Indoor Robot Navigation using a POMDP based on WiFi and Ultrasound observations

M. Ocaña, L. M. Bergasa, M. A. Sotelo and R. Flores

Department of Electronics
University of Alcala
Alcalá de Henares, 28805 Madrid, Spain
{mocana,bergasa,sotelo,flores}@depeca.uah.es

***Abstract* - . This paper presents a robot navigation system for indoor environments using a Partially Observable Markov Decision Process (POMDP) based on WiFi signal strength and ultrasound observations. The paper represents the first one in using WiFi sensor readings as an observation in a POMDP. We present an algorithm based on an EM-SLAM that we called WSLAM (Wifi Simultaneous Localization And Mapping) that is able to learn the observation and transition matrix in autonomous mode. With this algorithm we obtain a minimum calibration effort. We demonstrate that this system is useful to navigate in indoor environments with a real robot. Some experimental results are shown. Finally, the conclusions and future works are presented.**

Index Terms – indoor navigation, Markov Process, POMDP, WiFi observation, autonomous learning system WSLAM.

I. INTRODUCTION

The boom in wireless networks over the last few years has given rise to a large number of available mobile tools and their emerging applications are becoming more and more sophisticated by year. Wireless networks have become a critical component of the networking infrastructure and are available in most corporate environments (universities, airports, train stations, tribunals, hospitals, etc) and in many commercial buildings (cafes, restaurants, cinemas, shopping centres, etc). Then, new homes are slowly starting to add WiFi services in order to enable mobility to perform many routine tasks, in the known as intelligent houses. There are even emerging some projects about WiFi enabled cities as Paris, Barcelona, etc.

The recent interest in location sensing for network applications and the growing demand for the deployment of such systems has brought network researchers up against a fundamental and well-known problem in the field of the robotics as is the localization. Determining the pose (position and orientation) of a robot from physical sensors is not a trivial problem and is often referred to as “the most important problem to providing a mobile robot with autonomous capabilities” [1]. Several systems for localization have been proposed and successfully deployed for an indoor environment. Examples include infrared-based systems [2], various computer vision systems [3], ultrasonic sensors and actuator systems [4], physical contact based actuator systems [5] and radio frequency (RF) based systems [6].

Many mobile robot platforms use wireless networking to communicate with off-line computing recourses, human-machine interfaces or others robots. Since the advent of inexpensive wireless networking, many mobile robots have been equipped with 802.11b wireless Ethernet. In many applications, a sensor from which position can be inferred directly without the computational overhead of image processing or the material expense of a laser is of great use. Many robotics applications would benefit from being able to use wireless Ethernet for both sensing position and communication without to add new sensors in the environment.

WiFi location determination systems use the popular 802.11b network infrastructure to determine the user location without using any extra hardware. This makes these systems attractive in indoor environments where traditional techniques, such as Global Positioning System (GPS) [7] fail. In order to estimate the user location, wireless Ethernet devices measure signal strength of received packets. This signal strength is a function of the distance and obstacles between wireless nodes and the robot. Moreover, the system needs one or more reference points (Access Points) to measure the distance from. Unfortunately, in indoor environments, the wireless channel is very noisy and the radio frequency (RF) signal can suffer from reflection, diffraction and multipath effect, which makes the signal strength a complex function of distance. To overcome this problem, WiFi location determination systems uses a priori radio map (wireless-map), which captures the signature of each access point at certain points in the area of interest. These systems work in two phases: training phase and estimation phase. During the training phase, the system constructs the wireless-map. In the estimation phase, the vector of samples received from each access point is compared to the wireless-map and the “nearest” match is returned as the estimated user location.

WiFi location estimation techniques are divided into deterministic and probabilistic techniques. In the first one the physical area making up the environment is first divided into cells. Location is performed in the estimation phase selecting the most likely cell in order to determine which cell the new measurement fits best [8]. On the other hand, probabilistic techniques construct a probability distribution over the targets location for the physical area making up the environment. This last technique provides more precision with computational overhead. Some recent and representative works have appeared in this line. In [9] the authors utilize a Bayesian belief network to derive a posterior probability distribution over the target’s location.

In [10] a probabilistic approach using recursive Bayesian filters based on sequential Monte Carlo sampling is proposed. In both cases a laptop has been used for the localization tests and the best accuracy obtained is about 1.5 meters.

Typically the Bayesian approach is applied in the case when we have a grid-based representation of the environment. Another alternative for modelling the environment is with a topological map. In this case the localization is based on the fact that the robot automatically identifies nodes in the map from geometric environmental information.

For a global navigation system design, in which the objective is the guidance to a goal room and some low level behaviour perform local navigation, a topological discretization is appropriate to facilitate the planning and learning tasks. POMDP models provide solutions to localization, planning and learning in the robotic context. These models use probabilistic reasoning to deal with uncertainties, very important in the case of WiFi localization sensors, and a topological representation of the environment to reduce memory and process time of the algorithms.

In this paper, we present a probabilistic navigating system for a robotic platform in indoor environments using a POMDP based on WiFi signal strength and ultrasound measures. Firstly, we present an introduction to POMDP navigation systems and then we propose our POMDP based on WiFi and ultrasound observations. We experimentally demonstrate that the system performs well in real application. Also we present an algorithm based on an EM-SLAM that we called WSLAM. The WSLAM algorithm is able to learn the observation and transition matrix in autonomous mode minimizing the calibration effort. Finally we extract conclusions about it.

II. INTRODUCTION TO POMDP NAVIGATION SYSTEMS THEORY

While in a Markov Decision Process (MDP) the environment observation is free of uncertainty, in the real robotic systems, there are some uncertainties associated to their sensors observations. The MDP considers that only the effect of the actions has uncertainty.

When a MDP realizes an execution steps series and it goes along a different states (s_0, s_1, ...,s_n) executing an actions series (a_0, a_1, ...,a_n), the probability of being in a s_{t+1} state in the $t+1$ execution step is obtained as equation (1).

$$p\left(s_{t+1} \mid s_0, a_0, s_0, a_0, \ldots, s_t, a_t\right) = p\left(s_{t+1} \mid s_t, a_t\right) \qquad (1)$$

This expression indicates that the current state (s_{t+1}) depends only on the before state (s_t) and the before action (a_t) is known as *Markov Property*.

When a noisy sensor such as the WiFi signal strength is used, then observation with uncertainty are obtained from the sensors. This case is called as a partial observability.

The POMDPs are mathematic models that permit to characterize this type of systems. A POMDP is defined by the same elements than in a MDP: S (states set), A (actions set), T (transition function), R (recompense function); and also the next elements: O is the observations set ($o \in O$) and υ is the observation function [11].

A POMDP doesn't know its real state because the observation has uncertainty. A POMDP maintains a belief distribution called *Bel*(S) or *Belief Distribution (Bel)* over the states to solve it. This distribution assigns to each state s a probability that indicates the possibility of being the real state. This is the main reason to divide the control stage of a POMDP in two blocks, as can be seen in Figure 1:

1) State estimator: the input of this block is the current observations and its output is the Bel. This block obtains the probability over all possible states.

2) Politics: the input of this block is the current *Bel* and its output is the action to perform. This block obtains the optimal action to perform in the next execution step to maximizing the recompense (R).

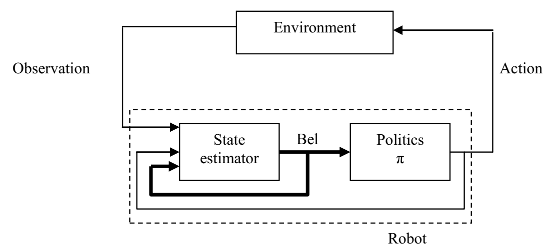

Fig. 1 POMDP structure

The Belief Distribution must be updated whenever a new action or observation is carried out. When an action a is executed and a new observation o is taken, the new probabilities became:

$$\mathrm{Bel}_t(s') = \eta \times p\left(o \mid s'\right) \times \sum_{s \in S} p\left(s' \mid s, a\right) \times \mathrm{Bel}_{t-1}(s), \forall s' \in S \qquad (2)$$

In the context of robot navigation, the states of the Markov model are the localizations (or nodes) of the topological representation of the environment. Actions are local navigations behaviours that the robot executes to move from a state to another, and observations are perceptions of the environment that the robot can extract from its sensors. In this case, the Markov model is partially observable because the robot may never know exactly which state it is in. To solve the POMDP model EM algorithm is used.

III. DESIGN OF OUR POMDP NAVIGATION SYSTEM

In this section we describe the design of our navigation system using a POMDP based on WiFi and ultrasound observations, this design include: the environment representation, the states set, the observations type selection, the possible actions of the robot and the transition and observation matrix.

A. Environment representation

In a topological representation of the environment, the discretization degree is the more important parameter to select because the process time depends directly from it.

In the topological map the nodes should be useful for the robot patrol application. In this case the robot must be able to navigate in an autonomous mode inside the corridors. The robot must be able to stop in front of all the office doors in order to come in the rooms. This last maneuver will be carried out in a teleoperated mode. As it's shown in Figure 2 for an environment example, the corridors are discretized into coarse-grained regions (nodes) of variable size. The limits of these nodes correspond to any changes in lateral features of the corridor (door, opening or piece of wall).With this background we have selected two types of nodes: office nodes (nodes that are in front of the offices rooms) and extreme nodes (nodes that are at the end of the corridor). The extreme sensor could to have a connection with a corridor in the left or in the right or with an ending room.

Figure 2 shows an example of the environment discretization and its topological representation. The extreme nodes are represented as a square and office nodes are represented as a circle.

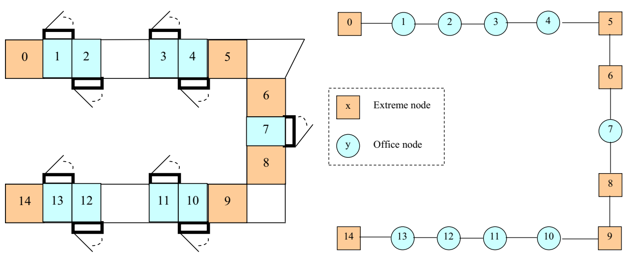

Fig. 2 Example of environment discretization and its topological representation

B. *States set*

With this topology, states of the Markov model are directly related to the nodes of the topological graph. Two states are assigned to each corridor node, one for each of the two main orientations (forward, backward) that the robot can adopt during corridor following. We denote as *forward* direction to the direction from the 0 node to the 14 one and *backward* in the opposite direction.

C. *Actions set*

The actions set has been selected to produce transitions from one state to another correspond to local navigation behaviors of the robot. We assume imperfect actions, so the effect of an action can be different of the expected one (this will be modeled by the transition model T).

The action set is very simple in our application owing to the configuration of the states and the local navigation system. Table I shows the action set.

TABLE I
ACTIONS SET

Action	Symbol	Function	States where it's possible to execute
Follow corridor	a_F	To continue through the corridor to the next state	Office nodes
No operation	a_{NO}	Used as a directive in the goal state	Office and extreme nodes
Turn around	a_T	To change the navigation direction	Extreme nodes
Turn right	a_R	To turn 90º to the right	Extreme nodes
Turn left	a_L	To turn 90º to the left	Extreme nodes

D. *The observations*

We select two kinds of observations in our model, the first one is the WiFi signal received measure observation (*obswifi*) and the second one is the ultrasound observation (*obsus*).

The *obswifi* used is obtained as the mean value of 60 samples of the signal strength, received in the WiFi robot interface, from all APs. This filtered is carried out in order to minimize the high noise of the WiFi signal measures. The number of samples has been obtained in a experimental way for optimal localization. The obswifi is then divided in *N* different observations ($obswifi_{AP1}$, $obswifi_{AP2}$, ..., $obswifi_{APN}$) one observation for each access point.

The mean signal is then rounded to integer value in order to obtain a discrete space of values. The possible values that can be obtained from the WiFi interface range from 0 to -99dBm, but we have changed the sign of the measure to obtain a useful observation to index in the observation matrix.

The *obsus* used is obtained from the ultrasound sensors. Four different observations are established: door in the left, door in the right, door in both sides and wall in both sides. In this manner the possible values are discrete and useful to index the observation. Figure 3 shows the combinations for the *obsus* and the codified values associated.

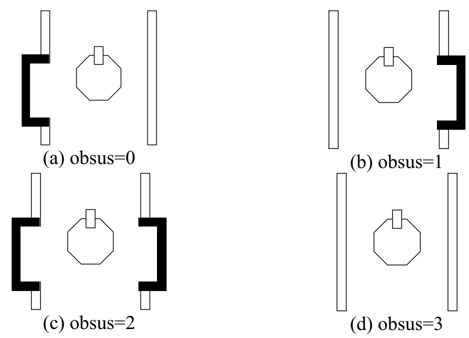

Fig. 3 Possible combinations for *obsus*: (a) door in left side, (b) door in right side, (c) door in both sides and (d) no doors detected

E. *Sensor fusion*

Observations from the WiFi and the ultrasound sensors are complementary. The first one obtains an estimation of the global localization and the second one obtains a good estimation of the local environment. The fusion of these observations will produce a good observability of states.

POMDP provide a natural way for using multisensorial fusion in their observations models ($p(\vec{o}\,|\,s)$) by mean of Bayes rule. Because these are independent observations, the observation model can be simplified in the following way:

$$\begin{aligned} p(\vec{o}\,|\,s) &= p(obswifi_{AP1},\ldots,obswifi_{APN},obsus\,|\,s) = \\ &= p(obswifi_{AP1}\,|\,s)\times\ldots\times p(obswifi_{APN}\,|\,s)\times p(obsus\,|\,s) \end{aligned} \tag{3}$$

At the equation (3) $\vec{o}$ is a vector composed by two types of observations: WiFi and ultrasound.

F. *Actions uncertainty model*

The actions uncertainty model represents the real errors or failures in the execution of the actions. The transition function *T* incorporates this information to the POMDP.

In our case, T is a matrix that represents the probability of reaching the state s_{t+1} when the robot is in the state s_t and it has executed the action a_t.

The matrix dimensions are *n_est x n_est,* where *n_est* is the total states number on the topological map.

If the action has no uncertainty, the robot executes an *F* action ("Forward"), the robot advances to the next state, this is the ideal case. In a real situation can appear some errors that introduce uncertainty in the actions. Some of them can be:

1) If a person is in the hollow of the door and he is blocking the depression of the door, the local navigator shall detect a depression with a width less than a door and then, the local navigator shouldn't validate it as a door. The error in this case is called *FF* ("Forward-Forward"), advance two states instead of one. This case can be repeated in two or more doors, then the error will be called *FFF* (for two doors blocked) *FFFF* (for three doors blocked) and so on.

2) If somebody leaves and object in a corridor with the same width that a door, the local navigator shall validate the object as a door. This error is called *NO* ("No-Operation"), the robot doesn't reach the next door.

G. Observations uncertainty model

The observations uncertainty model represents the real errors or failures of the sensor systems (ultrasound ring and WiFi interface). The observation function υ incorporates this information to the POMDP.

In this work, υ is a matrix for each observation (seven for the WiFi observations and one for the ultrasound one). The matrix dimensions are *n_est* x *obs_values*, where *obs_values* is the possible observation values on the current state.

The ultrasound observation uncertainty is bound to the same cases than in the actions uncertainty.

For the WiFi observations, the error sources are more complex. In indoor environments the WiFi signal is affected by different factors:

1) Reflections, refractions, and diffractions: that in indoor environments are knows as multipath fading.

2) Water resonant frequency: the WiFi technology is working in 2.4GHz, and this is the water resonant frequency, therefore, all persons in the environment can modify the signal strength received.

3) Free band frequency: we also remark that this frequency is in the free band frequency where several applications are working, such as: Bluetooth technology very common in wireless keyboards and mice.

Due to these factors the signal strength measure can be largely modified respect to the ideal value and this variation changes as function of the time. A deep study of these factors has been carried out by the authors in [12].

H. Training method for obtaining the transition and observation matrixes

In a lot of real systems using POMDPs, the values of the transition and observation matrixes are obtained with a simple deduction or with a priori expertise known [13][14][15].

In our case we use the ability of our low level controller to build an autonomous learning system. The robot navigates in autonomous mode with the ultrasound information only and with the initial state known storing the actions in the training action set, at each transition the robot obtains the WiFi and ultrasound observations, these information represent the training observations set. After that in an off-line stage we execute the SLAM based on Baum-Welch algorithm (EM algorithm) using the training sets in order to yield the transition and observation matrixes. We have called this technique WSLAM (Wifi-SLAM).

This process constitutes a learning phase in which the robot learns the transition and observation matrixes. Once this phase has finished a tracking phase is executed to track the robot using the before matrixes.

I. Politics π

There are different algorithms to solve the selection of the ideal action to execute in each state. In a POMDP the problem is more complex than in a MDP because we don't know the current state. In a POMDP we only maintain a belief distribution.

In [13][14] they solve the underlying MPD and then apply some different methods to select the optimal action.

In this work we use the *Most Likely State* (MLS) method to select the optimal action because the global observation, that provides the WiFi sensor, normally obtains a belief distribution with a maximum in the real state. This method selects the optimal action associated to the most probable state of the belief distribution (4).

$$a = \pi_{MLS}(\mathrm{Bel}) = \pi^*(\arg\max_S \mathrm{Bel}(s)) \qquad (4)$$

IV. Experimental results

First of all we describe the test-bed used for our navigation system and then we present some experimental results for the training and tracking phases to validate the proposed navigation system with the real robot.

A. Test bed

The test-bed was established on the 3rd floor of the Polytechnic School building, in the Electronic Department, at the University of Alcala. The layout of this zone is shown in Figure 4. It has dimensions of 60 m by 60 m with about 44 different rooms, including offices, labs, bathrooms, storerooms and meeting rooms.

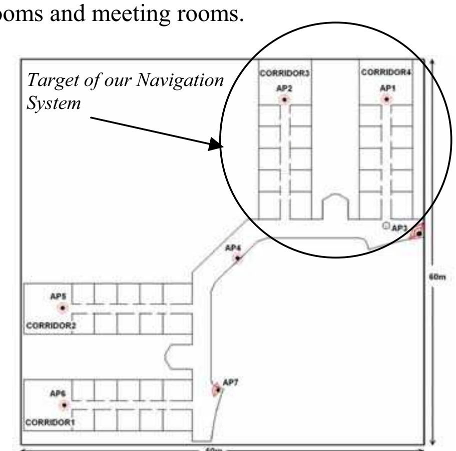

Fig. 4 Test bed environment. 3rd Floor of the Electronic Department

With a circle we remark our target test-bed. We suppose that the all results obtained in this area could be applied to the all environment, because the building and the WiFi Access Points (APs) are symmetrically distributed.

Seven Buffalo APs (WBRE-54G) were installed at the locations indicated in Figure 4, five APs were connected to omnidirectional antennas and 2 APs (AP3 and AP7) were connected to antennas of 120 degrees of horizontal beam-width. The APs acts as the wireless signal transmitters or base stations.

As mobile robot we have used a Pioneer 2AT of Activmedia robotics (Figure 5) with the following configuration: an embedded computer with a Pentium III 850MHz, a 16 ultrasound sensor ring, one Orinoco PCMCIA Gold wireless card with an omnidirectional Buffalo antenna placed above the robot. The operating system is Linux Red Hat 9.0, we modified the wireless tools of Jean Tourrilhes [16] and the patch of Moustafa A. Youssef for the Orinoco driver [17].

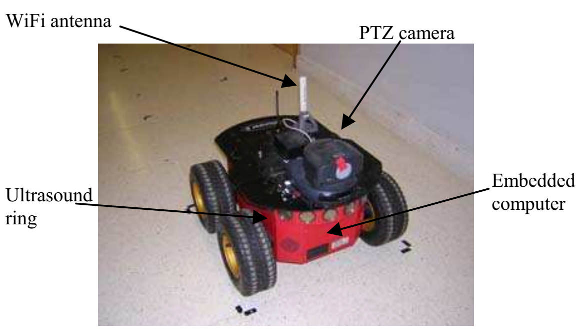

Fig. 5 Real robot used to test the navigation system developed

B. Training phase

In order to obtain the parameters of the POMDP (observation and transition matrixes) we have trained the system in an automatic way during nine training sets. We have compared the results of applying the SLAM-EM algorithm using Ultrasound observation, WiFi observation and WiFi+Ultrasound observations.

For the Ultrasound observation experiment, the results obtained aren't good. If we give the initial state to the system the algorithm converge with a 85% of true locations but the algorithm is not able to recover from a lost of state; if we don't give the initial state to the algorithm, this only estimates a 20% of the true locations.

For the WiFi observation experiment (*WSLAM)* if we give the initial state to the system, the algorithm converges with a 100% of true locations and the algorithm is able to recover from a lost of state; if we don't give the initial state, the algorithm estimates a 95% of the true locations.

If we use the algorithm with WiFi and Ultrasound observations, the ratings experiment a slightly improvement compared with the results using only WiFi observations. This difference could increase in case of there were a lot of people walking in the environment.

Figure 6 shows part of the results of these experiments without giving an initial state to the algorithm.

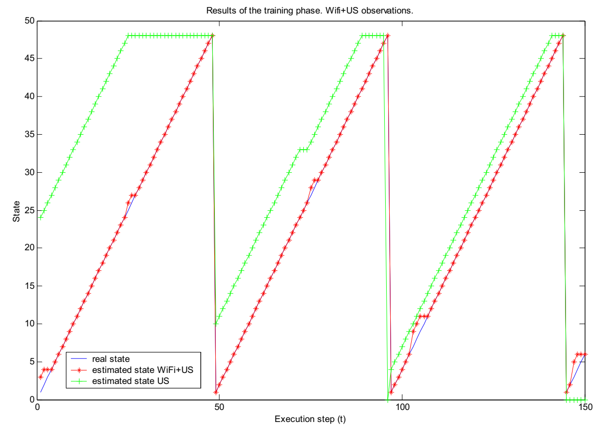

Fig. 6 Results of the training phase

C. Tracking phase

We have tracked the robot's route in several experiments. The results obtained up to date are described in the following.

First we have programmed a destination without give the initial state. The experiments that we have done are:

1) We have initialized the Bel as a uniform distribution and then we have tracked the route of the robot with the ultrasound observation, with the WiFi observation and with the both. In the first case the robot never achieves the target, in the second and the third case the robot achieve the target with a 95% of success.

2) We have initialized the Bel with weighted values in the places where the probability to initiate the robot is highest. With the ultrasound observation, the robot achieved the target in a 20% of the cases. With both observations the results were about 95% of success.

To check that the system is robust as function of time, we programmed several chained targets. The robot was navigating during 3 hours and we obtained a few localization errors as can be seen in Table II.

TABLE II
RESULTS FOR SEVERAL CHAINED TARGETS

		Number	Percentage
Successful	Direct path	26	65%
	Indirect path	12	30%
Failures	Incorrect target	1	2,5%
	Loops	1	2,5%

"Direct path" means that the target is reached following the ideal trajectory while "Indirect path" means that the target is reached after recovering maneuver.

The most remarkable feature of the system is not its successful performance itself that its ability to recover from observation failure situations as can be seen in Table II.

It's important to note that with the WiFi global observation added to the POMDP the algorithm converges in a few execution steps, around of 1 or 2 steps, while with the ultrasound observation only, the algorithm converges in a very higher number of execution steps (more than 30 steps).

As an example of the real application that we used for the tracking phase, we have represented in our control interface (Saphira) a color circles for the offices nodes and color squares for the extreme nodes. When the belief distribution is larger than 0.9 in the estimated state, then the color of the circle/square is green; when the Bel has a value between 0.3 and 0.9 the color is yellow, and when the Bel is lower than 0.3 the color is red. With this color is easy to track the robot. Figure 7 shows a sample of the control program.

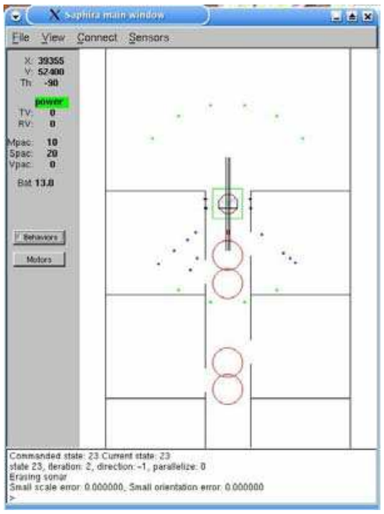

Fig. 7 Sample of the control program

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented a navigation system for indoor environment using a POMDP based on WiFi and ultrasound observations. According to the authors knowledge this is the first work that uses this kind of observation in a POMDP.

We present an autonomous algorithm to obtain the parameters of the POMDP. In this way we obtain the WiFi and ultrasound environment map with a minimum effort.

The probabilistic approach of the WiFi observation matrix models perfectly the noise of this sensor, because the WiFi observation is affected for several factors.

Adding these global observations to a POMDP we demonstrate that the localization algorithm converges faster than if we only use an ultrasound sensor.

With our system we have obtained a global navigation system that is useful in real robotic applications.

In the future, we propose to apply the system to obtain the WiFi map for a different mobile platform such a PDA carried by a man.

We will try to enhance the algorithm to be faster than actual, and then we will propose taking WiFi observation only in the interesting states, such as extreme nodes, obtaining a faster and stronger algorithm.

ACKNOWLEDGMENT

This work has been supported by grants IST-2001-34508 from the European Commission, TIC2002-10744-E (Special Action of the ADVOCATE II Project) from the Spanish Ministry of Science and Technology (MCyT) and DPI2002-02193 (SIRAPEM Project) from the Spanish Ministry of Science and Technology (MCyT). We would like to acknowledge the partners of the ADVOCATE II project. The consortium consists of the following partners: University of Alcalá, GETRONICS DECAN, Universidad Politécnica de Madrid, ATLAS Electronik GmbH, Ifremer, HUGIN Expert A/S, INNOVA S.p.A, and E-MOTIVE.

REFERENCES

[1] I. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle". IEEE Transactions on Robotics and Automation, 7(2): 193-204, 1991.

[2] R. Want, A. Hopper, V. Falco and J. Gibbons, "The Active Badge location system". ACM Transactions on Information Systems, 10, 91-102, January 1992.

[3] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera, Multi-Person Tracking for Easy Living". 3 rd IEEE International Workshop on Visual Surveillance, pp. 3–10, Piscataway, NJ, 2002.

[4] N. B. Priyantha, A. Chakraborthy, and H. Balakrishnan, "The Cricket Location Support System". Proceeding of the 6th ACM MobiCom, pp:155-164, July 2002.

[5] Abowd, "The Smart floor: A Mechanism for Natural User Identification and Tracking". Proceedings of the Conference on Human Factors in Computing Systems, pp. 1–6, Hague, Netherlands, April 2000.

[6] P. Bahl, and V. N. Padmanabhan, "RADAR: A, In-building RF-based User Location and Tracking System". Proceedings of the IEEE Infocom 2000, vol.2, pp. 775–784, Tel Aviv, Israel, March 2000.

[7] P. Enge and P. Misra, "Special issue on GPS: The Global Positioning System". Proceeding of the IEEE, pages 3-172, January 1999.

[8] A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang, "Location Sensing and Privacy in a Context Aware Computing Environment". IEEE Wireless Communications Magazine, pp. 10–17, October 2002.

[9] M. Youssef, A. Agrawala, A. U. Shankar, and S. H. Noh, "A Probabilistic Clustering-Based Indoor Location Determination System". Tech. Report, University of Maryland at College Park, CS-TR 4350, March 2002.

[10] M. Youssef, A. Agrawala, "Small-Scale Compensation for WLAN Location Determination Systems". University of Maryland at College Park.

[11] E.Lopez, R.Barea, L.M.Bergasa and M.S. Escudero "A human-robot cooperative learning system for easy installation of assistant robots in new working environments", Journal of Intelligent and Robotic System, Vol. 40, pp. 233-265. 2004.

[12] M. Ocaña, L.M. Bergasa, M.A. Sotelo, "Robust Navigation Indoor Using WiFi Localization", Proceedings of the 10th IEEE Internacional Conference on Methods and Models in Automation and Robotics, pp. 851-856, Miedzyzdroje, Poland, 30 August-2 September 2004.

[13] A.R. Cassandra, L.P. Kaelbling, J.A. Kurien, "Acting under uncertainty: Discrete Bayesian models for mobile robot navigation", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. 1996.

[14] S. Koenig, R. Simmons, "Xavier: a robot navigation architecture based on partially observable Markov decision process models", Artificial intelligence and mobile robots, pp. 91-122. 1998.

[15] I. Nourbakhsh, R. Powers, S. Birchfield, "Dervish: and office navigating robot", AI Magazine, vol. 16, nº2, pp. 53-60. 1995.

[16] http://www.hpl.hp.com/personal/Jean_Tourrilhes/

[17] http://www.cs.umd.edu/~moustafa/mapi/mapi.html