2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

A WiFi-based Software for Indoor Localization

Noelia Hernández*, Manuel Ocaña*, Sergio Humanes*, Pedro Revenga*, David P. Pancho† and Luis Magdalena†
*Department of Electronics, University of Alcalá, Madrid (Spain)
Email: {nhernandez,mocana,sergio.humanes,revenga}@depeca.uah.es)
†European Centre For Soft Computing, Mieres, Asturias (Spain)
{david.perez,luis.magdalena}@softcomputing.es)

***Abstract*—Indoor localization is increasingly required for applications like deployment of rescue teams in emergency situations, proactive care for the elders, and so on. The quick growing of coverage of WiFi networks makes WiFi technology a very promising choice for indoor localization. But, this localization should be linked to a map to be useful. This work presents an open-access software designed for that purpose. It is composed of two different applications, a desktop software for research purposes and an Android application for user friendly localization. We address the localization task as a high dimensional classification problem. So far, we have developed classifiers based on the classic Nearest Neighbour, Support Vector Machines (SVM) and fuzzy rule-based classifiers. This work is made in the context of the ABSYNTHE project which is aimed at creating human-robot teams. We show a use case of the new software in one of the scenarios of the ABSYNTHE project.**

I. Introduction

The ABSYNTHE project (Abstraction, Synthesis, and Integration of Information for Human-Robot Teams) [1] goal is the development of novel tools and approaches to facilitate communication and coordination in human-robot teams. To do so, a key issue is the collaborative creation and maintenance of high-level semantic descriptions of objects. All team members (humans and robots) share explicit and implicit semantics related to objects placed in their surrounding environment. Even though the way humans and robots perceive objects is very different, in order to achieve effective communication, some common interpretation is needed.

A lot of human activities, especially those involving planning and decision making, can be simplified thanks to the use of the tools developed in the ABSYNTHE project.

Localization is one of the most important information when human-robot teams are collaborating. Each team member must be aware of its own location but also of the location of the others. Hence, having an accurate localization system becomes a key issue. Moreover, information related to localization must be handled in a high abstraction level in order to be shared and useful for both humans and robots.

In this project we focus on indoor localization. Traditionally, the position information has been represented following two different approaches. On the one hand, distance-based localization (also known as metric or Cartesian) is usually adopted in robotics where localization is considered in a low abstraction level with the aim of estimating X-Y coordinates. On the other hand, the topology-based approach is preferred when people are involved in the localization or planning process. It allows a more human friendly communication than the metric approach, discretising the environment into nodes that correspond to differentiating features of the environment [2]–[4]. These approaches have been specially useful in localization systems where no movement models are available and topological information (e.g. been at a doorway) is more relevant than a metric one. Moreover, it is easier for robots to deal with the high-level topology-based localization than for humans tackling with the low-level distance-based approach.

Global Positioning System (GPS) [5] is the standard technology for outdoor localization, but there is not any standard technology for indoor localization yet. Different technologies have been tested for indoor localization: infrared sensors [6], computer vision [7], ultrasound [8], laser [9], radio frequency [10], cellular communication [11], etc.

Among existent wireless technologies (bluetooth, ZigBee, WiFi, etc), localization systems based on WiFi are arising as one of the most popular ones. This is probably due to the quick growing of coverage of WiFi networks. Most public buildings (hospitals, libraries, universities, museums, etc.) are already equipped with WiFi technology and measuring the WiFi signal strength is free of charge for every WiFi network. This fact allows to deploy a localization system based on WiFi without doing any modifications in the environment.

At present, there are some available indoor localization systems based on WiFi. One of the most famous ones is the Google localization service which combines GPS, WiFi and tower cell ID on indoor Google Maps [12] to provide guidance in buildings, but its accuracy is not enough to provide an indoor guidance service yet, according to our experience. Another alternative is Ekahau [13], that provides hardware and software for objects and people tracking, security and health monitoring among other applications, but it is a commercial software only available for its use with their own hardware.

In this work, we present a modular open-access software for WiFi-based indoor localization. It is composed of two different applications, a research desktop software and an Android application for user friendly localization. The modular nature of the software allows the use of different localization algorithms in both applications. So far, the localization algorithms have been developed using the classic Nearest Neighbour classifier, Support Vector Machines (SVM) and fuzzy rule-based classifiers, but the system is ready to take any other localization algorithm.

The rest of the manuscript is organized as follows. Next section presents an overview of WiFi localization systems. Section III describes our proposal. Section IV shows a case study on indoor WiFi Localization. Finally, Section V draws the main conclusions and points out some future works.

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

II. Wi-Fi Localization

Most of the commercial devices equipped with WiFi technology use 802.11b/g protocols which work at 2.4 GHz. This is a free frequency, where some other technologies such as bluetooth and microwave ovens work. The Received Signal Strength (RSS) is a noisy signal which decreases exponentially with the distance to the emitter. It has been used for localization purposes especially in outdoor environments. When working indoors, the RSS is strongly dependent on the building structure due to the multipath effect [14]. Another important issue is that the absorption of part of the signal by people moving around in the environment, which significantly diminishes RSS [15].

In addition, there are significant variations in the RSS when the WiFi device moves distances in the range of the wavelength ($\lambda = 12.5cm$). This effect makes very difficult to estimate the correct location because small variations in the position can lead to high RSS variations [16].

Moreover, most of the WiFi networks are deployed with the goal of maximizing connectivity, but disregarding localization tasks. The number of APs distributed over the environment is usually very high generating the so-called co-channel interference. This effect is crosstalk from two different APs using the same frequency (working at overlapped channels).

As result of all these effects, WiFi signal is extremely noisy. To overcome this problem, our previous work [16] proposed the use of a fuzzy rule-based system created using the open-source tool GUAJE [17] to improve the localization performance, obtaining an accuracy close to 90% in small environments.

To improve accuracy in larger environments, a hierarchical localization system [18] was designed using the classifiers generated with the Weka [19], [20] tool.

III. A New Software For Indoor WiFi Localization

This section presents the new software for topology-based localization that implements our previous work in [16] and [18]. For further reference about the training datasets, procedures and environments, please refer to the above mentioned articles. Two different applications have been designed:

- A desktop software for research purposes which allows to create new environments and to train and test different localization algorithms. Thanks to the software modularity, different localization algorithms can be evaluated on the same environments and compared to each other. The research software allows us to choose the best performing localization algorithm that can be loaded into the Android application.
- An Android application that allows users with a smartphone or a tablet obtain their position inside the environment and be guided to any position.

A. Research Software

This application was developed using C++ under Qt. Qt is a cross-platform application framework that is widely used for developing application software with a graphical user interface.

In this application, an environment is a map, or a group of maps, with user defined topological positions. The environments have to be created before the training stage, in order for the software to have all the necessary information to be able to train the localization system. The localization system will be trained for the environment to allow localization of the user in it. All these 3 steps, creation of the environment, training of the system for the environment and localization in the environment are performed into the research software. The training and localization stages allow the selection of different algorithms that can be executed for comparative purposes.

Fig. 1 shows a flow diagram of the software. This will be thoroughly explained in the next subsections where we will describe the process for new environments creation and both training and localization stages.

1) New Environments: This tool allows the creation of new environments. An environment stores all the necessary information to built the localization system. An environment can be composed of several maps (i.e. the maps of the different floors of a building) and the transitions between maps (i.e. stairs). The maps are composed of topological positions and the connections between them. New positions can be added to the different maps. When a new position is added, the user is asked to go to it, then the system measures the WiFi RSS from all the visible APs and stores it as data for the current position. A unique identifier is given to each position, but the user can also give a name to them facilitating their identification. This way, the user can ask for guidance to the "Entrance of Laboratory 1" instead of guidance to "Position 1". Finally, the positions can be linked to each other to allow or forbid transitions between them. This connections can be also defined between maps.

The environment creation is only available in the desktop software (Fig. 2). All the data collected during the environment creation is stored to be used by the training and localization modules. Once the environment has been stored, it can be opened for modifications, adding or removing maps, positions and connections between them.

The environment creation process can be summarized in the following steps:

- Create or open an environment: New environments can be created using the "New" button. Already created environments can be opened to be modified or used for training using the "Open" button.
- Add maps to an environment: New maps can be added to an environment using the "Add Map" button.
- Add positions to a map: New locations can be added to the selected map using the "Add Position" button. The user will be asked to move to the position to start the RSS measurements. While the system measures the WiFi signal, the user will be asked to click on the location of the position on the map, and to give, if desired, a name to it.
- Add connections between positions: The positions that are physically connected can be linked to indicate to the system the feasible transitions to improve the

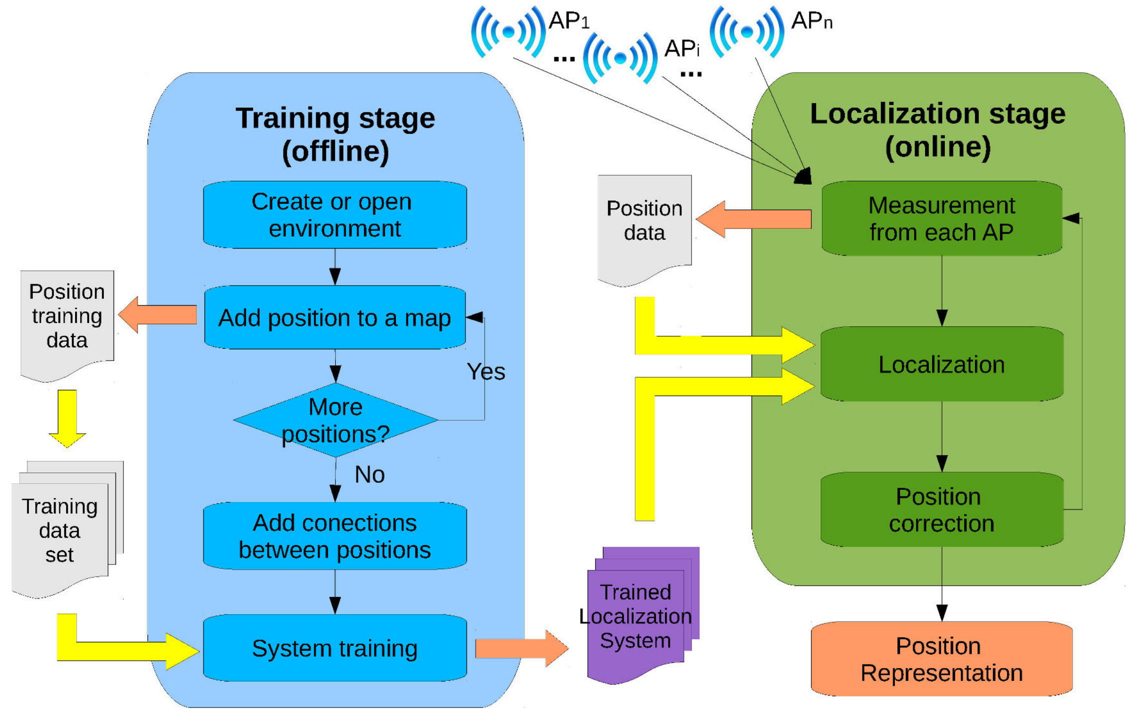

Fig. 1. Flow diagram of the software.

localization. Connections between positions from different maps are also allowed to permit movements between different maps.

- Save the environment: All the information collected in the previous steps are stored for future uses by pressing the "Save and train environment" or "Update and train environment" buttons.

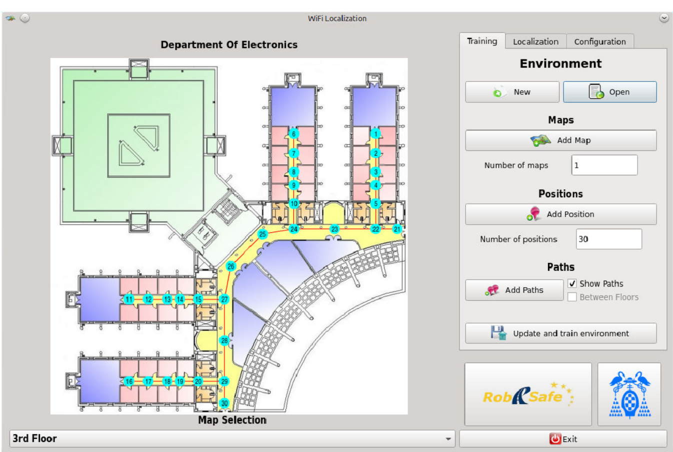

Fig. 2. Screenshot of the training stage in the desktop software.

2) Training stage: This is an off-line stage and it is only available in the desktop software (Fig. 2). Its goal is to train the localization system using all the information collected during the creation of the environment.

The localization system for the environment is trained by clicking the "Update and train environment" button. The training module is called and the system is trained using all the available algorithms, creating all the models that can be selected in the localization stage to obtain the device position.

Thanks to the modularity of the software new localization algorithms can be easily added. The system will be trained using the new algorithms for the selected environment and a new model will be created.

3) Localization stage: In this stage, the WiFi device will obtain its current position using the RSS from all visible APs on an on-line process. The set of classifiers trained in the previous stage can be now used to locate the device. This stage is available in both the desktop (Fig. 3) and Android (Fig. 5) applications. The localization stage comprises three steps as showed in Fig. 1:

- Measurement: The device measures the RSS from every AP and a sample in the necessary format is created to be used by the localization system.
- Localization: The previously created sample is classified using the selected localization method and the estimated position is provided. The localization module will be explained in Section III-C.
- Position correction: The position provided by the localization module can be corrected, if it is an unreachable position from the previous one, using the information about the connections between the positions.

B. Open-Access Localization Application

The open-access localization application [21] allows to locate an Android device using the system trained by the desktop software following the same structure as the one described in Section III-A3.

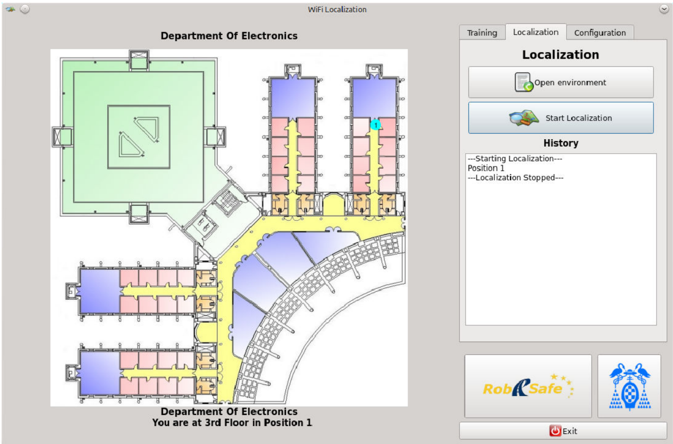

Fig. 3. Screenshot of the localization stage in the desktop software.

The position of the device can also be obtained by scanning a QR code as shown in the flow diagram in Fig. 4. These QR codes are distributed over the environment and contain the information related to the position where they are located.

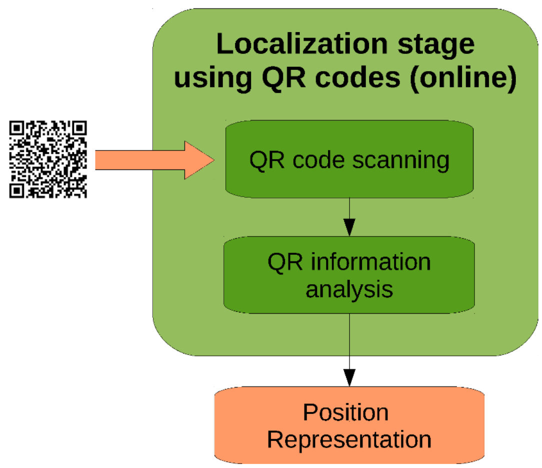

Fig. 4. Flow diagram of the QR localization.

Once the device location is obtained, using one of the methods previously described, the application provides guidance to a selected destination (Fig. 5) using the Dijkstra algorithm [22]. The application also allows the user to ask for a robot to guide him in certain environments. Currently, the localization and guidance application is available for the Polytechnic School of the University of Alcalá, and can be downloaded using the QR codes located at the entrances of the building.

C. Localization Module for the Desktop and Android Tools

The localization module allows the user to select the method, among the ones trained in the training stage, to obtain the current position of the device (Fig. 6).

Currently, four different learning methods are available built using two different open source tools: GUAJE and WEKA:

- GUAJE (Generating Understandable and Accurate fuzzy models in a Java Environment) [17] is a free software tool which implements the Highly Interpretable Linguistic Knowledge (HILK) methodology [23]. This fuzzy modelling methodology focuses on building comprehensible fuzzy classifiers. Applying fuzzy machine learning techniques HILK is able to automatically extract useful pieces of knowledge from experimental data. Such knowledge is represented by means of linguistic variables and rules under the fuzzy logic formalism. The fuzzy models generated by this software have been previously evaluated by the authors in [16]. The best configuration resulting of this evaluation has been used to build the localization algorithm to be used by the software. Specifically, we have used a Fuzzy Decision Tree rule induction algorithm with nine linguistic terms for each input variable.
    - Fuzzy Decision Tree (FDT) [24] generates a neuro-fuzzy decision tree from data which is translated into quite general incomplete rules (only a subset of input variables is considered). In addition, inputs are sorted according to their importance (minimizing the entropy). FDT is a fuzzy version of the popular decision trees defined by Quinlan in [25] and improved in [26].
- WEKA (Waikato Environment for Knowledge Anal-

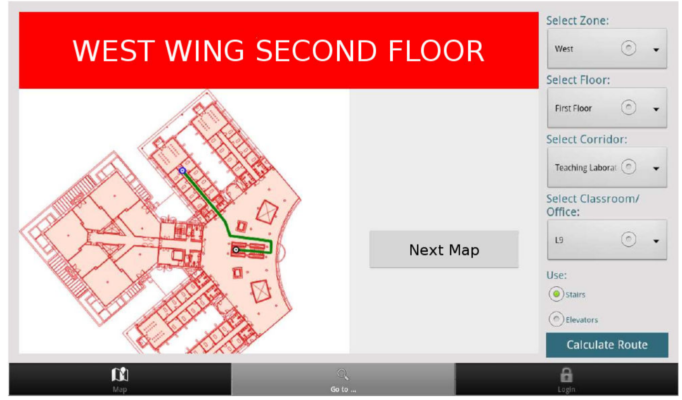

Fig. 5. Screenshot of the Open-Access Localization App.

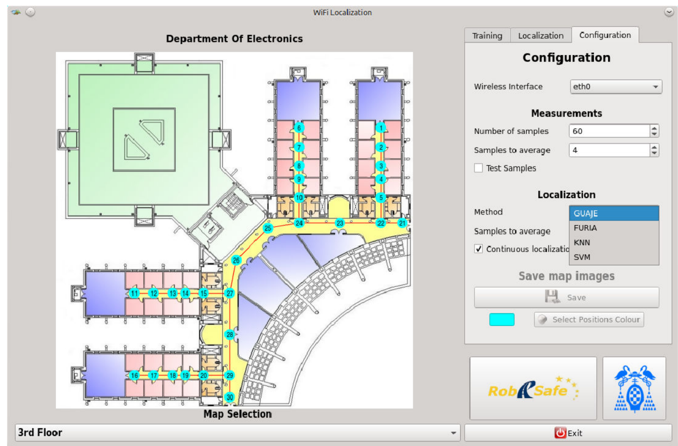

Fig. 6. Localization configuration in the desktop software.

ysis) [19], [20] is open source software issued under the GNU General Public License. It is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. The following classifiers have been used to develop the localization algorithms using the Weka tool, as explained in [18]:

- FURIA (Fuzzy Unordered Rule Induction Algorithm) [27] is a fuzzy modelling method which extends the well-known RIPPER algorithm [28], a state-of-the-art rule learner, while preserving its advantages, such as simple rule sets. In addition, it includes a number of modifications and extensions. In particular, FURIA learns fuzzy rules instead of conventional rules and unordered rule sets instead of rule lists. Moreover, to deal with uncovered examples, it makes use of an efficient rule stretching method.
- KNN (K-Nearest Neighbours) [29] is usually used as baseline to compare with indoor WiFi localization systems [30], [31]. It is a variation of the nearest neighbour algorithm where the most popular class of the k nearest examples is used for prediction.
- SVM (Support Vector Machines) [32] constructs an hyperplane or set of hyperplanes in a high-dimensional space which separates input classes. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

New methods can be easily added since the training and localization algorithms are separated modules of the software.

IV. Indoor WiFi Localization Use Case: The UAH Polytechnic School

This section presents a use case in the third floor of West wing of the Polytechnic School at the University of Alcalá (UAH) (Fig. 7). The environment has an area of $2400m^2$ and 30 significant topological positions (represented by circled numbers in Fig. 7) have been selected for the experiment.

The environment was created and trained using the desktop application on a laptop acquiring 1 sample per second. Two datasets (train and test) were collected manually on different days, one week apart, under real conditions. In the future an autonomous robot will be used to simplify the collection of the training data taking advantage of the ABSYNTHE project capabilities.

After the training process, the different localization algorithms were tested using the desktop application. Fig. 8 summarizes the results for the test dataset. The Y axis represents the accuracy of the system for the four available algorithms.

As can be seen, in the test SVM achieved an accuracy of approximately 85% overcoming the other algorithms. But,

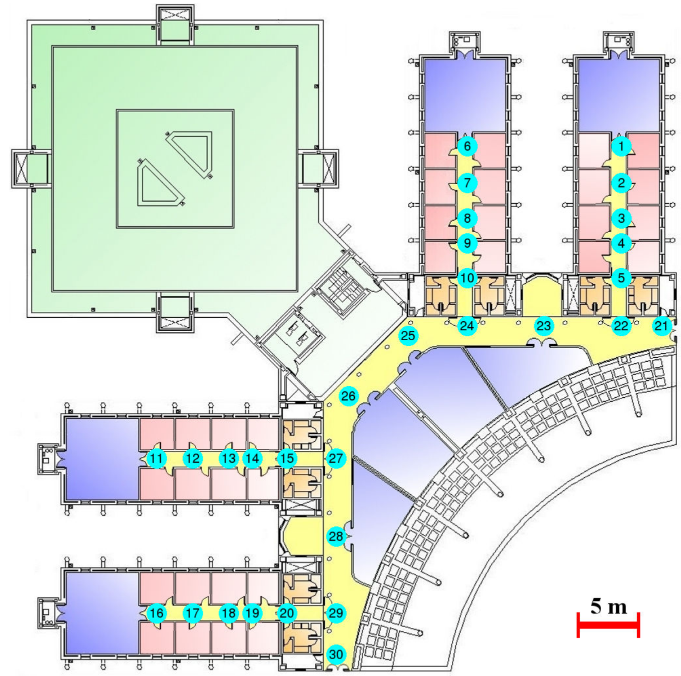

Fig. 7. Experimental environment.

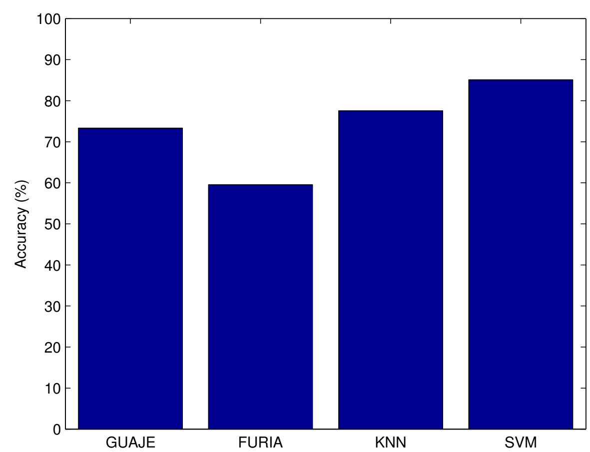

Fig. 8. Results for the use case.

some more test are needed in different-sized environments since the fuzzy classifiers generated using the GUAJE tool have been proved to achieve very good results in small environments.

After the training stage the system was tested using a tablet with the Android application and the SVM module 20 times. Almost all these tests were successful, except two of them where some problems with the robot navigation occurred. After these tests, the system was tested under real conditions in the context of the "Semana de la Ciencia" (Science Week). A tablet was given to the visitors at the main entrance and were asked to go to the show room in the Polytechnic School. Using the Android application the visitors could request assistance from a robot which will pick them up at their current location and guide them to the show room. The localization application was used by four different visitor groups in two different days: Three groups successfully reached their destination, but the communication with the robot failed during the visit of the fourth one and it never received the guidance request. The fourth group were still able to follow the guidance instructions on the tablet to reach their destination without the help of the robot.

V. Conclusions And Future Work

Two different applications has been created, a desktop software for research purposes and an Android open-access application for user friendly localization.

This software allows to create and train localization systems in new environments to be used for guidance. The localization module can be chosen from different options, the fuzzy classifiers created using the GUAJE tool and the FURIA, Nearest Neighbour and SVM classifiers created using the Weka tool. Any other software tool for classification design can be added to the localization software easily since the localization stage is an independent module. We have presented a use case test, showing that the designed WiFi localization system allows indoor localization and guidance.

As part of our future work we plan to add new environments to the open-access Android application. We also plan to improve the localization modules to reduce the error during localization, especially by testing the position correction module.

Acknowledgment

This work has been funded by the "Ministerio de Economía y Competitividad" through TIN2011-29824-C02-01 and TIN2011-29824-C02-02 (ABSYNTHE project), as well as by the University of Alcalá through CCG2013/EXP-066 (ROBOSHOP project).

References

[1] J. M. Alonso, K. LeBlanc, M. Ocaña, and E. Ruspini, "ABSYNTHE: Abstraction, synthesis, and integration of information for human-robot teams," in *International Workshop on Perception in Robotics, IEEE Intelligent Vehicles Symposium*, 2012, pp. P14.1–P14.6.

[2] B. J. Kuipers and Y.-T. Byun, "A robust qualitative method for spatial learning in unknown environments," in *Proceedings of the 7th National Conference on Artificial Intelligence*, 1988.

[3] D. M. Kortenkamp, "Cognitive maps for mobile robots: A representation for mapping and navigation," Ph.D. dissertation, University of Michigan, 1993.

[4] E. López, L. M. Bergasa, R. Barea, and M. S. Escudero, "A navigation system for assistant robots using visually augmented pomdps," *Autonomous Robots*, vol. 19, no. 1, pp. 67–87, 2005.

[5] P. Enge and P. Misra, "Special issue on GPS: The global positioning system," in *Proceedings of the IEEE*, vol. 87, no. 1, 1999, pp. 3–172.

[6] R. Want, A. Hopper, V. Falco, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, pp. 91–102, 1992.

[7] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," in *Proceedings of the IEEE International Workshop on Visual Surveillance*, 2000, pp. 3–10.

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the International Conference on Mobile Computing and Networking*, 2000, pp. 32–43.

[9] R. Barber, M. Mata, M. J. L. Boada, J. M. Armingol, and M. A. Salichs, "A perception system based on laser information for mobile robot topologic navigation," in *Proceedings of the IEEE Annual Conference of the Industrial Electronics Society*, 2002, pp. 2779–2784.

[10] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of the IEEE Computer and Communications Societies*, 2000, pp. 775–784.

[11] B. Parodi, H. Lenz, A. Szabo, H. Wang, J. Horn, J. Bamberger, and D. Obradovic, "Initialization and online-learning of RSS maps for indoor / campus localization," in *Proceedings of the IEEE/ION Position, Location, And Navigation Symposium*, 2006, pp. 164–172.

[12] Indoor Google Maps. (Online Web, accessed on January 2014) Available: http://maps.google.com/help/maps/indoormaps/.

[13] Ekahau - Wi-Fi Tracking Systems, RTLS and WLAN Site Survey. (Online Web, accessed on January 2014) Available: http://www.ekahau.com.

[14] E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: a comparative study," in *Proceedings of the First Annual IEEE Communications Society Conference on Sensor Ad Hoc Communications and Networks*, 2004, pp. 406–414.

[15] A. Bahillo, R. M. Lorenzo, S. Mazuelas, P. Fernández, and E. J. Abril, "Assessment of the shadow caused by the human body on the personal RF dosimeters reading in multipath environments," *Biomedical Engineering*, pp. 133–144, 2009.

[16] J. M. Alonso, M. Ocaña, N. Hernández, F. Herranz, A. Llamazares, M. A. Sotelo, L. M. Bergasa, and L. Magdalena, "Enhanced WiFi localization system based on soft computing techniques to deal with small-scale variations in wireless sensors," *Applied Soft Computing*, vol. 11, no. 8, pp. 4677–4691, 2011.

[17] J. M. Alonso and L. Magdalena, "Generating understandable and accurate fuzzy rule-based systems in a java environment," in *Fuzzy Logic and Applications*, ser. Lecture Notes in Computer Science. Springer, 2011, vol. 6857, pp. 212–219.

[18] N. Hernández, J. M. Alonso, M. Magro, and M. Ocaña, "Hierarchical WiFi localization system," in *Proceedings of the International Workshop on Perception in Robotics, IEEE Intelligent Vehicles Symposium*, 2012, pp. 21.1–21.6.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[20] J. C. Hühn and E. Hüllermeier, "An analysis of the FURIA algorithm for fuzzy rule induction," in *Advances in Machine Learning I*, 2010, vol. 262, pp. 321–344.

[21] ABSYNTHE Application. (2013) Android open-access localization application for the absynthe proyect. [Online]. Available: http://absynthe.depeca.uah.es/www/GPS-UAH/download-ciceroneUAH.html

[22] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[23] J. M. Alonso, L. Magdalena, and S. Guillaume, "HILK: A new methodology for designing highly interpretable linguistic knowledge bases using the fuzzy logic formalism," *International Journal of Intelligent Systems*, vol. 23, no. 7, pp. 761–794, 2008.

[24] H. Ichihashi, T. Shirai, K. Nagasaka, and T. Miyoshi, "Neuro-fuzzy ID3: A method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning," *Fuzzy Sets and Systems*, vol. 81, pp. 157–167, 1996.

[25] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[26] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *Journal of Artificial Intelligence Research*, vol. 4, pp. 77–90, 1996.

[27] C. Hühn and E. Hüllermeier, "FURIA: an algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[28] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the International Conference on Machine Learning*, 1995, pp. 115–123.

[29] D. Kibler and D. Aha, "Learning representative exemplars of concepts: an initial case study," in *Proceedings of the International Workshop on Machine Learning*, 1987, pp. 24–30.

[30] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd International Conference on Mobile systems, applications, and services*, 2005, pp. 205–218.

[31] B.-F. Wu, C.-L. Jen, and K.-C. Chang, "Neural fuzzy based indoor localization by kalman filtering with propagation channel modeling," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 812–817.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.