Multi-modal Active Visual Perception System for SPL Player Humanoid Robot

Francisco Martín, Carlos E. Agüero, José M. Cañas, and Eduardo Perdices

Robotics Lab (GSyC), Rey Juan Carlos University,
Fuenlabrada, Madrid 28923, Spain
{fmartin,caguero,jmplaza,eperdices}@gsyc.es

Abstract. Robots detect and keep track of relevant objects in their environment to accomplish some tasks. Many of them are equipped with mobile cameras as the main sensors, process the images and maintain an internal representation of the detected objects. We propose a novel active visual memory that moves the camera to detect objects in robot's surroundings and tracks their positions. This visual memory is based on a combination of multi-modal filters that efficiently integrates partial information. The visual attention subsystem is distributed among the software components in charge of detecting relevant objects. We demonstrate the efficiency and robustness of this perception system in a real humanoid robot participating in the RoboCup SPL competition.

Keywords: robot soccer, active vision, multitarget tracker, humanoid, attention.

1 Introduction

Several international robot competitions and challenges have emerged in the last years. For instance the DARPA Grand Challenge and Urban Challenge[6], RoboCup[10], the FIRA Robot World Cup[13], and the DARPA Robotics Challenge. They aim to foster AI and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. The motivation of competition makes them attractive, and they serve as proof of concept of current technological limits, pushing them further. In particular, RoboCup chose soccer as a central topic, which 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. It has the long term goal to develop a team of fully autonomous humanoid robots that can beat the human world champion team by 2050.

The RoboCup competition is organized into leagues according to the type of robots. In the Standard Platform League (SPL) [7] all teams use the same robot, and changes in hardware are not allowed, so the effort focuses on software. A fully featured SPL soccer player has to get information from the environment, mainly using its on board camera. It provides high volume data about the environment, and task relevant information must be extracted from them. Robot

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_40,

control decisions may be based exclusively on current image, but this suffers from some limitations. 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-localise and plan the next action: move, kick, search 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.

This paper presents the visual perception system of our humanoid robots participating in the SPL (SPiTeam[21]), composed by a visual memory and an integrated attention subsystem. The aim is to improve perception in order to make good decisions and so unfold good behaviors. The visual memory is composed of a collection of Joint Probabilistic Data Association Filters (JPDAF) [1] [4] that store and update the relative position of objects, like goals or the ball, relative to the robot. This approach has been designed to cope with partial and ambiguous observations. Partial observations occur when an object is occluded or when the camera field of view is limited. The observations can also be ambiguous, if the perceived feature is not unique due to symmetries or similarities with other objects. We also propose a novel attention subsystem that controls the head movement and continually shifts the focus of attention so the camera looks at different areas of the scene providing new images to feed the visual memory and to update the object state estimates. This visual perception system has been integrated in the behavior-based architecture of the humanoid robot, named BICA [11].

Next, we review the state of the art in world modeling, attention and multi-object estimation, with emphasis on the other teams of the RoboCup SPL. The two main components of the proposed perceptive system, the visual memory and the visual attention module, are described in detail in sections 3 and 4, respectively. The analysis of the experiments conducted is detailed in section 5, while discussion generated by this work is presented in section 6.

2 Related Works

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. The most commonly used filters are Kalman filter[22], its nonlinear variants Extended Kalman filter (EKF) or unscented Kalman filter (UKF)[20], particle filters[9], hybrid techniques, or even multi-modal algorithms[19].

A topic closely related to visual memory is the control of camera gaze. In the RoboCup environment, 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 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 a priori there is no object. A third group includes those using a specific component responsible for making this decision based on the requirements of active behaviors. There are attention mechanisms guided by utility functions based on the task the robot is currently doing[8], or salience-based schemes which increase with time[16] or time-sharing mechanisms, among others.

One SPL team [16] associates a tuple $< \rho, \theta, anchor >$ with each stimulus that the robot can detect. The $\rho[0, max\_dist]$ and $\theta[-\pi, \pi]$ indicate the relative distance and orientation to the stimulus, respectively. The $anchor[0, 1]$ indicates the confidence in the information of the stimulus. In this system, the behaviors define the importance of each stimulus. Depending on the importance defined for each stimulus and the values for *anchor*, 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. In this approach that only one of the objects receives the feature's update, while the other objects remain unchanged. This association is carried out according to a specific function, like Euclidean distance. In this work the search for new objects uses the same fixed pattern for head positions, independently of the type of the object to search.

One interesting approach [2] shares information among all the robots of a team. Each stimulus is labeled as valid, suspicious and invalid, depending on the confidence on the stimulus. This label is set to valid when an object is detected in the image of the robot, or suspicious if this information comes from a teammate. The stimuli which are labeled as valid do not need to be attended urgently. Those labeled as suspicious have to be inspected to check whether they are still valid. There are behaviors that define the importance of a stimulus. All stimuli are attended at all times. Regarding the visual memory, a mixed approach is used to estimate the goal positions. The estimate of the goal extracted from the self-localization system is mixed with the one captured in the last scan.

The approach proposed in [14] uses a multi-modal algorithm to estimate the positions of other robots. The observations are compared with the positions maintained by the world model using the Euclidean distance criterion. A similar approach was used in [15] but using the Mahalanobis distance between observations and objects in memory. The disadvantage of these algorithms is their full association between the observation and one of the objects.

In [18] and [5], the state of the robot is modeled using Monte Carlo Localization (MCL). The state includes both the robot's position and the ball's position. The aim of the active vision system is to minimize the entropy of the robot's state. Here the active vision is 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 system. Behaviors do not define the importance of the stimuli and do not modulate the active vision system in any way.

3 Multi-object Visual Memory

In BICA the humanoid robot intelligence is decomposed in perceptive components, which provide information and behavior components, that make control decisions taking into account such information. Some behavior components for the SPL are `SearchBall`, `GoToPoint`, and several shooting movements. Some perception components are `BallDetector`, `GoalDetector` and `LineDetector` which detect the relevant stimuli in the RoboCup scenario. These components are responsible for detecting the stimulus in the image, calculating its 3D position, updating the visual memory with this position, and providing information to the attention system.

Beyond the instantaneous detection in current image a visual memory is built and updated. The visual memory is formed by the composition of various independent JPDAF filters running in parallel, one for each object to be tracked. In order to avoid the additional uncertainty the self-localization algorithms and to be fast, the visual memory records the position and uncertainty of the objects in a coordinate system relative to the robot. For each object the last time it received a new observation is also stored. For the SPL humnanoid robot there are three JPDAF filters running in parallel: one for the ball, one of the team's goal, and another for the opponent's goal.

All filters are updated regularly based on the odometry generated by the robot and the detected features from perceptual components. In the initialization phase of each filter it is possible to set some features like the object motion model. For the ball, as can be kicked by other robots, the uncertainty of its estimate must increases if the robot stays still. As time passes without generating new observations, it becomes more likely that the ball changes its position. However, the goals will not move, so it makes sense not to increase their uncertainty if the robot remains motionless.

The general working environment for the JPDAF algorithms consists of several objects and several observations. These objects may have the same appearance, and therefore cannot be assigned to each estimator straightforward. The problem is to find out how we can associate each observation to each object. Using the notation of [17], we have a set of objects $X^k = x_1^k, \ldots, x_T^k$ at time k. The set of observations at that instant k is defined as $Z(k) = z_1(k), \ldots, z_{m_k}$. The entire set of possible associations between an observation j and object i, is defined as a joint association event θ to a specific set containing couples $(j, i) \in \{0, \ldots, m_k\} \times \{1, \ldots, T\}$. The set θ uniquely associates each observation with each object. The special empty observation $z_0(k)$ means that the object has not been perceived. The term Θ_{ji} expresses the set of all joint association events that pair j with the object i.

The core of the JPDAF algorithm is based on calculating the parameter β_j, whose mission is to measure the likelihood that the observation j belongs to object i. Equation (1) shows how to perform the calculation of the posterior probability.

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} P(\theta|Z^k) \tag{1}$$

The probability $P(\theta|Z^k)$ can be expanded according to equation 2 assuming the problem is Markovian and using the theorem of total probability. The full derivation of this formulation can be found in [17]. $P(\theta|X^k)$ evaluates the probability of a specific θ set given the current state of the tracked objects.

$$P(\theta|Z^k) = \alpha \int P(Z(k)|\theta, X^k) P(\theta|X^k) p(X^k|Z^{k-1}) dX^k \tag{2}$$

$P(Z(k)|\theta, X^k)$ of equation (2) specifies how likely the set of observations obtained with the current state of objects and a particular set of observation-object associations is. To derive this term, it is necessary to consider the case of a false positive. We call γ the probability that an observation is a false positive and the number of false positives on a θ is expressed as $(m_k - |\theta|)$. In turn, the probability associated with all false positives in at time k and a specific θ is $\gamma^{(m_k - |\theta|)}$.

For $P(\theta|X^k)$ it is assumed that all sets of pairs of associations are equally likely and, therefore, this term can be approached by a constant (as can be seen in [3]).

Taking into account the derivation of the previous two terms and assuming independence between observations, we have:

$$P(Z(k)|\theta, X^k) = \gamma^{(m_k - |\theta|)} \prod_{(ji) \in \theta} \int p(z_j(k)|x_i^k) p(x_i^k|Z^{k-1}) dx_i^k \tag{3}$$

Combining equations (2), (3) and (1) we get:

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} \left[\alpha \gamma^{(m_k - |\theta|)} \prod_{(ji) \in \theta} \int p(z_j(k)|x_i^k) p(x_i^k|Z^{k-1}) dx_i^k \right] \tag{4}$$

Once we know how to compute β_{ji} to weight each observation with each object, all we need is to describe how to update the Kalman filters to estimate each object. The prediction phase is performed by equation (5) and the correction phase is described by the equation (6).

$$p(x_i^k|Z^{k-1}) = \int p(x_i^k|x_i^{k-1}, t) p(x_i^{k-1}|Z^{k-1}) dx_i^{k-1} \tag{5}$$

$$p(x_i^k|Z^k) = \alpha p(Z(k)|x_i^k) p(x_i^k|Z^{k-1}) \tag{6}$$

This is where we introduce the factor β_{ji}, integrating over all the observations obtained and its association with the object i.

$$p(x_i^k|Z^k) = \alpha \sum_{j=0}^{m_k} \beta_{ji} p(z_j(k)|x_i^k) p(x_i^k|Z^{k-1}) \tag{7}$$

3.1 Ball Object

The ball is one of the objects stored in visual memory. Although there is only one ball, in practice and due to calibration problems or lighting, it is possible to detect several balls in the same frame in addition to the correct ball. The JPDAF ball algorithm is configured as a single object, i.e., you can not create or destroy new estimates, only one remains. This causes the correct observation to be weighted with a high value of β. The comparison between the correct observation and false positives greatly benefits the correct observation and, therefore, the rest have less influence on the estimate.

3.2 Goal Objects

Unlike the ball filter, the JPDAFs for each goal are set to hold two independent objects simultaneously, one for each post. Although the tracking algorithm maintains an independent estimation for each post, there is a restriction that can be applied and improves the accuracy of the estimate. This constraint imposes the posts are always at the same distance from each other, in our case 1400mm.

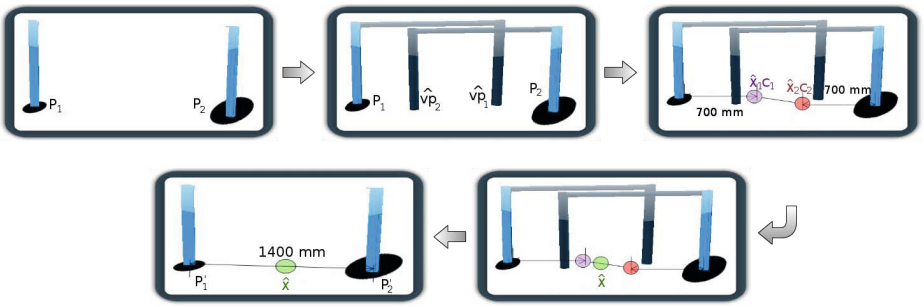

Fig. 1. Diagram of the algorithm for the calculation of goal's center

The idea of the algorithm is to assume that each post is correctly estimated and therefore we can infer the opposite virtual post position $v\hat{p}_i$. Along with the position of each post and the opposite ones virtually generated, two hypotheses with centers of the goals at $\hat{X}_i$ are created. The end point where you estimate the center of the goal would be in a straight line connecting these two hypotheses. Although the midpoint of the two hypotheses could be a first approximation to the solution, it does not take into account that the original estimations for each post can have different uncertainties. Therefore, it is preferable to weight the uncertainty as each of the hypotheses. Equation 8 calculates the desired point $\hat{X}$ assuming $\hat{X}_i$ is the position of the hypothesis i and its covariance associated with C_i. Finally, the original estimations of each post P_i are adjusted according to the calculated midpoint and the restriction of known distance from the midpoint of each post obtaining P'_i. Algorithm 1 summarizes the steps for calculating the

Algorithm 1. Optimization to improve the calculation of the goal center

1: Update posts estimations P_i using JPDAF algorithm
2: **for** $i \leftarrow 1, 2$ **do**
3: $\hat{vp}_i$ = Generate the virtual opposite post
4: $\hat{X}_i$ Calculate the hypothesis between P_i and the vp_i
5: **end for**
6: **for** $i \leftarrow 1, 2$ **do**
7: $\hat{X}$ = Obtain the corrected center applying weighting the two Gaussians $(\hat{X}_1, C1))$ and $(\hat{X}_2, C2))$
8: Using $\hat{X}$ as the center of the net, generate P_i' according with the width constraint
9: **end for**

central point of the goal. Furthermore, Figure 1 shows an example of the whole adjustment process made to the position of each object.

$$\hat{X} = \hat{X}_1 + C_1[C_1 + C_2]^{-}1(\hat{X}_2 - \hat{X}_1) \tag{8}$$

4 Visual Attention

Robots equipped with cameras have a limited field of view. Visual stimuli may not all be present simultaneously in the image perceived by the camera of a robot. For this reason, the robot has to search in the scene by varying the orientation of the camera. The visual stimuli are incorporated into the visual memory, as described in section 3, and they should be checked periodically to update their position. The visual attention system is responsible for performing the scanning for visual stimuli in a scene, and it verifies and updates the already collected stimuli.

Some actuation components have perceptual requirements, i.e., a set of visual stimuli to be aware of. These perceptual requirements are met by activating the perception components responsible for detecting each one of these stimuli and by setting the importance, in the range (0.1], for each visual stimulus.

Our visual attention system receives attention requests from each of the components activated with interest in some objects. Each of these components have different requirements that may conflict with each other, so the visual attention component acts as a referee assigning control of the attention fairly among all visual stimuli. Figure 2 (left) shows how some actuation components send their attention requirements about the visual stimuli in the scene. For each visual stimulus received, the visual attention attention system chooses the highest value received and then it normalizes this value by the sum of all importances. At the left side of figure 2 (left), the value of importance for the goals is 0.5 $(max(0.5, 0.25))$, for the ball is 1.0 $(max(0.75, 1.0, 0.5))$ and 1.0 for the lines.

If an actuation component sets a need for a visual stimulus, and then becomes inactive, the visual attention system must adapt to this new situation. The components are iterative, and their mechanism of deactivation is simply stop running

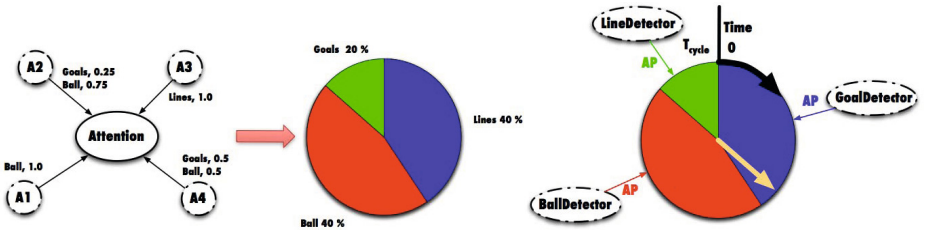

Fig. 2. Example of attention Setup. left: Actuation components (A1, A2, A3 and A4) send their perception requirements to the Attention component, which calculates the importance relationship among the object. Right: The attention system receives desired points to focus the camera. When it is the ball's turn, the attention system asks the component in charge of detecting the ball (red), and when it is the goal's turn, the attention system asks to the component in charge of detecting the goal (blue).

them silently. Because of this, each component in each iteration should resend their perceptual requirements. The visual attention system discards those stimuli that are not refreshed frequently, recalculating the importance relationship among visual stimuli.

The visual attention system implements a time-sharing policy, a time slot is assigned to each visual stimuli. This time slot is dedicated to seeking and tracking the corresponding stimulus. Figure 2 (right) shows how the turns are managed. On each turn, the visual attention system asks the component responsible for the stimulus what to do. The answers come in the form of attention points (AP). These attention points are three dimensions coordinates where the camera should focus. The only feedback sent to each detector is when the camera is pointing at the requested attention point.

It is remarkable that the visual attention system does not decide where to focus the camera, how to seek or track an object or when it is considered that a stimulus has been found or when it is lost. The novelty of this system is that these tasks are delegated to the corresponding perceptive component (detector):

- The attention component notifies to the detector components when the camera is focusing to the desired attention point. If the object is not at the position where it should be, the detector is responsible to guess if it is lost, using the last detection time. Some detectors could me more robust to occlusions, depending on the object characteristics.
- If an object is considered lost, or its position is not known, the detector generates attention points sequentially in the positions where the object can be found. Whenever a requested position is reached, the attention system reports this event, and the detector generates the next attention point. It is important that the focal points generated depend on which object you are looking for: the ball can only be at a point on the ground, but it is more effective to find the goals at the points on the horizon. Actually, it is

even possible to use the self-location information to decide where to generate search points. In any case, the selection of these attention points is done by each detector, which is specialized to look for the object.
– When an object is detected and the current slot corresponds to its detector, decisions may be different. In the case of the ball, being a very dynamic object, the remaining time slot can be dedicated to track the object.

More details and a comparison between several developed attention mechanisms can be found at [12].

5 Experiments

An extensive experimentation has been carried out to validate the system described in this article. We have used the real NAO robot in the real SPL environment as shown in Figure 3. The Nao robot is a medium-sized humanoid robot with 58 cm. of height, with 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.

Fig. 3. Experimental setup. The environment is the real SSL field and the robot is equipped with a pattern in order to be detected by a ground truth system.

During the experiments, we collected a great amount of data used to analyze the visual attention subsystem and the visual memory. The data collected during the execution of the experiments is stored in a log file for offline processing. Meanwhile, we have adapted the SSL [23] ground-truth system that captures the correct position of the dynamic elements of the field (robots and ball). This system is composed of two ceiling cameras mounted above the center of each half field. During the experiments, the robot is equipped with a visual pattern easily detected by the cameras. 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 in the perception of the robot.

Visual Memory Accuracy Experiments. The first experiment makes a statistical analysis of the accuracy obtained in estimating the current stimuli in the visual memory. The stimuli analyzed are the ball and one of the goals. For each stimulus, we calculate the error (measured as the Euclidean distance between the real and the estimated position) and the standard deviation of the estimate over time. There have been two different tests, one with a static robot and a second with the robot in motion. The attention settings for each test are labeled in the key of the graph. Figure 4 shows the data extracted from the experiment, where data analyzed from the static robot are located in the upper image and the data extracted from the robot in motion are displayed in the lower image. Besides the error in the estimate and standard deviation, Figure 4 also shows on a horizontal bar the intervals where each object of attention has the attention turn.

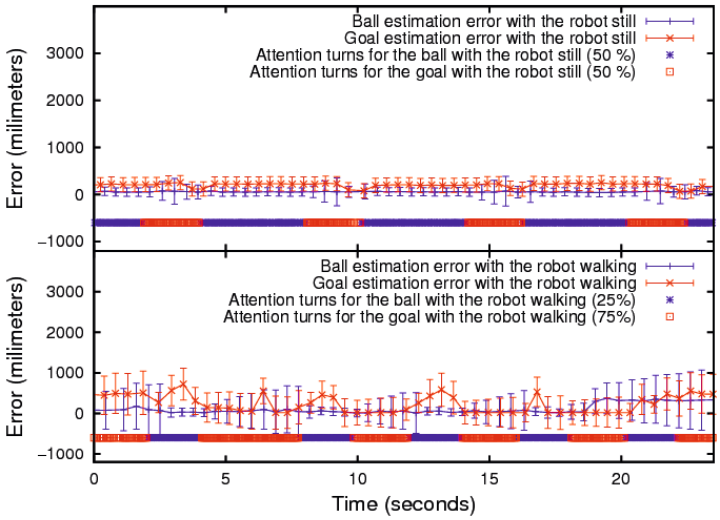

Fig. 4. Accuracy and standard deviation of the visual memory estimation

One interesting result is how the standard deviations vary while the robot remains static (Figure 4 upper). In the case of the goal (displayed in red), the uncertainty remains constant even when there is lack of attention. However, in the ball case (displayed in blue), the standard deviations increase during intervals in which no features are received from the ball. When the robot moves (Figure 4 lower), the dynamic of the stimulus is automatically configured with different parameters and this affects how the uncertainty grows when the head does not look at the goal and, consequently, no features are received. If we pay attention at the lower blue curve on Figure 4, we notice that the standard deviations are larger and grow faster when there are no features perceived, compared with the static robot experiment.

Attention Experiments. The next experiment analyzes how the visual attention system simultaneously detects, confirms and tracks two different elements: the ball and goal. In some tests that are part of this experiment, the robot is stationary, and in others the robot is moving. Perceptual configurations are different, giving in some cases more attention to the ball, and in some cases greater attention to the goal.

As described in Section 4, attention is distributed in the stimuli detectors. The search is different in the case of the ball (it looks at the ground) and the goal (it searches the horizon line). Likewise, the ball tracking is different in the case of the ball (all the time available is used to track and update the ball position) and the goal (when the goal is detected and its position is updated, it yields the rest of this time). Finally, the ball is a manipulable element, so its uncertainty increases over time as the robot moves. Instead, the goal is static, and its uncertainty only grows when the robot moves.

Figure 5 (right) shows the evolution of the uncertainty (standard deviation of estimate) in the detection of the ball and the goal when the robot is stopped. In further experiments we will analyze the accuracy of the system, but in this case we want to show how the attention system keeps low uncertainty of the estimates. This graph shows how in the configurations with the ball importance set to 100%, the average uncertainty remains very low, as the robot performs a constant tracking of the ball, without paying attention to other stimuli. The standard deviation of ball estimate does not increase too much as the goal importance increases. In the case of the goal, the uncertainty is high when the robot is not actively looking for this element, decreasing as the goal detector increases its importance.

Figure 5 (left)shows how the uncertainty is much greater when the robot walks. When the importance of the ball is less than 50%, the uncertainty increases to 750mm. The goal uncertainty, however, remains adequate when its importance is greater than 50%.

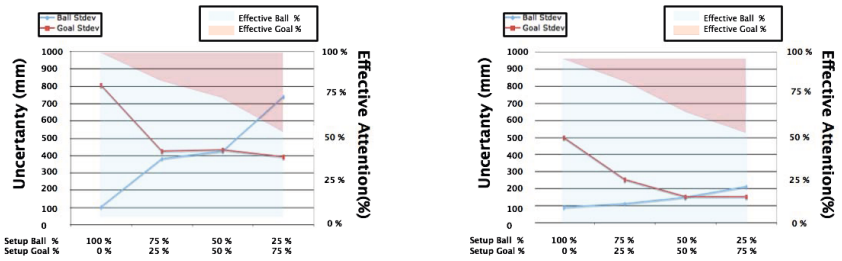

Fig. 5. Attention experiment with the robot **walking** (left) and the **static** robot (right). The graph shows the standard deviation of the ball and goal estimates, depending on the attention modulation. The colored area shows the actual attention distribution.

Figure 5 show (colored areas) the real value of time spent on each item. The percentage of time spent on the ball is always greater than on the goal due to its internal implementation. The ball always take the most of all the available time, while the goal detector yields the extra time.

Robustness Experiments. The next set of experiments measures the performance of visual memory under undesirable situations, such as false positives or false negatives observations. During the tests the robot has remained static. The first experiment consisted in estimating the position of the ball in the presence of false positives. A second ball located inside the robot's field of view was used to generate false positives. In the second experiment we used a single ball that was periodically occluded. The third experiment evaluates the behavior of the goal estimator against false positives. We used an extra goal post, which was perceived by the robot's camera during certain frames. Finally, a fourth experiment analyzes the behavior of the estimator of goals to false negatives caused artificially occluding one of the goal posts.

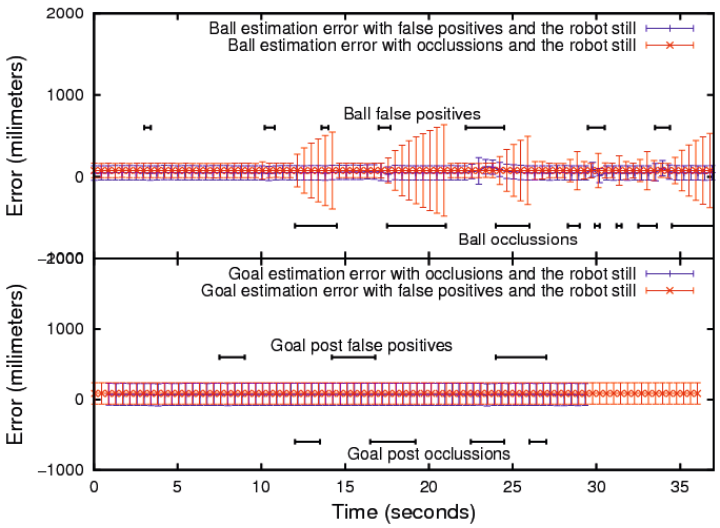

Fig. 6. Robustness against false positives and false negatives for the ball and goal

Figure 6 shows a summary of data extracted from the experiments. In Figure 6 (upper) it is displayed the error of the ball estimator under false positives and false negatives. In Figure 6 (lower) we show the same information for the goal. In both graphs we have labeled the time intervals for each experiment in which false positives and false negatives appeared.

The graphs show that the estimators have high robustness, due to low error variation when the false positives or false negatives arise. A rapid increase in ball

position uncertainty is noticeable during ball occlusions due to the movement mode associated with the ball.

Eficiency Experiments. The next experiment measures the visual memory and the attention system CPU consumption. In this experiment we run the main behavior for playing soccer (`Striker` component). Moreover, `Striker` executes the attention (`Attention` component) and visual memory behaviors (`VisualMemoryBall` and `VisualMemoryGoal`), along with several other components that analyze images (`Perception`), self-localized the robot itself (`Localization`), among others. During the experiment a log file stores information about the average execution time of each component. Thus, we can see the average execution time of the visual memory and attention components and we can compare their values with the time required by other components. Figure 7 shows the results obtained.

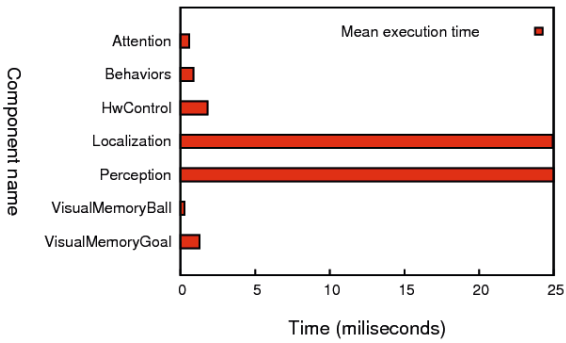

Fig. 7. CPU consumption of the `Attention`, `VisualMemoryBall` and `VisualMemoryGoal` components compared with the rest of components running on the robot

The average execution time of the component that implements the visual memory for the ball is 0.21 ms. In turn, the visual memory required for the goal is 1.5 ms due to greater number of features received compares with the ball. In turn, the attention component consumes 0.47 ms. to execute one of its iterations. Looking at Figure 7 we can see how the results are comparable to the fastest components contained in our architecture and thus they are appropriate to ensure real time execution.

6 Discussion and Future Lines

In this work we have presented the visual perception system developed for a humanoid robot participating in the Standard Platform League of RoboCup.

This system is composed of a visual memory and an attention subsystem. The visual memory has been designed to store the relative position of the objects, like goals or the ball, around the robot. We use several multi-modal JPDAFs to update the object features from the camera images. The images coming from the robot camera fed this visual memory. The attention mechanism moves the robot's head in order to visually explore the environment for new objects, to reobserve the existing ones or to cover all robot surroundings. Following a time sharing approach this attention component combines several perceptive needs of the soccer application like seeing the ball or seeing the goals to self-localize. A time slot is reserved for each relevant stimuli. Inside it, the head control is distributed among the active perception components for detecting and tracking the different stimuli.

Advantages of this visual memory system include time persistence and better integration of partial information, due to a multi-modal algorithm. It has broader scope than the single current camera image and the object estimations are more reliable than instantaneous estimations for occlusions, false negatives and false positives which usually appear in the images. The visual memory provides more and more reliable information about the robot surroundings than the current camera image alone. Taking both into account, memory and current image, the robot can take better behavior decisions.

The advantages of the attention subsystem include the convenient combination of perception requirements, usually contradictory, and the delegation of control commands to the components specialized in each object. The perception requirements are combined in a fair time sharing distribution among different stimuli. This solves the need of looking at the ball and looking at the goals from time to time. This organization allows the independent development of different behaviors, because their perceptive requirements can be met regardless other behaviors, so there is no need to consider interferences at this level.

The experimentation confirms the advances provided by this work. The visual memory system keeps object estimates robust to errors in perception, such as false positives or false negatives. The different tests carried out on the visual attention subsystem have shown how perceptual behavior requirements are met. During the experiments, we have shown how this system shares the turns between stimuli and their uncertainty is kept low. In addition to laboratory experimentation, this system was also used in the RoboCup-2011 by the SPiTeam during real games. It is difficult to provide quantitative results of the participation. Instead, from a qualitative point of view, the results of the perceptual system were very satisfactory, since the software operated as expected.

We are extending this work in several directions. First of all, an occlusion detection mechanism might be developed. When a new object has been detected, we could check whether it is lined up with another object estimated before. In the positive case, we could modulate the growth of uncertainty of the occluded object in a much slower way, probably because the object remains in its original position but it has been occluded by the new object. Other future directions to extend the system is the introduction of other robots (teammates or opponents)

into the visual memory and the use of sensor information from other teammates to update the robot visual memory.

Acknowledgements. This research has been partially sponsored by Community of Madrid through the RoboCity2030 project (S2009/DPI-1559). The authors also would like to thank all the members of the URJC Robotics Group, SPiTeam and CORAL Robot Lab who have collaborated in this work.

References

1. Bar-Shalom, Y., Fortmann, T.E.: Tracking and data association. Mathematics in Science and Engineering, vol. 179. Academic Press Professional, Inc., San Diego (1987)
2. Coltin, B., Liemhetcharat, S., Mericli, C., Tay, J., Veloso, M.: Multi-humanoid world modeling in standard platform robot soccer. In: Proceedings of 2010 IEEE-RAS International Conference on Humanoid Robots, Nashville, TN, USA, December 6-8 (2010)
3. Cox, I.J., Hingorani, S.L.: An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(2), 138–150 (1996)
4. Cox, I.J.: A review of statistical data association techniques for motion correspondence. International Journal of Computer Vision 10, 53–66 (1993)
5. Czarnetzki, S., Kerner, S., Kruse, M.: Real-time active vision by entropy minimization applied to localization. In: Ruiz-del-Solar, J. (ed.) RoboCup 2010. LNCS (LNAI), vol. 6556, pp. 266–277. Springer, Heidelberg (2010)
6. DARPA. DARPA Urban Challenge, `http://archive.darpa.mil/grandchallenge/`
7. The Robocup Federation. Robocup Standard Platform League (2010), `http://www.tzi.de/4legged/bin/view/Website/WebHome`
8. Guerrero, P., Ruiz-del-Solar, J., Romero, M.: Explicitly Task Oriented Probabilistic Active Vision for a Mobile Robot. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS (LNAI), vol. 5399, pp. 85–96. Springer, Heidelberg (2009)
9. Isard, M., Blake, A.: Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In: Proc. 5th European Conf. Computer Vision, vol. 1, pp. 893–908 (1998)
10. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E.: Robocup: The robot world cup initiative. In: ICJAI 1995 - Workshop on Entertainment and AI/ALIFE (1995)
11. Martín, F., Agüero, C., Cañas, J.M., Perdices, E.: Humanoid soccer player design. In: Papic, V. (ed.) Robot Soccer, pp. 67–100. INTECH (2010)
12. Martín, F., Rubio, L., Agüero, C., Cañas, J.M.: Effective visual attention for behavior-based robotic applications. In: Proceedings of the 2013 Workshop on Agentes Físicos (2013)
13. Federation of International Robot-soccer Association FIRA (2011), `http://www.fira.net`
14. Ratter, A., Hengst, B., Hall, B., White, B., Vance, B., Sammut, C., Claridge, D., Nguyen, H., Ashar, J., Pagnucco, M., Robinson, S., Zhu, Y.: rUNSWift Team Report 2010. Technical report, University of New Wales (2010)

15. Röfer, T., Laue, T., Müller, J., Fabisch, A., Feldpausch, F., Gillmann, K., Graf, C., de Haas, T.J., Härtl, A., Humann, A., Honsel, D., Kastner, P., Kastner, T., Könemann, C., Markowsky, B., Riemann, O.J.L., Wenk, F.: B-human team report and code release (2011), http://www.b-human.de/downloads/bhuman11_coderelease.pdf
16. Saffiotti, A., LeBlanc, K.: Active perceptual anchoring of robot behavior in a dynamic environment. In: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), San Francisco, CA, pp. 3796–3802 (2000), http://www.aass.oru.se/~asaffio/
17. Schulz, D., Burgard, W., Fox, D., Cremers, A.B.: People tracking with mobile robots using sample-based joint probabilistic data association filters. I. J. Robotic Res. 22(2), 99–116 (2003)
18. Seekircher, A., Laue, T., Röfer, T.: Entropy-based active vision for a humanoid soccer robot. In: Ruiz-del-Solar, J., Chown, E., Plöger, P.G. (eds.) RoboCup 2010. LNCS (LNAI), vol. 6556, pp. 1–12. Springer, Heidelberg (2010)
19. Stone, L.D., Corwin, T.L., Barlow, C.A.: Bayesian Multiple Target Tracking, 1st edn. Artech House, Inc., Norwood (1999)
20. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). MIT Press (September 2005)
21. Universidad de León y Universidad Rovira i Virgili Universidad Rey Juan Carlos, SPITeam's Application for qualification to RoboCup-2011 (2011)
22. Welch, G., Bishop, G.: An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill (2004)
23. Zickler, S., Laue, T., Birbach, O., Wongphati, M., Veloso, M.: SSL-Vision: The Shared Vision System for the RoboCup Small Size League. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS, vol. 5949, pp. 425–436. Springer, Heidelberg (2010)