Local robot navigation based on an active visual short-term memory

Julio Vega, Jose María Cañas, Eduardo Perdices

***Abstract*—Vision devices are today one of the most often used sensory elements in autonomous robots. Some of their hindrances are the difficulty in extracting 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 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 one of the most successful sensing modalities used in mobile robotics. It would seem to be the most promising one for the long term. Computer vision research is currently growing rapidly, both in robotics and in many other applications, from surveillance systems for security purposes to the automatic acquisition of 3D models for Virtual Reality displays. The number of commercial applications is also increasing, like traffic monitoring, parking access or face recognition. And we feel that it is well worth continuing with work on the long-term problems of making robotic vision systems.

Vision is the sensor whose main skill lies in giving information about which and where are the objects that the robot is finding over its path. And, although we must be wary when comparing a robot with a biological organism [Nehmzow, 1993], what is clear is that sight is the main sense used by animals when they want to move around the environment.

Humans have 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? First, some parts of a scene perhaps are not accessible to a single sensor are viewable by a moving device. In humans, movable eyes and head give us almost a full panoramic range of view. Second, 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 them 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 work we describe an overt attention system for a mobile robot endowed with a pan-tilt camera, whose will let it to find paper arrows on its surroundings 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 life and salience that drives the behavior of the attention system over time. That way, our system will continuously keep relevant objects around the robot -such as arrows or parallelograms- in its visual short-term memory and it will know where they are located.

In the next section some related works about vision based navigation and attention systems are described as context of our proposal. Our system description has been divided in two sections, one explaining the visual memory and another one describing the attention subsytem. Some experiments have been carried out with a real robot to validate our approach, they are commented in section V. We end this paper with some conclusions and future lines.

All of them are with University of Rey Juan Carlos.

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

II. Related works

Vision has been used in robotics almost from its beginning. In the last years its use is increasing, mainly because of the reduction in the camera price, the available computing power and the potential of cameras as source of information about robot surroundings. Many issues have been taclked in the intersection of computer vision and robotic fields. For instance, vision-based control or navigation, vision-based map building and 3D representation, vision-based localization, object recognition, attention and gaze control among others. We will review some examples here.

Regarding vision based control and navigation, Remazeilles et al.[Remazeilles *et al.*, 2006] presented the design of a control law for vision-based robot navigation. The particularity of this control law is that it does not require any reconstruction of the environment, and it does not force the robot to converge towards each intermediary position in the path.

Recently, Srinivasan [Srinivasan *et al.*, 2006] presented a new system to increase accuracy in the optical flow estimation for insect-based flying control systems. A special mirror surface is mounted in front of the camera, which is pointing ahead instead of pointing to the ground. The mirror surface decreases the speed of motion and eliminates the distortion caused by the perspective. Theoretically, the image should present a constant and low velocity everywhere, simplifying the optical flow calculation and increasing its accuracy. Consequently, the system increases the speed range and the number of situations under which the aircraft can fly safely.

Badal [Badal *et al.*, 1994] reported a system for extracting range information and performing obstacle detection and avoidance in outdoor environments based on the computation of disparity from the two images of a stereo pair of calibrated cameras. The system assumes that objects protrude high from a flat floor that stands out from the background. Every point above the ground is configured as a potential object and projected onto the ground plane, in a local occupancy grid called Instantaneous Obstacle Map (IOM). The commands to steer the robot are generated according to the position of obstacles in the IOM.

Goldberg [Goldberg *et al.*, 2002] introduced a stereo vision-based navigation algorithm for the rover planetary explorer MER, to explore and map locally hazardous terrains. The algorithm computes epipolar lines between the two stereo frames to check the presence of an object, computes the Laplacian of both images and correlates the filtered images to match pixels from the left image with their corresponding pixels in the right image. The work also includes a description of the navigation module GESTALT, which packages a set of routines able to compute actuation, direction, or steering commands from the sensor information.

Regarding map building and self-localization maybe the most succesful approach in last years has been the MonoSLAM from A. Davison [Gerardo Carrera y Davison, 2011]. The detection of relevant points in the image and a fast Extended Kalman Filter allow the system to continuously estimate the camera 3D position and orientation and the 3D position of such points. The localization results are impressive. The quality of the maps, mainly as collection of 3D points, was not so good at the beginning but they have improved it even with dense maps in real time [Newcombe y Davison, 2010].

Mariottini and Roumeliotis [Mariottini y Roumeliotis, 2011] presented a strategy for active vision-based localization and navigation of a mobile robot with a visual memory within a previously-visited area represented as a large collection of images. This strategy can disambiguate the true initial location among possible hypotheses by controlling the mobile observer across a sequence of highly distinctive images, while concurrently navigating towards the target image.

Gartshore [Gartshore *et al.*, 2002] developed a map building framework and a feature position detector algorithm that processes images on-line from a single camera. The system does not use matching approaches. Instead, it computes probabilities of finding objects at every location. The algorithm starts detecting the objects boundaries for the current frame using the Harris edge and corner detectors. Detected features are back projected from the 2D image plane considering all the potential locations at any depth. The positioning module of the system computes the position of the robot using odometry data combined with image feature extraction. Color or gradient from edges and features from past images help to increase the confidence of the object presence in a certain location. Experimental results tested in indoor environments set the size of the grid cells to 25 mm 25 mm. The robot moved 100 mm between consecutive images.

In autonomous robots it 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 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 *salience 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 salience map.

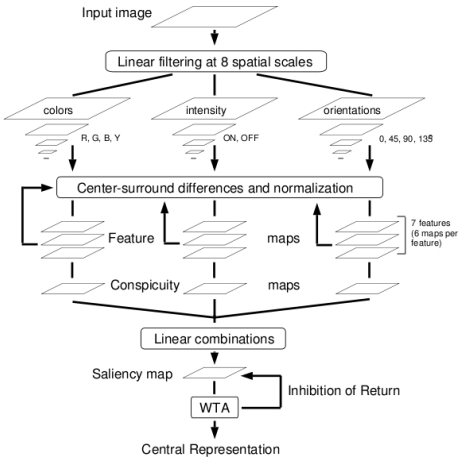

Fig. 1. Salience map as proposed by Itti

Hulse [Hulse *et al.*, 2009] 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. They argued that visual information has to be subsequently processed by a number of cortical and subcortical structures that place it: 1) in context of attentional bias within egocentric salience maps; 2) the aforementioned IOR inputs from other modalities; 3) overriding voluntary saccades and 4) basal ganglia action selection. Thus, biologically there is a highly developed, context specific method for facilitating the most appropriate saccade as a form of attention selection.

Arbel and Ferrie presented in [Arbel y Ferrie, 2001] 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. The new measurements, accumulated over time, are used in a one-step-ahead Bayesian approach that resolves the object recognition ambiguity, while it navigates with an entropy map.

III. 3D Visual Memory

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

The first stage of the system is the 2D analysis, which detects 2D segments 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 in conjunction with the attention system. Some previous versions of the system are described in [Vega y Cañas, 2009; 2010]. First, an object detector is responsible of identifying basic shapes in the current image. Second, the prediction mechanism will allow the system to predict how the stored items will appear in next images, reducing the computational cost of image processing for them. Third, a 3D reconstruction block is responsible to obtain 3D instantaneous information from objects in the current image and merging them with the objects already stored in the visual memory.

A. 2D Image Processing

The main objective of this part of the system is to extract 2D straight segments as a basic primitive. These primitives are handled by the 3D reconstruction module. The 2D detection module, 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. It also can be used to confirm/refute the stored instantaneous objects. The current image is useful to confirm structures previously observed partially.

The first step to simplify the image is an edge filter, by using the Canny algorithm. Subsequently we apply the Hough transform to extract only straight segments. To implement these techniques, we use the OpenCV library. In the Figure (2) we see the reconstruction of 3D segments before and after of Hough postprocessing.

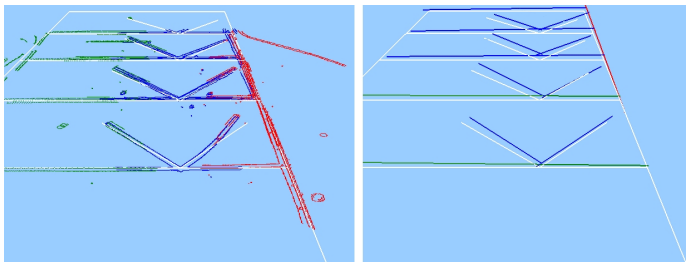

Fig. 2. 3D segments reconstruction, before and after postprocessing

B. Predictions

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.

We have used our library called Progeo, which provides *projective geometry* capabilities given a calibrated camera. So each 3D visible object is stored and made its projection on the image plane (see Figure 3). The system refutes/corroborates such segments predicted, comparing one of these segments with those obtained by the Hough Transform. This comparison leads to three sets of segments, as seen in Figure 4.

C. 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

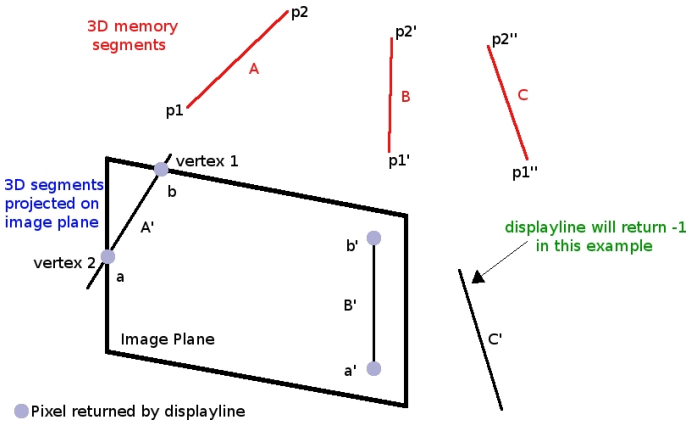

Fig. 3. Segments projection onto the image plane

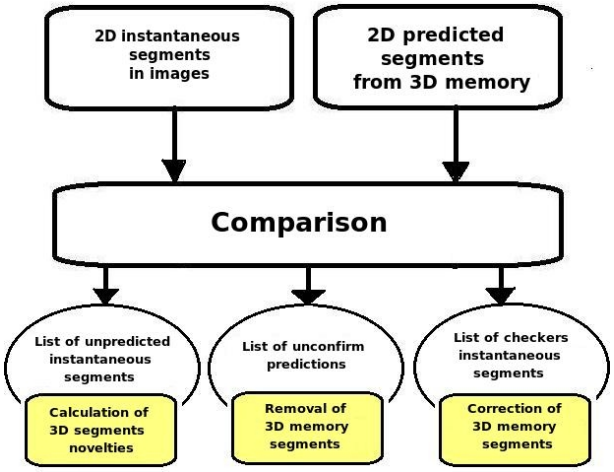

Fig. 4. Match between predicted and instantaneous segments

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 compares 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 5 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 four coordinate systems to define the geometric model, as we can see in Figure 6:

- The absolute coordinate system whose origin lies somewhere in the world where the robot is moving.
- 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 to which the camera is attached to. 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.

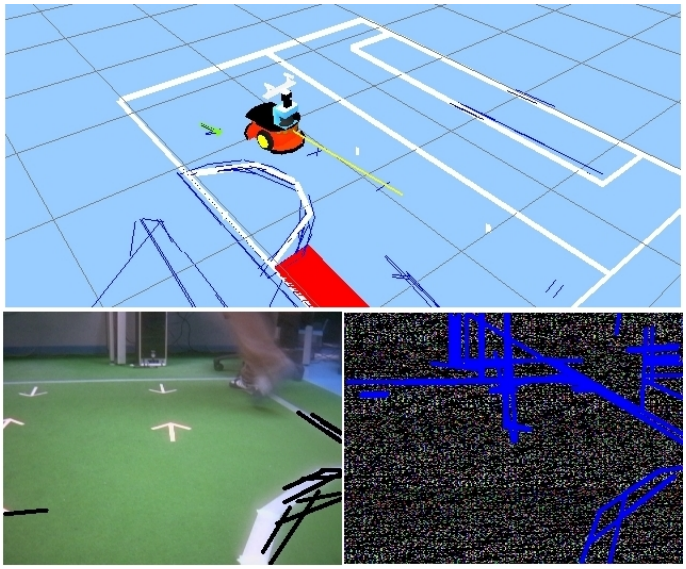

Fig. 5. 3D Scene Reconstruction, predicted and instant segments

D. Inserting segments into the 3D visual memory

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). The most basic form of structure is the segment. Thanks to the memory we can establish relationships between them to make up more complex elements such as arrows, parallelograms, triangles, circles or other objects.

To store a segment we have a structure called *Segment3D*, consisting in a start and end point and a pointer to other possible structures of which it can be part of: Arrow3D or Parallelogram3D.

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 the longest length of its predecessors, and the orientation of the more recent, as probably it is more consistent with reality (the older ones tend to have more noise due to errors robot odometry).

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

E. Predictions: deletion and correction of segments

As mentioned before, the 2D analysis system returns different subsets of segments, as the result of comparison between

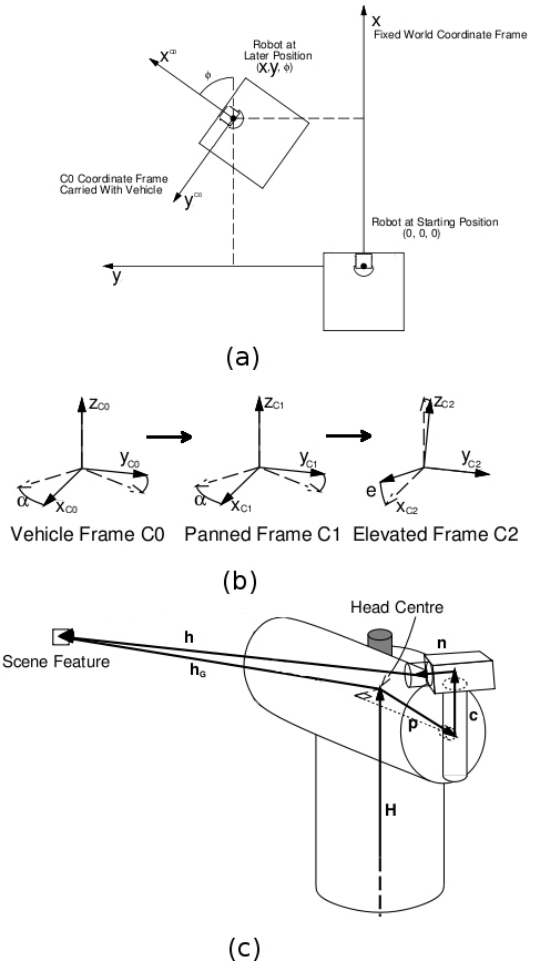

Fig. 6. Coordinate systems used to define the geometric model

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 in 3D which might replace an 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 element deletion process is based on the same principle, but here there are more rigorous restrictions. So, the replacement process is a priority compared to deletion.

F. Perceptual hypothesis generation

Our object model consists of a set of segments whose vertices can belong to more abstract structures like parallelograms. The vertices are labeled with 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 instance, for parallelograms the minimum number of visible vertices is small, with three points we are able to estimate the fourth one.

The segments and their corresponding vertices are used to detect parallelograms checking the connection between them and the parallelism conditions. 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 incomplete or intermittent segments. 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 7-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 7-a. In this situation, we have a collection of parallelograms spread on the floor. The robot, after several snapshots, captures what is around itself: those parallelograms, and noise extracted by the segmentation algorithm. However, our parallelogram hyphotesis generation is able to extract only the real parallelograms, avoiding such noise.

Similarly, we can abstract other abstract objects such as arrows.

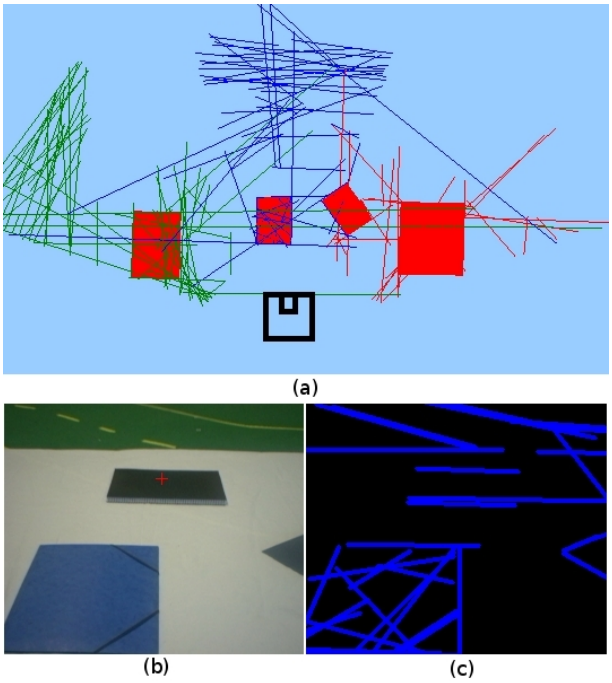

Fig. 7. Generation of hypothesis: parallelogram with occlusion

IV. VISUAL ATTENTION SYSTEM

In the previous section we have described in detail the operation of placing objects on the robot 3D visual memory. Now we will describe the visual attention mechanism implemented based on two of object attributes: *salience* and *life*. The salience is used for deciding where to look at in every moment, while life is the mechanism for forgetting an object, deleting it from memory, when it has disappeared from the scene.

In addition, we have designed a mechanism to control the camera movements, to track objects, and another mechanism to explore new unknown areas from the scene.

A. Gaze control: salience dynamics

Some sort of decision-making mechanism to indicate to the system where to look at in the next instant. Each object in the visual memory has its coordinates in the scene representation. It is desirable to control the movement of the pan-tilt unit to direct the focus to that position periodically in order to reobserve that object. To control the movement of the pan-tilt unit, we introduced the dynamics of salience and attention points. They mainly represent the detected objects in the scene. Each one contains a 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 at that point.

Anything that attracts attention or stands in a given situation is salient. The focus may be changing over time to look at all the salient points. In this system salience indicates how attention selects the next object to be visited. Each memory element has an associated salience, which grows over time and vanishes every time that element is visited. The focal point with highest salience will be the next to be visited. If the salience is low, it will not be visited now.

$$Salience(t) = \begin{cases} Salience(t-1) = 0 & \text{if object attended} \\ Salience(t-1) + 1 & \text{otherwise} \end{cases}$$

When a point is visited, its salience is set to 0. A point that the system has not visited recently calls more attention than one which has just been attended. The system is thus similar to the behavior of a human eye, as pointed by biology studies [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 consider that 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 assigned 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.

We assume that a detected object will be found near the location where it was previously.

B. Tracking of a focused object

When the look-sharing system chooses the attention point of a given object, it is going to be looking for a certain time (3 seconds), tracking it if it moves spatially. For this tracking, and to avoid excessive oscillations, we use a *P-controller* (Fig. 8) to control the speed of the pan and tilt and thus continually focus that object on the image center. This driver orders high speeds to the pan-tilt unit if the focus of attention is far from the predicted position; or lower speeds if it requires small corrections. The controller follows next equations.

$$v(Pan) = \begin{cases} 0 & \text{if} \quad \epsilon < 0.3 \\ K_p \cdot (P_t - P_a) & \text{if} \quad 0.3 \leq \epsilon < M_p \\ M_p & \text{if} \quad M_p < \epsilon \end{cases}$$

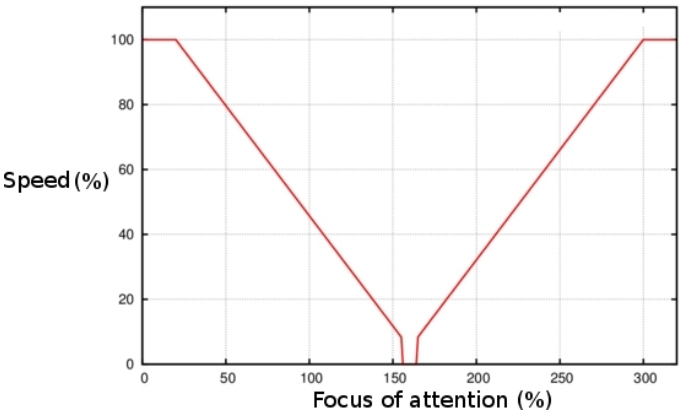

Fig. 8. P-controller mechanism

$$v(Tilt) = \begin{cases} 0 & \text{if} \quad \epsilon < 0.1 \\ K_p \cdot (T_t - T_a) & \text{if} \quad 0.1 \leq \epsilon < M\_t \\ M_t & \text{if} \quad M_t < \epsilon \end{cases}$$

Where: K_p is the P control gain, T_t is the Tilt of the target, T_a is the actual Tilt, P_t is the Pan of the target, P_a is the actual Pan, M_t is the maximum Tilt and M_p is the maximum Pan.

C. Exploring new areas of interest

At any time, it may be interesting to look for new objects in the scene. For that search our system will insert periodically (every *forcedSearchTime*) scanning points with high salience in memory. This search is especially interesting at the beginning, when there are many unknowns areas of the scene where there can be objects of interest.

The scanning points can be of two types: random and systematic ones. The first type are assigned uniformly random coordinates *(pan, tilt)* within the pan-tilt range (pan = [-159, + 159], tilt = [-31, +31]). Systematic scanning points follow a regular pattern to finally cover the whole scene around the robot.

The attention points, whatever their type, have a high initial salience in order to be quickly visited with the camera and thereby check whether any object of interest is found around it. In such a case that object will enter into the memory and into the gaze sharing module.

As we discover objects, the desire to explore new areas will decrease in proportion to the number of already detected objects.

D. Representation of the environment: life dynamics

As already discussed in previous sections, our visual attention system guides the search and tracking of objects within the scene. The objects may eventually disappear from the scene, and then they should be removed from the memory in order to maintain coherence between the representation of the scene and the reality.

To forget such old elements, we have implemented the *life* dynamics. With this mechanism the system can know whether

an object has left the scene or 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 less visited. When the life of an object is below a certain threshold, it will be discarded.

Every time the attention system visits an object, its life increases one point, with a maximum limit to provide 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.

$$Life(t) = \begin{cases} min(MAX_{LIFE}, Life(t-1) + \Delta) \\ \qquad \text{if object observed} \\ Life(t-1) - 1 \\ \qquad \text{otherwise} \end{cases}$$

E. Attentive system operation

The objects of the environment surrounding the robot guide the movements of the camera. So the mechanism of attention is so far *bottom-up*. Besides, the underlying mechanism of *top-down* attention is that existing relevant objects are only those that have a certain appearance given by the task at hand: in our examples, 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 [Arkin, 1998].

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).
- Tracking object (state 3).

Periodically the system updates the salience and life attributes of the objects that have already stored in memory following previous equations. It checks whether any of them is already outdated, because its life is below a certain threshold. If not, it increases its salience and reduces its life.

Based on the initial state (or state 0), the system asks whether there is any attention point to look at (in case we have an object previously stored in memory) or not. If so, it goes to state 1. If not, it inserts a new scanning attention point into memory and goes back to state 0.

In state 1 the task is to complete the movement towards the absolute position specified in 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.

V. Experiments

Our experiments were performed with a real Pioneer 2DX robot (by MobileRobots), endowed with a Dell laptop with an Intel Centrino processor at 1.7 GHz. and Linux Ubuntu 8.04 (hardy) as operating system. It also has installed a pan-tilt unit (46-17.5 Unit Pantilt Directed Perception) with a pan range of [180, -180] and a tilt range of [31, -80] degrees. It works at a minimum speed of 0.0123 deg./sec. and a maximum speed of 300 deg./sec. on both axes. The pan tilt unit has a firewire iSight camera (by Apple) on top, with autofocus and a field of view of 60 and 40 degrees in 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 did a thorough systematic search for information from the environment. The system commanded saccades to the pan-tilt. These movements are short, accurate and fast, just enough time to examine whether there is any interesting object in the current image received from the camera or not. After a certain time, the system begins to detect segments (see Figure 9).

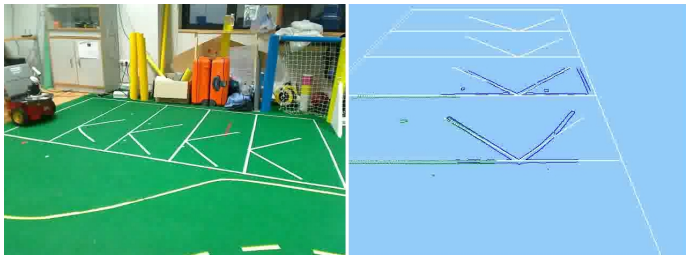

Fig. 9. Land lines reconstruction. Initial stage

After several glimpses the robot is able to plausibly reconstruct the detected segments along its path (see Figure 10). The visual memory periodically performed some post-processing by which unique and refined segments were obtained. In this experiment they perfectly fit those in reality (Figure 2).

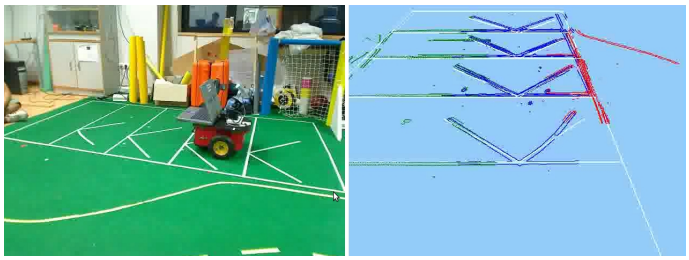

Fig. 10. Land lines reconstruction. Final stage

B. Parallelograms

In the second experiment, besides finding segments of the environment, the system can abstract parallelograms given the characteristics of all segments in the scene.

The *forced-SearchTime* period was 5 seconds, so every 5 seconds new exploring scanning points where inserted in the visual memory. This process is repeated some times, until the robot begins to detect objects of interest in the scene (see Figure 11). When it begins to have several elements

(parallelograms) in memory (see Figure 12), the *forced-SearchTime*is increased. This feature allows to look longer at already detected objects and less to explore new areas or search for new objects. With this increased time finding new objects will become increasingly rare as we have already detected more and more items.

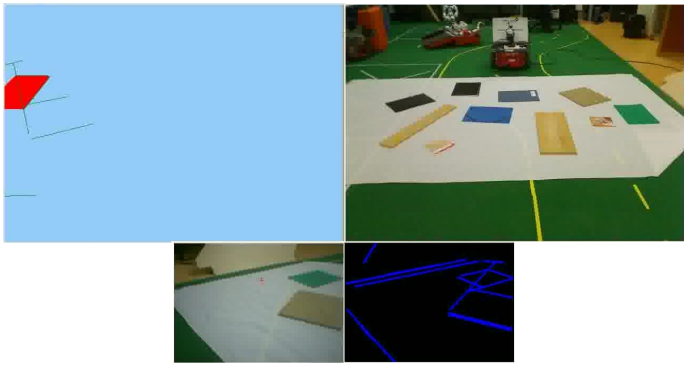

Fig. 11. Parallelograms recognition. Initial stage

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).

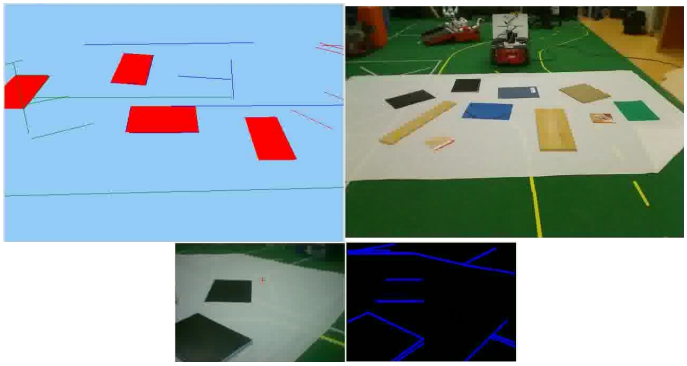

Fig. 12. Parallelograms recognition. Final stage

In Figure 13 the mode of action of the two competing dynamics, salience and life, is shown. It corresponds with a situation where the system has detected two elements. We can see in Figure 13-a how the salience of both evolve. 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 at it.

The evolution of life on both objects, when both remain on the scene, is shown in Figure 13-b. Its operation is inverse to the salience, that is, every time the system visits an item, its life is increased one point, with a maximum limit score of 100 points to provide saturation.

Figure 13-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.

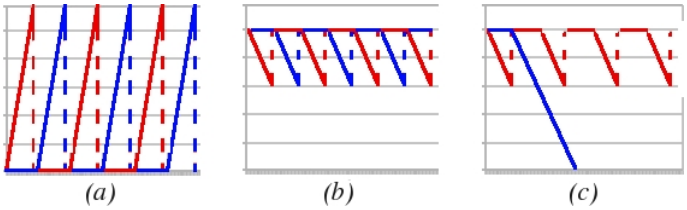

Fig. 13. Time evolution of the salience (a), Life (b) and Life of a disappearing object

C. Arrows as navigation landmarks

In this 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 14 shows when the robot recognizes the arrow as such, having been previously detected the segments which compound it.

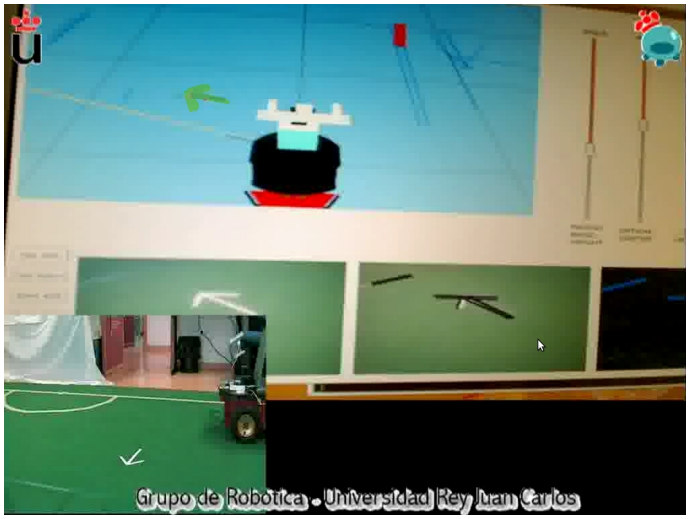

Fig. 14. Recognition of arrows as a mark of direction

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 14). Also, once detected, it automatically guides the direction of the robot (see the yellow line of the robot showed in Figure 14).

In Figure 15 we have mixed objects of different types (parallelograms and arrows) and they are recognized and stored in the 3D visual memory. Also, upon detection of several arrows in the robot's environment, it will only consider the nearest one as navigation landmark and will follow its direction.

D. Robot occlusions

This experiment shows how the system behaves in case of temporary occlusions. They happen very often in real environments where there are dynamic objects which can obstruct the robot field of view.

Fig. 15. Recognition of parallelograms and arrows

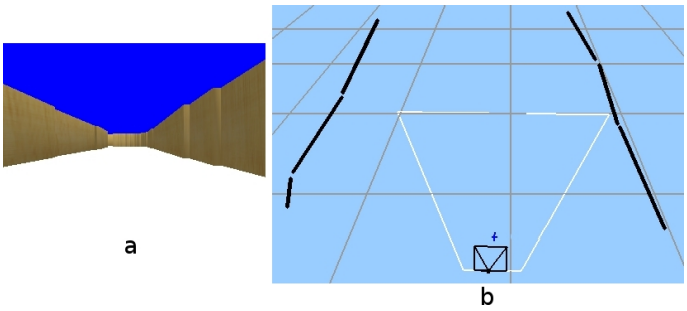

Fig. 16. (a) Obstacle free situation; (b) Short-term memory after a while

The initial situation is showed in figure 16-a. After a few seconds, our robot has recovered environment information thanks to the short-term memory and the visual attention system. This information is showed in 16-b, and it is broader than current camera field of view.

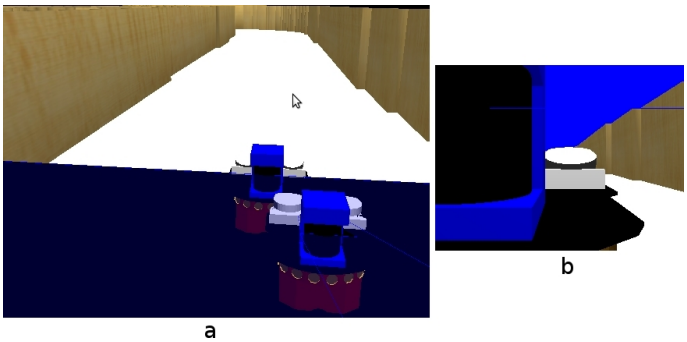

Fig. 17. (a) Situation; (b) Field of view

After a while another robot appears (as showed in Figure 17-a) occluding the field of view of our robot (as showed in Figure 17-b), so it is unable to see anything. This situation continues for some time while the second robot moves away from our robot (18-a,b).

This hindrance is solved by our system because of the

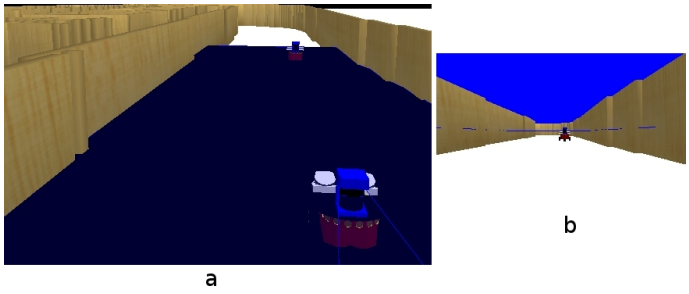

Fig. 18. (a) Situation; (b) Field of view

persistence of the short-term visual memory. As we discussed in section IV-D, the memory is refreshed over time. If it is inconsistent, that is, if what the robot sees does not match with the information stored in memory, we give a confidence interval until this situation is solved and new observations let's confirm or discard the objects in visual memory.

VI. Conclusions

In this paper we have presented an overt visual attention system whose purpose is to find objects in the scene surrounding the robot and to track them. We have developed a mechanism with two concurrent dynamics for gaze control: life (or object quality) and salience. They are defined for objects in 3D, not just the current image. The salience of objects increases in time and is set to zero every time the camera looks at it. The element with the highest salience is the next to be visited and the system controls the pan-tilt unit to look at it with the camera. This double dynamics offers a time sharing in which the robot sequentially looks at all the objects. It also accepts search 3D positions to explore for new relevant objects.

The life of objects decreases in time and grows every time the object appears in the image. Those objects with life above certain threshold are a coherent representation of the items in the scene. Objects with life below another threshold are forgotten and discarded, preventing the robot to pay attention to objects that are no longer there. The system has some patience before forgetting an item, this way the system is robust to some false negatives due to occlusions.

Since the scene is greater than the 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 the robot knows they are there and can take them into account for better movement decisions.

Several experiments have been carried out with the visual memory, both in a real robot and in a simulated one. The robot navigates through its environment using this attentive visual memory. They show that the attention behaviors generated are quite similar to a human visual attention system.

With regard to future lines, we are working in extending the visual memory to properly represent dynamic objects. Currently the memory manages slow object movements as far as it observes the new object position close to the old one and can do the matching. Faster movements are not properly

managed, as they cause the creation of *new* objects in the new position and a ghost object remains in its old position before disappearing.

We are also working on playing with different salience dynamics for different object types. For example, let it grow faster in some objects recognized as obstacles, navigation beacons or objects very interesting for the task at hand. This would let robot to achieve a safe navigation.

ACKNOWLEDGEMENTS

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

REFERENCES

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

[Arkin, 1998] C. Arkin. *Behavior-based robotics*. MIT Press, 1998.

[Badal *et al.*, 1994] S. Badal, S. Ravela, B. Draper, y A. Hanson. A practical obstacle detection and avoidance system. *Proc. of 2nd IEEE Workshop on Applications of Computer Vision, pages 97104*, 1994.

[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.

[Gartshore *et al.*, 2002] R. Gartshore, A. Aguado, y C. Galambos. Incremental map buildig using an occupancy grid for an autonomous monocular robot. *Proc. of Seventh International Conference on Control, Automation, Robotics and Vision ICARCV, pages 613618*, 2002.

[Gerardo Carrera y Davison, 2011] Adrien Angeli Gerardo Carrera y Andrew J. Davison. Lightweight slam and navigation with a multi-camera rig. In *Proceedings of the 5th European Conference on Mobile Robots ECMR 2011*, pages 77 – 82, September 2011.

[Goldberg *et al.*, 2002] S.B. Goldberg, M.W. Maimone, y L. Matthies. Stereo vision and rover navigation software for planetary exploration. *Proc. of IEEE Aerospace conference Proceedings, pages 52025,52036*, 2002.

[Hulse *et al.*, 2009] M. Hulse, S. McBride, y 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.

[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.

[Mariottini y Roumeliotis, 2011] G.L. Mariottini y S.I. Roumeliotis. Active vision-based robot localization and navigation in a visual memory. *Proc. of ICRA*, 2011.

[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.

[Newcombe y Davison, 2010] Richard A. Newcombe y Andrew J. Davison. Live dense reconstruction with a single moving camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 1498 – 1505, San Francisco, California (USA), June 2010.

[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.

[Remazeilles *et al.*, 2006] A. Remazeilles, F. Chaumette, y P. Gros. 3d navigation based on a visual memory. *Proc. of ICRA*, 2006.

[Srinivasan *et al.*, 2006] V. Srinivasan, S. Thurrowgood, y D. Soccol. An optical system for guidance of terrain following in uavs. *Proc. of the IEEE International Conference on Video and Signal Based Surveillance (AVSS), pages 5156*, 2006.

[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.

[Vega y Cañas, 2009] J. Vega y J.M. Cañas. Sistema de atencin visual para la interaccin persona-robot. *Workshop on Interaccin persona-robot, Robocity 2030, pp. 91-110. ISBN: 978-84-692-5987-0*, 2009.

[Vega y Cañas, 2010] J. Vega y J.M. Cañas. Memoria visual atentiva basada en conceptos para un robot mvil. *Robocity 2030, pp 107-128, Madrid, September 15th 2010. ISBN: 84-693-6777-3*, 2010.

[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.