknowledgments

This work was supported by the project S2009/DPI-1559, RoboCity2030-II, from the Comunidad de Madrid and by the project 10/02567 from the Spanish Ministry of Science and Innovation.

7. References

[1] D. H. Ballard, Animate Vision, Artificial Intelligence, Vol. 48, pp57-86, 1991.
[2] L. Itti, Models of bottom-up and top-down visual attention, PhD dissertation, California Institute of Technology, 2000.
[3] V. Navalpakam and L. Itti, Modelling the influence of task on attention, Vision Research, Vol. 45, pp205-231, 2005.
[4] J. K. Tsotsos et al., Modeling visual attention via selective tuning, Artificial Intelligence, Vol. 78, pp507-545, 1995.
[5] L. Itti and C. Koch, Computational modelling of Visual Attention, Nature Reviews Neuroscience, Vol. 2, pp194-203, 2001.
[6] D. Marocco, and D. Floreano, Active vision and feature selection in evolutionary behavioral systems, Proc. of Int. Conf. on Simulation of Adaptive Behavior, pp 247-255, 2002.
[7] S. Vijayakumar et al., Overt visual attention for humanoid robot, Proc. of Int. Conf. on Intelligence in Robotics and Autonomous System (IROS-2001), pp 2332-2337, 2001.
[8] L. Itti et al., A saliency-based search mechanism for overt and covert shifts of visual attention, Vision Research, Vol. 40, pp 1489-1506, 2000.
[9] C. Soyer et al., APES: attentively perceiving robot, Autonomous Robots, Vol. 20, pp 61-80, 2006.
[10] C. Breazeal and B. Scassellati, A context-dependent attention system for a social robot, Proc. of Int. J. Conf. on Artificial Intelligence, pp.1146-1151, 1999.
[11] Y. Sun and R. Fisher, *Object based visual attention for computer vision*, Artificial Intelligence, Vol. 146, N. 1, pp 77-123, 2003.

[12] F. Orabona, G. Metta and G. Sandini, *Object-based visual attention: a model for a behaving robot*, Proc. of Int. Workshop on Attention and Performance in Computational Vision WAPCV-2005, 2005.

[13] A. Zaharescu, A. L. Rothenstein and J. K. Tsotsos, *Towards a biologically plausible active visual search model*, Proc. of Int. Workshop on Attention and Performance in Computational Vision WAPCV-2004, pp 133-147, 2004.

[14] M. Hulse, S. McBride and M. Lee, *Implementing inhibition of return; embodied visual memory for robotic systems*, Proc. of the 9th Int. Conf. on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems. Lund University Cognitive Studies, 146, pp 189 - 190, 2009.

[15] T. Arbel, F. Ferrie, *Entropy-based gaze planning*. Image and Vision Computing, vol. 19, no. 11, pp 779-786, 2001.

[16] P. Guerrero, J. Ruiz-del-Solar, M. Romero, *Explicitly Task Oriented Probabilistic Active Vision for a Mobile Robot*. In RoboCup 2008: Robot Soccer World Cup XII, Springer-Verlag, pp 85-96, 2009.

[17] A. Saffiotti and K. LeBlanc, *Active perceptual anchoring of robot behavior in a dynamic environment*, Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2000, San Francisco, CA, pp 3796-3802.

[18] A. Seekircher, T. Laue, and T. Röfer. *Entropy-based active vision for a humanoid soccer robot*. In, RoboCup 2010: Robot Soccer World Cup XIV, number 6556 in Lecture Notes in Artificial Intelligence, pp 1-12. Springer; Heidelberg; http://www.springer.de/, 2011.

[19] S. Czarnetzki, S. Kerner and M. Kruse. *Real-time active vision by entropy minimization applied to localization*. In RoboCup, volume 6556 of Lecture Notes in Computer Science, pp 266-277. Springer, 2010.

[20] H. Kitano et al., *RoboCup: The robot world cup initiative*. In ICJAI-95 – Workshop on Entertainment and AI/ALIFE, 1995.

[21] F. Martin, C. E. Agüero, J. M. Cañas and E. Perdices, *Humanoid soccer player design*. In Robot Soccer, pp 67-100. InTech 2010. Available: http://www.intechopen.com/books/robot-soccer/humanoid-soccer-player-design

[22] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso, *SSL-Vision : The Shared Vision System for the RoboCup Small Size League*. In RoboCup 2009 : Robot Soccer World Cup XIII, pp 425-436,2009.