# Effective visual attention for behavior-based robotic applications

Francisco Martín, Luis Rubio, Carlos E. Aguero and José María Cañas

***Abstract*—Behavior-based architectures is one of the most popular alternatives to make an autonomous robot to carry out a task. This article presents a way to organize attention capabilities in architectures based on behaviors when the source of information is a camera with a limited field of view. In these architectures, the visual requirements are set by the active behaviors, and the attention system has to cover all the robot surroundings to perceive all the objects of interest. The system described in this paper controls the gaze using a control algorithm based on salience which continuously selects where the camera should look at. Several experiments have been performed with a humanoid robot in order to validate them and to have an objective comparison in the context of the RoboCup, where the robots have several perceptive needs and object tracking that must be satisfied and may not be fully compatible.**

***Index Terms*—behavioral robotics, Active robot vision, Sensor attention, Humanoid robot.**

## I. Introduction

AUTONOMOUS mobile robotics is a very extensive research field, since the robots have to move around their environment to perform a task, mainly using the information provided by their sensors. In recent years, the use of cameras as a primary sensor has grown enormously. They are cheap sensors that provide a lot of information: not only the information stored in the image, but the camera position relative to the robot and its environment. Although there are omnidirectional cameras, most cameras equipped on robots have a limited field of view, and has to move continuously to detect relevant objects at all times.

One of the most successful alternatives to generate actuation in autonomous robots is behavior-based architectures. In these architectures, a complex behavior usually arises from the combination of simple behaviors. Simple set of behaviors can vary depending on the situations that the robot is facing. Each of the simple active behaviors may require to perceive one or more objects and to define the importance of each object for a correct operation. For example, a humanoid robot has to navigate in an office environment to find an object. This general behavior can be decomposed into a simple behavior to look for the object, and a global navigation behavior, which requires the perception of landmarks for self-localizing. In this architecture, there must be a mechanism to decide where to move the robot camera to meet the needs of self-localization and object search.

An attention mechanism of human vision system has been 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 been always compared to the performance of several animals, including humans, in simple visual search tasks. In last years, biological models are moving to 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 been typically 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. Search of autonomous vehicles in outdoor scenarios for military applications [4], and search for traffic signals inside the images from the on-board car cameras are just two sample applications. One interesting concept is salience.

The 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 the images [6].

Most successful systems define low level salient features like color, luminance gradient or movement [8]. Those features drive the robot attention following an autonomous dynamics in a close loop with the images. 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, that is, 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 new focus of attention.

In our scenario, visual representation of interesting objects in robot's surroundings improve the quality of humanoid behavior as its control decisions may take more information into account. This poses a problem when there are several

Francisco Martín is with Rey Juan Carlos University.
E-mail: fmartin@gsyc.es

Luis Rubio is with Rey Juan Carlos University.
E-mail: l.rubiorod@gmail.com

Carlos E. Aguero is with Open Source Robotics Foundation.
E-mail: caguero@osrfoundation.org

José María Cañas is with Rey Juan Carlos University.
E-mail: jmplaza@gsyc.com

objects and do not lie completely into the camera same field of view. Some works use omni directional vision, and they have been successfully applied in problems like visual localization or soccer behaviors in RoboCup middle size league. Other approaches use a regular camera and an overt attention mechanism [3][13], which allows for rapid sampling of a several areas of interest. This is the only available approach in the RoboCup SPL humanoid league.

In this paper we describe an attention system for a humanoid robot endowed with a camera on its head, which can be oriented at will independently from the robot base. This algorithm combine several perceptive needs from behaviors and controls the camera movements in order to keep all of them satisfied, providing them with enough images along time to achieve its perceptive goal.

Following this introduction, the second section describes some related works. Our attention mechanism and the robot control architecture where it has been developed is presented. In the fifth section we describe the experiments carried out on a real humanoid robot to validate this work. Finally some brief conclusions end the paper.

### A. *Related work*

One of the concepts widely accepted in visual attention is the salience map. It 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 (as color, 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 behavior the decision on the positioning of the head. Second, those which continuously moved 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. A third group include those using a specific component responsible for making this decision based on the requirements of active behaviors. 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.

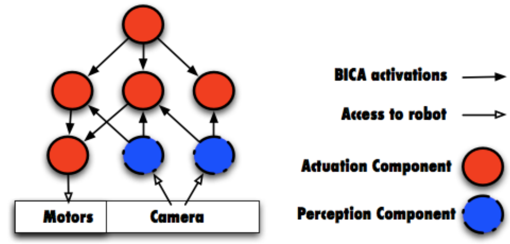


Fig. 1. Activation components tree in BICA architecture

In one SPL team [17] the behaviors define the importance of each stimulus. Depending on the importance defined for each stimulus and the value of its current associated uncertainty, the active vision system decides which stimulus to focus on at any time. The behaviors 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 modeled 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 focus at all times. This approach makes emphasis in 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 makes to decrease the entropy of system. In these approach behaviors do not define the importance of the stimuli and they do not modulate the active vision system in any way.

## II. Visual Attention Architecture

The software of our humanoid robot is organized with a behavior-based architecture. It is implemented in a component oriented software architecture, named Behavior-based Iterative Component Architecture (BICA) [21]. Components are independent computation units which periodically execute at a pre-configured frequency. Every component has an interface to modulate its execution and retrieve the result of the computation. Behaviors in BICA are defined by the activation of perception components and actuation components. Actuation components take movement decisions, send commands to the robot motors, or locomotion system, or activate other actuation components. They run iteratively to periodically update their outputs. Perception components take data from robot sensors or other perception components and extract information. They basically provide information to the actuation components. The output of a perception component is refreshed periodically and can be read from many other components in the system.

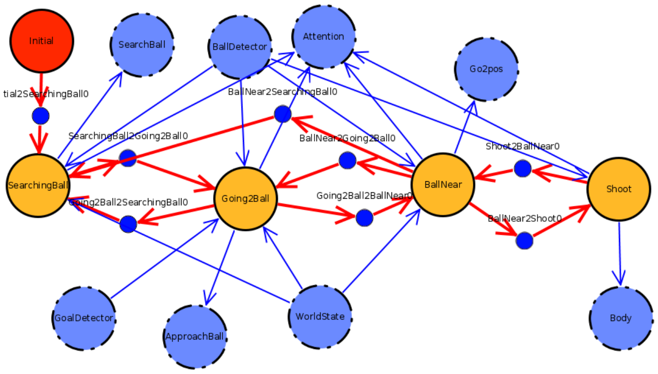


Fig. 2. Component functionality coded by a finite state machine (states are colored in orange). Each state activates different component sets (component activation in blue)

Not all the perception capabilities of the robot must be active at the same time, consuming computing resources. Even more, the whole set of behaviors that the robot is able to eventually perform is not suitable to deal with the current situation. Only a subset of behaviors and perception units are relevant to the current situation. In BICA each component is activable and deactivable at will, so it remains inactive until the situation demands it, when another component activates it. Typically an actuation component activates the perception components it requires and the child actuation components (if any) that implement its control decisions. This activation chain creates a dynamic component tree to cope with the robot's current situation. Figure 1 shows an component activation tree with both perception and actuation components. Some of these components can be encoded as finite state machines. In each state, the set of active components may be different (Figure 2).

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

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 3, are behaviors, detectors and the attention controller. The generic attention system operation is described as follow:

- Behaviors 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. Behavior are software components which runs iteratively activating another 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 no longer must be perceived an object. The interface between the controller behavior and attention is through the call setImportance(object, importance). For example, to perceive the object A with a 90% importance, a behavior make the call setImportance (”A”, 0.9).
- Detectors are software components specialized on detecting perceptive objects using the information from the camera (Figure 4). They are part of the execution in cascade initiated by high-level behaviors. Once detected an object in the image, 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 by using extra information like the Z coordinate position, which is 0 for the object in the floor. 3D position of the object is incorporated into a visual memory (represented in Figure 5) distributed among the detectors. Visual memory uses a set of extended kalman filter, 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: there are static and dynamic objects, with unique or multiple instances, those which are best detected on the horizon or that can be anywhere. When behaviors set the attention controller to perceive the object A, it requests (using `get)` 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 null value if it is not necessary to perceive it again (this is common for static and unique 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 request a point to look at, it returns the same point in the list. If the controller informs (using

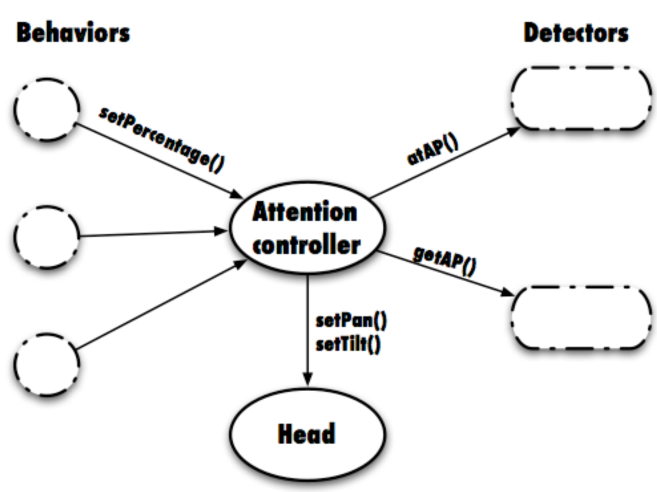


Fig. 3. Attention system architecture. The attention controller receives the importance values for each detector. It also asks to the detectors for 3D attention points and informs when a attention point is reached. Robot's neck angles are calculated from each 3D attention point.

atAP() method) that the point has been already reached. It replies the next point in the list, and so on.

- The controller of the attention is responsible for combining the perceptual requirements from behaviors 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 we introduced above, the interface with the detectors consists of two calls: getAP() and atAP(). The first requests from a detector a point 3D where to direct the camera, and the second informs that the attention point has been reached.

This description defines in general terms how is the attention system, but it keep open many questions: how the controller set the turns using on the importance of each object? How a detector defines which point in the search list is better 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.

The salience-based algorithm selects the active detector depending on the difference between the detection quality and the importance established for it by behaviors. The quality of the information of a detector is the average of the quality of the detector objects that it has to detect (the inverse of the object estimation uncertainty). If the item is unique, the detector quality depends on the uncertainty of the object only. 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 case of dynamic objects and slowly if they are static) and robot motion.

Figure 6 illustrates the mechanism of arbitration. Behaviors define the importance of perceiving the object A and B. These values are used as reference (qref(A) and qref(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 priority for the controller attention. In Figure 3, we can observe how the quality of the detector B decreases faster than the one of the detector A. This may be because the object that detects B is dynamic, and the object that detects A is static.

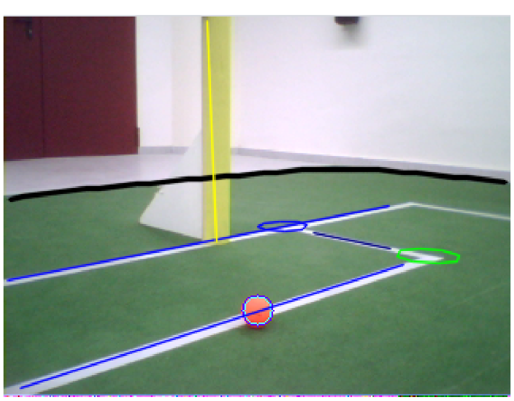

Fig. 4. Each detector detects a particular object.

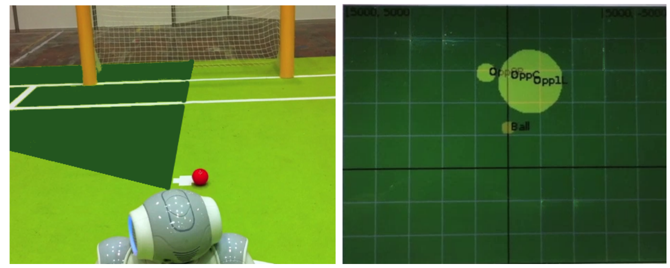

Fig. 5. A visual memory system stores the position and uncertainty of the detected elements.

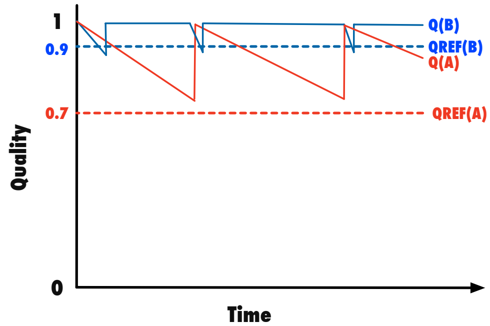


Fig. 6. Evolution of the quality for two different detectors.

Once presented how the system selects which object attends to, we define what to do to perceive a particular object. 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. Salience of the attention points covered by the field of vision of the camera 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 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 7 shows an example where three detectors are active (A, B and C). The attention points for each detector are represented with a different color and a shape, and each point has an identifier that indicates which detector belongs to, and in brackets the value of salience. The detector A is responsible for searching an object that is unique, while the detector C is responsible for detecting multiple objects in the same class. In the 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. The detector B has detected the desired object (represented by colored circles) in the image,

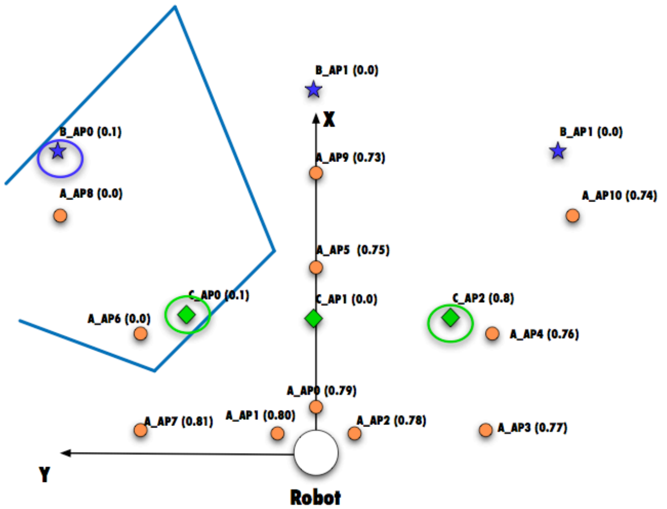


Fig. 7. Cenital view of the robot space. Attention points from different detectors are represented with different colors and shapes. The blue parallelogram represents the camera field of view. Circles represents the detection of the objects. Each attention point is labeled with an id and the salience value.

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 object that it is looking 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 object, when the 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.

## III. Experiments

The whole system described above is general purpose, and is not limited to any particular application. We have applied this system on two very different applications: Tracking multiple human faces in human-robot interaction, and in the RoboCup environment. The environment of the RoboCup (Figure 8) is a standard testbed where most of mobile robotics technologies can be evaluated and contrasted. It is a very dynamic challenging environment where teams of autonomous robots coordinate to play soccer in a 6x8 meters field where relevant items are identified by their shape and color. Robots must solve the classical problems (navigation, self-localization, perception, locomotion, ...) efficiently and robust to collisions, kidnappings, ...

We have carried out several experiments to validate the system described in this article. We have used the real NAO robot in the RoboCup environment. During the experiments, the robot is equipped with a visual pattern easily detected by the cameras, as we can see in Figure 9. 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.

We have compared the salience last algorithm with two other algorithms: Round Robin and Fixed Pattern:

- The Round Robin algorithm plans the attention giving each detector a time slot. The duration of this time slot is proportional to the importance defined by the behaviors. If the attention controller receives different importance values for the same detector, it chooses the maximum of the values received. In the slot assigned to a specified detector, it is asked for an attention point. 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 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 10 shows an example of this algorithm. Behaviors define the importance of each type of objects. In this Figure the behavior 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 sets 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.
- Fixed Pattern algorithm moves the camera using a list of attention points (Figure 11) shared by all the detectors. Attention system moves the camera slowly ( pi / 3 per second) visiting the attention points, regardless of whether the objects of the active detectors are perceived or not. If there is a detector critical for the robot operation, the last known position of the object is pushed back

Fig. 8. Complex behaviors in dynamic environments where the main sensor is a small field of view camera.

Fig. 9. Ground truth system based on cenital cameras and visual patterns.

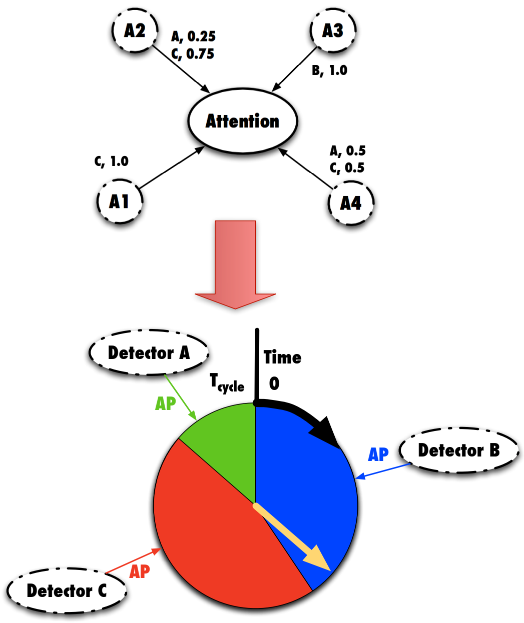


Fig. 10. Attention algorithm based on a Round Robin approach.

in the shared list of attention points.

During experimentation, active behaviors set equal importance to detect the ball and the two goalposts. To compare the efficacy of different algorithms, we measure the uncertainty of each object. When an object is not in the viewing area of the camera, its uncertainty increases depending on the time that has passed since that is not perceived and the robot movement (this movement has a high uncertainty). When an object is perceived in the image, its position is corrected and its uncertainty decreases.

In the first experiment the robot is stationary while the ball is moved manually between various points (Figure 12). This experiment tested how to manage loosing a moving object, and how to recover from this. In the second experiment, the robot moves toward the ball (Figure 13). This experiment verifies how the movement affects to attention objects uncertainty increases rapidly, and how the interaction with an object is managed.

Figures 14, 15 and 16 shows the result of one trial of the experiment 1 using the Round Robin, Fixed Pattern and Salience

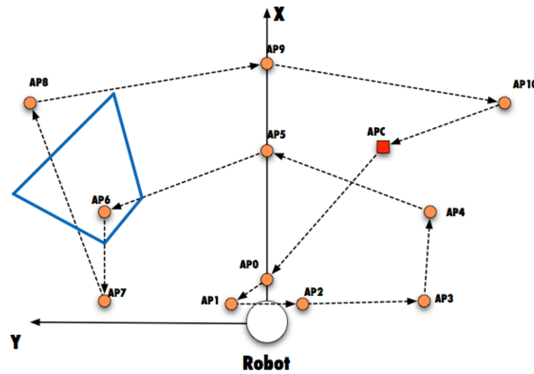


Fig. 11. Attention algorithm based on a Fixed Pattern approach.


Fig. 12. Experiment 1: Robot stopped and the ball is moving around three positions.

Fig. 13. Experiment 2: Robot is moving to the ball.

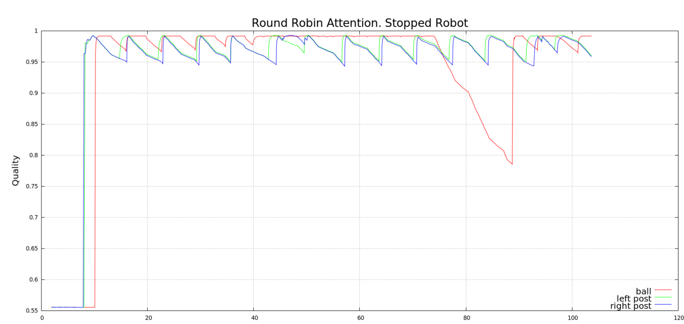


Fig. 14. Estimation quality evolution of three objects using the Round Robin attention algorithm while the robot is stopped and the ball is moving.

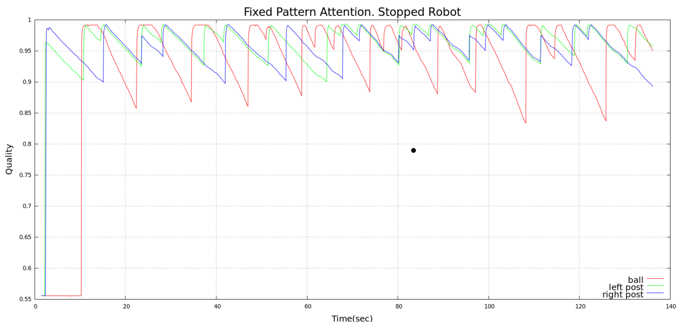


Fig. 15. Estimation quality evolution of three objects using the Fixed Pattern attention algorithm while the robot is stopped and the ball is moving.

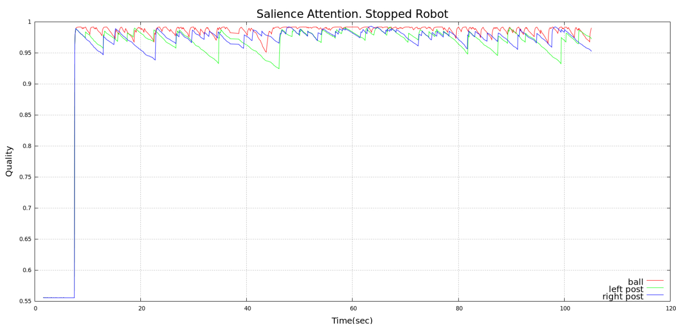


Fig. 16. Estimation quality evolution of three objects using the Salience attention algorithm while the robot is stopped and the ball is moving.

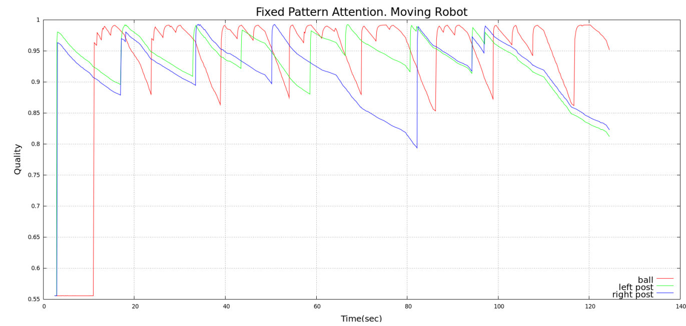


Fig. 18. Estimation quality evolution of three objects using the Fixed Pattern attention algorithm while the robot is moving to the stopped ball.

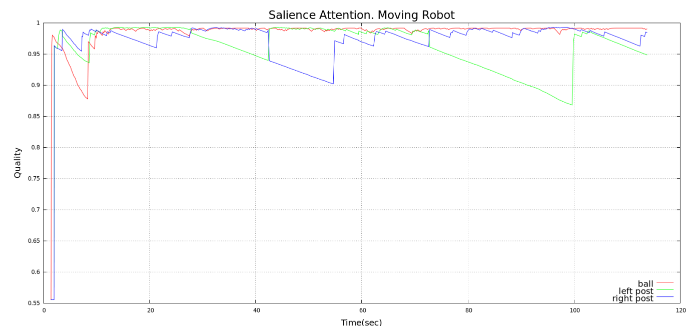


Fig. 19. Estimation quality evolution of three objects using the Salience attention algorithm while the robot is moving to the stopped ball.

algorithms, respectively. The estimation quality (inverse of uncertainty) is always high using the salience algorithm, which means that the objects are efficiently tracked. Other algorithms shows worst results, and even the ball is lost (and recovered) using Round Robin approach.

Figures 17, 18 and 19 shows the result of one trial of the experiment 2 using the Round Robin, Fixed Pattern and Salience algorithms, respectively. The estimation quality The estimation quality of goalposts decreases rapidly, but global results are again better in the Salience approach. The results of several trials are summarized in table 20.

## IV. Conclusions and further work

In this work we have presented an efficient attention mechanism for behavior-based architectures, 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 shifts 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 the ball, which is not always compatible with the importance of looking at the goals. This organization allows the independent development of different behaviors, because their perceptive requirements can be met regardless of other behaviors.

The RoboCup SPL environment has been selected as a validation scenario. The variety of objects to track and its different nature (stationary objects as the goal posts and moving objects as the ball or the robots) represents a good benchmark for the proposed approach. We have presented indications that the system works efficiently, and improves commonly used options to address this problem.

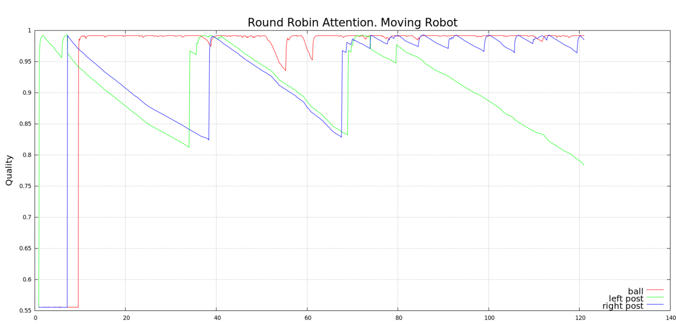


Fig. 17. Estimation quality evolution of three objects using the Round Robin attention algorithm while the robot is moving to the stopped ball.

| **Attention Algorithm** | | Ball | Left Post | Right Post |
|---|---|---|---|---|
| Round Robin | *Stop* | 0.9736 | 0.9763 | 0.9715 |
| | *Move* | 0.9887 | 0.9009 | 0.9386 |
| Fixed Pattern | *Stop* | 0.9471 | 0.9667 | 0.9561 |
| | *Move* | 0.9497 | 0.9372 | 0.9209 |
| Salience | *Stop* | 0.9871 | 0.9719 | 0.9758 |
| | *Move* | 0.9829 | 0.9519 | 0.9806 |

Fig. 20. Summary of the experiments.

We are extending this work in several directions. First, it is interesting to investigate a best combination of the different needs of active behaviors. Second, this algorithm is suitable to carry out a distributed attention system among multiple robots, which improves the search time of objects.

## 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, pp 205-231, 2005.

[4] J. K. Tsotsos et al, Modeling visual attention via selective tuning, Artificial Intelligence, Vol. 78, pp 507-545, 1995.

[5] L. Itti and C. Koch, *Computational modelling of Visual Attention*, Nature Reviews Neuroscience, Vol 2, pp 194-203, 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, J. Conradt, T. Shibata and S. Schaal, *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 and C. Koch, *A saliency-based search mechanism for overt and covert shifts of visual attention*, Vision Research, Vol. 40, pp 1489-1506, 2000.

[9] C. Soyer, H. Bozma and H. Istefanopolus, *APES: attentively perceiving robot*, Autonomous Robots, Vol 20, pp61-80, 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, pp77-123, 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. WoWorkshop 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. Rofer. *Entropy-based active vision for a humanoid soccer robot*. In J. Ruiz del Solar E. Chown and P.G. Ploeger, editors, RoboCup 2010: Robot Soccer World Cup XIV, number 6556 in Lecture Notes in Artificial Intelligence, pages 112. 2010.

[19] S. Czarnetzki, S. Kerner, and M. Kruse. *Real-time active vision by entropy minimization applied to localization*. In Javier Ruiz del Solar, Eric Chown, and Paul-Gerhard Plger, editors, RobuCup, volume 6556 of Lecture Notes in Computer Science, pages 266277. Springer, 2010.

[20] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, *RoboCup : The robot world cup initiative*. In ICJAI-95 ? Workshop on Entertainment and AI/ALIFE, 1995

[21] F. Martin, C. E. Agero, J. M. Caas and E. Perdices, *Humanoid soccer player design*. In Robot Soccer, pp 67–100. InTech 2010.

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