DRAFT. Accepted in WAF'2004
March 2004 (University of Girona, Spain)

Robot localization using WiFi signal without intensity map

Oscar Serrano José María Cañas Vicente Matellán
Luis Rodero
Universidad Rey Juan Carlos, 28933 Mostoles (Spain)
{oserran,jmplaza,vmo,lrodero}@gsyc.escet.urjc.es

Abstract

This paper describes a method to estimate the position of a mobile robot in an indoor scenario using the odometric calculus and the WiFi energy received from the wireless infrastructure. This energy will be measured by wireless network card on-board a mobile robot, and it will be used as another regular sensor to improve position estimation. The Bayes rule will be used to accumulate localization probability as the robot moves on. In this paper several experiments in a university building are shown. The two major contributions of the presented work are that the self-localization error achieved is bounded, and that no significant degradation is observed when the theoretical WiFi energy at each point is taken from radio propagation model instead of an a priori experimental intensity map of the environment.

Keywords: Robot localization, Bayes rule, Wireless networks.

1 Introduction

Localization in mobile robotics can be defined [7] as the problem of determining the position of a robot. This information is essential for a broad range of mobile robot tasks, as long as the robot behavior may depend on its position. The aim of localization is to estimate the position of a robot in its environment, given a map of the environment and local sensorial data. Robot localization has been recognized as one of the most fundamental problems in mobile robotics [8].

The robot odometric sensors play a critical role to solve the localization problem in wheeled robots, as they provide information about robot movements. In that way, if the initial position is known, the current position could be calculated. Unfortunately, these sensors are noisy and accumulate errors over time [4]. They are accurate enough for local movements but are not suitable for long term localization. Several techniques have been proposed to improve the localization accuracy. For instance, the use of landmarks of known position and Kalman filtering [14]. Without such beacons or environment tailoring, other sensors like sonars, laser and even vision sensors may provide some indirect information about robot position given a map. They have been successfully integrated in a probabilistic framework and offer good localization results [10, 13, 12]. More recently sampling methods (MonteCarlo) have been introduced in the probabilistic approach to speed up the estimation, giving also good results [6].

Our approach is based in the use of the signal power level received from the wireless Access Points (AP) deployed nowadays across hotspots (airports, stations, etc.) of our cities. The main interest in the use of these information in indoor robotics research is based in the fact that they are a cheap, and non intrusive method. The infrastructure has already been deployed (for instance most universities, as our, are endowed with wireless Access Points), it may cover big areas and does not require any additional hardware or environmental engineering to work.

The work described in this paper shows our experiments in the localization problem using data from the wireless network and from the odometry of the robot.

This approach is not new, there are also other works in the literature that have used WiFi signal in localization problems. Some of these experiments have trusted the WiFi signal so much, that they simply triangulate the position, translating power received into distance to the AP. However, these experiments have shown that using this method only poor resolution can be achieved, see [15] or [5],

The combination of odometric data and the WiFi energy received has also been explored by Jason Small [9], Andrew M. Ladd [2] or Sajid M. Siddiqi [3]. They have developed different statistical methods to calculate the location probability. These methods are based on models of the actions that the robot can perform (turn, going forward, etc.) and the predicted observation after these actions had been performed. This prediction is based on an a-priori map of the WiFi energy.

The main difference between our work and these ones is that we are introducing the use a theoretical model of indoor propagation of the signal from the APs. This avoids the need of manually building a "wireless energy map" of the environment. Using this theoretical model we are getting localization error similar to the obtained with the energy map, as we will show in section 3, and it results on an easier implementation.

The rest of the paper will be organized as follows. Second section explains the theory underneath our experiments. Third section is devoted analyzing the results obtained and last section is summarizing the conclusions we have reached.

2 Probabilistic localization

The aim of the experiments described in this paper is to locate a robot inside the indoor scenario shown in figure 1, populated with several access points used for wireless communication. We follow the probabilistic approach, a technique probed widely successful in robot localization with sensors as sonars or laser [10, 11].

In this approach the localization evidence is stored as the probability of being in each possible location, which generates a "probability grid". Such probabilities are continuously updated with the information coming from sensor observations and movement commands [13]. The Bayes rule is used to fuse new information with that already stored. The best position estimation is that of highest accumulated probability.

As stated in (equation 1), the probability of the robot be in the location x is defined as the conditioned probability of such location given all the past observations from robot sensors, which add some indirect position information. Considering only independent observations, at least Markovian independence (equation 2), and following the analysis of [13], the probability can be expressed and computed in an incremental fashion (equation 3).

$$p(x(t)) \quad = \quad p(x(t)/obs(t), obs(t-1), ...) \tag{1}$$

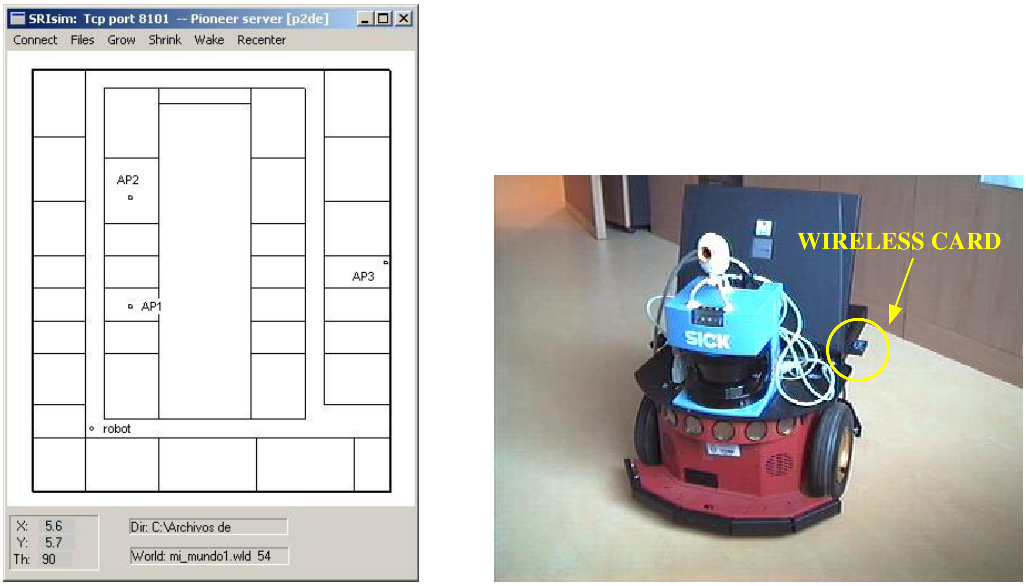

Figure 1: Indoor scenario with 3 access points (left) and our robot with wireless card (right)

$$p(x(t)) = p(x(t)/obs(t)) * p(x(t)/obs(t-1), obs(t-2), ...) \quad (2)$$
$$p(x(t)) = p(x(t)/obs(t)) * p(x(t-1)) \quad (3)$$
$$p(x(t)) = p(x(t)/mov(t-1), x(t-1)) * p(x(t-1)) \quad (4)$$

The robot movements are integrated in the probability estimation following a given *action model*, as shown in (equation 4). That model stores the probability of being at position x if the robot was previously at position $x(t-1)$ and it makes the movement $mov(t-1)$ at time $t-1$.

In practice, the effect of such action model is an evidence displacement over the probability grid. Such displacement follows the robot movement, and in our work is computed from the encoders readings. Some gaussian noise in translations and rotations in added to take into account slippages and encoders deviation from real movement. The noise blurs the evidence displacements.

The sensor observations also modify the probability grid. In (equation 3) $p(x(t)/obs(t))$ represents the posterior sensor model, which contains all the position information carried by the observation, in a probabilistic way. Sometimes this sensor model can be inferred from the a priori sensor model, $p(obs(t)/x(t))$, which contains the probability to obtain the given sensor measurement $obs(t)$ in time t if the robot were at position x at time t.

We use the WiFi energy measurements as the main sensor observations. In a typical indoor scenario there are several access points which provide wireless connectivity to the computers inside, mainly mobile ones. The wireless hardware in such computers provides the WiFi energy of the radio signal received, which is used at low level to choose the closest access point to actually perform the data transmission. We take advantage of such capability to use the wireless card as a *WiFi sensor*. The *WiFi measurement* is then defined as a vector with several components: number of visible access points, and signal and error energy levels for each one. In right picture of the figure 1 the wireless card of our robot is remarked.

An ad-hoc posterior sensor model has been developed to integrate their indirect position information. In order to a WiFi energy measurement to provide some location information it has to be compared with the expected value at each location. A normalized distance function measures the similarity between the expected reading and that actually obtained by the WiFi sensor. It returns 0 to two identical values, and 1 to two completely different readings.

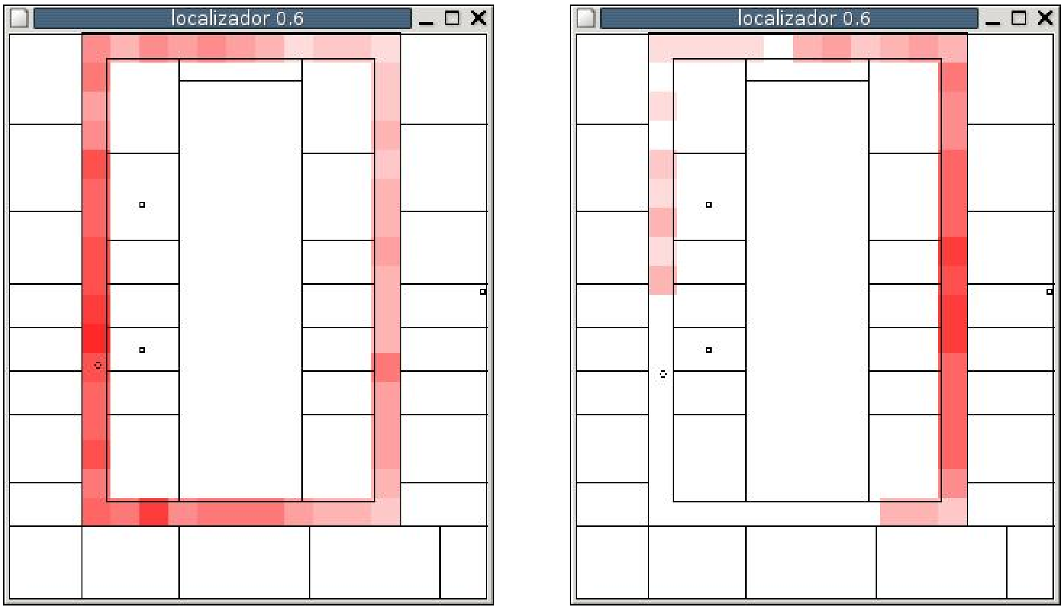

Figure 2: WiFi energy maps for Access Points 1 and 3

The intuition here is that the closer the values, the more likely the location is. If expected and observed readings differ significantly, it is not likely the robot be currently in that location. Otherwise the values would be similar. Nevertheless, if expected and observed readings are similar then the likelihood of the robot being at that position is high. Putting it in another way, the current WiFi energy provided by the wireless network card will raise the probability of the locations with a similar expected energy and will low that of the locations with a very different energy value.

We have tested two probabilistic models to add WiFi data. The first one obtains the expected energy value from an a priori compiled energy map of the environment. Such map was built taking various measurements in all possible locations and collecting the energy received at each one. The second sensor model gets the expected energy value from a theoretical WiFi propagation model. These two models are described in detail in the next two sections.

2.1 A priori WiFi energy map

The a priori energy map that the robot uses is in fact a set of maps, one for each access point, so we have three maps for our test scenario. Figure 2 shows those for access point 1 and 3. The energy tends to be lower at further locations from the access points, as it can be expected, however this attenuation was much lower than expected.

The sensorial model used was $p(x/obs(t)) = 1 - d(t)\sigma$, where σ is an amplification factor and $d(t)$ is computed as the percentage of energies from the sensor reading vector that fall close to its

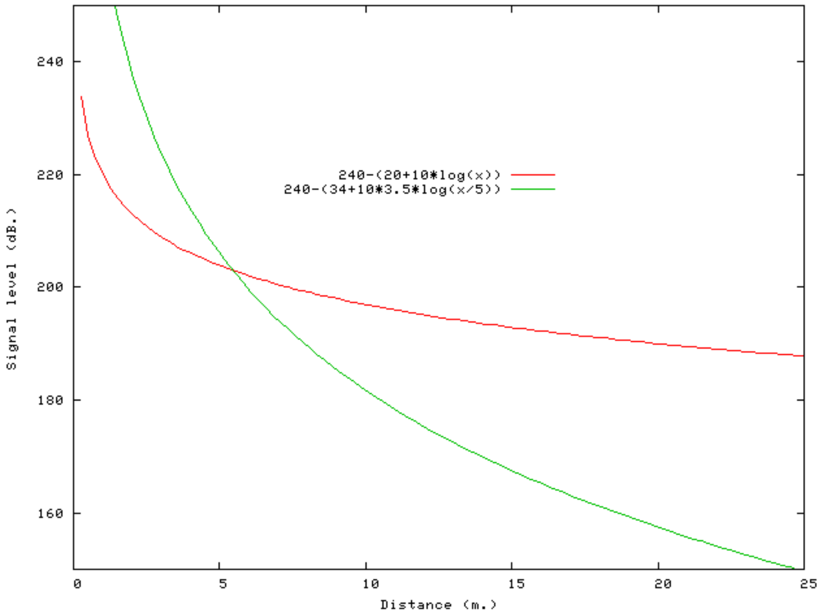

Figure 3: Breakpoint model function with exponent 3.5 after breakpoint at 5 m.

corresponding element in the expected vector. A given threshold is set to consider two energies as close enough.

The maps were built moving the robot through all possible positions and storing the measured WiFi energy values for each access point. In practice the robot position is bounded to the corridors of our Computer Science Department, so no reading was stored inside the different rooms. We discretized the world into 2x2 meter square cells. This distance was chosen because we couldn't find significant differences in the signal level for smaller distances.

2.2 WiFi propagation model

In the second WiFi sensor model the expected value was obtained from a propagation model for the WiFi energy. In particular it follows the equation 5, where distance function decreases exponentially with the differences in energy. In this equation r^i_x is the expected signal level (using the propagation model) in position x for AP i, and r^i_{obs} is that measured by the WiFi sensor.

$$d(t) = e^{-(\sum_{i=0}^{AP}(\frac{|r^i_{obs}-r^i_x|}{100})\sigma)^2} \tag{5}$$

There have been a lot of research about propagation of radio signals in indoor environments, we have chosen the breakpoint model [1]. Our model will compute the signal level for each point as the equation 6. This is a free space loss model that takes only into account the distance from the emitter, and ignores any walls in between. Two different regions are used, before and after a given breakpoint, which was set at 5 meters for our experiments. The figure 3 shows the predicted energy level. It always follows the lower line, keeping a smooth drop for distances under the breakpoint, but falling down at higher rate further.

$$signal\_level = \begin{cases} 240 - (20 + 10 * \log(l)) & \text{if l<=5 meters} \\ 240 - (34 + 10 * 3.5 * \log(l/5)) & \text{if l>5 meters} \end{cases} \quad (6)$$

Figure 4: Propagation models for AP1 and AP3

The use of the model for this experiment allow us to obtain the expected WiFi energies at every location of the grid without actually moving acquiring the data in a previous phase. This is very convenient as the map building phase is very tedious, and takes time. Figure 4 shows the appearance of expected energy readings for access points 1 and 3 according to the propagation model.

In this case, we can easily calculate the energy map for the whole building, not just for the corridors as in the previous approach. We do not need explicit permission to enter private rooms, etc. In this way the experiments made using this approach use the whole map, which are harder conditions (larger grid and more possible errors) than in the previous approach, however, as we will point out in the next section, results are quite similar.

3 Experiments

The localization performance of the probabilistic approach using the two different WiFi sensor models previously described was tested over a simulator. The quality of the algorithm will be measured as the localization error, given that the simulator provides the ground truth and our algorithm continuously delivers a position estimate. For the sake of comparison we also include the deadreckoning estimate, that is, the position computed from encoders, without any correction.

The module architecture of the software is shown in figure 5. We used the standard simulator of our Pioneer robot, SRIsim, which simulates the effect of movement commands and gives the

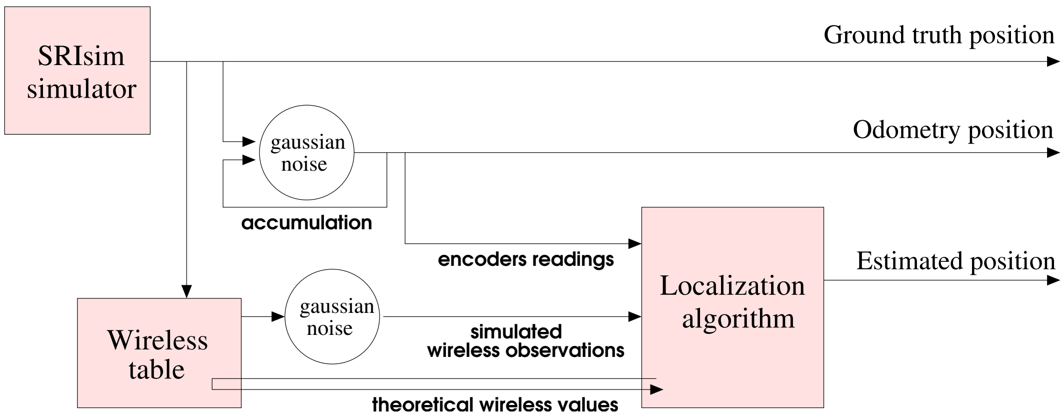

Figure 5: Software architecture for the experiments

real location of the robot (*ground truth position* in figure 5), and the position computed from the encoders (*odometry position* in figure 5). It also provides the simulated sonar and laser readings, but they were ignored in our experiments. We have also coded a wireless module to simulate the WiFi readings using the measures taken by the robot to make the energy map described in section 2.1.

Given the ground truth position, the WiFi readings are always taken from the experimental energy map, with some noise added to provide the *observed* energies (*simulated wireless observations* in figure 5). The localization algorithm uses only such signal level from the WiFi sensor and the odometers readings to continuously compute its position estimate following one of the approaches described in last section. It also accesses the wireless table to get the theoretical wireless reading corresponding to each plausible location.

In the experiments, we let the robot move freely through the Computer Science Department of our university, using a wander behavior. As shown in figure 1 that area is made up by two 34 meter long hallways in the east and west and two of 22 meters in the north and south forming a rectangle surrounded by several rooms of different shapes and sizes. In the same figure three access points can be seen, which were deployed time ago before our experiments.

3.1 Localization using a WiFi energy map

In this experiment the average localization error in 4 random runs of 400 cycles each is 1.08 meters with a standard deviation of 0.70 meters, the error is smaller than 1.5 meters in 82.27% of the times and is smaller than 3 meters in 97.9%. We are going to limit the problem to the localization of a P2DXE robot in the corridors of the Computer Science department.

The left graph of figure 6 shows the distribution of the localization error of this algorithm compared to the accumulated odometric error. The X-axis is the time in seconds and the Y-axis is the distance in centimeters. The accumulated odometric error starts at 0 and is not bounded, however using the WiFi information (lower graph) the error is bounded around a maximum of 3 meters.

The shape of the error using the wireless network is similar to the shape of the odometric error graph because the motion model also relies on the data provided by odometric sensors. There can also be observed some discontinuities in the error graph of our localizator. This is due to the fact the information provided by the wireless card is strong enough to change the belief

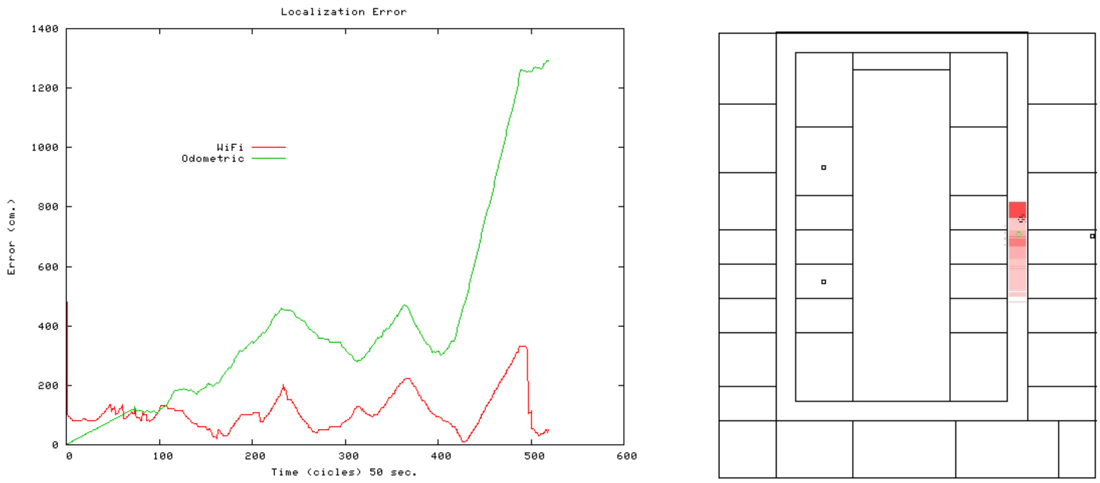

Figure 6: Localization error evolution and evidence after 250 seconds using a WiFi energy map

based on the measurements of the odometric sensors to a position closer than the previous one where the distance function is returning smaller values.

In the right picture of figure 6 can be seen a snapshot of the interface of the localizator algorithm working. The probability cloud is represented in a scale of colors being the darker ones the positions with higher beliefs. The real position of the robot is also shown as a small circle.

3.2 Localization using a WiFi propagation model

Using the theoretical propagation method we have obtained an average error of 2.00 meters with a standard deviation of 1.38 meters, 42% of the times the error is smaller than 1.5 meters and 83% is bellow 3 meters. These data were obtained making the same experiments that in the previous case.

Figure 7, which is the equivalent to figure 6 but using the propagation model, shows the distribution of the error of the localizator versus the odometric error. After the initial steps in which the robot it still updating its beliefs, the error is bounded around 2 meters.

There are more discontinuities than in previous experiment because our model is now also giving signal measures to the points outside the corridor, as can be seen in figure 7. This makes it possible for the localizator to believe that the robot is outside the corridor which forces sudden changes in the beliefs. Most of this sudden changes in our localizator are happening when odometric errors are growing too fast which carries the beliefs to positions far away from the right one and in most cases outside of the corridor which forces these changes.

In the right picture of figure 7 it can be seen how our localizator algorithm works. This time there are probabilities clouds outside the corridor but not in the rooms because we are assigning near 0 probability to positions in walls (they are impossible) and the propagation of evidences that Markov localization makes give very low probabilities to locations inside the rooms, as well as producing the stripes that can be observed.

The setup in this experiment is exactly the same as in the previous one, the only change is in the sensorial model. With this new sensorial model we are getting a better localization than

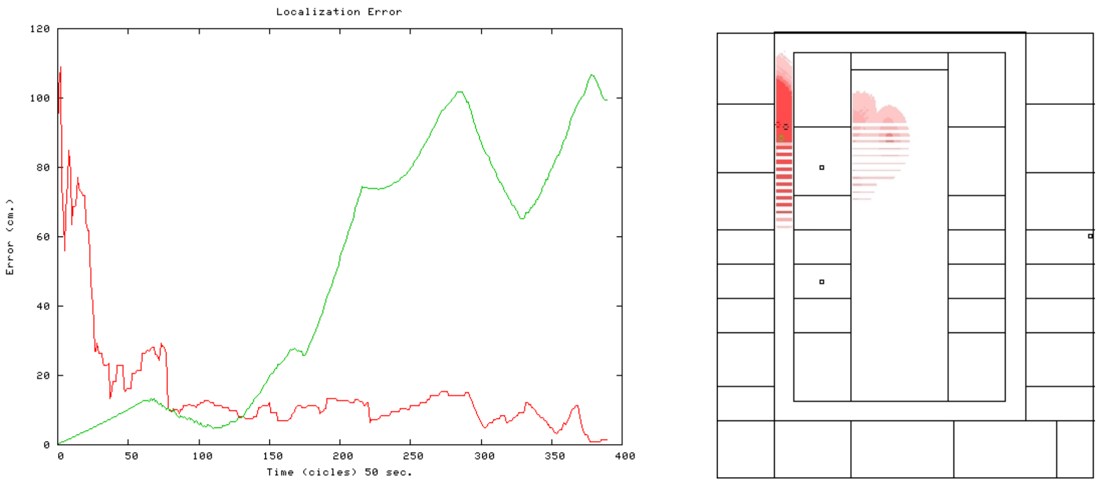

Figure 7: Localization error evolution and evidence after 50 seconds using propagation model

with the previous one. The average error in 4 random runs of 400 cycles each is just of 1.53 meters, with a standard deviation of 1.52 meters. 61% of the times the error is under 1.5 meters and 94.1% is smaller than 3 meters.

4 Conclusions and future work

A probabilistic localization algorithm has been presented which uses a WiFi network card as another sensor in combination with odometric. The card measures the energy received from different communication Access Points in the environment. Experiments described in previous section have shown that, even though the wireless information isn't very discriminative, it's clear it helps in the robot localization task, improving the accuracy estimation over the odometric itself.

In this paper we have also shown that the power signal received from the wireless Access Points can be exploited without any a priori experimental energy map. We have used a Breakpoint model of indoor radio propagation obtaining similar results or even better than the ones reported in the literature, and in our experiments, with a priori energy maps. This kind of methods, without using energy maps, are the most promising ones as they are the only ones that don't require any additional information about the wireless network apart from the positions of the Access Points.

Experiments in the simulator have shown that at the beginning the algorithm with the theoretical propagation model offers localization errors higher than the algorithm with the experimental energy map. But they also indicate that after the initial cycles, error is bounded around 2 meters and the position estimates are better than using the map.

We also argue that though it has been said that orientation affects signal strength [2], in our experiments the changes are slight and it's not possible to use this information to obtain a good estimation of the orientation.

In the near future we plan to repeat these same experiments using the real robot instead of simulator, in order to validate these preliminary conclusions.

References

[1] A.Clarke. A reaction diffusion model for wireless indoor propagation, 2002.

[2] K.E.Bekris A.M.Ladd and A.Rudys. Robotics-based location sensing using wireless ethernet. *MOBICOM'02*, pages 23–26, 2002.

[3] Sajid M. Siddiqi Gaurav S. Sukhatme Andrew Howard. Experiments in monte-carlo localization using wifi signal strength. In *Proceedings of ICAR 2003*, Coimbra, Portugal, July 2003.

[4] J. Borenstein and L. Feng. Umbmark - a method for measuring, comparing, and correcting dead-reckoning errors in mobile robots. Technical Report UM-MEAM-94-22, University of Michigan, 1994.

[5] R.V. Ferré E.G. Villegas. Localización en redes wlan 802.11: desarrollo e implementación de una solución basada en traps snmp. *XIII Jornadas TELECOM I+D*, 2003.

[6] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte Carlo localization: efficient position estimation for mobile robots. In *Proceedings of the 16th AAAI National Conference on Artificial Intelligence*, pages 343–349, Orlando (Florida, USA), July 1999.

[7] G.Wilfong I.Cox. *Autonomous robot vehicles*. Springer Verlag, NW, 1990.

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

[9] A.Smailagic J.Small and D.P. Siewiorek. Determining user location for context aware computing through the use of a wireless lan infrastructure, 2000.

[10] Reid Simmons and Sven Koening. Probabilistic navigation in partially observable environments. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, Montreal (Canada), July 1995.

[11] S.Thrun. Probabilistic algorithms in robotics. 21:93–109, 2000.

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

[13] Sebastian Thrun, Jens-Steffen Gutmann, Dieter Fox, Wolfram Burgard, and Benjamin J. Kuipers. Integrating topological and metric maps for mobile robot navigation: a statistical approach. In *Proceedings of the 15th AAAI National Conference on Artificial Intelligence*, pages 989–995, Madison (Wisconsin), July 1998.

[14] F. Wawak, F. Matía, C. Peignot, and E.A. Puente. A framework for the integration of perception and localisation systems over mobile platforms. In *Proceedings of the 3rd European Robotics, Intelligent Systems and Control Conference EURISCON*, Athens, June 1998.

[15] M.Oliver X.G.Riu and B.Bellalta. Implementación de un servicio de microlocalización wlan. *XIII Jornadas TELECOM I+D*, 2003.