Attentive visual memory for robot navigation

Julio Vega, Jose María Cañas, Eduardo Perdices

***Abstract*—Vision devices are today one of the most often used sensory elements in autonomous robots. Their main difficulty is to extract useful information from the captured images and the small visual field of regular cameras. Visual attention systems and active vision may help to overcome them. This work proposes a dynamic visual memory to store the information gathered from a continuously moving camera onboard the robot and an attention system to choose where to look at with such mobile camera. The visual memory is a collection of relevant task-oriented objects and 3D segments, and its scope is wider than instantaneous field of view of the camera. The attention system takes into account the need to reobserve objects in the visual memory, explore new areas and test cognitive hypothesis about object existence in the robot surroundings. The system has been programmed and validated in a real Pioneer robot that uses the information in the visual memory for navigation tasks.**

***Index Terms*—Visual attention, object recognition and tracking, active vision, camera model, autonomous navigation.**

I. INTRODUCTION

COMPUTER vision is being the most successful sensing modality used in mobile robotics, it would seem to be the most promising robotic sensor for the long term. While international computer vision research is currently growing rapidly, attention is moving away from its use in robotics towards a multitude of other applications, from face recognition and surveillance systems for security purposes to the automatic acquisition of 3D models for Virtual Reality displays. This is due to the greater demand served by these applications and the successes being enjoyed in their implementation; they are certainly the immediate future of computer vision, and some hugely impressive systems have been developed. However, we feel that it is well worth continuing with work on the long-term problems of making robot vision systems.

Vision is the sensor which is able to give the most completely information about *what* and *where* an object can be found by a robot. Although we must be cautious in comparing robot and biological systems ([Nehmzow, 1993]), it is clear that it is the main aid to navigation for many animals.

Humans are in possession of an active vision system. This means that we are able to concentrate on particular regions of interest in a scene, by movements of the eyes and head or just by shifting attention to different parts of the images we see. What advantages does this offer over the passive situation where visual sensors are fixed and all parts of images are equally inspected?

- Parts of a scene perhaps not accessible to a single realistically simple sensor are view-able by a moving device. In humans, movable eyes and head give us almost a full panoramic range of view.
- By directing attention specifically to small regions which are important at various times we can avoid wasting effort trying always to understand the whole surroundings, and devote as much as possible to the significant part; for example, when attempting to perform a difficult task such as catching something, a human would concentrate solely on the moving object and it would be common experience to become slightly disoriented during the process.

Active vision can be thought as a more task driven approach than passive vision. With a particular goal in mind for a robot system, an active sensor is able to select from the available information only that which is directly relevant to a solution, whereas a passive system processes all of the data to construct a global picture before making decisions; in this sense it can be described as data driven.

The emerging view of human vision as a *bag of tricks* ([Ramachandran, 1990]); a collection of highly specialised pieces of *software* running on specialised *hardware* to achieve vision goals, rather than a single general process, seems to fit in with active vision ideas when a similar approach is adopted in artificial vision systems. High-level decisions about which parts of a scene to direct sensors towards and focus attention on can be combined with decisions about which algorithms or even which of several available processing resources to apply to a certain task. The flexibility of active systems allows them to have multiple capabilities of varying types which can be applied in different circumstances.

In this paper we report on an overt attention system for a mobile robot endowed with a pan-tilt camera, whose will let it to find arrows and navigate through the 3D-space avoiding obstacles. This system performs an early segmentation on color space to select a set of candidate objects. Each object enters a coupled dynamics of liveliness and saliency that drives the behavior of the system over time. That way, our system will continuously keep relevant objects -such as arrows, parallelograms, faces- around the robot and it will know where are they located.

II. STATE OF THE ART

Visual attention has two clearly marked stages: first, considered pre-processing, is one in which objects are extracted -which meet certain characteristics- within the visual field; and the second, called focused attention, is the identification of those objects.

In autonomous robotics is important to perform a visual attention control. The cameras of the robots provide a large flow of data you need to select what is interesting and ignore what does not; this is the main goal of visual attention. There are two aspects of visual attention: *overt attention* and *covert*

All of them are with University of Rey Juan Carlos.

E-mails: julio.vega@urjc.es, jmplaza@gsyc.es, eperdices@gsyc.es

attention. The aim of covert attention ([Tsotsos *et al.*, 1995]), ([Itti y Koch, 2001]), ([Marocco y Floreano, 2002]) is to select interesting information within an image. Overt attention selects from the environment surrounding the robot, beyond the field of view, those objects of interest, and it looks at them ([Cañas *et al.*, 2008]).

The visual representation of the interesting objects around the robot can improve the quality of the robot's behavior and the ability to handle more information when making their decisions. This poses a problem when those objects are not in the immediate field of vision. To solve this problem, some studies used omnidirectional vision, in others using a regular camera and a mechanism for overt attention ([Itti y Koch, 2001]), ([Zaharescu *et al.*, 2005]), which enables fast-to-take samples of a very broad area of interest. The use of a camera in motion to facilitate object recognition was proposed by ([Ballard, 1991]), and has been used, for example, to distinguish between different forms in the images ([Marocco y Floreano, 2002]).

One of the concepts widely accepted in the work area is the *saliency map*. It is found in ([Itti y Koch, 2001]), as a covert visual attention mechanism, independent of the particular task to be performed and composed by all visual stimuli that attract attention from the scene. In such work is considered purely a form of "bottom up", where, as we see in Figure 1 in each iteration the different scene-descriptive maps (as colors, intensities or directions) compete between each other. Then, they merged into conspicuity maps (one for each feature) and eventually will form a unique and representative saliency map.

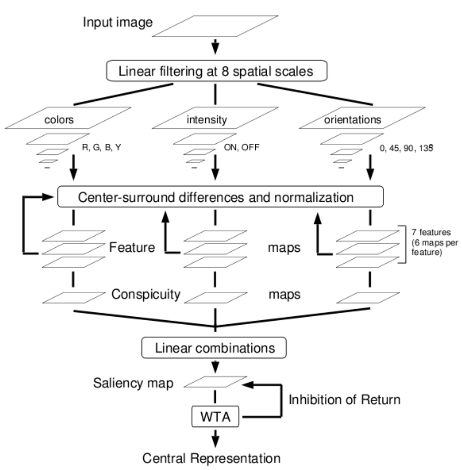

Fig. 1. Training Scheme salience map

III. 3D Visual Memory

The goal of our system is doing a visual track to the various basic objects in the scene surrounding the robot. Therefore, it must attract new objects, sharing a focus between them and removing existing memory once they have disappeared.

The first stage of the system is the 2D analysis, which detects 2D segments and character of human face present in the current image. Then the 3D reconstruction algorithm places these objects in 3D space according to the *ground-hypothesis*; that is, we suppose that all objects are flat on the floor. And finally, the 3D memory system stores their position in 3D space, calculates perceptual hypotheses and generates predictions of these objects in the current image perceived by the robot.

In this section we will see the various components of our 3D visual memory system, implemented for the attention system. On one hand the object detector, which is responsible for identifying the basic shapes and human faces which exist in the current image. On the other hand, the prediction mechanism will allow to the system predicts in advance stored items, easing the computational cost. And also highlight the algorithm of perceptual hypothesis generation about the items stored, allowing the system to abstract complex objects. And all supported on visual memory, which forms the core of the whole attention system. This allows to extend the field of vision to the whole scene surrounding the robot, not just instant visual field.

A. 2D Image Processing

The main objective of this part of the system is to extract 2D straight segments as a basic primitive, and human features. These primitives are handled by the 3D reconstructor (discussed in the next section). The 2D detection system, in turn, is connected to the 3D memory directly, in order to save computation time of reconstruction of objects that may already be stored in memory, also it can be used to confirm/refute the stored instantaneous objects. Also, the current image is useful to confirm structures previously displayed partially.

The first step to simplify the image is a edge filter, by using Canny algorithm. Subsequently we apply the Hough transform to extract only straight segments. In the case of detection of human faces we use the detector based on AdaBoost and Haar filters ([Viola y Jones., 2001]) and ([Lienhart y Maydt., 2002]). To implement these techniques, we use the OpenCV library.

In the Figure (2) below we see the reconstruction of 3D segments before and after of Hough postprocessing.

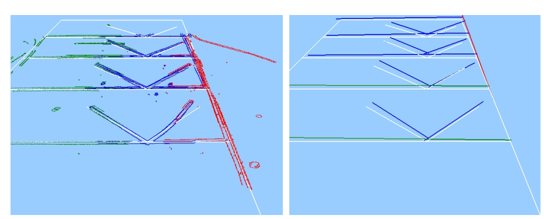

Fig. 2. 3D segments reconstruction, before and after postprocessing

1) Predictions: As we discussed previously, the 2D analysis system is connected directly to the 3D visual memory to alleviate the computational cost due to image analysis. So before extracting features of the current image, the system makes the prediction of those objects stored in the 3D memory which should be visible from the current position.

To do this processing we used the library called Progeo, by our Robotics Group, which provides *projective geometry* capabilities given a calibrated camera. So each 3D visible object is stored and made its projection on the image plane.

Then, system refutes/corroborates such segments predicted, compared one of these segments with those obtained by the Hough Transform. This comparison leads to three sets of segments, as seen in Figure 3.

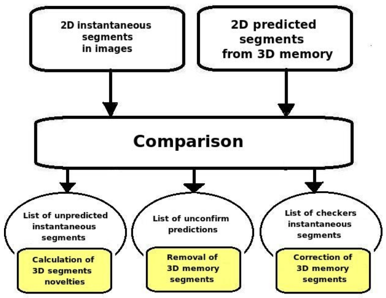

Fig. 3. Match between predicted and instant segments

B. Instantaneous reconstruction with 3D segments

The above mechanism extracts a set of 2D segments which must be located in 3D space. To do this, and as we have already mentioned, we rely on the idea of *ground-hypothesis*. Since we have one camera, we need a restriction which will enable to estimate the third dimension. We assume that all objects are flat on the floor.

Once we have the 3D objects, and before inclusion in the 3D memory, post-processing is needed to avoid duplicates in memory due to noise in the images. This postprocessing compare the relative position between segments, as well as its orientation and proximity. The output is a set of 3D segments situated on the robot coordinate system. Figure 4 shows the 3D scene with objects reconstructed by the system, the segments detected in the current image and the segments predicted from such a position.

We use a total of four sets of coordinate systems to define the geometric model:

- The absolute coordinate system whose origin lies somewhere in the world by moving the robot.
- The system located at the base of the robot. The robot odometry gives its position and orientation with respect to the previous system.
- The system relative to the base of the pan-tilt unit, which is attached the camera. It has its own encoders for its position at any given time, with pan and tilt movements with respect to the base of the robot.
- And finally we have the coordinate system of the camera itself, displaced and oriented in a particular mechanical axis from the pan-tilt unit.

C. Inserting segments

3D memory comprises a dynamic set of lists which stores information about the different types of elements present in the scene (position, type or color). From the most basic form of structure, segment, and thanks to the memory, we can establish relationships between them to make up more complex elements such as arrows, objects, parallelograms, triangles or circles.

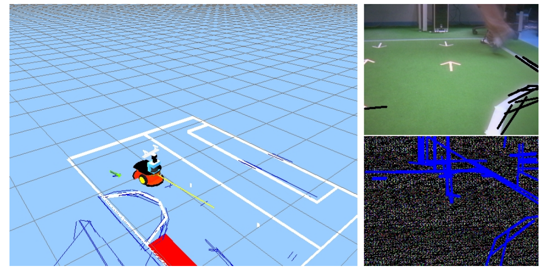

Fig. 4. 3D Scene Reconstruction, predicted and instant segments

To store a segment we have a structure called *Segment3D*, consisting start and end point and a pointer to other possible structures that can form part of it: Arrow3D (arrow) or Parallelogram3D (grid structure).

The process of incorporating 3D memory segment basically consists of comparing each segment individually calculated in the snapshot with those already stored. In case of nearby segments with similar orientation, the system combines these segments into a new one taking most of the length of his predecessors, and the orientation of the more recent, is probably more consistent with reality (the older ones tend to have more noise due to errors robot odometer).

To make this fusion process computationally lighter, the system has a segment cache where there are only those segments closest to the robot (in a radius of around 4 m.). Its implementation is basically a dynamic list of pointers to these segments. As we see, the system always works with subsets of features, which are pointers to the overall 3D memory elements.

D. Predictions: deletion and correction segments

As mentioned earlier, the 2D analysis system returns different subsets of segments, as the result of comparison between instant and predicted segments (from 3D memory).

If a segment is identified in the current image and it does not match the predictions, the system creates a new one which might replace the existing one (replacement or correction) under certain restrictions. To reflect this process, system has a parameter called *uncertainty* which will increase as the time segment remains in memory.

The deleting elements process is based on the same principle, but here there are more lax restrictions. So the replacement process is a priority compared to deletion.

E. Perceptual hypothesis generation

Our object model consists of a set of segments whose vertices can belong to more abstract structures. Not only that, but there are vertices labeled by the number of segments that are tied to them. This requires an object model for cases in which certain vertices are not visible at any given time. For example, for any parallelogram, the minimum number of visible vertices is small, with three points we are able to estimate the fourth.

The segments and their corresponding vertices are used to detect parallelograms checking the connection between them and the possible parallel ones. The analysis of the angles formed by each segment provides information about how the segments are connected to each other. In addition, this feature can be used to merge segments which are incompletely or intermittently. Similarly, we can extract the position of a possible fourth vertex using the information about other edges and/or possible parallelogram vectors. This capability makes our algorithm robust against occlusions, which occur frequently in the real world. Figure 5-b, c illustrates an example of occlusion that is satisfactorily solved by our algorithm. The results of reconstruction of parallelograms can be seen in Figure 5-a.

Similarly, we can abstract other objects such as arrows. These objects are crucial to the system we discuss, because they help the robot to navigate in the direction indicated by the arrow.

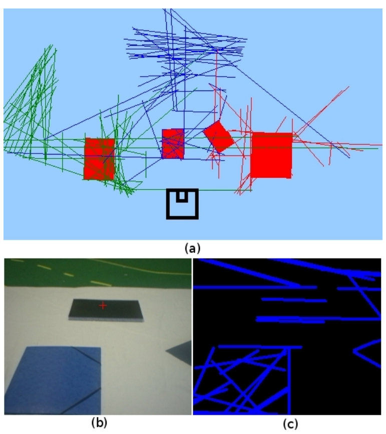

Fig. 5. Generation of hypothesis: parallelogram with occlusion

IV. Visual attention system

In the previous section we have described in detail the operation of placing objects on the robot visual memory, which have certain attributes. Well, now we will describe the visual attention mechanism implemented based on two of these attributes: *salience* and *life*. On the one hand, the saliency is used for deciding where to look in every moment, while life is the mechanism for *forgetting* an object which has disappeared from the scene.

In addition, we have a mechanism to monitor these movements of the camera, implemented as a *P-controller*, and another mechanism that allows us to explore new unknown areas from the scene.

A. Look distribution. Salience

Once established the coordinates of the scene representation for any object, it is necessary to properly control the movement of the pan-tilt unit to direct the focus to that position. Moreover, given the existence of various objects detected and located in local memory of the scene, we must have some sort of decision-making mechanism to indicate to the system where to look in the next instant.

To govern the movement of the pan-tilt unit, we introduced the dynamics of salience and attention points. These represent the detected objects in the scene. Each contains the position in the 3D scene (X, Y, Z), which is translated into mechanical commands to the pan-tilt unit in order to direct the focus to this element.

Salience is anything that attract attention or stands in a given situation, hence the focus may be changing over time. In this system the salience indicate how attention has the next object to be visited. Each memory element has an associated salience, which grows over time and vanishes every time you visit. Thus, if we have a focal point with a high salience will be the next to be visited because it is a point that draws attention, if the salience is low, will not be visited.

One way to determine the salience every attention point has, is in relation to the time this point is not visited. When a point is visited, its salience is set to 0. By contrast, a point that you have not been visited call more attention than one who has attended recently. The system is thus the behavior of a human eye, and that studies of biology ([Itti y Koch., 2005]), when the eye responds to a stimulus that appears in a position that has been previously treated, the reaction time is usually higher than when the stimulus appears in a new position.

The designed algorithm allows the system to alternate the focus of the camera between the different objects in the scene according to their salience. In our system, we considered all objects have the same preference of attention, so all of them are observed during the same time and with the same frequency. If we assign different priorities to the objects, we could establish different rates of growth of salience. This would cause the pan-tilt unit to pose more times in the object whose salience grows faster.

The problem of how to revisit a point is solved considering the spatial interpretation, because we consider it is a problem of evolution of the hypothesis in the time among detections in t and $t + n$ (where n is the time a point is not visited). We have assumed that a detected object will be found near where it was previously.

B. Follow-up motion

When the look-sharing system choose a focal point, it is going to be looking for a certain time (3 seconds), even following it if it moves spatially. For this monitoring, and to avoid excessive oscillations and have a more precise control over pan-tilt unit, we decided to implement a *P-controller* to control the speed of the pan and tilt and thus continually focus on the target face image. This driver allows command P or high speed, proportional to the pan-tilt unit, if the focus of attention to be targeted is far from the current position; or lower speeds if it requires small corrections.

C. Exploring new areas of interest

At any time, it may be interesting to look for new objects in the scene. Our system will insert periodically (every *tiempoBusquedaForzada*) scanning points with high salience in memory. This search could be interesting, especially at the beginning of the process, when it still unknowns areas of the scene where there are objects of interest.

The scanning points can be of two types: random and travel. The first generation consists in assigning random coordinates *(pan, tilt)* within the pan-tilt range (pan = [-159, + 159], tilt = [-31, +31]). That way, the system ensures that all areas of the scene will be visited. Thus, these points will range from the lowest position of pan-tilt to the highest position, and also with the tilt coordinate.

The attention points, whatever their type, have a high initial salience in order to be visited more quickly and thereby check whether they exist any object of interest. Then these points will remain in memory. This is the way in which new objects entering the system: they are inserted into the memory and so they enter the look-sharing system.

There will be a proliferation of points of exploration in the beginning, because the most interesting process in that moment is to find areas of interest in the scene as we start from the absolute ignorance of the environment. As we discover objects, the desire to explore new areas will decrease in proportion to the quantity of these ones.

D. Representation of the internal environment. Life

As already discussed in previous sections, our visual attention system is always guided by tracking objects within the scene. It can keep track of several objects which have been detected over time and stored in memory, alternating between them, though not within the immediate field of view of the camera. The objects may eventually disappear from the scene, which should be removed from the system to maintain consistently the representation of the scene with reality.

To accomplish this task of forgetting old elements, we have implemented the *life* dynamic. With this mechanism the system can know if an object has left the scene or if it is still there. The life operation is the reverse of the salience, that is, a frequently-visited object will have more life than one which has just visited. When the life of an object is below a certain threshold, it will be discarded.

To implement this dynamic, every time you visit an object, its life increases a little, with a maximum limit to avoid saturation. The life of unobserved objects will decrease over time. Thus, when the life of an object exceeds a certain threshold, which is still on the scene, whereas when is below it is gone.

E. Attentive system design

The objects of the environment surrounding the robot guide the movements of the camera, so the mechanism of attention is *bottom-up*. Besides, the mechanism of *top-down* is that existing relevant objects are those that have a certain appearance: human faces, parallelograms or arrows. This tendency to look at objects with a certain aspect is similar to the bias detected by ethologists in animals with respect to certain stimuli ([Tinbergen, 1951]).

The visual attention system presented here has been implemented following a *state-machine* design, which determines when to execute the different steps of the algorithm. Thus, we can distinguish four states:

- Discuss next goal (state 0).
- Saccade is completed (state 1).
- Analyze image (state 2).
- Follow-up object (state 3).

The operation is as follows. Initially what we do, over time, is going to update the possible objects that have already stored in memory. On the one hand to check if any of them is already outdated, because its life is below a certain threshold, and secondly, increase the salience and reduce life.

Based on the initial state (or state 0), the system asks if there is a goal to look (in case we have an object previously stored in memory) and if so, we go to state 1. If not, create a new attention point, and fit it into memory. Back to state 0.

In state 1 the task is to complete the move towards the absolute position specified by the state 0. Once there, we go to stage 2 where we will analyze whether there are relevant objects or not. In any case, it passed the state 0 and back again.

From state 0 to 3 will only pass if the target set last found an object, in which case it can track it. This is precisely the purpose of the state.

V. Experiments

Our experiments were performed with a real robot Pioneer 2DX (by ActivMedia Robotics), on which is mounted a Dell laptop with an Intel Centrino processor at 1.7 GHz. and Linux Ubuntu 8.04 (hardy) as operating system. Also it has installed a pan-tilt unit (46-17.5 Unit Pantilt Directed Perception) with a pan range of [180, -180] and a tilt one of [31, -80] degrees. It is able to work since a minimum rate of 0.0123 deg./sec. and a maximum rate of 300 deg./sec. on both axes. In turn, it has placed a firewire iSight camera (by Apple) with autofocus and a focal distance of 60 and 40 degrees horizontal and vertical respectively. The power to the pan-tilt unit is supplied by the base of the robot, and it is serial-port commanded.

A. 3D floor reconstruction

In this first experiment, the robot has no knowledge of the environment. Initially, and as already mentioned, it will do a thorough systematic search for information from the environment. Thus, the system must command saccades to the pan-tilt. These movements are short, accurate and fast, just enough time to examine whether there is some interest object in the current image received with the camera. After a certain time, the system begins to detect segments (see Figure 6).

After several glimpses the robot is able to plausibly reconstruct the detected segments in its path (see Figure 7). Finally the memory is subjected to a post-processing by which we obtain unique segments, coinciding with what exists in reality (Figure 2).

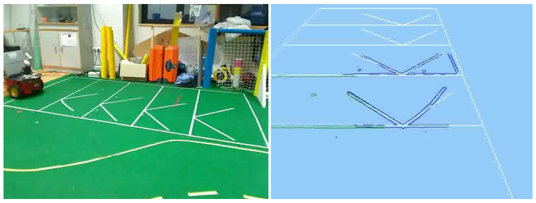

Fig. 6. Land lines reconstruction. Initial stage

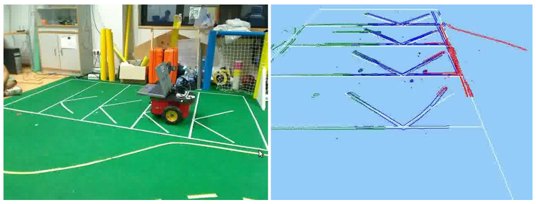

Fig. 7. Land lines reconstruction. Final stage

B. Parallelograms

In the second experiment we also consider the absolute ignorance of the environment surrounding the robot. In this case, besides finding segments of the environment, the system can abstract parallelograms given the characteristics of all segments in the scene. The forced scan time is 5 seconds, after which the system will scan the scene enforced. This process is repeated for some time, until the robot begins to detect objects of interest in the scene (see Figure 8).

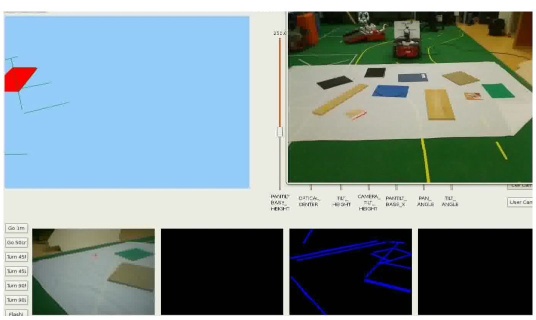

Fig. 8. Parallelograms recognition. Initial stage

When it begins to have several elements (parallelograms) in memory (see Figure 9), the forced scan time is increased. This technicality allows to maintain the look time longer at objects it has detected and also the possibility for further searches. Through this mechanism, the system finds step by step almost all the items in the scene (note that some objects cause problems because of their texture).

As you notice, what we get with this increased time is that as we are detecting more and more items, finding new ones will become increasingly rare. However, when the system make a forced exploration, it will also explore new areas, and then return the item to be visited over time.

Finally, we reflected on the graph below (Figure 10) the mode of action of the two competing dynamics as explained above: salience and life. Correspond to the time the system has two elements detected.

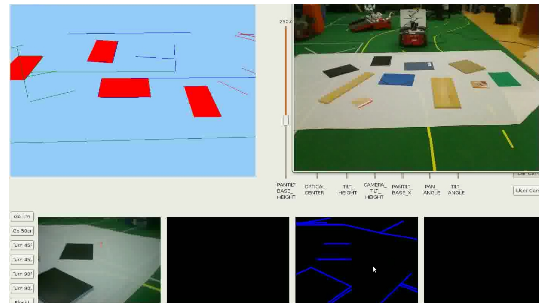

Fig. 9. Parallelograms recognition. Final stage

In such a situation, we can see in Figure 10-a how to evolve the salience of both. When the system is following an item (blue) its salience decreases, while the other item stored in memory (red) increases until it wins the competition and forces the system to look there.

The evolution of life on both objects, when both remain on the scene, is shown in Figure 10-b. Its operation is inverse to the salience, that is, every time the system visits an item, its life is increased slightly, with a maximum limit to avoid saturation.

Figure 10-c reflects a situation in which we occluded one of the two elements, so that the system fails to detect it as such and, therefore, its life begins to fall. When its value is below a certain threshold, the object is discarded and not re-visited.

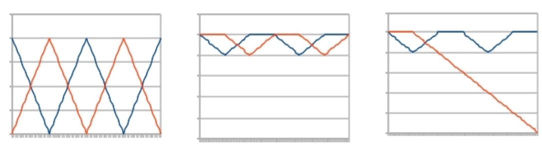

Fig. 10. Plot of time evolution of the system. (a) Salience, (b) Life and (c) Life to the disappearance of one of them

C. Arrows as a mark of direction

In this last experiment we rely on the same ideas given above, but in this case we focus on the recognition of arrows in the environment, and the use of this item as a mark of direction for robot navigation. Figure 11 shows when the robot recognizes the arrow as such, having been previously detected the segments which compound it.

Given the characteristics of an arrow, the system is able to abstract the concept *arrow* and represent it as such in the 3D memory (see the green arrow showed in Figure 11). Also, once detected, it automatically guides the direction of the robot (see the yellow line of the robot showed in Figure 11).

Finally, in Figure 12 we have mixed objects of different types (parallelograms and arrows) and they are recognized and stored in the 3D visual memory. Also, upon detection

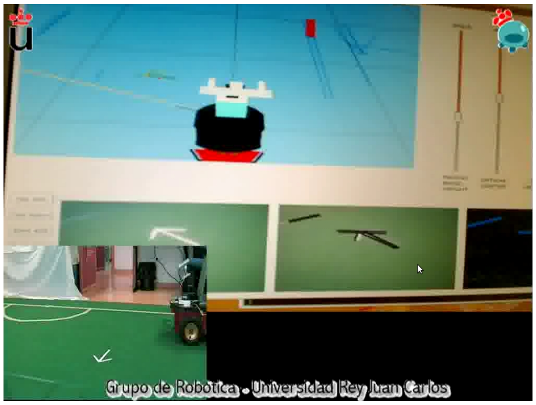

Fig. 11. Recognition of arrows as a mark of direction

of several arrows in the robot's environment, it considers the nearest primary. Hence, its target is now different, because the new arrow is closer than it had previously established as a target.

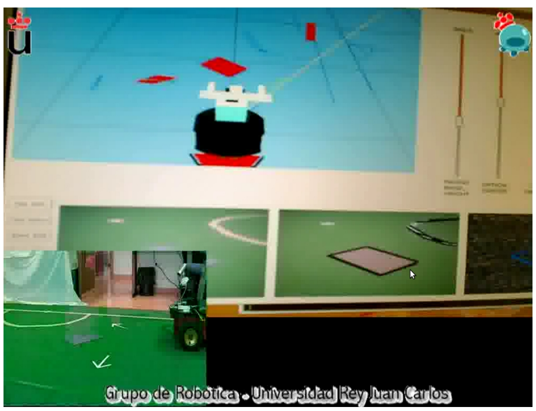

Fig. 12. Recognition of parallelograms and arrows

VI. Conclusions

In this paper we have presented a global visual attention system whose purpose is to find objects or abstract concepts by the scene surrounding the robot, tracking them. For this purpose we have developed a mechanism for concurrent dynamic between life and salience, in which the element with the highest salience is the next to be visited and, therefore, directs the movement of the pan-tilt unit at all times. So the robot continues looking for all the objects which we consider interesting. And the life dynamic allows to have a coherent representation of the items on scene, thus preventing the robot to pay attention to objects that are no longer there.

Moreover, since the scene is greater than the immediate field of view of the robot camera, we implemented a 3D visual short-term memory. This memory has facilitated the internal representation of information around the robot, since objects may be placed in positions that the robot can not see at any given time but where it knows there are elements of interest.

The different experiments carried out show that the attention behaviors generated are quite similar to a human visual attention system.

Another aspect is the forgetting of items that have disappeared from the scene, thereby avoiding ghosts in the memory representative of the environment. However, take several attempts have failed to consider the disappearance of an object, because sometimes an object may not be detected by sporadic occlusions. Although the detection algorithm presented is usually quite robust to different lighting conditions.

One possible improvement in this work could be the use of visual attention for the robot navigates autonomously, asides to recognize and abstract correctly the objects from the environment surrounding itself. The system could move the camera to detect the different visual elements and navigation marks (arrows) as well as potentially dangerous obstacles (such as walls). Once all the elements were detected, the system would include in its internal representation of the world, and would share the look between them.

And at that point we could play comfortably with the salience parameter so that, for example, grow faster in the objects recognized as obstacles and/or navigation beacons. This would let robot to achieve a safe navigation.

REFERENCES

[Ballard, 1991] D. H. Ballard. Animate vision. *Artificial Intelligence 48, pp. 57-86*, 1991.

[Cañas *et al.*, 2008] J.M. Cañas, M. Martínez de la Casa, y T. González. Overt visual attention inside jde control architecture. *International Journal of Intelligent Computing in Medical Sciences and Image Procesing. Volume 2, Number 2, pp 93-100, ISSN: 1931-308X. TSI Press, USA*, 2008.

[Itti y Koch, 2001] L. Itti y C. Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience 2, pp. 194-203*, 2001.

[Itti y Koch., 2005] L. Itti y C. Koch. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[Lienhart y Maydt., 2002] R. Lienhart y J. Maydt. An extended set of haar like features for rapid object detection. *IEEE ICIP 2002, volume 1, pp. 900-903*, 2002.

[Marocco y Floreano, 2002] D. Marocco y D. Floreano. Active vision and feature selection in evolutionary behavioral systems. *Proc. of Int. Conf. on Simulation of Adaptive Behavior (SAB-7), pp. 247-255*, 2002.

[Nehmzow, 1993] U. Nehmzow. Animal and robot navigation. *The Biology and Technology of Intelligent Autonomous Agents*, 1993.

[Ramachandran, 1990] V. S. Ramachandran. Visual perception in people and machines. *A. Blake, editor, A.I. and the Eye, chapter 3. Wiley and Sons*, 1990.

[Tinbergen, 1951] N. Tinbergen. The study of instinct. *Clarendon University Press, Oxford UK*, 1951.

[Tsotsos *et al.*, 1995] J. K. Tsotsos, S. Culhane, W. Wai, Y. Lai, y N. Davis. Modeling visual attention via selective tuning. *Artifical Intelligence 78, pp. 507-545*, 1995.

[Viola y Jones., 2001] P. Viola y M. Jones. Rapid object detection using a boosted cascade of simple features. 2001.

[Zaharescu *et al.*, 2005] A. Zaharescu, A. L. Rothenstein, y 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*, 2005.