ARIEL: Advanced Radiofrequency Indoor Environment Localization: Smoke Conditions Positioning

Jose Vicente Marti Aviles
Dpto Ingenieria y Ciencia de los Computadores
Universitat Jaume I
Castellon, Spain
vmarti@icc.uji.es

Raul Marin Prades
Intelligent Robotics Laboratory
Universitat Jaume I
Castellon, Spain
raul.marin@icc.uji.es

***Abstract*—Indoor sensor location is a complex task. In normal circumstances laser meters, ultrasonic meters or even image processing may be used to estimate the position of a given node at a particular moment.**

Indoor localization in low-visibility conditions due to smoke is one of the goals that has been studied within the EU GUARDIANS project (http://vision.eng.shu.ac.uk/mmvlwiki/index.php/GUARDIANS). When the density of the smoke grows beyond the 25%, optical sensors such as laser and cameras are not efficient anymore. In these scenarios other sensors must be studied, such as sonar, radar or radiofrequency signals. In this paper we describe the ARIEL method, which uses ZigBee and Wifi signals combinations to localize a mobile sensor in a building such as a warehouse, office or campus. Moreover, the system presents a high intensity LED panel that can be activated via ZigBee in order to have a fine grained localization to get into doors and other points of interest. In addition, a digital compass and a RFID reader are used as a help to the above.

Fingerprinting methods are an alternative to accurate localization of mobile sensors and actuators in indoor environments, which learn a radio map for a given scenario and use this information for calculating the position of a given node. In fact, when using other conventional methods in complex scenarios that may present irregular geometries and materials, fingerprinting techniques can be a very good alternative. Moreover, although they need a previous training of a knowledge database for each scenario, once this is done the method runs in a quite stable and accurate manner without needing any sophisticated hardware.

***Keywords*-Location; Transmitter; ZigBee; WiFi; RSSI; Fingerprinting**

I. State of the art

Localization methods of mobile sensors and actuators is an active research field presenting several fundamental techniques. In fact, for indoor applications, the GPS is either not accessible or not practical to use [1][2].

First of all, some works use the laser range finder as a way to localize a mobile system in an environment [3][4]. This solution is quite straight away when the geometrical map of the building is very well known, including the furniture. Other works focus on using visual landmarks to localize the mobile systems through vision cameras [5][6]. These two alternatives are very accurate in situations of good visibility (e.g. non smoke conditions), although expensive to implement.

Moreover, in the sensor networks community, we can find several cheaper localization methods. In fact, several methods have recently been proposed for determining the position of mobile nodes by measuring radio signals (time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), signal strength) [7][8]. In particular, the TDOA method can use a radio signal combined with a sonar. By measuring the difference time of flight of both the radio and the sonar signal we can estimate the distance between the transmitter and the receiver in a very accurate manner. Some extra work must be done to avoid the effect of reflections.

A. Fingerprinting versus Analytical Localization methods

On the other hand, radio signals can be affected by the geometry of a given indoor scenario as well as by the materials used in the building and furniture. For these situations where the sensors can move in very different scenarios it is possible to enhance the localization method accuracy by applying fingerprint techniques [9][10][11]. This means that the system builds a radio behavior database for the whole room, which is used afterwards to recognize the real position of a node by applying pattern recognition techniques. These methods have the drawback that building such a database is very time consuming, and moreover, this requires more memory and processing power in the nodes to perform the recognition in real time. On the other hand, these methods simplify hardware on the nodes by just using a radio transmitter/receiver (e.g. ZigBee, WiFi, etc.).

In this article, we will propose a combination of methods for sensor location in some unfriendly environments, as fire scenarios with smoky atmosphere (the typical place where fireman actuations are developed), where no other techniques are suitable. In fact, we are going to use the following methods (with their respective estimated accuracy in parentheses):

- Map navigation with a laser-meter (a few decimeters)

Figure 1. Mobile erratic platform block structure

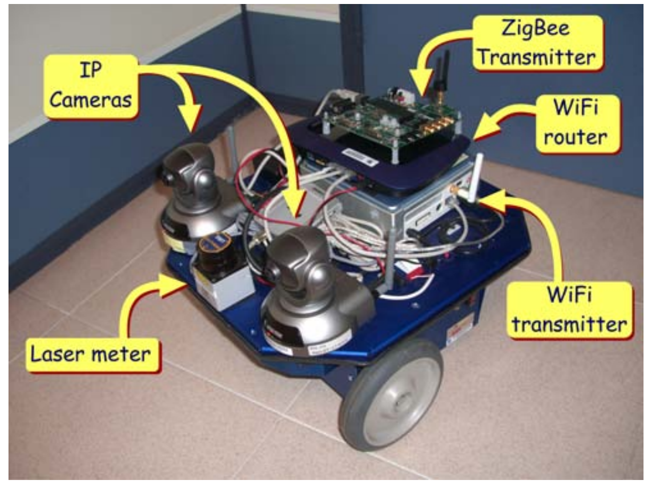

Figure 2. A mobile erratic platform carrying all sensors used

- RSSI measurement with fingerprinting
  - WiFi based location (8 meters)
  - ZigBee based location (1.5 meters)
- Image recognition (a few centimeters)

In a known environment where a map can be used, an easy way to navigate is the laser-meter based method. However, there are symmetric buildings where a low-precision location system may help to correctly estimate the actual position, i.e. the building wing, by suggesting the area where the robot is located. Since the WiFi infrastructure is usually present in public buildings, and even in private ones, and may be used without extra cost, this will be the low-precision location system. ZigBee requires specific devices but provides better accuracy than WiFi due to the greater number of measures - changing the transmitter power and channel- taken for every position. In addition, when approaching to a specific point, as a door, a fire extinguisher, etc. a LED panel (see Figure 3) can be activated by a ZigBee beacon in order to be seen by two cameras and triangulate its position to reach the specific point with enough precision.

II. Hardware description

The hardware platform used in the experiments consists of a mobile robot with all the sensors mounted on it, as shown in Figure 2. The robot structure is shown in Figure 1 where one can appreciate that every sensor is connected to a mini-PC installed in the robotic platform. A wireless router performs the communication task, both between many robot's devices and with the external world.

A. ZigBee location

The system used is based on the CC2430 and CC2431 Texas Instruments microcontrollers. The transceiver uses the ZigBee standard, with the capacity to obtain the RSSI (Received Signal Strength Indicator) from every received packet. Moreover, 16 different channels can be configured with 256 different power levels. This fact has been used to increase the number of samples and packets sent between the beacons and the mobile sensor to improve the localization method efficiency.

On the other hand, the CC2431 microcontroller includes the *Location Engine* system that estimates the distance between each beacon and the transmitter by knowing the original signal intensity and the propagation coefficient in the medium it crosses. Then, by using three or more beacons the system can triangulate the transmitter's position. This will allow us to compare the proposed fingerprinting localization method performance with the mentioned *Location Engine*.

The experiments have been performed by using four transmitters in known positions (beacons) and one mobile transmitter located on top of the robotic platform (see Figure 2). The whole sensor network information comes to the mini-PC, which calculates in real-time its own position. Moreover, the system can dinamically switch when a sensor becomes a beacon or when a beacon becomes a mobile sensor. This makes possible to implement cooperative algorithms for mobile sensor behaviors.

In summary, two different types of communication modules (nodes) are involved in the measurements:

- SRF04 (Evaluation board): This is the board containing the node installed in the robot and is communicated with the mini-PC through a RS232 interface (see Figure 1). It may assume one of the following two roles:
  - Controller: It is the node that is going to calculate its own position by sending and receiving some packets to the beacons.
  - Beacon: These are the nodes located in known positions. They answer to the controller in order to measure the RSSI and perform the calculations. A robot may act as a beacon when is positioned beside a tag (panel or RFID).

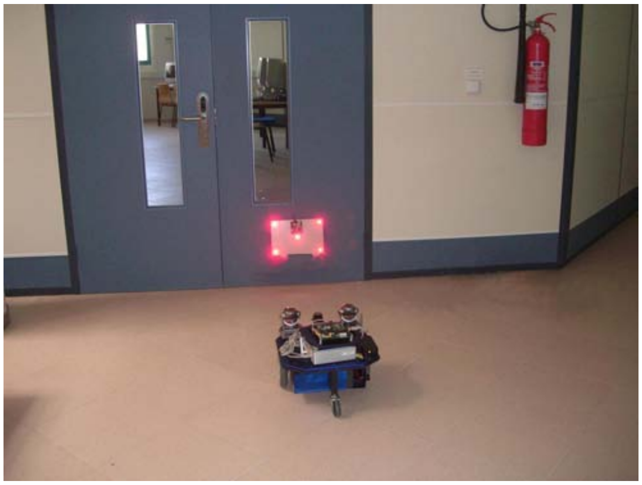

Figure 3. Mobile Erratic-Videre Platform approaching to a LED panel for fine grained localization

- SOCBB (System On Chip Battery Board): This is the most simple board to hold a CC243x. They are going to be always located in a fixed position in the building with two possible functionalities:
  - Simple beacon: This node is a beacon that only returns to a controller every packet received.
  - LED panel controller: As an additional function, the beacon is connected to a LED panel and may activate it when a controller asks to do it.

B. WiFi

The sensor related to the WiFi measurements consists in the mini-PC wireless card adapter. By performing a search for wireless networks, one obtain a list of WiFi routers, their RF channels and the received signal strength.

An arrangement has been made to be able to process this information with the ZigBee location method. The location error obtained is greater than the reached with the ZigBee transmitters because, in that case, several channels and transmitter power levels are used while the WiFi transmitters are using a single channel and power level.

C. Image recognition

Two IP cameras are installed in the robot. These cameras are operated by an application in the mini-PC and used to take images when searching for the LED panel (see Figure 3).

D. Accessories

The robot may be equipped with an RFID tag reader and/or a digital compass sensor, both to make the location task more accurate.

The locations where a beacon should be positioned when performing ZigBee measurements may be signaled with RFID tags. Then, when a mobile robot is approaching to the beacon point, it can activate its RDIF reader and when the tag has been read, the beacon position has been reached with accuracy. Then, the robot may change its role from navigator to beacon and cooperate to the new navigator positioning method.

A digital compass or IMU (Inertial Motion Unit) may help the MCL method (Montecarlo Localization [12]) to compare the robot actual orientation with the calculated one using the laser measurements and the building map. In addition, when approaching to a LED panel, the IMU will provide information about the expected direction where the panel is located and, then, the direction in which the cameras should be oriented to view the panel and perform the visual localization.

Figure 4. Scenario 1, Classroom: B1, B2, B3, and B4 shows the beacons positions

III. Proposed Fingerprinting Experimentation

In this section we will describe the performed experiments. In general, the proposed fingerprinting method works in two phases: (A) *Training* and (B) *Location estimation*. Let us see every phase in detail.

A. ZigBee Training

The training procedure involves taking RSSI measurements in different locations. For this paper we have used three different scenarios: classroom (Figure 4), corridor (Figure 5) and garden. In these measurements, beacons are placed in known positions and the transmitter (mobile sensor) is located at every position in the scenario, using a certain density mesh (typically one meter by one meter). Moreover, data packet transmissions are made in different channels and using different power levels, to enhance the efficiency of the recognition. In fact, for this experiment we used six channels and four power levels in order to cover the whole parameter range provided by Texas Instruments transmitters.

Specifically, for given used channels, the corresponding frequencies can be calculated with equation (1):

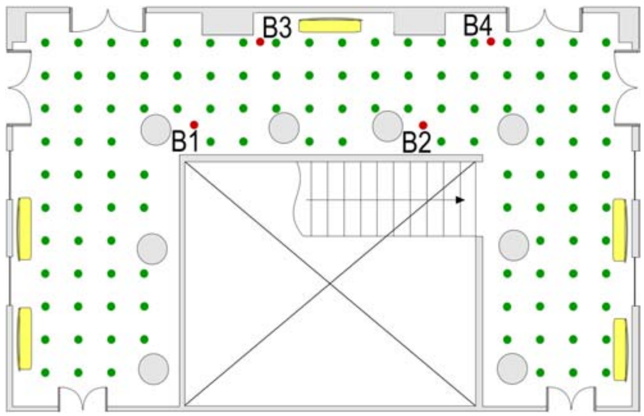

Figure 5. Scenario 2, Corridor

$$F = 2405 + 5 * (ch - 11) \quad MHz \tag{1}$$

Where ch is the channel number and must take a value between 11 and 26. Then, channels 11, 13, 16, 19, 22 and 26, used in this experiment, correspond to the frequencies shown in Table I. Also, the different power levels used can be seen in Table II, where first and last values are, respectively, the maximum and the minimum power the transmitter can generate.

Channel	Frequency
11	2405 MHz
13	2415 MHz
16	2430 MHz
19	2445 MHz
22	2460 MHz
26	2480 MHz

Table I
CHANNEL - FREQUENCY MATCHING

Value	Gain
255	0,6 dB
95	-0,4 dB
19	-5,7 dB
3	-25,2 dB

Table II
VALUE - POWER MATCHING

For every combination of beacon, channel, and signal power, five packets are sent from the transmitter (mobile sensor), which are sent back to the transmitter with the RSSI information. Moreover, the information sent by the beacons to the transmitter uses always the same power level (95, corresponding to -0,4 dB as stated in Table II).

To perform the training procedure the mobile sensor is placed on every position of the scenario, in order to store in the database every RSSI for every combination of beacon, channel, and signal power.

Every packet sent by the transmitter to a beacon contains a four bytes payload as shown in Table III, where the second one contains the beacon number an the rest are reserved for future use.

Byte	Value
1	1
2	i
3	0
4	0

Table III
TRANSMITTER PACKET CONTENTS

When a packet from the transmitter is received, every beacon calculates its RSSI and returns as confirmation a packet with a four bytes payload, as we can see in Table IV, where the beacon x and y coordinates are sent in the first and second bytes. The third byte contains the beacon identification number and the fourth byte contains the obtained RSSI value in -dB (i.e. a positive number between 0 and 90).

Byte	Value
1	x(i)
2	y(i)
3	i
4	dB

Table IV
BEACON PACKET CONTENTS

If a confirmation packet from the beacon is not received by the transmitter in a configurable amount of time, the transmitter sends a retry packet. This operation is repeated a configurable number of times. Finally, if no response is received, the transmitter sets the RSSI to a minimum value of -90 dB for this particular combination of power, channel and beacon.

This process is repeated five times for every channel, power and beacon combination. Once the five measurements are made, a packet which has the structure explained in Table IV is sent to the receiver module with the results.

For every transmitter position, five different channels and four power levels are used against four beacons. This represents a total of 96 different transmissions, each of them producing a packet with the structure shown in Table V.

All this packets are forwarded by the receiver node to the base station PC, through the RS232 serial port. The PC adds to each packet the transmitter real coordinates (previously introduced by hand as reference) and generates a new entry in the signal strength database. This information will contain the transmitter characterization for every position in the scenario.

byte	value
1	Beacon x coordinate.
2	Beacon y coordinate.
3	Beacon number.
4	Power level used by transmitter.
5	Channel used.
6	Signal strength. First packet received by beacon.
7	Signal strength. First packet received by transmitter.
⋮	⋮
14	Signal strength. Fifth packet received by beacon.
15	Signal strength. Fifth packet received by transmitter.
16	Kind of packet indicator.

Table V
RECEIVER PACKET CONTENTS

Once the whole scenario has been measured, some calculations with the received data are done in order to condense the radio map. For this, several possibilities have been studied, which are in fact a work in progress. For example, for every five values set obtained for each location, channel, power and beacon a mean is calculated, reducing with this procedure the amount of information to a fifth. In an alternative procedure, the maximum and minimum values have been discarded and then a mean calculated with the three remaining values.

B. ZigBee Location Estimation

Once the database is trained for a given scenario, the location estimation procedure comes up, which consists of locating the transmitter (mobile sensor) in every location of the scenario. For each position, the obtained RSSI values set is compared with every set stored in the database, calculating some proximity parameters.

The proposed method is a modification of the *k-nearest neighbors* pattern recognition method. For this, we calculate the k-nearest samples in the radio map that have a bigger similarity with the sample obtained at the current mobile sensor position. Then, the recognition result is the more repeated position in this k-nearest vector.

Having in mind this base, the following modifications have been implemented, in order to increase the whole system performance.

- Once we have a RSSI's sample (array) for the current position, we give more weight to the RSSI values received by the beacons than the one calculated from the packets received by the transmitter, since the transmitter changes its signal power and beacon does not. Then beacons will receive different values for different power while transmitter will theoretically receive every confirmation packet with the same signal strength. There are then two parameters (*wfb* - weight factor for beacon and *wft* - weight factor for transmitter) to adjust this.
- Two values do not need to be equals to be considered a RSSI match. In fact, the parameter (*er* - equivalence radius) sets the maximum distance between two signal strength values to be considered identical.
- In addition to matches, for every couple of compared values (last measurement and database stored) the difference between them is calculated and stored. This value will provide extra information for recognition since the smaller this value, the better the match.
- As a result after completing the comparison, eight candidates will be obtained, sorted by match and difference values. Depending on the matching level, the first candidate or the one with more candidate neighbors will be selected. To decide if two candidates are neighbors, the distance between them is calculated and then compared with the parameter *mnd* - maximum neighbor distance.

Then, for every transmitter position one will go over every location stored the in database and calculate the two values (*matches* and *difference*). The matches (*M*) value will be obtained from the equation (2):

$$M = \sum_{b=1}^{4}\sum_{p=1}^{4}\sum_{c=1}^{6}\{|SSB(b,p,c) - MSB(b,p,c)| < er\} * wfb + \tag{2}$$

$$+\{|SST(b,p,c) - MST(b,p,c)| < er\} * wft$$

while the difference (*D*) value will be obtained by evaluating the equation (3):

$$D = \sum_{b=1}^{4}\sum_{p=1}^{4}\sum_{c=1}^{6}|SSB(b,p,c) - MSB(b,p,c)| * wfb + \tag{3}$$

$$+|SST(b,p,c) - MST(b,p,c)| * wft$$

where:

- *b* - beacon id (1...4)
- *p* - power id (1...4)
- *c* - channel id (1...6)
- *SSB* - Stored value for signal strength received by beacon
- *MSB* - Measured value for signal strength received by beacon
- *SST* - Stored value for signal strength received by transmitter
- *MST* - Measured value for signal strength received by transmitter
- *er* - Equivalence radius
- *wfb* - Weight factor in measures received by beacon
- *wft* - Weight factor in measures received by transmitter
- $A < B$ - Takes a '0' value if the expression is true and a '1' value if it is not

C. Choosing the best candidate

Two intermediate values are calculated to help in the best candidate selection:

- *cd*: First candidate difference result respect second candidate difference result.
- *cm*: First candidate matches result respect second candidate matches result.

In both cases, a higher value indicates a better result for the first candidate respect the second candidate.

Four parameters are established as limits to decide:

- *cd_or* and *cm_or* are limit values for *cd* and *cm*. First candidate will be selected if ONE OF THEM is overcome by the calculated value.
- *cd_and* y *cm_and* are limit values for *cd* and *cm*. First candidate will be selected if BOTH OF THEM are overcome by the calculated value.

In other words, if at least one of the two following conditions is accomplished, the first candidate will be selected as the transmitter's nearest location.

$$(cd > cd\_or) OR (cm > cm\_or) \tag{4}$$

$$(cd > cd\_and) AND (cm > cm\_and) \tag{5}$$

Otherwise, a two-dimension array called *dist* containing distances between every candidates will be calculated with the equation 6:

$$dist(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{6}$$

and, from this array, one list *numneighbors* is made to store the number of neighbors of every candidate. Two candidates are considered neighbors if they are closer than *mnd*, thus the array *dist* is searched for every candidate and one neighbor added every time a value less or equal to *mnd* is found.

Once these calculations are made, the candidate with more neighbors will be selected as the best result.

As an additional result, a mean with the selected candidate and its neighbors coordinates is provided, with an extra weight (configurable in parameter *cp* - central point weight) for candidate position.

This method has been compared with the *Location Engine*. It is an analytical method and uses two parameters depending on signal propagation in actual environment (the signal strength to 1 meter distance and the attenuation coefficient). With the obtained measurements this two parameters have been calculated for each scenario and *Location Engine* properly configured to obtain its best performance.

D. WiFi location method

We are going to feed the previously described ZigBee 'Selected' location method with RSSI values from WiFi routers to perform location based in this technology.

The information received when performing a search for wireless networks contains, with other informations, the router ID (a 48 bit number), the ESSID (the network name), the RF frequency and the received signal strength and quality. These two last parameters depend on each other. The signal strength is in dB and is a negative value between -1 and -95. The more negative the value, the weaker the signal. To have a positive number to describe this property of the signal, a calculation is made and the 'quality' parameter is obtained by adding 95 to the 'signal strength' parameter value. The 'quality' parameter is, then, a positive number between 0 and 94 directly proportional to the received signal strength. This information will be received from every wireless router able to communicate with the robot.

For each robot position where a WiFi measurement is made, a list with the router ID and its quality parameter value will be made. Since the number and identity of the routers reached may change depending on the robot position, for every building explored we are going to make a general list containing every router reached from any point of the building.

In the ZigBee method, previously explained, every measurement consists of 96 couples of values (signal strength received from transmitter and signal strength received from beacon for each beacon-power-channel combination), while in the WiFi method each measurement has an undetermined number of sets of one value (as many as routers have been reached by the robot).

To make the WiFi data set compatible with the ZigBee method, we are going to assign a value of zero to the signal strength received from the beacon and set the signal strength received by the transmitter to the 'quality' parameter value. The beacon-power-channel combination will be equivalent to the router ID, so we will be able to register up to 96 routers by building or positioning area.

This method will, obviously, need a previous training phase to fill the database. In the navigation phase, the router ID list will be necessary to be sure that the equivalent 'quality' values are compared accordingly. In addition, when processing WiFi data, the beacon signal strength values will be not used to obtain a shorter calculation time.

Due to the variable number of routers 'visible' from the robot's position, the precision of this method is variable and depends on the area the robot is located. In order to have some information about the precision of the calculated position, a second training phase is conducted to build a table with the position error for each coordinate in the scenario. This will provide an extra information in the navigation phase. Figure 7 shows this information.

E. Visual localization method

When approaching to a LED panel equipped point of interest, such as a door, the system will send a message to the related beacon (one of the LED panel controller SOCBB) to activate the LEDs and then the image recognition phase will begin.

Using the digital compass sensor as a help to know the direction where the panel is expected to be, the robot will start a searching navigation of panel using the IP cameras. Then, the triangulation algorithm will start to calculate the relative position.

For visual localization of the LED panel, a template tracking method has been implemented. This template is matched at each iteration on the video stream during the localization phase of doors and special points of interest. An homography between the template and its match in the current image is computed and used for transforming the LEDs points to its actual pose ([13] [14]).

IV. Experimentation Results

These three methods (Location Engine, Selected and WiFi) have been tested in three different scenarios (except the WiFi method in the 'garden' scenario because the robot detects a very little number of routers there). For every transmitter position, the three methods have been used.

Then, distances between the actual position and the one obtained by every method (i.e., the positioning errors) have been calculated. From this information, sum, mean and typical deviation for every scenario and method have been calculated. All this values are shown in the Table VI for the ***classroom*** scenario, in Table VII for the ***corridor*** scenario and in Table VIII for the ***garden*** scenario.

Location	Location Engine	Selected	WiFi
Point 1	3.16	1.00	8.26
Point 2	3.21	0.00	7.00
Point 3	3.55	0.00	15.33
⋮	⋮	⋮	⋮
Point 53	13.21	0.00	4.29
Point 54	11.99	3.00	3.93
Point 55	16.40	2.00	3.00
ErrSum	440	81	385
ErrMean	8.00	1.47	7.91
StdDev	4.09	1.80	6.52

Table VI
POSITIONING ERROR. RESULTS IN **CLASSROOM** SCENARIO

Figures 6 and 7 compare in a visual way the different methods results in different scenarios. It is easy to see that *Location engine* and *WiFi* methods have similar precision, but the last one uses the building wireless communications system while the first one needs specific hardware. It is not so easy to appreciate that the *WiFi* method shows groups of similar values on its precision, due, as signaled previously, to the different number of routers that communicate with the robot depending on the building area. This behavior will allow to make a table with the expected precision in every robot location.

Location	Location Engine	Selected	WiFi
Point 1	6.04	3.00	8.43
Point 2	3.28	2.00	8.43
Point 3	3.00	3.16	8.00
⋮	⋮	⋮	⋮
Point 120	16,93	1.00	3.61
Point 121	16,37	2.24	4.29
Point 122	16,52	5.00	3.94
ErrSum	1308	255	1120
ErrMean	10.72	2.09	8.14
StdDev	7.15	1.54	5.88

Table VII
POSITIONING ERROR. RESULTS IN **CORRIDOR** SCENARIO

Location	Location Engine	Selected	WiFi
Point 1	2.16	0.00	-
Point 2	1.72	0.00	-
Point 3	2.25	1.00	-
⋮	⋮	⋮	⋮
Point 97	1.66	0.00	-
Point 98	0.85	1.00	-
Point 99	1.34	2.00	-
ErrSum	179	95	-
ErrMean	1.83	0.97	-
StdDev	0.77	0.77	-

Table VIII
POSITIONING ERROR. RESULTS IN **GARDEN** SCENARIO

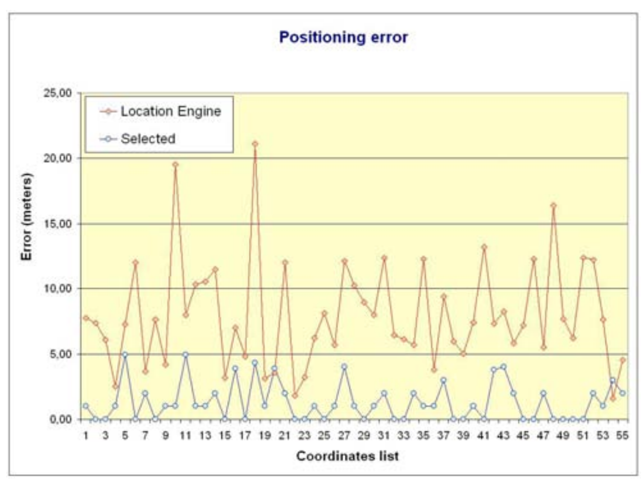

Figure 6. Positioning error comparison between **Location Engine** and **Selected** methods in **classroom** scenario

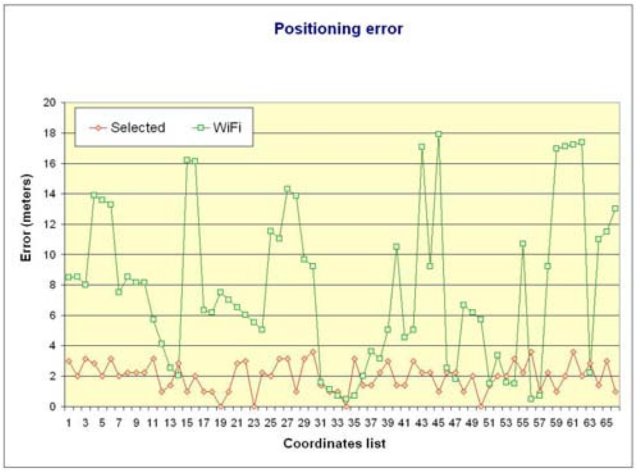

Figure 7. Positioning error comparison between **Selected** and **WiFi** methods in **corridor** scenario

V. Conclusions and Future Work

The different location methods developed and integrated in the robotic platform constitute a wide offer, allowing to use one or several of them at every moment, depending on the required accuracy and the visibility level in the scenario. When navigating in an area without points of interest with good visibility, one can use the map based method with the laser meter helped by the IMU and the WiFi localization method. In critical areas with not enough visibility to use the laser meter where a high accuracy is desired to reach a point of interest, the ZigBee method provides the required precision to reach a point that makes visible the LED panel and complete the positioning by visual methods.

It is necessary to consider that the proposed *selected* and *WiFi* methods require a previous training for every given scenario, and more hardware resources in the sensor nodes in order to perform the calculations in the pattern recognition phase.

The paper has shown a proposed fingerprint algorithm for enhancing the efficiency of localization methods in indoor environments with irregular scenarios. The method increases the performance of a k-nearest neighbors recognition and shows very good results in every tested scenario. Combined with high luminosity visual localization and RFID may allow a robot to navigate in a smoky atmosphere and reach specific points of interest to help firemen.

Future work will focus on managing the described methods and decide which one to use in every moment or how to mix them to an optimum system performance.

Acknowledgments

This work has been funded in part by the EU-VI Framework Programme under grant IST-045269- "GUARDIANS" of the EC Cognitive Systems initiative, the Spanish Ministry of Education and Science (MEC) under grant AP2006-00222, the Bancaja-UJI research program under grant RETA (P1-1B209-39) and the Spanish National CICYT under grant TIN2009-14475-C04.

References

[1] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *Personal Communications, IEEE*, vol. 7, no. 5, pp. 28 –34, oct 2000.

[2] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001*, p. 10 pp., jan. 2001.

[3] G. Cen, N. Matsuhira, J. Hirokawa, H. Ogawa, and I. Hagiwara, "Mobile robot global localization using particle filters," *ICCAS 2008. International Conference on Control, Automation and Systems, 2008*, pp. 710 –713, oct. 2008.

[4] M. Hentschel, O. Wulf, and B. Wagner, "A GPS and laser-based localization for urban and non-urban outdoor environments," *IROS 2008. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008*, pp. 149 –154, sept. 2008.

[5] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *Signal Processing Magazine, IEEE*, vol. 19, no. 2, pp. 17 –29, mar 2002.

[6] S. Atiya and G. Hager, "Real-time vision-based robot localization," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 6, pp. 785 –800, dec 1993.

[7] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," vol. 3, march-3 april 2003, pp. 1734 – 1743

[8] Z. Shan and T.-S. Yum, "Precise localization with smart antennas in ad-hoc networks," nov. 2007, pp. 1053 –1057.

[9] Q. Yao, F.-Y. Wang, H. Gao, K. Wang, and H. Zhao, "Location estimation in ZigBee network based on fingerprinting," dec. 2007, pp. 1 –6.

[10] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piche, "A comparative survey of wlan location fingerprinting methods," march 2009, pp. 243 –251.

[11] S. Outemzabet and C. Nerguizian, "Accuracy enhancement of an indoor ann-based fingerprinting location system using particle filtering and a low-cost sensor," may 2008, pp. 2750 –2754.

[12] Y. Wang, D. Wu, S. Seifzadeh, and J. Chen, "A moving grid cell based MCL algorithm for mobile robot localization," in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, 2009, pp. 2445 –2450.

[13] D. Dementhon and L. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, no. 1/2, pp. 123–141, June 1995.

[14] M. Prats, P. Martinet, A. P. del Pobil, and S. Lee, "Robotic execution of everyday tasks by means of external vision/force control," *Intelligent Service Robotics*, vol. 1, no. 3, pp. 253–266, 2008.