Visual Based Localization for a Legged Robot*

Francisco Martín, Rafaela González-Careaga, Pablo Barrera, Jose María Cañas and Vicente Matellán
GSyC, Escuela Superior Ciencias Experimentales y Tecnología
Universidad Rey Juan Carlos, Móstoles, Madrid
Email: {fmartin,rafaela,barrera,jmplaza,vmo}@gsyc.escet.urjc.es

***Abstract*— This paper presents a visual based localization mechanism for a legged robot. Our proposal, fundamented on a probabilistic approach, uses a precompiled topological map where natural landmarks like doors or ceiling lights are recognized by the robot using its on-board camera. Experiments have been conducted using the AIBO Sony robotic dog showing that it is able to deal with noisy sensors like vision and to approximate world models representing indoor office environments. The two major contributions of this work are the use of this technique in legged robots, and the use of an active camera as the main sensor.**

***Index Terms*— mobile robot navigation, localization, legged robots**

I. Introduction

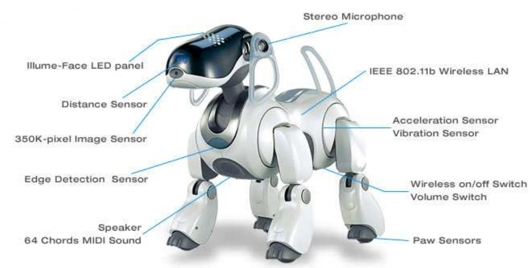
One of the basis for mobile robot operations, like position based navigation or motion planning techniques, is the localization capability [1]. This can defined as the ability of a robot to determine its position in a map using its own sensors. Many works have been developed to estimate the robot location. Unfortunately, the most of existing algorithms have been designed for robots equipped with wheels, where local precise odometric information can be achieved.

The solution we present solves the problem for a legged robot were odometric information is not reliable. The information needed for the localization has been obtained by the robot external sensors, mainly from the robot's camera. The odometry is a key in the majority of the localization works with wheeled robots ([2], [3]). A similar approach to ours, but applied to a wheeled robot, is exposed in [4], where an office environment is divided topologically into states.

Our approach is based on *Markov localization* [5], a well-known probabilistic technique that maintains a probability density (belief) over the entire states space where the robot moves. This technique is also used in many others works as [6], where sensor as ultrasonic or infrared sensors are use to determine the obstacles around the robot and [7], but always applied to wheeled robots. In the first case with sensors as ultrasonic or infrared sensors to determine the obstacles around the robot.

In section II the fundamentals of the implementation of this technique is shown. There are another approaches that use another well-know sampling algorithm for localization *Monte Carlo*, but it has reported [8] not to be enough effective in noisy environments as the ours. Despite of this, we have not discarded to evaluate this algorithm in the future.

This work has been partially financed by Spanish Government under project DPI2004-07993-C01-01



(a) Sony AIBO ERS7. Front (image from AIBO site)

(b) Sony AIBO ERS7. Back (image from AIBO site)

Fig. 1. AIBO ERS7 Anatomy

Depending on the state granularity defined over the map, the connectivity information and the angles inference, we consider a localization method as *topological* or *metric*. For instance, in [5] a metric approach is used, with a linear resolution between 10 and 40 cm and the angular resolution set between 2 and 5 degrees. This resolution leads to a vast amount of states, even in not so big maps (7.200.000 states for $30x30m^2$ map) which need some techniques like sampling for being computable.

Our approach is *topological* because the connectivity information is important for our navigation task, and the exact robot position is not so important . The office environment division into states is similar to [9] and [10], where the set of nodes is built depending on the observations that can be obtained in each place of this environment, but again, these approaches are used to locate wheeled robots.

With very few exceptions ([5]), most of the approaches are *passive*, i.e. neither the position nor sensor orientation can be controlled. Our approach is *active*, setting the sensor orientation to get the information we wish from the environment.

The platform where this system has been developed is the Sony AIBO robot. The Sony AIBO ERS7 robot (figure I) is a completely autonomous robot which incorporates an embedded MIPS processor running at 576MHz, and 64MB of main memory. It gets information from the environment mainly through a 350K-pixel color camera and 2 infrared sensors. Another of the AIBO locomotion main characteristics is its dog aspect with four legs. AIBO also incorporates IEEE 802.11b Wireless LAN card.

The main reason to choose this robotic platform for our research is its generalization as low cost legged robotic platform. Our group is committed with the use of common platforms and the availability of source code, letting research claims be checked by peers.

The rest of this paper is organized as follows: in section II we make a brief review of Markov Localization technique used. In section III we describe our model and its components, showing the experiments and results in section IV. Finally, we will expose our conclusions in section V.

II. Markovian localization framework

Localization based on indirect information provided by the robot sensors (sonar, laser, etc.) has been successfully integrated in the probabilistic framework and has shown good results [2], [11]. In particular, sampling methods that speeds the estimation [5], are currently the most popular methods.

In our work, we have used a Partially Observable Markov Decision Processes (POMDP) where a probability distribution Bel, over all the possible locations $S = \{s_1, s_2, ...\}$, is defined at a time t, so $Bel_t(S = s)$ will represent belief of being in the state s at the time t.

Depending on the knowledge about the initial localization of the robot $Bel_0(S)$ will be uniformly distributed if the initial state is not known, or will be centered in a state if the initial position is known.

The belief actualization is divided in two atomic steps. In the *movement step* an action is executed by the robot. The belief is modified according to the action executed. In the *observation step* the belief is updated according to the observations taken from the sensors. In each robot movement these two steps are executed sequently. The description of these two steps is presented as follows:

Movement step. Robot motion is modelled by the probability $p(s'|s, a)$. This is the probability of reaching state s' if an action a is executed at state s. To obtain the a priori belief for the whole set of states $Bel_t(S')$ Bayes update is assumed. When an action is executed, and before it is corrected by the data from the sensors, we apply:

$$Bel_t(s') = \sum_{s \in S} p(s'|s, a) \cdot Bel_{t-1}(s), \quad \forall s' \in S \tag{1}$$

Observation step. To calculate the corrected belief $Bel_t(S)$ we take $p(o|s)$ as the probability of getting the observation o being in the state s and we operate, as it is described in [4], as follows:

$$Bel_t(s) = p(\mathbf{o}|s) \cdot Bel_t(s'), \quad \forall s, s' \in S \tag{2}$$

If there are many observations and they are independents between them, we can use them as a product of independents terms:

$$Bel_t(s) = p(o_1|s) \cdot p(o_2|s) \cdots p(o_n|s) \cdot Bel_t(s'), \forall s, s' \in S \tag{3}$$

Obviously, $p(o|s)$ has to be known. In our case, this information is inferred from the map of the environment. The way it is calculated is described in next section.

III. Our Model

Summarizing, our localization method needs three components to be defined:

1) The environment map and how it is translated to a set of states.
2) A set of actions the robot can perform and the probabilistic action model.
3) A set of observations a robot perceives from its environment and its probabilistic model related to states.

A. The state space

Our robot is going to move around in an indoor office environment made up by corridors and rooms, as shown in figure 2. This environment has been manually coded as a set of topological nodes representing places with similar characteristics. For example, a node could be a continuous corridor region where there isn't doors in the right neither in the left.

Once the set of nodes has been defined, each node is divided in four different states, representing the same robot position with four orientations with angular resolution of 90º (north, east, south, and west).

In the top left diagram of figure 2 we can see a portion of office environment. At the top right of the same figure it has been divided in nodes attending to their characteristics. So, a region where there are doors at both sides of the corridor is different to the region where there is only one or to the region there is none. The division is guided by the number of doors or ceiling lights the robot can sense in each position, because they are the landmarks the robot is able to perceive from its raw camera images.

Note this is a topological division, where nodes represent areas of different sizes. States are then defined over the nodes with different orientation. In this way, for example, in figure 2 node 4 is divided into states 4 to 7, node 5 into states 8 to 11 and so on (lower part of figure 2):

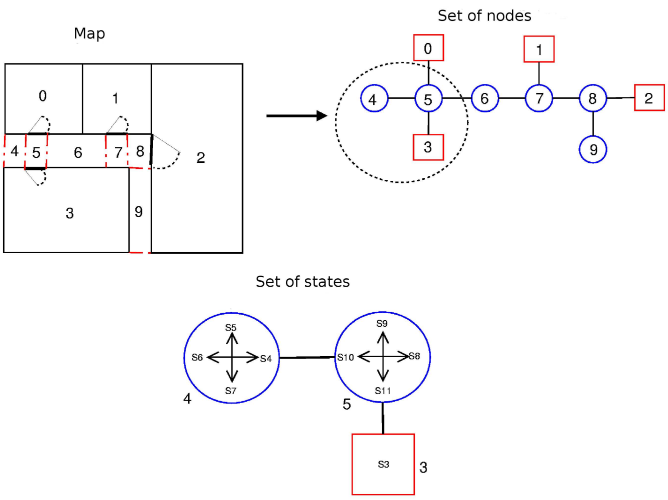


Fig. 2. From map to states

B. Action model

The action primitives we implement in this work are: to turn 90º on the left, to turn 90º on the right and go forward the necessary time to reach the next state with the same orientation. The model supports uncertainly in the movement primitives. In figure I we can see the uncertainly in action execution. If the robot execute the turn 90º in the left, for example, we have in mind that the robot could do nothing(N), to turn correctly(T) or to turn more than 90º(TT). If the movement is go forward, the robot could do not reach the next state nothing(N), reach the next state (F), to pass slightly the correct state (FF) or to make a really big travel (FFF).

Turn Left	N: 0.15	T: 0.70	TT: 0.15	
Turn Right	N: 0.15	T: 0.70	TT: 0.15	
Go forward	N: 0.20	F: 0.6	FF: 0.15	FFF: 0.05

TABLE I

UNCERTAINLY IN ACTION EXECUTION.

In figure 3 two nodes from the map are shown: the robot starts at position A and performs the action $a_{forward}$. The robot moves forward, but with this movement it does not reach the next state, being in position B instead of C. In the forward action, to do nothing (N) do not implies the robot has not moved. It implies the robot has not reached the next state, but in the model this is taken as the same.

When the robot execute a action primitive, i.e. when the robot moves (this is called *actuation phase*) our system updates the belief as it is shown in equation 4. The action

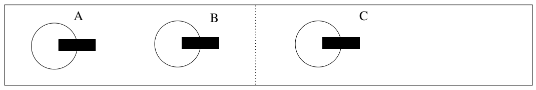


Fig. 3. Transition of states

model defines $p(s'|s,a)$ as the probability of to reach state s', starting at state s and executing the action a:

$$p(s'|s,a), \forall s \in S, \forall a \in A \quad (4)$$

$$A = \{a_{\{F\}}, a_{\{T_L\}}, a_{\{T_R\}}\}$$

This probability $p(s'|s,a)$ will be our action model and it is calculated a priori depending on the possible action the robot can perform in the state space and the table I.

The robot must be centered in the corridor as much as possible, and oriented to the wall or to the corridor in order to a correct observation of the environment. If we turn on the left 90º, we want the action to be *atomic*, i.e. turn 90º or turn 0º. Mov 60º, for example, is not desirable for the sensing tasks. For correcting orientation errors, after an action was performed, a correction phase is needed.

To improve action primitives accuracy the robot correct its position after an action is executed. For this purpose, the robot obtain infrared measures with distinct angles respect to its body position, turning the head the angles we want. There are two possibles situations depending on the first infrared measure taken with 0º of deviation with respect the body: if the robot does not detect any obstacle, we suppose to be oriented faced to the end of corridor, almost pararell to wall. On the other hand, if we detect any obstacle, we consider it faced to wall. Let's describe these situations:

Faced to wall. If the robot is faced to corridor, we obtain infrared measures getting the distance to an obstacle in -30, -15, 0, 15 and 30 degrees (see figure III-B) with respect the body, turning the dog's neck to orientate the infrared sensor. As we see in figure 6, we can obtain the 2D points of the obstacles obtained in their measures. By the minimum quadratic method, we can obtain the angle with the wall which will be used for correcting the robot position.

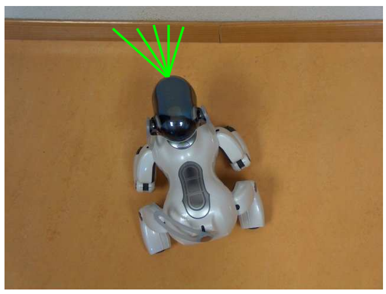


Fig. 4. The robot takes infrared measures when it is faced to wall with -30, -15, 0, 15 and 30 degrees

Face to corridor. The method is the same as before, but the angles are -90, -80, -70, 70, 80 and 90 degrees (see figure III-B) . We can obtain two lines corresponding to the walls of the corridor. With these lines we can correct the robot orientation and position with respect the corridor.

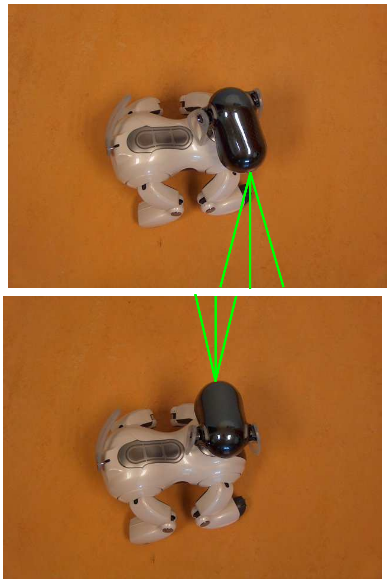


Fig. 5. The robot takes infrared measures when it is faced to corridor with -90, -80, -70, 70, 80 and 90 degrees

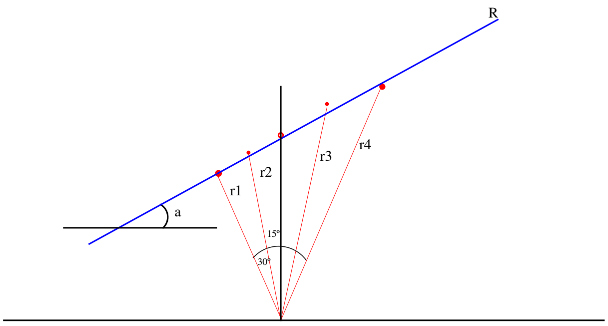


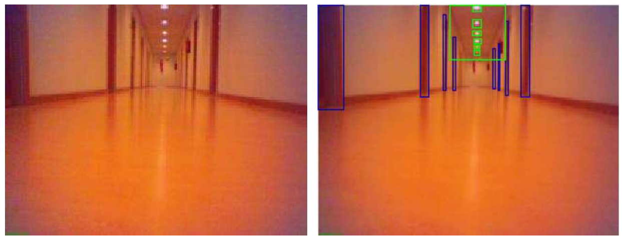
Fig. 6. When we have calculated the 2D points corresponding to sensor measures, we calculate by Minimum quadratic method the line corresponding to an obstacle. The angle a will be used to correct the robot position

C. Sensor Model

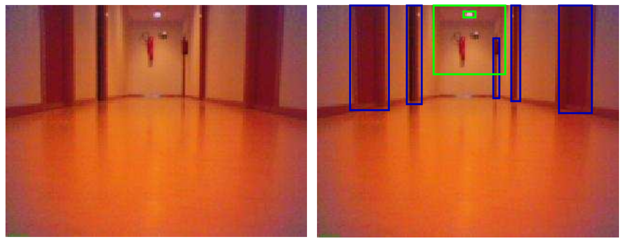
Our sensor model take three types of sensations from the image taken by the robot's camera: textbfdepth. The main target for this observation is measure how far the robot is from the wall when it is orientated to the end of the corridor. For this purpose we detect the number of ceiling lights that the robot perceive. If the number of ceiling lights is high, the robot is far from the end. If this measure is low, the robot is near to the end. In figure III-C we can see the original image and the image with the ceiling lights and doors detected.

Doors. Due a color analisys of the image, the robot is able to count the number of doors it can observe ahead. The doors must be vertical to the floor and the jambs must be pararell between them. If a image region comply with these specifications, it is assumed to be a door.

near landmark. This observation give us information about which landmarks the robot found around itself. We define landmarks as the doors, walls or corridor that are situated in the right, left and front side of robot. For



(a) Detecting 6 ceiling lights and 8 doors



(b) Detecting one ceiling light and 5 doors

Fig. 7. Image information extraction results

example, a observation could detect there is a door at the left side, a wall at the right side and in front of the robot there is a corridor.

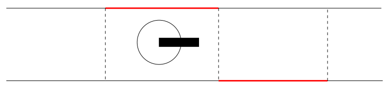


Fig. 8. Detecting the landmarks surrounding. In this situation the robot must know it is face to the corridor, it has a wall on its right and it has a door on its left

To take this observation, the robot detects if there is an obstacle ahead with an infrared sensor. If there is one obstacle, it must determine if it is a wall or a door by analyzing and image taken from its camera. The robot repeats the same operation turning its head left and right.

Once the data is collected, we apply the equation 3 for correct the belief as follows,

$$Bel_{subsequent}(s) = p(\mathbf{o}|s) \cdot Bel_{previous}(s), \forall s \in S \quad (5)$$

$$\begin{aligned} Bel_{subsequent}(s) &= p(o_{ceilinglights}|s \cdot \\ & p(o_{doors}|s) \cdot \\ & p(o_{nearlandmarks}|s) \cdot \\ & Bel_{previous}(s), \forall s \in S \end{aligned} \quad (6)$$

IV. Experiments

In order to verify the correct operation of our approach, we will realize several experiments in a corridor of an office environment. In figure 9 we can see the corridor that we have used for the experiments and how we have topologically divided it into nodes. Afterwards we divide again the set of nodes into states. This office environment is very simetric and that is why this scenario entails much more difficulty for the localization system.

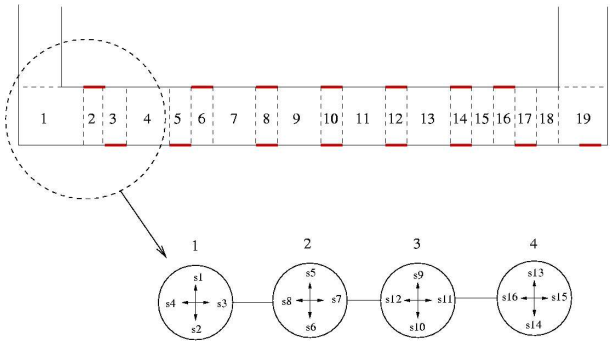


Fig. 9. Map divided into nodes

For the experimental results, we use the error function shown in equation 7, where $state_{mayor}$ denotes the state with the greatest belief and $state_{actual}$ is the robot actual position. The *distancex* is measured as the number of steps needed to reach a state from another.

$$
\begin{aligned}
error \quad = \quad & abs(prob(state_{mayor} - prob(state_{actual}))) \cdot \\
& \cdot distance(state_{mayor}, state_{actual})) \qquad (7)
\end{aligned}
$$

A. Ability for recovery of action error

In the first experiment we want to verify if the system is robust to action errors. The system must be able to detect when the movement was wrong using its sensors, and recover from this situation.

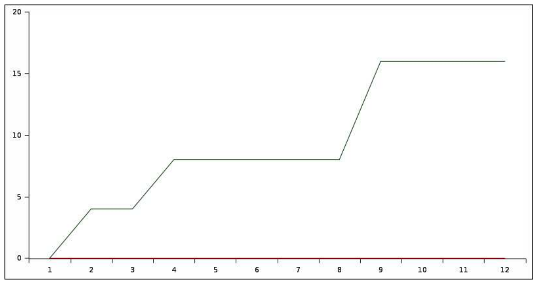
For this purpose, we situated the robot in state 15 (see map in figure 9) and we ordered it to go forward along the corridor. The robot knows where it is at the beginning, in other words, the probability distribution is concentrated in the state 15. Due to the imperfections of the actions, the error localization increases in each step. In the figure 10(a) the green line is the localization error if the movements was perfect, and the red line is the localization error using our approach. As we see in the graph, using our localization the error is 0 because the system recover in all the cases from action errors.

B. Speed in localization

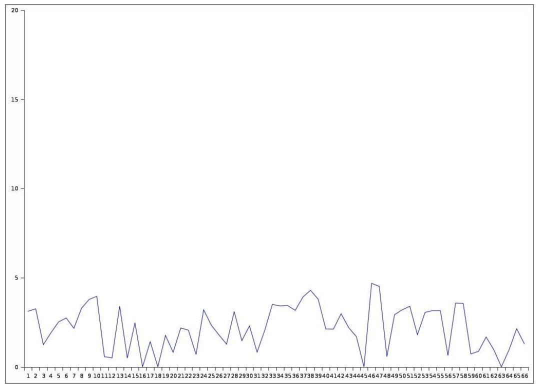
In this experiment the robot does not know where it is at the beginning, so the first time the probability distribution is uniform. This experiment was realized with a lot of sensor noise because there were a lot of people walking along the corridor. Despite this difficulty, the robot is able to be localized with a small error in a few movements and can recovery to sensor error quickly, as we see in figure 11(a)-11(d).

In 11(a) the robot starts at node one and the distribution (painted in green) is uniform along all the nodes. For this explanation we will talk about *node* instead of *states*, which is actually what we use in our model, to simplify this explanation. So, a node will be padded in green depending on the belief of the state situated in this node, orientated on the right. When the robot moves forward it reach to the node 2 (Fig. 11(b)) and it takes data from its sensors. With this data the model evolves and the probability is concentrated in state 2 and 17, because these two states have almost the same observation properties. The robot goes forward, but an error occurs and the robot reaches node 4, instead of node 3. This anomaly is observed in the model and it is corrected in the *observation phase*, as we see in 11(c). In the last movement the robot reaches the node 5 and then the simetry is broken, concentrating the probability in the node 5, as we see in Figure 11(d).

Figure 10(b) shows the error evolution during the experiment.



(a) Experiment IV-A: Error in the localization

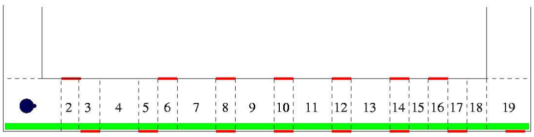


(b) Experiment IV-B: Recovering from sensor errors

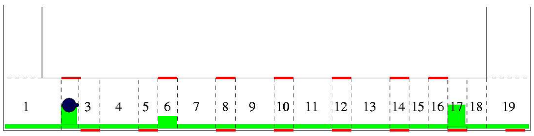
Fig. 10. Experiments results

V. Conclusions

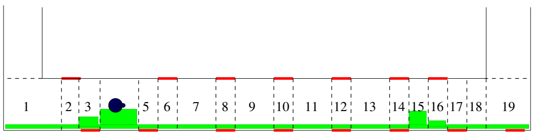
This article has presented the preliminary results for an approach to localization of legged robot, using mainly the vision as an active input sensor to extract characteristics from the environment. We have shown that the robot is able to localize itself even in environments with noise produced by the normal activity in a real office.



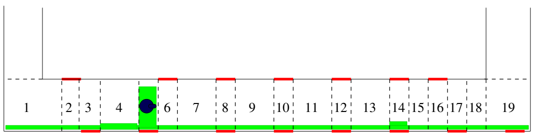
(a) Initial state for the experiment. The belief (green) is the same for every states.



(b) After the first movement was finished, there is a couple of states where the robot could be, due to the corridor simetry.



(c) In the next step the movement was no correct (it skips the node 3), but the robot can localize itself by the observation. The simetry still do not let us know where the robot is.



(d) In this step the simetry is broken and the robot knows its position.

Fig. 11. Experiment done in a corridor. The amount of green in each state represent the belief in it.

The data obtained from sensors, mainly the camera, is very rich and let a fast convergence from an initial unknown state where the belief over the set of states is uniform. Also we demonstrated that the robot can detect action failures when it is localized, and recover from them in a efficient way.

The set of observations have been descriptive enough to be efficient in the localization process. The way we determine the number of doors and ceiling lights the robot can perceive has been the key for the localization system.

Despite these results, there are some limitations that deserve future research. One of the key limitations arises from the low accuracy in the localization due to the granularity of the large areas defined as states in the map building. Maybe granularities near to the metric approximation could be more useful for many indoors applications.

We believe that probabilistic navigation techniques hold great promise for getting legged robots reliable enough to operate in real office environments. Although the experiment results shows the system works, but for a correct evaluation an immediate task is to realize more experiments to evaluate this system in a more complex and longer scenarios.

REFERENCES

[1] B. J.Borenstein and L.Feng, *Navigating mobile robots: Systems and techniques*. MA: Ltd. Wesley, 1996.

[2] R. Simmons and S. Koening, “Probabilistic navigation in partially observable environments,” in *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, Montreal (Canada), July 1995.

[3] D. R. The, “Topological robot localization by training a vision-based transition detector.” [Online]. Available: citeseer.ist.psu.edu/283978.html

[4] M. E. L. Guillén, “Sistema de navegacion global basado en procesos de decisión de markov parcialmente observables. aplicación a un robot de asistencia personal.” Ph.D. dissertation, Escuela Politécnica. Universidad de Alcalá, 2004.

[5] D. Fox, W. Burgard, and S. Thrun, “Markov localization for mobile robots in dynamic enviroments,” *Journal of Artificial Intelligence Reseaarch*, 1999.

[6] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, “Acting under uncertainty: Discrete bayesian models for mobile robot navigation,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996. [Online]. Available: citeseer.ist.psu.edu/cassandra96acting.html

[7] T. V. Franck Gechter and F. Charpillet, “Robot localization by stochastic vision based device,” in *Cybernetics and Informatics (SCI)*, 2001.

[8] P. Guerrero and J. R. del Solar, “Auto-localización de un robot móvil aibo mediante el método de monte carlo,” Instituto de Ingenieros de Chile, Tech. Rep., 2003.

[9] D. Radhakrishnan and I. Nourbakhsh, “Topological localization by training a vision-based transition detector,” *IROS*, 1999.

[10] R. P. Illah Nourbakhsh and S. Birchfield, “Dervish, an office-navigating robot,” *AI Magazine*, 1995.

[11] S.Thrun, “Robotic mapping: A survey,” *Technical Report CMU-CS-02-111*, 2002.