Visual Based Localization of a Legged Robot with a topological representation

Francisco Martín[1], Vicente Matellán[2],
José María Cañas[1] and Carlos Agüero[1]
[1]*Rey Juan Carlos University*
[2]*León University*
Spain

1. Introduction

One of the fundamental tasks for a mobile robot is the ability to determine its location in the environment. This ability is the basis for mobile robot operations, such as position based navigation or motion planning techniques (Borenstein et al, 1996). This problem is known as the *localization* problem.
Robots have to perform a great variety of tasks in an indoor office environment, such as delivering mail, guiding people from one room to another, and so on. In order to perform these tasks, the robot does not need precise information about its position, which could be obtained from a *geometric environment representation* approach. The approximate, but reliable, information about its position is successful.
In this work we use a markovian technique with a topological representation of the environment. This method will be conducted by a legged robot, which will be detailed in this section.
The office environment has been divided into states, in similar way to previous works as (Nourbakhsh et al, 1995) and (Kuipers, 2000) have, depending on the observations (explained later), that can be obtained in each part of this environment. So, a state could be a large portion of a corridor, for instance, where the sensory information is similar. The main advantage is the simplification of the localization process, reducing the number of states where the robot could be. On the other hand, the disadvantage is the low level of accuracy in the information about the robot's position, but this is not a problem for performing the robot tasks, as we mentioned above.
Our work has been developed using the Sony AIBO ERS7 robot (Fig. 1). This is an 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. AIBO's main locomotion characteristic is its canine aspect with four legs.

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

Fig. 1. Sony AIBO ERS7. (image from AIBO site)

Another key factor in this work is the limited and noisy sensory information available, obtained mainly from a camera. The legged robot locomotion influences are notable in the camera, allocated in the robot's head. This makes the image acquisition noisy. In the literature (Enderle, 2000), (Enderle, 2001), (D.Schultz & Fox, 2004) and (Veloso et al, 2000) face localization problem in legged robots using vision, but most of these works carry out their experiments in reduced and very "engineerized" environments, mainly in the Robocup four-legged league, in which landmarks are designed to be easily detected and do not present simetry or multiplicity. On the other hand, our work has been tested in a large office environment using only natural landmarks. They are harder to detect and can be similar in different locations in the environment, presenting simetry in many cases.

Most topological based localization works (e.g. (Simmons and Koening, 1995) and (Radhakrishnan and Nourbakhsh, 1999) have been developed using wheeled robots with sonar, laser sensors or omni-directional cameras. These kinds of sensors are much more reliable than cameras, and they obtain 360º sensory information from the environment. These sensors and accurate odometry information make the methods used in these works inappropriate for our settings. A legged robot displacement depends greatly on the floor conditions, which makes odometry unreliable. A non-grid topological approach is used in a SLAM work, (Choset and Nagatani, 2001), where the map is a one-dimensional set of curves that captures the salient topology of the environment. Once again, this approach is unusable in our platform because of its sensor model.

The localization method proposed in this paper is based on a markovian approach (Fox et al, 1999). This is a traditional approach to solve the localization problem, however, most previous works using this approach are based on metric representation (Kosecká and Li, 2004) (where a wheeled robot is used and extracts local invariant features from images) using wheeled robots (López et al, 2003). The Markovian approach has proved to be an effective solution for mobile robot localization in large unstructured environments (Cassandra et al, 1996) (Gechter et al, 2001). This method calculates a probability density (belief) over the entire space of states (nodes of the topological map in this case).

There are other approaches that use sampling algorithms for localization. For instance, the Monte Carlo approach does not calculate a probability density over the entire set of states. This approach only calculates the localization probability of a set of samples named particles, allocated along the space of states. The amount of particles in a state determines

the robot's position. This approach has been used in applications (e.g. (Sridharan et al, 2005), (Guerrero and Solar, 2003) and (Thrun et al, 2001)) where a large number of states does not allow the use of non-sampling algorithms, mainly for scalability reasons. In our work, the space of states is small because we represent many geometric positions in the same topological state.
This paper presents the practical steps needed to make the Markovian method effective and reliable for a legged robot using its on-board camera as the main sensor over a large not-engineered indoor environment. In summary, this paper presents three ma jor contributions:

1. The topological representation of large indoor office environments in states and their use as localization representation for a legged robot.
2. The development of a probabilistic localization technique for legged robots, where odometry is not reliable or accurate.
3. The use of noisy and non omni-directional sensors (mainly cameras) for detecting natural landmarks in indoor office environments, such as doors and ceiling lights.

The remainder of this chapter is organized as follows: in section 2 we give a brief description of the Markov process developed. In section 3 we describe our model and its components, showing the experiments and results in section 4. Finally, we will put forward our conclusions in section 5.

2. The Markovian localization framework

Localization based on indirect information provided by the robot sensors (sonar, laser, etc.) has been successfully integrated into the probabilistic framework and has exhibited good results (Thrun, 2003). In particular, sampling methods that speed up the estimation (Fox et al, 1999) are currently the most popular ones (Simmons and Koning, 1995).
In our work, we have used a markovian approach in which a probability distribution Bel over all the possible locations $S = \{s1 , s2 , ...\}$ is maintained. $Bel_t (S = s)$ represents the belief of being in state s at time t. Depending on the knowledge of the initial localization of robot $Bel_0 (S)$, the initial state will be uniformly distributed,

$$Bel_0(s_i) = \frac{1}{|S|} \tag{1}$$

or will be centered in a state where the initial position j is known,

$$Bel_0(s_i) = 0 + \delta, \forall i \neq j, \delta << 0.01 \tag{2}$$

$$Bel_0(s_j) = 1 - \delta, \delta << 0.01 \tag{3}$$

The belief *Bel* actualization is divided into two atomic steps. The first one is the movement step applied when an action has been performed by the robot. The belief is modified according to the action performed. The second one is the observation step, in which the belief is updated according to the observations taken from the sensors. This process is iterative and is executed every time a new movement is performed. These two steps in more detail are:

- **Movement step**. Robot actions are modelled on the probability $p(s'|s, a)$ (action model). This is the probability of reaching state s' if an action a is executed in state s. To obtain *the a priori* belief for the whole set of states $Bel_t(S')$, Bayesian updating is assumed. When an action is executed we apply equation 4.

$$\boldsymbol{Bel_t(s')} = \sum_{s \in S} p(s'|s, a) \cdot Bel_{t-1}(s), \forall s' \in S \tag{4}$$

- **Observation step**. To calculate the updated belief $Bel_t(S)$, we take $p(o|s)$ (sensor model) as the probability of getting observation o to be in state $\underline{s}$ and the operation is as described in [13]. When a new independent observation is obtained, the belief is updated using equation 5.

$$\boldsymbol{Bel_t(s) = p(o_1|s)} \cdot p(o_2|s) \cdots p(o_n|s) \cdot Bel_t(s'), \forall s, s' \in S \tag{5}$$

$$\boldsymbol{Bel_t(s)} = p(\mathbf{o}|s) \cdot Bel_t(s'), \forall s, s' \in S \tag{6}$$

3. Our Model

Our localization method requires three components to be described, and these will be defined in more detail in this section:

1. The map and how it is translated into a set of states.
2. A set of actions that the robot can perform and their probabilistic action model related to states $p(s'|s, a)$.
3. A set of observations that the robot perceives from the environment, and its probabilistic model related to states $p(o|s)$.

3.1 The state space

We name possible locations of the robot as "states". These states are defined over an indoor office environment (see Fig. 2) made up of corridors (represented in the set of nodes as circles) and rooms (represented as boxes). Nodes are defined as places with similar characteristics and can group together one or more states. In this way, for example, a particular node could be a continuous corridor region where there are no doors to the right or left. Another node could be another continuous corridor region where there is a door to the right and a wall to the left, and so on.

In order to translate a metric map in the stated representation, first, nodes are created, according to the previous definition. Once the set of nodes has been defined, each node is divided into four different states, representing the robot's position with four orientations: North, East, South, and West.

In the top left diagram of figure 2 we can see a portion of an office environment. At the top right of the same figure, the same area has been divided into nodes attending to their characteristics. For example, a region where there are doors on both sides of the corridor is different to the region where there is only one. This division is guided by the number of doors and ceiling lights the robot can sense in each position. These natural landmarks are the ones that the robot is able to perceive more easily from its raw camera images according to our experimentation.

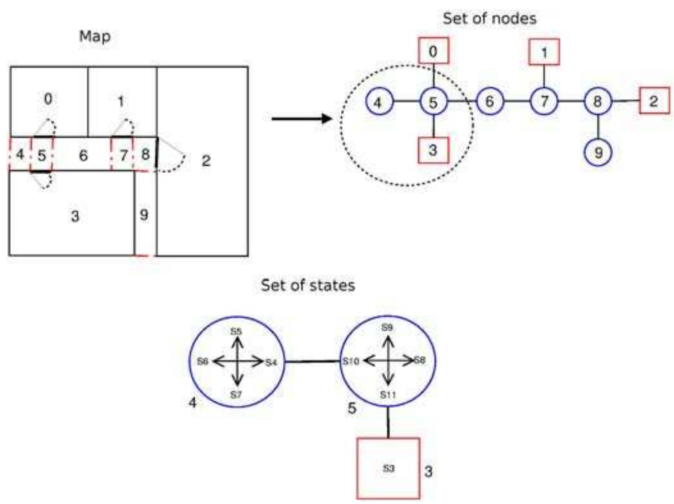

Fig. 2. The set of states is built from the map and its node decomposition

Note this is a topological organization, 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):

Turn Left	N: 0.15	T: 0.70	TT: 0.15	TTT: 0.0
Turn Right	N: 0.15	T: 0.70	TT: 0.15	TTT: 0.0
Go Forward	N: 0.20	F: 0.6	FF: 0.15	FFF: 0.05

Table 1. Uncertainly in action execution

3.2 The state space

The action primitives we have implemented in this work are: to turn 90º to the left $a_{\{TL\}}$, to turn 90º to the right $a_{\{TR\}}$ and go forward $a_{\{F\}}$ until the next state with the same orientation is reached. In table 1 we show the uncertain model we have used to characterize the action execution. If the robot executes "go forward", for example, we have in mind that the robot could either do nothing(N), reach the next state (T), or go to the second state in that direction (TT) or even three states (TTT).

When the robot executes an action primitive, i.e. when the robot moves, it updates the belief as shown in (4). The action model defines $p(s'|s, a)$ as the probability of reaching state s', starting at state s and executing action a:

$$p(s'|s,a), \forall s \in S, \forall a \in A = \{a_{\{F\}}, a_{\{T_L\}}, a_{\{T_R\}}\} \tag{7}$$

This probability $p(s'|s, a)$ will represent our action model and it will be calculated *a priori*, depending on the possible action the robot can perform in that state space according to table 1.
The robot's navigation is also an important aspect of this model:

- The robot must be centered in the corridor, as much as possible, in order to get correct observations from the environment.
- The robot must avoid bumping into any dynamic obstacle and, of course, walls and doors.
- Transitions from one state to the next must be detected.

We use information from camera and infrared sensors to achieve these ob jectives:

- The robot's head is moving horizontally in an arc of [-90º ,90º] (figure 4) at 1 Hz.
- The robot is continuously obtaining infrared information in each arc position. The robot will detect any obstacle to avoid collision. It will also use this information in order to be centred in the corridor.
- The robot will capture an image if the angle's head is near 90º or -90º(blue zone in figure 4). It lets us know what elements (doors or walls) there are on both sides of the robot. The robot maintains a memory of the elements that have been detected at both sides. When any of these elements change, the state will be changed too. For example, if the robot detects a door on its right and a wall on its left, it will be aware of a state change when it detects a wall on its right, for instance.

Fig. 3 shows the ground-truth path that the robot has followed along the corridor, avoiding the obstacles (persons and walls) and centering itself in it. This information has been manually recorded placing marks on the floor behind the robot at constant rate of 1 mark every 10 seconds and interpolating the path.

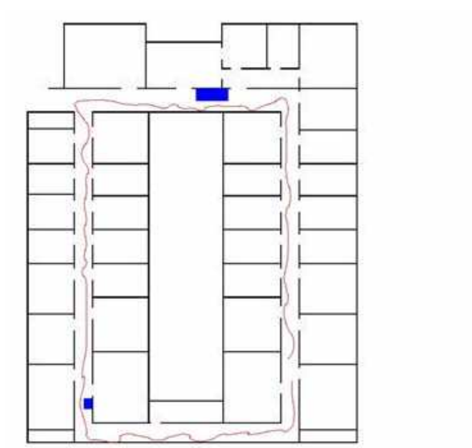

Fig. 3. Ground-truth path followed by the robot.

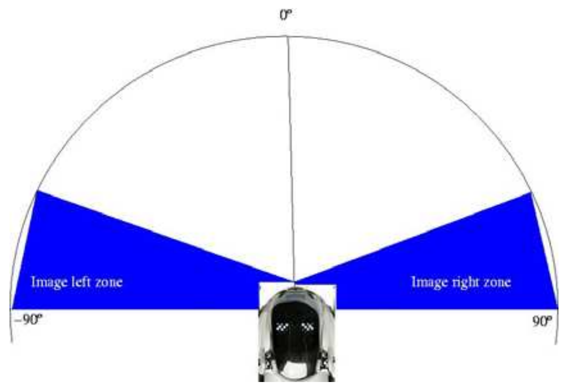

Fig. 4. Active sensing. The robot senses the whole area with its infrared sensor. It analyzes the images captured in the blue area.

3.2 Sensor Model

Our sensor model takes three types of sensations from the image obtained by the robot's camera:

- **Depth**. The main goal of this observation is to measure how far the robot is from the wall when it is oriented toward the end of the corridor. For this purpose we detect the number of ceiling lights that the robot perceives. If the number of ceiling lights is high, the robot is far from the end. If this number is low, the robot is near to the end. In figure 5 we can see the original image and the image that shows the detection of ceiling lights (green boxes).
- **Doors.** The robot is able to count the number of doors it can observe ahead using a color filter. The doors are supposed to be perpendicular to the floor and the jambs parallel to them. If a region of the image fulfills with these specifications, it is assumed to be a door. In figure 6 we can see the original image and the image with the result of the door detection (blue boxes).
- **Nearby landmarks**. This observation gives us information about which landmarks are around the robot. We define landmarks as the doors or walls that are situated on the right and left sides of the robot. For example, an observation may be that there is a door on the left side and a wall on the right side.

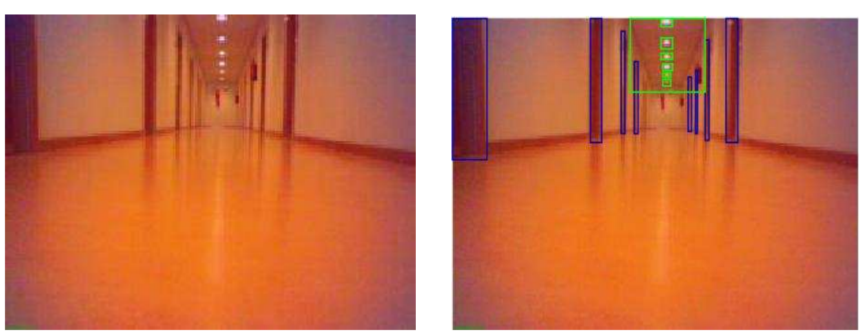

Fig. 5. Detecting 6 ceiling lights and 8 doors.

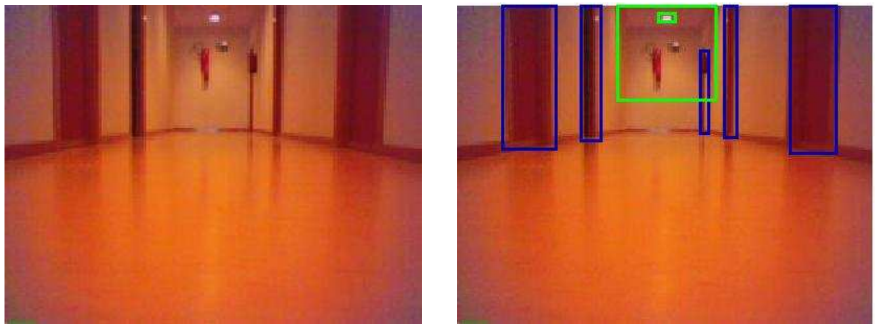

Fig. 6. Image information extraction results.

Once the data is collected, we apply equation (5) to correct the belief, according to the information obtained by the sensors. The types of sensations described before are independent from them, so we can apply equation 6.

$$Bel_{\boldsymbol{subsequent}}(s) = p(\mathbf{o}|s) \cdot Bel_{previous}(s), \forall s \in S \tag{8}$$

$$\begin{aligned} Bel_{\boldsymbol{subs}cqucnt}(s) \quad = \quad & p(o_{ccilinglights}|s \cdot \\ & p(o_{doors}|s) \cdot \\ & p(o_{nearlandmarks}|s) \cdot \\ & Bel_{previous}(s), \forall s \in S \end{aligned} \tag{9}$$

4. Experimental setup and results

Several experiments were conducted to test the quality of the localization algorithm. The experiments were carried out in our office environment in our normal daily work. So, the environment is noisy because of the normal activity: people walking in the corridor, doors opening, and so on. This sometimes produces erroneous sensory information that will show how robust the model being used is. This is a requirement presented in section 1.

According to the desiderata presented in section 1, we evaluate the robot's localization performance based on:

- The robot's robustness to detect state changes.
- Overall accuracy.

4.2 Test for detecting state changes

In order to test the state change detection, we designed a task in which the robot was required to go forward in the corridor and to stop when a new state was detected. Fig. 7 shows an example of this test. The correct state change detections are displayed as blue circles and the wrong state change detections are displayed as red boxes. We obtained 82% accuracy in state change detections. This information has been obtained by an operator monitoring the process and verifying when the robot has correctly detected the new state.

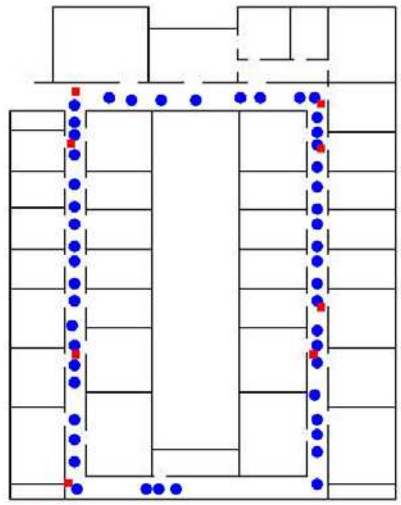

Fig. 7. Correct state change detection (blue circles) and the incorrect state change detection (red boxes).

4.2 Accuracy

To test the overall accuracy, we have designed two experiments: in the first one the initial position is not known, and in the second one the initial position is known. In each experiment we measured two values:

- **Error in localization**. To measure the error in localization, we count the nodes that a robot must traverse from the position that the localization algorithm calculates as the most probable (The mode of the *Bel* distribution), for the robot's actual position.
- **Entropy**. The entropy of the states probabilities is a usual measure for determining when the belief about the robot pose is concentrated in few states. This doesn't mean the robot is well localized. This is only a dispersion measure that indicates when the belief would determine the robot pose. When this value is near 0, the Bel probabilities are accumulated in a single node, which is considered to be the robot's position. If the entropy value is near 1, the robot's position cannot be determined and the Bel information is not useful (the error is not meaningful neither). We must consider situation in which continuous errors in perception lead robot pose converge into a wrong pose with low entropy.

$$\boldsymbol{H}(Bel) = -\sum_{s \in S} Bel(s) log(Bel(s)) \tag{10}$$

4.2.2 Initial position unknown

In this test (Figure 8), the initial position is not known. Figure 9 shows how the error and the entropy (figure 10) evolve in the localization process. At the initial step, the entropy is high, because it is distributed in several states, making it difficult to determine the robot's actual

position. In state 2, the entropy is low, which means that the belief is concentrated in few states (typically two in our case,
representing symmetry in the environment), showing instability in the error graph. When error stabilizes at step 15 and the entropy is still low, the robot has localized itself. Some localization error can happen due to observation errors or motion errors (this increases entropy), but the system recovers quickly.

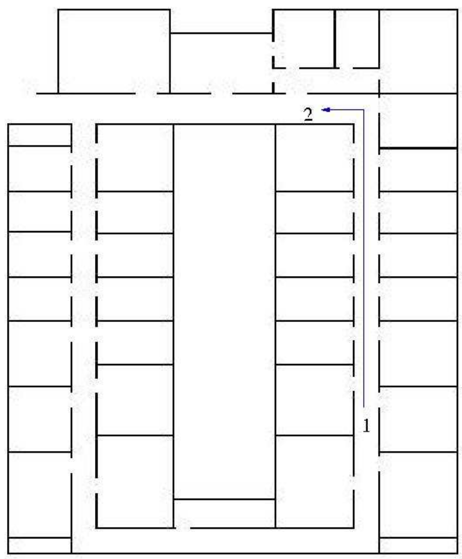

Fig. 8. The robot moves from position 1 to position 2. The robot does not know its initial position.

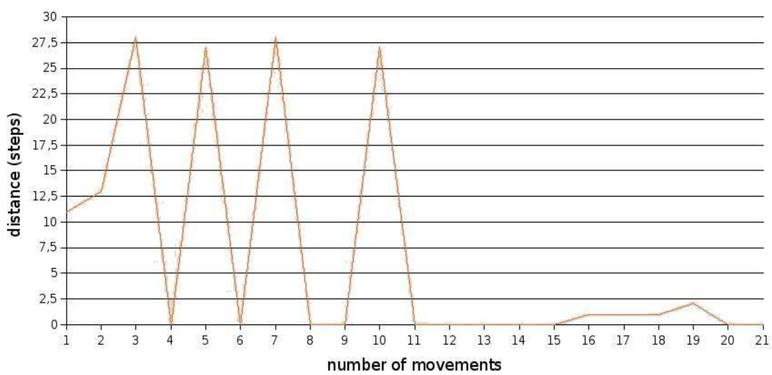

Fig. 9. Error measured as the distance from Bel mode to the actual robot position.

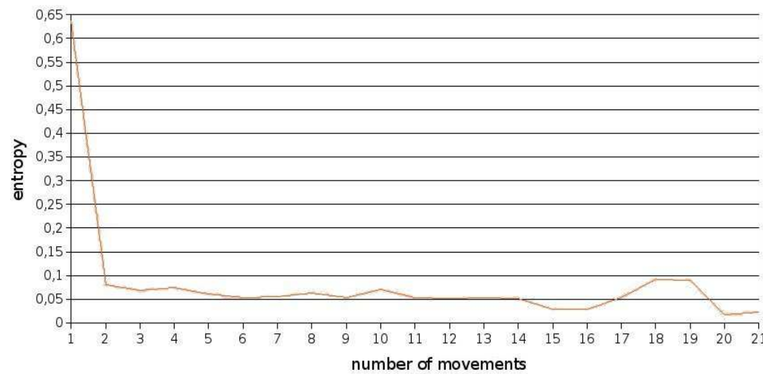

Fig. 10. *Bel* entropy

4.2.2 Initial position known

In this test (Fig. 11), the initial position is known. The robot starts at a known position and this is why the error (Fig. 12) starts at low values. Some perception errors in state 24 cause the robot to get lost for a few movements, but when these errors disappear, the robot recovers its position knowledge.

The experimentation results show interesting conclusions. First of all, our approach works and is ableto be carried out with the localization task. The experimentation results also show that the knowledge of the initial state does not influence the process in the long term.

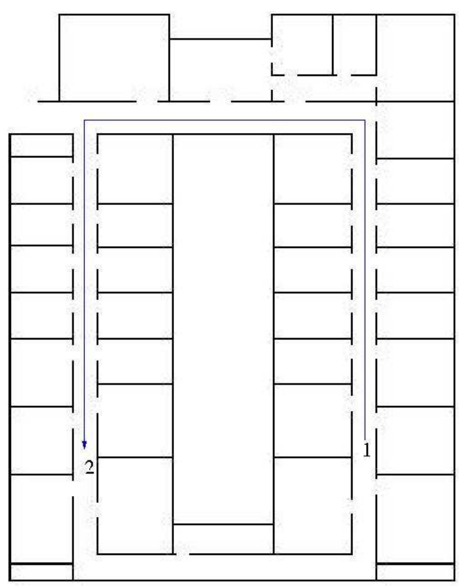

Fig. 11. The robot moves from position 1 to position 2. The robot knows its initial position.

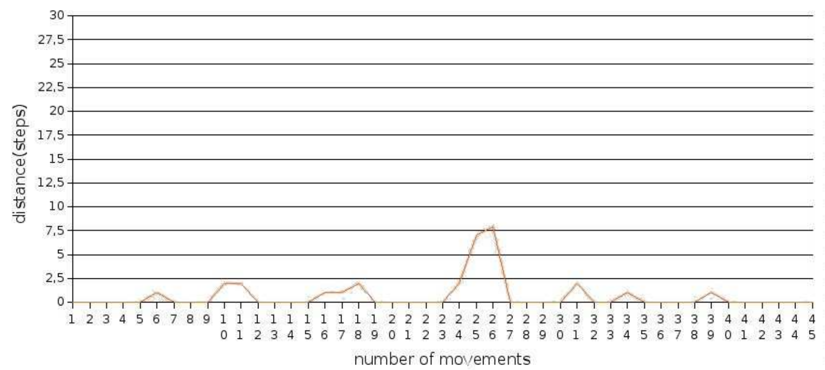

Fig. 12. Error measured as the distance from Bel mode to the actual robot position.

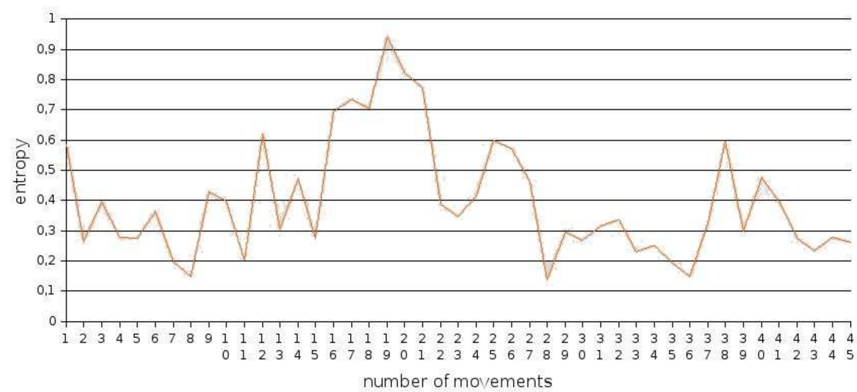

Fig. 13. *Bel* entropy.

5. Conclusions

In this chapter we have presented the performance of a localization method of legged AIBO robots in not-engineered environments, using vision as an active input sensor. This method is based on classic markovian approach but it has not been previously used with legged robots in indoor office environments. We have shown that the robot is able to localize itself in real time even inenvironments with noise produced by the human activity in a real office. It deals with uncertainty in its actions and uses perceived natural landmarks of the environment as the main sensor input.

We have also shown that data obtained from sensors, mainly the camera, is discriminate enough and allows a fast convergence from an initial unknown state, in which the belief has been distributed uniformly over the set of states . We have also shown that the robot can overcome action failures while localizing, and it recovers from them in an efficient way.

The set of observations we have chosen have been descriptive enough to be efficient in the localization process. We think that the way we determine the number of doors and ceiling lights has been the key for the success of the localization system. This approach depends on the a priori environment knowledge, but we are studying new types of useful observations to make thisapproach applicable in other environments where this a priori knowledge is not available.

Despite these results, there are some limitations that deserve future research. One of the key limitation 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 indoor applications.

We believe that probabilistic navigation techniques hold great promise for enabling legged robots to become reliable enough to operate in real office environments.

6. References

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

R. Nourbakhsh, R. Powers, and S. Birchfield, *Dervish - an office-navigating robot.*, AI Magazine, vol. 16, no. 2, pp. 53–60, 1995.

B. Kuipers, The spatial semantic hierarchy, Artif. Intel l., vol. 119, no. 1-2, pp. 191–233, 2000.

D.Schultz and D.Fox, *Bayesian color estimation for adaptative vision-based robot localization*, in IROS-04, Sept. 2004.

S. Enderle, M. Ritter, D. Fox, S. Sablatnog, G. K. Kraetzschmar, and G. Palm, *Soccer-Robot Localization Using Sporadic Visual Features*, in Intel ligent Autonomous Systems 6 (IAS-6) (E. Pagello, F. Groen, T. Arai, R. Dillmann, and A. Stentz, eds.), (Amsterdam, The Netherlands), pp. 959– 966, IOS Press, 2000.

S. Enderle, H. Folkerts, M. Ritter, S. Sablatnog, G. K. Kraetzschmar, and G. Palm, *Vision-Based Robot Localization Using Sporadic Features*, in Robot Vision (R. Klette, S. Peleg, and G. Sommer, eds.), vol. 1998 of Lecture Notes in Computer Science, Berlin, Heidelberg, Germany: Spinger-Verlag, 2001.

M. Veloso, E. Winner, S. Lenser, J. Bruce, and T. Balch, *Vision-servoed localization and behavior-based planning for an autonomous quadrup legged robot*, in Proceedings of the Fifth International Conference on Artificial Intel ligence Planning Systems, (Breckenridge, CO), pp. 387–394, April 2000.

R. Simmons and S. Koening, *Probabilistic navigation in partially observable environments*, in Proceedings of the 1995 International Joint Conference on Artificial Intel ligence, (Montreal (Canada)), pp. 1080–1087, July 1995.

D. Radhakrishnan and I. Nourbakhsh, *Topological localization by training a vision-based transition detector*, in Proceedings of IROS 1999, vol. 1, pp. 468 – 473, October 1999.

H. Choset and K. Nagatani, *Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization*, IEEE Transactions on Robotics and Automation, vol. 17, pp. 125 – 137, April 2001.

D. Fox, W. Burgard, and S. Thrun, *Markov localization for mobile robots in dynamic environments,* Journal of Artificial Intel ligence Research, vol. 11, pp. 391–427, 1999.

J. Koseck´a and F. li, *Vision based topological markov localization*, in Proceedings of the 2004 IEEE International Conference on Robotics and Automation, (Barcelona (Spain)), Apr. 2004.

M. E. López, L. M. Bergasa, and M.S.Escudero, *Visually augmented POMDP for indoor robot navigation, Applied Informatics*, pp. 183–187, 2003.

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.

F. Gechter, V. Thomas, and F. Charpillet, *Robot localization by stochastic vision based device*, in The 5th World Multi-Conference on Systemics, Cybernetics and Informatics - SCI 2001. The 7th International Conference on Information Systems Analysis and Synthesis - ISAS 2001, Orlando, FL, USA, Jul 2001.

M. Sridharan, G. Kuhlmann, and P. Stone, *Practical vision-based monte carlo localization on a legged robot*, in IEEE International Conference on Robotics and Automation, April 2005.

P. Guerrero and J. R. del Solar, *Auto-localización de un robot móvil aibo mediante el método de monte carlo*, Anales del Instituto de Ingenieros de Chile, vol. 115, no. 3, pp. 91–102, 2003.

S. Thrun, D. Fox, W. Burgard, and F. Dallaert, *Robust monte carlo localization for mobile robots*, Artif. Intel l., vol. 128, no. 1-2, pp. 99–141, 2001.

S. Thrun, *Robotic mapping: a survey*, pp. 1–35, 2003.