Perception planning and execution control for a visual and landmark-based navigation task

[1]Antonio Marin-Hernandez, [2]Michel Devy, [1]Fernando M. Montes-Gonzalez

[1]Facultad de Física e Inteligencia Artificial
Universidad Veracruzana
Sebastián Camacho No. 5, C.P. 91000, Xalapa, Ver., Mexico
{anmarin, fmontes}@uv.mx

[2]LAAS-CNRS
7, Av. du Colonel Roche
31400, Toulouse, France
michel@laas.fr

***Abstract* – This paper deals with indoor navigation based on visual landmarks. The robot is equipped only with a PTZ camera (pan, tilt and zoom control), used in a learning step, to detect and learn landmarks such as posters, signs or other characteristic objects in the environment. To reach a goal the robot generates a path, so it could always locate itself with respect to one or two landmarks. For this work we propose a strategy for optimally control of the PTZ camera, thus saccadic motions are minimized during the transition steps between the trajectory stages controlled by different landmarks. Depending on the position of learnt landmarks a perception planner selects along the path, the better landmarks to be tracked and to be used to control the robot motion. During the execution of this plan, some unforeseen events (occlusions, tracker disruptions, obstacle avoidance) make mandatory to modify either the perception plan or the generation of a new plan. Several tracking modalities can be selected according to the landmark characteristics, also to the current illumination conditions. The strategy we are proposing is initially validated, using a simulation environment, then, on a robot equipped with a Sony PTZ camera.**

Keywords*: indoor navigation, landmark tracking, active vision, perception planning, execution control.*

I. INTRODUCTION

Navigation is a classical topic in mobile robotics. It is sometimes considered that this problem is now solved, and many research teams are now interested on other challenges like cognitive robotics, multi-robot coordination or human robot interaction. Despite many improvements had been proposed in the past ten years, yet a robot equipped with cheap sensors, cannot execute safely complex motions on long distances in cluttered environments. In this study only indoor navigation is considered; thus, for a mobile robot only equipped with an active or PTZ camera, the pan, tilt, focus, aperture and zoom configuration can be controlled to adapt the view conditions to the current scene. On the other hand, our robot is also equipped with ultrasonic sensors used for obstacle detection and avoidance functions which are not within the scope of this paper. As a consequence we focus on different strategies to optimally control the PTZ camera when the robot executes a trajectory towards a goal.

In writing, several navigation modes have been proposed, especially metrical and topological ones. Here, only the metrical navigation approach is considered. The trajectory is a curve on the ground. A path planner requires a free space representation (typically an occupancy grid) to generate a trajectory curve on the ground. Then, the robot requires a landmark map (typically, a stochastic map) to locate itself when it executes the motion along this curve. When unforeseen obstacles are detected the robot can locally deform the curve as an elastic band.

In this paper landmarks are discriminatory objects that the robot has previously detected, characterized by an appearance-based representation and located in a world-reference frame. This learning process has been presented in [1], and during motion, landmarks must be detected and tracked along the trajectory. The tracker must switch from a landmark to another depending on the camera field of view and the robot position with according to landmarks. The navigation task using a passive camera, mounted on the robot has been described in [2] for the metrical mode also in [3] for the topological mode, and different tracking modalities have been proposed in [4]. This paper is focused on the selection of the best strategy to make more robust the navigation task using the PTZ camera with respect to the landmark positions and the trajectory.

In order to carry on our experiments two steps must be considered: planning and execution control. For the former, the robot analyses the available knowledge to decide at every point of the nominal trajectory what the optimal camera configuration is. In [5], this perception planning function has been presented: some simulations were shown to validate the selected strategy. Section 3 summarizes our contribution and proposes some improvements. The section 4 is concerned with the execution of the perception-plan when the robot moves along the path considering different events that make mandatory an adaptation of the original plan or the generation 'on the fly' of a new one. Then, section 5, presents experimental results. Finally, in section 6 we offer a discussion related to our approach and a comparison of to the state of the art. Below, the overview of this navigation approach based only on monocular vision and on landmarks is explained.

II. Overview on our navigation approach

In relation to visual navigation many works are devoted to visual servoing with a trajectory defined as a path in an image data base [6] or as a sequence of vision-based motions [7]. Here, our trajectory is defined as a geometrical curve in the task space (a 2D curve on the ground plane). This so-called metrical approach requires several environmental representations that must incrementally be built during an a priori learning stage. The robot executes a SLAM function [8] to learn the visual-landmark positions with respect to a world-reference frame, and builds a global occupancy grid using a range sensor to learn the free space model. This representation is shown in Fig.1

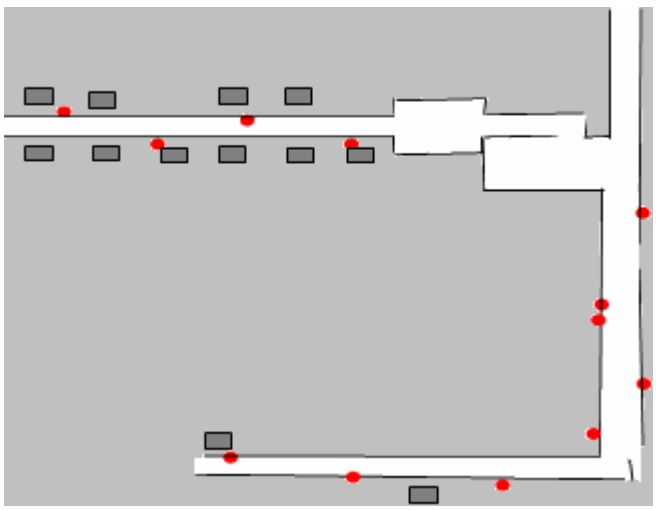


Fig.1. A part of our environment model: free space is white; red spots are detected posters; grey rectangles are detected doors (not used here).

When the robot has to move in the environment towards a given goal, a two-steps planning function is executed to generate a landmark-based trajectory. First a path planner selects a trajectory as a curve in the free space, from the current robot position to the current goal. Then a perception planner selects at every point of this curve, at most, two landmarks to be tracked by the PTZ camera and to be used for the robot localization. Next, the robot path is split in successive trajectory pieces defined by one or two landmarks. It could be considered as a trajectory in the PTZ camera configuration space.

The execution control of a landmark-based trajectory, must tackle several problems:

- Unforeseen obstacles on the trajectory: in our context, a person sharing the task space with the robot, or an object (such as a chair…) that has been moved after the learning step. The elastic band concept is used to deform the initial path; the tracking plan must be adapted or generated again.
- The initialization of successive landmark trackings: on every trajectory piece, the robot must track one or two landmarks. When approaching the end of the current piece of the plan; a switching function is activated to search the next landmarks to be tracked and to select the best tracker modalities with respect to current illumination conditions.
- The tracker control that is required to recover from any tracking disruption. For example, tracking can be disrupted for many reasons: in our context of a mobile camera used to track static landmarks, the most commeon are occlusions (another object hides the landmark from the current camera position); distraction (another object very similar from the one to be tracked, produces some ambiguities); and large target image motions due to large camera motions (mainly, during robot rotations).
- The execution on-line of a landmark-based localization function to measure the error between the current robot position and the estimated on the trajectory. The control law minimizes this error.

In previous work we have presented, several tracking modalities [4, 9]. Here the robot uses only for localization planar quadrangles (mainly posters, Fig.2), so only the template tracker is executed. In our work, the prediction step of the tracker can be simplified: the target image motion is predicted from the estimated camera motion given by the odometry and the gyro from robot motion (X, Y, and heading), also by encoders from the optical axis and the focal length.

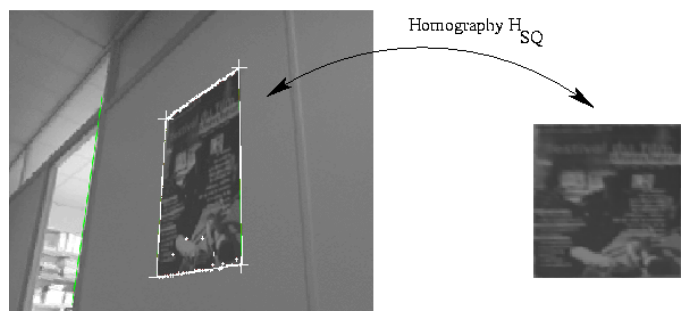


Fig.2. A poster and its invariant representation used for both tracking and localization.

In the two following sections, the perception planner and the execution controller for a trajectory are presented.

III. Perception Planning

A. *Landmark Visibility*

To estimate the current robot position, we use quadrangular-planar landmarks. A quadrangular planar landmark **L** can be represented by its four corners $\mathbf{p}_j$ with $j \in [1, 4]$. Contrary to the bi-valued visibility function commonly used, the visibility is defined as a continuous real valued function.

Be **x,** the observer position in 3D space, then the visibility V_j for each corner $\mathbf{p}_j$ of the landmark is defined by:

$$V_j = \cos(\phi_j) \tag{1}$$

where ϕ is the angle between vectors **d** = **x** - **p** and **N** is the normal vector to the planar surface (Fig 3).

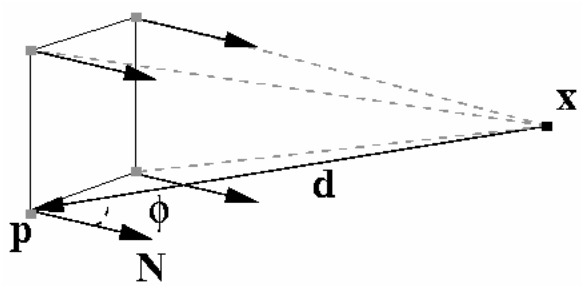


Fig. 3.Vectors and position taken into account for the visibility of a planar landmark.

Be $\mathbf{L}_i$ = {$\mathbf{p}_{i1}$, $\mathbf{p}_{i2}$, $\mathbf{p}_{i3}$, $\mathbf{p}_{i4}$}, the set of corners of a quadrangular landmark *i* with normal vector $\mathbf{N}_{Li}$. The visibility V_{Li} of the landmark $\mathbf{L}_i$ is quantified as:

$$V_{Li} = \frac{\sum_{j=1}^{n_p} V_{ij}}{n_p} \tag{2}$$

where V_{ij} is the visibility of the *j*-th corner of the landmark *i*, and n_p is the number of points considered, here $n_p = 4$.

If visibility value $V_{Li} = 0$, it means that the line of sight is parallel to the planar surface. Negative visibility values means a rear viewpoint.

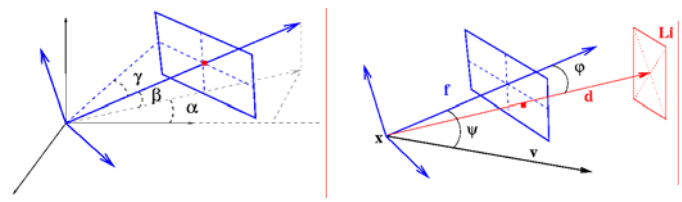
With this definition, visibility depends only on the angle between the orientation (normal vector) and the observer position, whatever the distance is to the target point. It means mainly that, proper visual characteristics of the observer (e.g. the visual sharpness) are not considered by this criterion. Therefore, depending on camera parameters, a point can be visible but not perceptible. Proper visual characteristics will be later incorporated in utility measurements.

It is important to note that, for a planar landmark, criterion (2) gives its optimal value (V_{Li} = 1.0), at the frontal position to the normal vector and at the infinite distance from the landmark. However, as it could be proved, in a frontal view when the distance from observer to target is greater than the target dimensions in one order of magnitude, visibility values are greater than 0.995, which are very acceptable values.

B. Landmark Utility

The utility function of a given landmark $\mathbf{L}_i$ is the product of various terms *K*. Each one of these *K* terms is joined to some criterion that depends on some characteristics: viewer orientation, visual sharpness, landmark size and/or specific criteria for the given task. The *K* terms are normalized between 0 (null utility) and 1 (optimal utility).

Two terms are always used: the first one to assess the size of the landmark on the viewer's image, and the other one to compare the alignment between the optical ray camera-landmark and the optical axis (camera orientation). Fig. 4 shows the camera model and variables used for utility computation.



(a) (b)

Fig. 4. Camera model: a) internal angles, α = pan angle, β = tilt angle and γ = view field (zoom), b) external angles φ angle between **f** and **d**, ψ angle between **v** and **f**.

The size of the landmark projected on the image depends on the dimensions of the landmark, and then directly on the possible configurations of the field of view, know as the zoom (parameter γ). The field of view angle is measured on the horizontal line of the camera-reference frame. Vertical angle is typically ¾ of γ angle.

The first criterion K_1 is defined by:

$$K_1^{Li} = 1 - \frac{\left|\gamma - \gamma_i^*\right|}{\max_{\gamma}\left|\gamma - \gamma_i^*\right|} \tag{1}$$

where | | represents the absolute value, γ is the field of view of the camera (Fig. 2a) and γ_i* is the optimal field of view at the current distance for the landmark *i*. Optimal field of view depends on the visual task, here the landmark localization and tracking. It must be defined from a statistical analysis of many experiments: typically, γ_i* will be selected so that the landmark projection must be large enough to improve the localization accuracy, but cannot be too large to make more efficient the tracking. This term could be modified to give a penalty for extreme positions of the zoom. Thus, the system will prefer the close landmark with a mean zoom than the further one with the minimal field of view.

It is important to notice that, although the line of sight observer-landmark exists (where visibility is not zero), the landmark could be out of the image [i.e. the optical axis points to another direction (Fig 2b)]. Our second utility term considers the angle between these two vectors, and then is defined as:

$$K_2^{Li} = \cos\varphi_i \tag{3}$$

where φ is the angle formed between the optical axis **f** and the line of sight vector for the *i*-th landmark $\mathbf{d}_{Li}$. The utility is maximal when the two vectors are aligned (no specularity).

The utility function is a product of the different *K* terms. Considering a navigation task, a third utility term, presented in [5], promotes configurations where the optical axis and the next robot orientation are aligned, so that the camera

anticipates a robot rotation. Finally, the utility for a landmark $\mathbf{L}_i$ at position $\mathbf{x}_t$ on the path will be expressed by:

$$U_i^{\mathbf{x}t} = \left(\prod_{j=1}^{m} K_j^{Li} \right) V_i(\mathbf{x}_t) \qquad (4)$$

where m is the number of utility terms, in our case m = 3. A better visibility of $\mathbf{L}_i$ from configuration $\mathbf{x}_t$, improves the utility of this couple ($\mathbf{L}_i$, $\mathbf{x}_t$).

If two or more landmarks are visible from a given position $\mathbf{x}_t$, then the utility for this set of landmarks is the sum of utility functions for each one of them:

$$U_c^{\mathbf{x}t} = \sum_{i=1}^{n} U_i^{\mathbf{x}t} \qquad (5)$$

where U_i is the individual utility of the landmark $\mathbf{L}_i$.

C. Visual Planning

To find the best camera modality configuration along the trajectory, the path is sampled in uniform pieces, limited by points ($\mathbf{x}_0 \ldots \mathbf{x}_t \ldots \mathbf{x}_N$). All the subsets of visible landmarks are generated at a given point of view $\mathbf{x}_t$, and then the utility for each subset is computed. In order to reduce complexity of the planner and to be realistic with respect to the real time constraints, the maximal number of elements per subset is limited. Here it is considered that simultaneously the robot tracks only two visible landmarks.

After some mathematical manipulation, the optimal Camera-Landmark alignment for the pan and tilt angles, in the general case, is obtained by:

$$\alpha = \tan^{-1}\left(\frac{\sum_{i=1}^{n_v} C_i \sin\alpha_i}{\sum_{i=1}^{n_v} C_i \cos\alpha_i} \right), \quad \beta = \tan^{-1}\left(\frac{\sum_{i=1}^{n_v} C_i \sin\beta_i}{\sum_{i=1}^{n_v} C_i \cos\beta_i} \right) \qquad (1)$$

where n_v is the number of landmarks at the current subset, $C_i = K_{Li}{}^1 V_{Li}$, and α_i and β_i are the pan and tilt angles of $\mathbf{d}_i$.

Once the utility-computation is made, for all subsets of each sampled point along the path, is possible to find the set of camera-modalities for the entire path that maximizes utility. However, the modality plan does not assure a smooth transition between modalities.

In order to avoid erratic motions of the camera and its zoom (saccadic movements) that could perturb tracking landmarks and its detection. The modality plan is improved by a dynamic programming algorithm. For each point $\mathbf{x}_t$, all the possible sets of the visible landmarks are determined (one or two). A graph is created where such a set is a node; edges are created between all the sets created for $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$, and labeled with a cost which is proportional to the distance between the two related configurations of the PTZ camera. The optimal perception plan will minimize the global cost of the path in this graph, i.e. will minimize the camera motions (Fig 5).

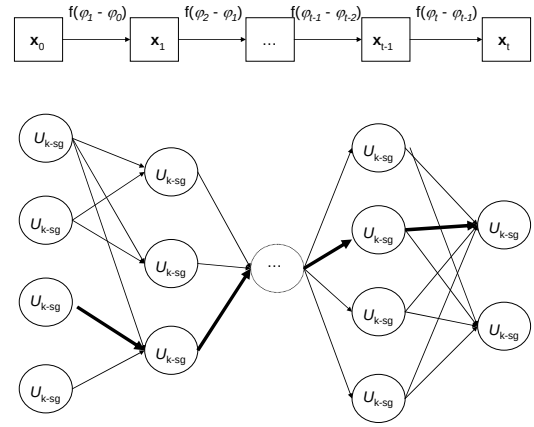


Fig. 5. Generation of the perceptual plan by a dynamic programming algorithm, dark arrows represent the optimal path.

The complete strategy is summarized as follow:

```
Generation of the planned path
Path sampling (x ={x1,…,xt,…})
For each xt
  Calculate the subsets C of visible landmarks
  For each subset
     Calculate the total utility Uc
     If t > 0 then
        Compute the cost to go from all subsets
        at t-1 to the current subset
Apply dynamic programming to get the optimal visual
path.
```

IV. Execution Control of the Perception Plan

Many authors have proposed reactive strategies in order to optimize the utility of an active camera. Take for instance the work of Toyoma [11, 12, 13] and Rasmussen [14], which propose several probabilistic models to select the best target to be tracked and the best tracker modalities. Our approach is different because is a combination between the planning step and the execution control, hence the on-line selection is limited up to the recover-point where is possible to cope with unforeseen conditions.

Once the optimal perception plan has been generated, a post-processing is applied to the group of successive points $\mathbf{x}_t$ associated to the same set of landmarks. Finally, the camera trajectory is expressed as a sequence of switching positions between trajectory pieces:

$$\mathbf{S} = (\mathbf{s}_0 \ldots \mathbf{s}_t \ldots . \mathbf{s}_P), \text{ with } P < N.$$

On each piece ($\mathbf{s}_t$, $\mathbf{s}_{t+1}$), one or two landmarks ($\mathbf{L}_{t1}$, $\mathbf{L}_{t2}$) have been selected to be tracked (L_{t2} could be NULL).

Next the architecture of the path controller is described. Then, the fuzzy controller for the PTZ camera is presented, and finally landmark switching and error recovery.

A. The path controller.

The execution control is implemented by six modules presented on figure 6.

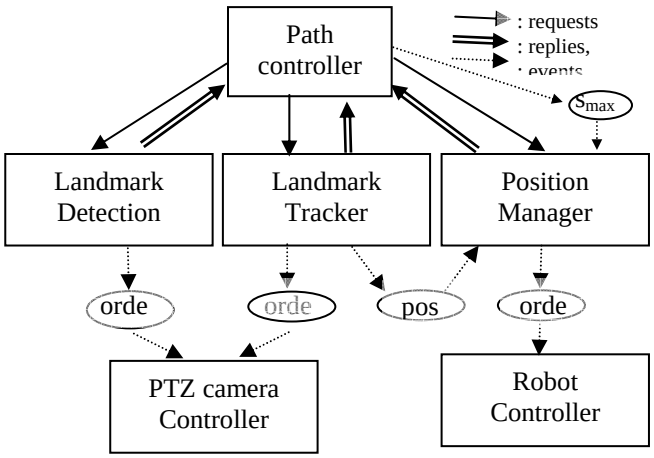


Fig. 6. Modular architecture for the execution control of a perception plan.

The Path Controller module is an asynchronous module that initially receives the perception plan. It activates the Landmark Tracker, and the Position Manager modules, and in turns it sends requests to the Landmark Detection module. Then, the path controller waits for events or replies generated by all these modules.

The Landmark Tracker module is running a main loop at 5Hz (even when the template tracker, used to track a poster, can be executed at 15Hz). This tracker receives the labels ($\mathbf{L}_{t1}$, $\mathbf{L}_{t2}$) of the landmarks that has to track. Initially, it adapts the tracker modality with respect to the current illumination conditions. At each period, an image is acquired and the Tracking function is executed:

- If at least one landmark is found, it is exploited by two functions. (1) The Locate function computes the current robot position from one or two tracked landmarks. This position is written in a shared-memory read by the Position Manager module. (2) The Center function computes new orders for the PTZ camera, so the landmarks are centered in the image and seen with optimal resolution. Orders are written in the shared-memory read by the PTZ Camera Controller module.
- If tracking fails, then the Landmark tracker sends a reply to the path controller and then it stops.

The Position Manager module is periodic (25Hz, like the Robot Controller module). After its launching, it receives the nominal trajectory to be executed. At each period, the Position Manager estimates the robot position from data provided by the odometry and landmark-based localization. Then, it generates orders for the Robot controller module with respect to the error and the nominal trajectory. The Position Manager reads in a shared-memory a position $\mathbf{s}_{max}$ that has to be monitored; if the robot position (expressed as a curvilinear abscissa on the path) overpasses $\mathbf{s}_{max}$ then an event is sent to the Path Controller module.

The Landmark Detection module is an asynchronous process. When approaching $\mathbf{s}_{t+1}$, the path controller sends a request in order to search the next landmarks to be tracked ($\mathbf{L}_{t+1,1}$, $\mathbf{L}_{t+1,2}$), while the tracker still continues on ($\mathbf{L}_{t1}$, $\mathbf{L}_{t2}$). The Landmark Detector module generates orders to the PTZ camera to enlarge the current field of view until the new landmarks are visible.

The PTZ Camera Controller is a periodic module. It reads orders from Landmark Tracker and Detection modules. This controller selects executive orders for the pan, tilt and zoom configurations. Normally, the path planner selects a landmark sequence where conflicts are avoided: fields of view required by the two modules, must overlap.

B. *A fuzzy controller for the PTZ Camera.*

The fuzzy controller is divided in two main parts: One for the control of the zoom or field of view angle, and the second to control pan and tilt angles (fig. 7)

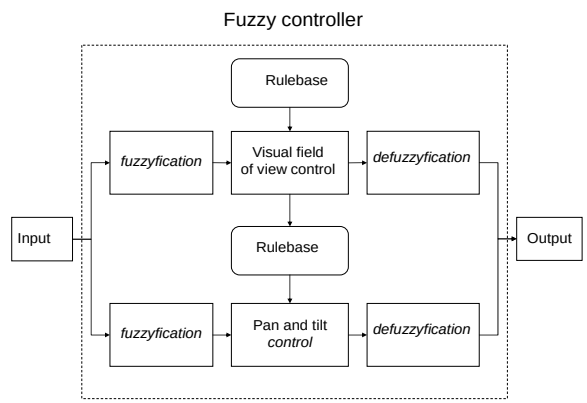


Fig.7. Fuzzy controller

In order to detect and track objects in current images, a proper image-size is required. The first part of the controller is dedicated to this task. We use as input data the difference (or error) between the current and the desired size and the rate of change of the diagonal in the bounding-box containing the objects within the image (e_d, $\dot{e}_d$). The output variable is the speed v_γ which should be applied to the camera encoders to arrive to the desired bounding-box size on the image. These input and output variables are quantified in tree states or linguistic descriptors (table I).

TABLE I
QUANTIFICATION OF INPUT AND OUTPUT VARIABLES

	Input		Output
	e_d	$\dot{e}_d$	v_γ
Linguistic Descriptors	Negative	Negative	Negative
	Zero	Zero	Zero
	Positive	Positive	Positive

The rule base is generated combining all possible values of linguistic descriptors for input variables, resulting in a 3x3 rule matrix giving a possible value for the output variable as is shown in table II.

TABLE II
RULE MATRIX

		e_d		
		Negative	Zero	Positive
$\dot{e}_d$	Negative	P	Z	P
	Zero	P	Z	N
	Positive	N	Z	N

The output value for the output variable is obtained by *RSS* (*Root-Sum-Square*). This method guarantee values for the output variable all over the continuous output interval.

The second part takes into account the position of the center of the bounding box (x, y), and the rate of change as input variables. Linguistic descriptors are divided in five states, given as a 5x5 matrix rule, for each one of the speeds given to motors that drive the pan and tilt angles. This last part also depends on results from the previous zoom part as is show in fig. 7.

C. *Landmark switching*

A simplified pseudo-code of the path controller is given below. We assume that only one landmark is detected and tracked, and then for each trajectory piece ($\mathbf{s}_t$,$\mathbf{s}_{t+1}$) a list of landmarks is sorted according to their utilities, has been determined by the perception planner, so that the recovery procedure for a tracker disruption or a detection failure, consists in searching the next landmark in the list.

TABLE III
SIMPLIFIED PSEUDO CODE

```
t=0 ; n=1; LandmarkDetector.Search (L01) ;
PosManager.Activate (Trajectory);
Smax = s1 - δ;

Do
    Wait (event);

    If (event == 'Detector ok') then
        LandmarkTracker.Activate (Ltn)

    Else if (event == 'Tracker disruption') ||
            (event == 'Detector failure')  then
        n = n+1; LandmarkDetector.Search (Ltn);

    Else if (event == 'Transition')
        t = t+1;
        if (t ≠ P) then
            n = 0; Smax = st - δ;
        LandmarkDetector.Search (Ltn);

Until (t == P)
```

The parameter δ depends on the robot speed and on the field of view change required during a transition; it could be adapted on-line. Some typical transition configurations are shown on figure 8: if many landmarks have been detected in the environment, the same landmark is tracked during two trajectory pieces, e.g. ($\mathbf{L}_1$, $\mathbf{L}_2$) tracked, then ($\mathbf{L}_2$, $\mathbf{L}_3$), then ($\mathbf{L}_3$, $\mathbf{L}_4$)…

Fig.8. Landmark switching: two configurations with the current tracked landmark projected close to the image center, and the next one, partially viewed.

V. EXPERIMENTAL RESULTS

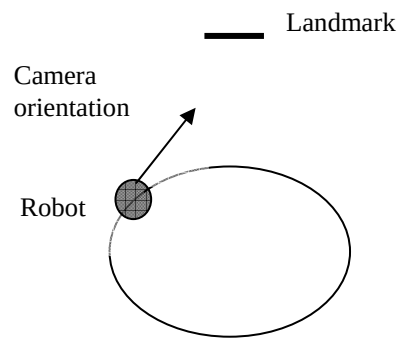


Fig.9: Validation of the PTZ Camera Controller module (top view)

First, our navigation approach has been validated using simulations, with a generator of synthetic images on a virtual world. These results have been presented in [5]. Then, experimental results were obtained using a Nomadic robot equipped with a Sony EVI-D31 camera. Transitions between different trackers were described in [4]. The metrical and landmark-based navigation mode have been proven valid by different tests. Firstly, a circular trajectory in an open space was executed, which always keeps a landmark in the image center with the same resolution (Fig.9). Secondly, the robot executes a trajectory in a corridor with the detection and tracking of sequential landmarks along the corridor walls (Fig.10).

Fig 9: The navigation in the top corridor (Fig.1) using sequential landmarks detected along the corridor.

VI. Conclusion

In this paper, we present a metrical landmark-based approach for the planning and control of indoor navigation. An environmental model is built during an initial learning step: where a landmark-based map and the free space model are made. The robot is equipped with a PTZ camera, used for detection, characterization, recognition and tracking of landmarks. Our study emphasizes the use of a tracking function and a camera control that are required when a planned path is executed to keep landmarks in the current processed image. Detection of sequential landmarks set the optimal view conditions whatever the context: obstacles, occlusions, illumination problems…

Further experiments are in progress to validate this approach when the robot executes several times a loop. We are planning to develop a new model where the world reference-frame is avoided and the definition of the whole trajectory is determined as connected pieces related to particular landmarks sets. Finally, it is important to notice that such a definition will require different representations that avoids the use of a global stochastic map, and will exploit a topological graph together with other planning strategies.

Acknowledgment

The authors thank J.B. Hayet and F. Lerasle for results exploited in this paper (Fig.1) on landmark detection and robot localization.

References

[1] J.B. Hayet, F. Lerasle, and M. Devy. “Visual landmarks detection and recognition for mobile robot navigation”, Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'2003), Madison (USA), Vol.II, pages 313--318, June 2003.

[2] V. Ayala Ramirez, J.B. Hayet, F. Lerasle, and M. Devy “Visual localization of a mobile robot in indoor environments using planar landmarks”, Proc.2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'2000), Takamatsu (Japan), November 2000.

[3] P. Ranganathan, J.B. Hayet, M. Devy, S. Hutchinson and F. Lerasle. “Topological navigation and qualitative localization for indoor environment using multisensory perception”, Robotics and Autonomous Systems, Vol.41, p.137-144, November 2002.

[4] A. Marin Hernandez, V. Ayala Ramirez, and M. Devy. “Collaborative tracking for robotic tasks”, Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'2003), Las Vegas (USA)}, pages 272--277, October 2003.

[5] A. Marin Hernandez, M. Devy, and V. Ayala Ramirez. “Visual planning for autonomous mobile robot navigation”, Proc. Fourth Mexican Int. Conf. on Artificial Intelligence (MICAI’2005), Monterrey, Mexico, November 2005.

[6] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control" IEEE Trans. on Robotics and Automation, 18(4):534-549, August 2002.

[7] R. Pissard-Gibollet, K. Kapellos, P. Rives and J.J. Borrelly, "Real-Time Programming of Mobile Robot Actions Using Advanced Control Techniques", 4th Int. Symp. on Experimental Robotics (ISER’95), Stanford, USA, June 1995

[8] R. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty”, Int. Journal of Robotics Research (IJRR), 5(4):56--68, 1987.

[9] A. Marin-Hernandez, M. Devy, and V. Ayala-Ramirez. “Comparison and evaluation of real-time vision tracking methods for autonomous mobile robot tasks”, Proc. 3rd Mexican Int. Conf. on Artificial Intelligence (MICAI'2004), Morelia, Mexico, 10p., April 2004.

[10] M. King, B. Zhu, and S. Tang, “Optimal path planning,” *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.

[11] H. Tagare, K. Toyama and J.G. Wang, "A Maximum-Likelihood Strategy for Directing Attention during Visual Search", IEEE Trans. on Pattern Analysis and Machine Intelligence(PAMI), 23(5), pp.490-500, 2001

[12] K. Toyama and A. Blake, "Probabilistic Tracking With Exemplars in a Metric Space ", Int. Journal of Computer Vision (IJCV), 48(1), pp.19, 2002.

[13] K. Toyama and G. Hager, "Incremental focus of attention for robust vision-based tracking", Int. Journal of Computer Vision (IJCV), 35(1), pp.45-63, November 1999.

[14] K. Rasmussen and G. Hager, "Joint Probabilistic Techniques for Tracking Multi-Part Objects ", IEEE Trans. on Pattern Analysis and Machine Intelligence(PAMI), 23(6), pp.560-576, 2001.