Neurocomputing 338 (2019) 372–380

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Automatic wireless mapping and tracking system for indoor location

André Sales Mendes, Gabriel Villarrubia, Javier Caridad, Daniel H. De La Iglesia, Juan F. De Paz*

Faculty of Science, University of Salamanca, Plaza de la Merced s/n, 37002 Salamanca, Spain

ARTICLE INFO

Article history:
Received 29 January 2018
Revised 1 July 2018
Accepted 12 July 2018
Available online 9 January 2019

Keywords:
Neural networks
Indoor location system
Mapping and tracking

ABSTRACT

Automatic vehicle tracking systems ease the completion of numerous tasks in different fields. Moreover they can automatically capture information, this feature allows to perform location tasks. These systems can be implemented at airports, in shopping centers and in other large buildings; in this way, wireless network scans will serve as a basis for the creation of signal maps that can be used in indoor location systems. This work proposes an automatic people tracking system which also allows to map Wi-Fi networks in order to localize people indoor. In order to operate the system, information on vehicle movement was used to capture signal maps, with the aim of reducing the need to perform manual calibration and thus, improving the updating of information. The final location is determined by combining information provided by wireless networks, Bayesian networks are employed for this task.

1. Introduction

In the last years, considerable developments have been made in the field of indoor location systems; they have become more precise and can now locate users in real-time. These advances have allowed to leverage location systems in different fields; they can be found in case studies in the areas of medicine [1,2,10], employee monitoring [3], robots [4,5] etc. It is necessary to consider both the technology and the algorithms that can be used in location systems. The current trend in location systems is that they should operate, not one, but various technologies, in order to be able to apply information fusion techniques in the final calculation of a users' position. The purpose of this work is to create a system that will allow to combine the information provided by the different location mechanisms. Moreover, it intends to facilitate the process of location by means of the automatic mapping of signal levels.

Advances in system location have include both hardware and software. In the hardware part, systems with different technologies have been developed, such as Wi-Fi, RFID, Bluetooth, ZigBee or the analysis of electromagnetic fields. These systems have often been combined with other technologies, such as inertial systems. The main problem presented by these systems is low precision, in order to improve it, new algorithms based on fingerprint [12,16] have been applied. These algorithms have been chosen because they allow to locate users more precisely on the basis of the measurement of changes in the signal levels [12,20,21]. The main problem presented by the fingerprint technique is the need to make calibrations of the environment; this means that before the system can be used, a lot of time and efforts are required in order to maintain the information on signal levels updated [12].

This work proposes a multi-agent system which allows to locate devices in indoor spaces by operating iBeacon and Wi-Fi signals. The location system integrates iBeacon in order to determine scanning points, these scanning points are prefixed beforehand as the key points for calibration. At the points at which Bluetooth is detected, automatic scanning of the levels of Wi-Fi and iBeacon signals is carried out with the aim of creating a signal map and in order to make the use of fingerprint feasible. Moreover, the information on signal levels obtained from the iBeacons will be used in order to establish the aisle in the supermarket at which the user is located. After determining the aisle, the device is placed inside it thanks to the measurements made. Once we have these measurements, an application of the Bayesian signal distribution model is performed, obtaining a series of probabilities of stay at a calibration point. The system uses these probabilities to triangulate and calculate the final position.

In addition, time series will be applied in order to reduce the oscillation in the location of users. The proposed system has been used in a supermarket, for this purpose a shopping trolley was developed; the trolley tracks the users path and also guides the users to the place they want to reach. Without leaving out the traditional functions of this device, it is an element that is perfectly integrated with its environment, capable of detecting users and their gestures,

* Corresponding author.
E-mail addresses: andremendes@usal.es (A.S. Mendes), gvg@usal.es (G. Villarrubia), jch@usal.es (J. Caridad), denihiglesias@usal.es (D.H. De La Iglesia), fcofds@usal.es (J.F. De Paz).

https://doi.org/10.1016/j.neucom.2018.07.084
0925-2312/

this allows the trolley to always follow the correct user. The trolley independently follows the user and helps him in his shopping tasks. Moreover, it tells the system where the user is located at all times. All the movements made by the trolley are calculated autonomously, with the data provided to the Tablet by diverse sensors. This tablet is in charge of calculating the movements; these calculations are passed on to the microcontrollers which change these values into the energy that is provided to the motors for movement. This article is structured in the following way: related work is reviewed in Section 2, Section 3 describes the proposal, Section 4 shows the case study and finally in Section 5 the results and conclusions obtained are outlined.

2. Heuristics applied to optimization

Currently, location systems are widespread and their function varies depending on the case study. Outdoor location systems are usually based on the use of the Global Navigation Satellite System (GNSS), such as GPS, GLONASS, Galileo etc. [6], these systems often use different satellite networks to establish the final position. In initial works, we can see how sensor networks, consisting of beacons and tags, are deployed in order to locate objects in enclosed spaces. Along with the evolution of technology, indoor location systems have also been evolving. At the beginning it was possible to find location systems based on RFID technology; this was due to its low cost and because its features allowed to operate active and passive tags, although passive tags had limited capacities.

RFID based systems were implemented in different environments, for example, for the tracking of patients in hospitals [2,7] or to track objects in general [8]. With the appearance of new technologies, such as ZigBee, it was possible to create indoor location systems with greater precision [9]. Unlike Wi-Fi or Bluetooth, ZigBee never came to be integrated in mobile phones, this is why there have been Wi-Fi [10,11]and Bluetooth [12,13]. based indoor location systems.

Apart of looking at the signal level that reaches the devices by operating numerous technologies, different sensors have also been used, such as a camera or accelerometers, in order to implement the systems. In work [14], natural visual markers and inertial systems are used jointly in order to calculate the location of the user. Code based visual markers could be used as an alternative for finding the users' location [15]. New trends in location systems make use of magnetic fields [16], sound and lighting in order to detect the location of the user [17].

When dealing with networks formed by beacons and tags, not only the technology that will be used has to be considered, but also the location algorithm which will be used in the location process. In the case of wireless networks, it is common to use the received signal strength indicator (RSSI) [11], to determine the distance of the device according to the variation of the signal, although more information could be obtained, such as the signal quality LQI (Link Quality Indicator) [18]. There are numerous alternatives, such as signpost, which associates a tag with the beacon that is receiving the strongest signal level; on some occasions the tag itself will emit the signal that is received by the beacons and in others, a beacon will emit the signal and it will be received by other beacons. Some algorithms intend to establish the location more precisely by considering the attenuation of the signal with the distance [11] trilateration and multilateration based algorithms [19], function in this way. The main difficulty that these techniques display is when there are obstacles that attenuate the signal, for this reason the use of fingerprint [12,20,21] is more advisable.

Other studies that propose location systems with Wi-Fi, Bluetooth, magnetic field, sounds etc. use fingerprint. With fingerprint, it is necessary to carry out the calibration processes which allow to relate a particular location with sound, magnetic field or signal levels [12,16]. In order to carry out the calibration process, inertial systems can be used in the same way that they are used during location [16]. This will make the calibration process easier [12] as it will be possible to establish points of origin, destination and to detect the steps of the user, in this way, relating location with signal levels. In recent works, it is common to find location systems that combine inertial systems with Wi-Fi [12,21].

In this article, we intend to simplify the calibration process by using iBeacons and by measuring signal levels automatically in the locations in which they are detected. The use of these iBeacons allows to update signal levels automatically and in addition to integrate new measurement points more simply.

3. Proposal

The Project was carried out using a multi-agent system which allows to control both the hardware of the trolley as well as the functionality of the location system. A multi-agent system has been chosen due to the possibilities it gives when including new functionalities dynamically. Fig. 1 shows the multi-agent system with its organizations. The trolley organization is in charge of operating the vehicle and allows for its movement, it includes the following agents for this purpose: The User detection agent, in charge of detecting and calibrating the user; the Movement agent, gives movement to the trolley through the actuators; the Obstacle detection agent prevents collision while the trolley is moving and finally, the PID agent controls the movement of the trolley. The location organization is in charge of carrying out the calibration and location processes by employing different machine learning techniques. The iBeacon and Wi-Fi agents are in charge of making a wireless scanning. The calibrate and location agents make use of the iBeacon and Wi-Fi agents when obtaining signal levels, both agents receive the scanning data from iBeacon and Wi-Fi and use this information to calibrate and locate, beside the location agent make use of the classifier agent that contains different classifiers in order to calculate the likelihoods belong to each calibrate point. Lastly, there is a database organization, which has a data agent that is responsible for managing the information in the database.

This work is done using a multi-agent system with the aim of generating a structure in which future improvements can be easily implemented. For example, it will be possible to model user behavior, analyze the sales dynamics in supermarkets, as well as propose an inter-relationship evolving system between the different organizations in the shopping cart.

Within each of the organizations, the agents share general information in such a way that it is common to all of them, and the variation in it causes changes in their behavior.

3.1. Description of the trolley's hardware system

This section briefly explains the functioning of the designed autonomous vehicle, as well as its main hardware components. Considering the electronic side of the design, we can distinguish the following elements: DC motors and their drivers for the management of movement, obstacle sensors, a central control unit, a device used to acquire images of the environment (Kinect), Bluetooth and Wi-Fi connection modules in order to make scans and finally a battery that powers the entire system.

The Microsoft Kinect device is a crucial element when tracking a user in the supermarket, making the following process much simpler. This device allows to recognize individuals that appear within the camera's . The API provided by the manufacturer allows to recognize and localize the skeleton of those individuals that appear in an image. This characteristic makes it possible to develop end services for users, which do not require the use of a remote control. To power the Kinect, the ML2596 voltage regulator was

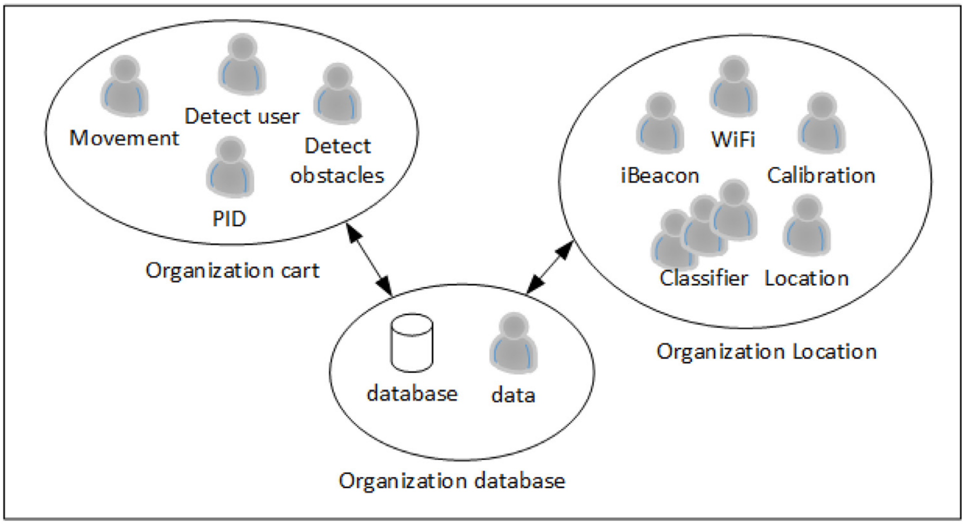


Fig. 1. Multi-agent system with system organizations.

used, in order to adjust the voltage of the battery since it can reach 14 V and the voltage of the Kinect is 12 V.

The device that is in charge of processing the images acquired by the Kinect camera, is the Surface Tablet. It should be remarked that it was necessary to use a tablet with the Windows operating system, since the application used in this case study has been developed with C#.

DC motors and motor drivers, which are connected to the tablet using the Arduino microcontroller, via the USB port, are used to move the trolley.

The computations of motion kinematics for following the user are calculated in the tablet because they are of high computing cost for an 8-bit microcontroller, such as the Arduino.

In order to detect people and objects moving on the sides, ultrasonic distance sensors are used, they allow to detect obstacles easily and efficiently. In order to make scans for subsequent location, a USB Wi-Fi &Bluetooth dongle is used. This allow to make scans with a different card to the one that is used to keep the Wi-Fi and Bluetooth connection established for the iBeacons scanning.

The connections between electronic components, that are used in the prototype, can be seen in the diagram in Fig. 2. An USB HUB has been used for the connections between the Surface Tablet, Arduino, Kinect and the USB dongle.

3.2. Kinematics of movement

The application running on the tablet, is responsible for determining user movement information that appears in the image and to calculate, at every instant, the power that is to be applied to the motors. The calculated power is transmitted to the motors in a uniform way by using a microcontroller and a PID (proportional, integral, derivative). The computational load of the system resides, therefore, in the tablet, however the use of an 8-bits microcontroller Arduino is necessary in order to be able to communicate with the motors through an analog signal. DC current motors are, therefore, the actuators that allow to move the trolley in one direction or another. To change the direction of the trolley, the motor moves in one direction or in the opposite one. The power and speed that has applied by the motors, depend on both the load the trolley is carrying, its distance to the user and the speed at which the user is moving in the supermarket. In order to avoid sudden movements and adapt the speed to each situation, a PID controller was implemented, to control the power applied to the motors at each instant.

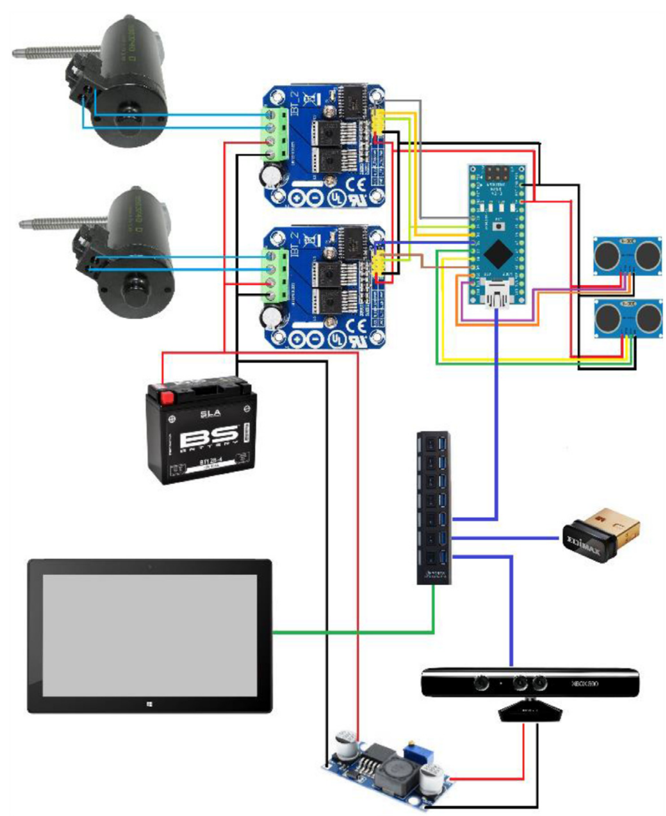


Fig. 2. Diagram of the electronic prototype.

The PID controller is a closed loop control system, which constantly evaluates the output of a variable in order to be able to compare it with a reference signal and calculate the difference between both (this value is called error). This value is introduced into the calculations with the aim of taking the output signal to the reference signal.

The PID controller is operated with the following formula:

$$u\,(t) = K_P e(t) + \frac{K_P}{T_i}\int_o^t e(t)dt + K_P T_d \frac{d}{dt} e(t) \tag{1}$$

In formula (1) we can clearly observe the three terms that contribute to the control of PID. K_P is the proportionality constant, T_i is the integrative constant and T_d is the derivative constant. Similarly, $e(t)$ is the error function and $u(t)$ is the output for the microcontroller. To find a solution to (1), a very simple heuristic method is implemented: once the signal error is obtained, $e(t)$ proportional control is applied immediately to K_P, in order to correct it. In this case, the signal is modified into its multiple in order to correct the error. This multiple is called gain and it will define a more or less rapid approach to the desired value. Once the control is close enough to the objective, the error will decrease and we will need to include a function that will evaluate the evolution of the error over time, modifying our correction of error. This evaluation is carried out by the integral part of the system, which accumulates all the errors that are produced over time. Due to the definition we made for this term, we will never look for T_i values close to zero. Lastly, it is necessary to evaluate the speed of change of this error, and how it is adapted to the desired value, for which we will determine T_d.

In this system, in particular, the PID controller is implemented in 4 stages where the distance between the trolley and the user is evaluated and the error is the difference between the distance of the KINECT device and the distance to the user, which is to be 25 cm. The PID controller is implemented in 4 stages, by means of a heuristic process based on a set of constrained values for the three constants.

First the proportional part is analyzed and the K_p parameter is tuned. This is done by placing T_i at its maximum value and T_d at its minimum. Then, the integral part is analyzed in order to compensate for the disturbances and keep the distance at around 25 cm. One of the key issues is that the value of T_i does not approach 0, since this will generate instability in the system. Once previous values are established, we increase T_d in order to get a faster response, while maintaining previous characteristics.

This control is implemented in the Arduino microcontroller through the motor drivers and the PWM pulses calculated previously by the PID controller.

3.3. Calibration system

The calibration system is created semi-automatically. The key for carrying out the calibration consists of the iBeacons, which are introduced into the environment. The location of these iBeacons is established beforehand and is based on the distribution of obstacles. Each iBeacon is associated with a point in the plan and the MAC address is stored. The iBeacon devices are continuously scanned until the level of the RSSI signal is higher than the established threshold. In that instant, the scanning of Wi-Fi networks and iBeacons begins. The signal levels of RSSI, Wi-Fi networks and iBeacon are stored. For each point p on the plan represented by the coordinates x_p, y_p a set of values r is taken as indicated in the following equation:

$$w^p = \{w_1^p, \dots, w_r^p\} \tag{2}$$

Each of the measurements obtained at time t follows the structure indicated in the following equation:

$$W_t^p = (bssid_1,\ rssi_1^p(t), bssid_2,\ rssi_2^p(t), \dots, bssid_s,\ rssi_s^p(t), x_p, y_p) \tag{3}$$

Thus for the point o, there will be a set of values w^p in different instant of time t.

3.4. Wireless network location system

The location system operates in two stages, first, the section in which the user is located is determined. In order to carry out the detection of the user, the only information required is the one from the iBeacons. The iBeacons are associated to locations and locations are associated to sections, thus, the user is located in the last section in which an iBeacon was detected, with a higher level of signal than the predefined threshold.

Once the section in which the user is located is determined, the system goes on making a location within the section. In order to do the location within the section, supervised learning is carried out by means of the measurement points obtained in the previous step. In order to calculate the position of the users, continuous scanning of the Wi-Fi signal levels and the iBeacons is performed. The process of calculating the location of the Wi-Fi networks and iBeacons follows the scheme shown below.

Once the measurements for the sections are obtained, as indicated in Section 3.3, the system proceeds to carrying out a training. Before the training, it is necessary to modify the measurement data, in order to keep the number of input parameters constant. It should be considered that the measurements obtained for each point may not have a determined BSSID, due to the fact that the signal does not reach the device that is performing the scan. In order to carry out supervised training by means of a Bayesian network, it is necessary to keep the number of inputs constant for all the BSSID that are not detected. They are assigned with a default value which indicated the Wi-Fi and iBeacon signals are out of reach. The following BSSID have been obtained during the calibration process.

$$BSSID = \{bssid_1,\ bssid_2, \dots, bssid_n\} \tag{4}$$

So that the set of measurements obtained for the point p and time t is defined according to (5), here m refers to the value entered when the access point is not visible from p.

$$W_t^p = (bssid_1,\ rssi_1^p(t), bssid_2,\ rssi_2^p(t), \dots, bssid_n, \\ rssi_n^p(t), x_p, y_p)/rssi_d(t) = m \forall bssid_d \notin w_k(t) \tag{5}$$

Therefore, from (5), (2) is redefined as

$$W^p = \{W_1^p, \dots, W_r^p\} \tag{6}$$

Therefore, thanks to (6) the Bayesian network has the same inputs for all the points in which the calibration is carried out, if we denote as RB_W the trained Bayesian network from the data measured in all z points where W is defined according to the following equation:

$$W = \{W^1, \dots, W^z\} \tag{7}$$

When a new tuple is obtained with the RSSI values of each of the access points by the following equation:

$$W_t^* = (bssid_1,\ rssi_1^p(t), bssid_2,\ rssi_2^p(t), \dots, bssid_n, \\ rssi_n^p(t), x_p, y_p)/rssi_d(t) = m \forall bssid_d \notin w_k(t) \tag{8}$$

We proceed to making an estimate of the probability that the measurement W_t^*corresponds to each of the z points in the section, obtaining therefore probability values.

$$P_t = \{p_t^1, \dots \infty, p_t^z\} \tag{9}$$

Where

$$\sum_{i=1}^{z} p_t^i = 1 \tag{10}$$

In addition, the coordinate point p represented as x_p, y_p is also known. From the values given in (9) and the coordinates of each of the points, we proceeded to calculate the estimate of the estimated location at time t defined as $(x,\ y)\ (t)$ by triangulation in (11).

$$(x,\ y)(t) = \left(\frac{1}{z} \sum_{i=1}^{z} x^i \cdot p_t^i,\ \frac{1}{z} \sum_{i=1}^{z} y^i \cdot p_t^i \right) \tag{11}$$

Table 1
List of the electronic components.

Units	Description
1	Microsoft Kinect 360
1	Super Zoom Kinect
1	Surface Pro Tablet 128GB
1	Arduino Nano
2	IBT_2 Driver Motors
2	Motors
2	HC-SR04 Ultrasound Sensors
1	YTX9-BA Battery
1	WIFI Dongle/Bluetooth USB Endimax EW-7611UL
1	ML2596 Voltage Regulator

Because the RSSI signal levels oscillate, even if no movement occurs, it is necessary to smooth the displacements made to avoid jumps and also limit the maximum displacement, according to time. Therefore, the value calculated in (11) is not the final position calculated, as the first step the value of the jump is limited as indicated in (12) where d is defined according to (13)

$$(x,y)(t) = \begin{cases} \dfrac{x(t)+x(t-1)}{d/M}, \dfrac{y(t)+y(t-1)}{d/M} & d > M \\ (x,y)(t) & eoc \end{cases} \tag{12}$$

$$d = \sqrt{(x(t)-x(t-1))^2 + (y(t)-y(t-1))^2} \tag{13}$$

Finally, it is necessary to calculate the time series that reduces the oscillations, for which the previous k measurements are considered, calculating for it a weighted average where the last value has 80% of the weight on the final position and the remaining 20% is distributed among the measurements as indicated in the following equation:

$$\begin{aligned}(x,y)(t) = {} & (x(t)\cdot 0.8 + 0.2\cdot(x(t-1)\cdot 0.8\ldots),\\ & y(t)\cdot 0.8 + 0.2(y(t-1)\cdot 0.8\ldots)\end{aligned} \tag{14}$$

4. Case study

The prototype made for the case study is equipped with a Microsoft Kinect 360 camera, which allows to capture the position of the user and to obtain a cloud of points which is used to detect obstacles that are within its reach. In order to detect the obstacles that are not within the camera's view, HC-SR04 distance sensors have been used, these are installed on the sides of the back of the device, given that there are more probabilities that the trolley will crash with these sides when making a turn. These sensors are connected to Arduino, which sends the distances between the trolley and the objects located at the sides, to an application executed from Surface. The central element of the system is Microsoft's Surface Tablet; it executes the application that is in charge of obtaining the position of the user and the cloud of obstacle points. With these data, the Tablet constantly makes calculations of the direction in which the trolley has to move and of the power that should be sent to the motors through the PID, and it sends it to Arduino.

The Tablet is also in charge of scanning the Wi-Fi networks and the iBeacons through a USB adapter from the Edimax brand, EW-7611UL model, which incorporates Wi-Fi and Bluetooth. The Arduino microcontroller is in charge of receiving the information from the application executed on the Surface Tablet, about the direction where the trolley should move in each instant. The microcontroller receives this information to apply current to the DC motors, using IBT_2 model drivers for DC motors.

Table 1 provides a list of all the electronic components, the number of units needed in the system.

In Fig. 3 we can see the installation point of the different components that have been described previously in the location system. This device has the appearance of a shopping trolley, so it naturally fits the supermarket environment; in this way it was possible to evaluate its performance with real users, who saw the device as an intrinsic element of their everyday experience at the supermarket.

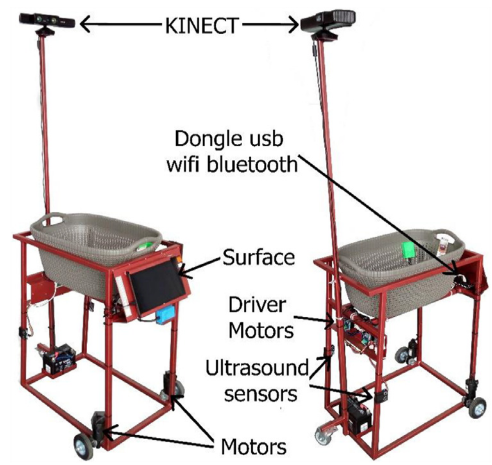


Fig. 3. The trolley's hardware components.

The main task of the trolley is to follow the users in their shopping process. For this reason, it is based on the detection of the movements and gestures made by the users. If the user wants the trolley to follow him, he needs to carry out the synchronization process. The trolley detects gestures in the synchronization process; the gesture chosen for the synchronization and desynchronization of the trolley, is the lifting of two hands. The user has to keep his hands lifted for two seconds. Once the trolley is synchronized, it establishes a continuous detection of the user's gestures. To do this, the device uses the position of the skeleton, captured by the Kinect. Once the position of the user is obtained, the device maps the position of the obstacles in its range of movement, using the cloud of points provided by the Kinect. Obstacles located on the sides are detected using ultrasound sensors. When the position of both the user and the obstacles is known, the device proceeds verifying if the user is at a greater distance than 25 cm from the trolley, if this is so, the device calculates a route that will allow the trolley to get closer to the user without colliding with other users or obstacles. If the user stops and the trolley is at a distance inferior to 25 cm, then it stops as well. In order to stop the trolley from following the user, the desynchronization process has to be carried out. The procedure is the same as when synchronizing the trolley; the user has to lift his hands and keep them lifted for 2 seconds, once desynchronized the trolley stops moving and remains in the same position in which it was desynchronized. If one user is synchronized, it will not be possible for another user to get synchronized with the trolley; a second synchronization will be rejected by the device. In Fig. 4 shows the synchronization, desynchronization steps and the task of following users, more in detail.

The synchronization process follows a pre-established workflow. The first thing to be done is to detect a user. Once a user is detected, his relative position is evaluated. If the user is performing the chosen synchronization gesture, the system synchronizes with him if it was at rest. After synchronization, the loop returns to the beginning with user detection. The position of the user who is now synchronized in the system is re-evaluated. As the user now does not perform the synchronism signal, the system picks up user's position, the obstacles around them and performs the tracking movement towards the user. This cycle is repeated until the synchronized user performs the synchronization gesture again, at this point the system returns to the idle state.

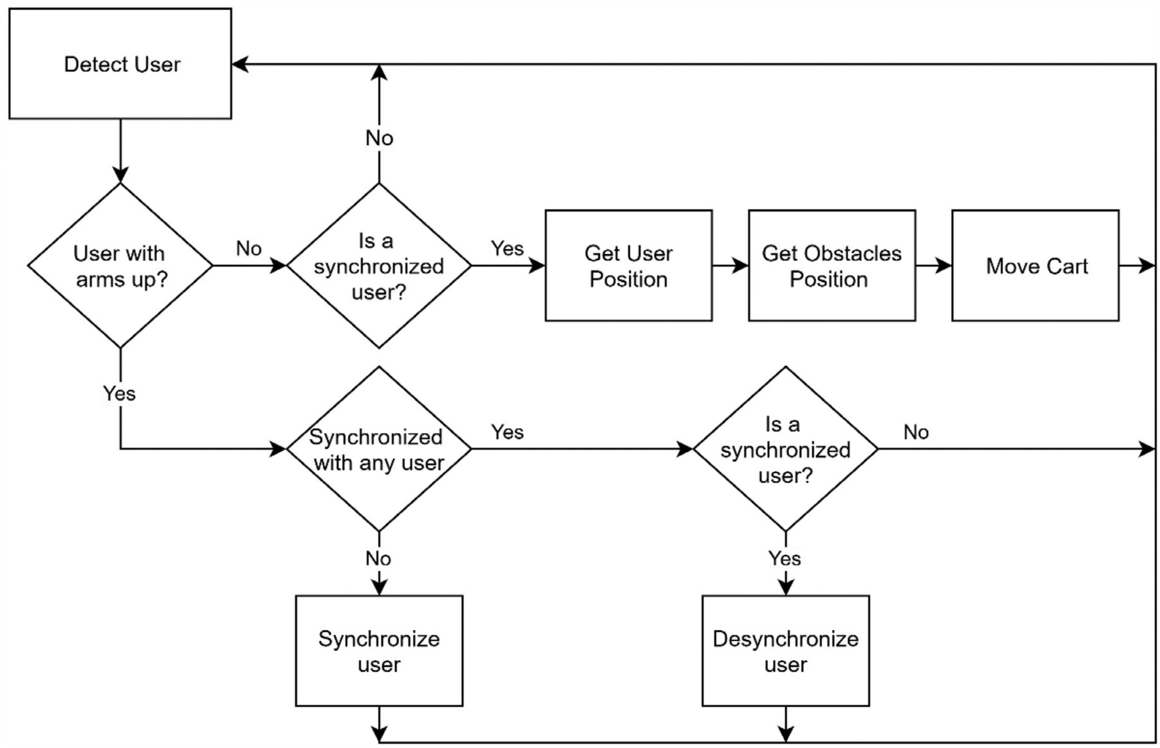


Fig. 4. Synchronization, desynchronization and user following steps.

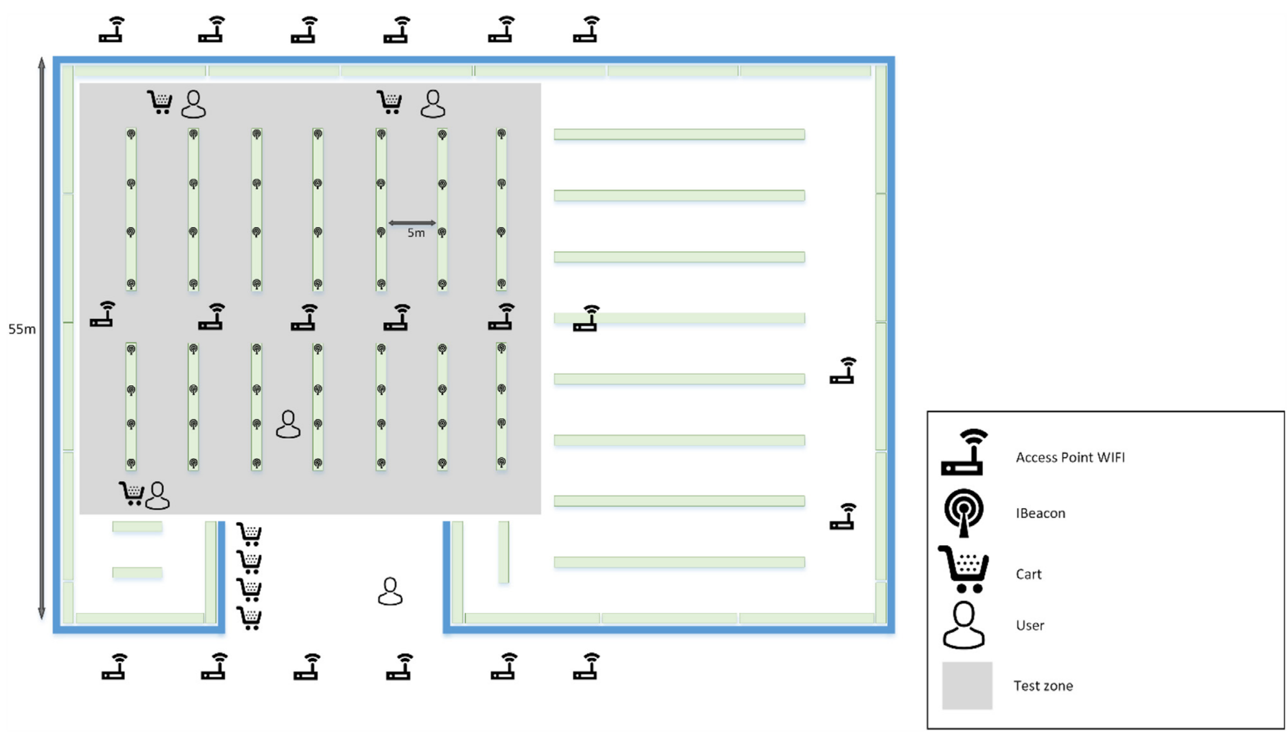


Fig. 5. A distribution map of the iBeacons and Wi-Fi networks.

In order to carry out the calibration process, iBeacons were placed on the supermarket's shelving, as shown in the Figure. The iBeacons were located in a way that there was direct vision with no obstacles at least between two of them, they were placed at the beginning and at the end of each aisle. On the other hand, the Wi-Fi routers were installed at the facility beforehand, therefore, this infrastructure was not modified.

5. Results and conclusions

The system evaluation method consists of mapping and locating an area of a supermarket in order to analyze the performance of the system. In the tests, different phases have been carried out to verify the operation of the different characteristics. Some features, such as user synchronization or motion control using a PID

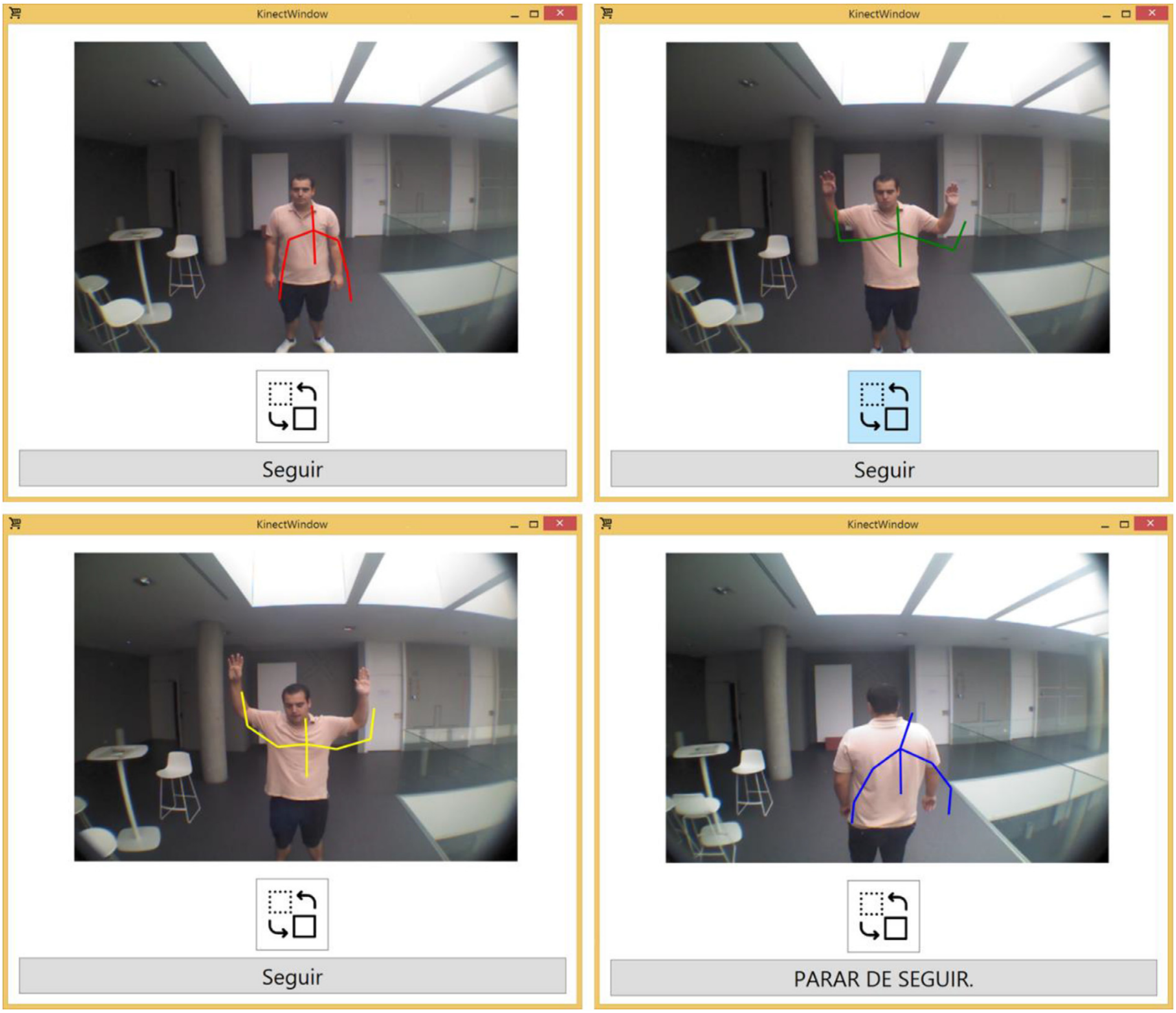


Fig. 6. The user synchronization and desynchronization window. (a) User detection, (b) User synchronization (c) User synchronized (d) Following the user.

controller, are evaluated using heuristic techniques and progressive refinements are made. However, in other areas such as position calculation, an extensive statistical analysis is required.

The system was tested in 14 aisles, with a total of 56 iBeacons and 20 Wi-Fi routers. In the tests, the synchronization, desynchronization and location systems were tested. In order to simplify the tests, an area of the supermarket, which we can see in Fig. 5, was chosen for the case study.

First, the functioning of the synchronization and desynchronization system is analyzed. In Fig. 6, we can see the steps needed to perform the synchronization. The four synchronization stages are represented by different skeleton colors. In Fig. 6a, the red skeleton is used to represent the users that are not synchronized. In Fig. 6b, green is used during the 2-s synchronization process. When the skeleton is yellow, as in Fig. 6c, the user can drop down his hands. At last, when is blue as in Fig. 6d, the user is already synchronized. Until the user does not desynchronize himself, the skeleton remains blue and the trolley follows the user.

In order to analyze the functioning of the system, it was calibrated as indicated in Fig. 4. The scanning frequency for Wi-Fi networks is approximately of 4.6 scans per second, while the iBeacons scan themselves 1.6 times per second. The iBeacon emission frequency was configured at 300 ms. Every time the Wi-Fi scan is finished, the system proceeds storing the values obtained for Wi-Fi and the last iBeacons registered. The system does the same procedure with the iBeacons. During the calibration phase, a total of 1596 values was obtained.

The first step was to determine the efficiency in identifying the supermarket aisles, each aisle was considered to be a different sector that is identified by the iBeacons associated to that aisle. In order to analyze average error, the use of different techniques was analyzed in order to establish the probabilities of belonging to each point and proceed to making the final triangulation. During the testing phase, a total of 10,824 measurements of Wi-Fi networks and iBeacons were made, for each of these measures were marked the x coordinate and the corresponding shelving. The measurements were taken on several routes and they were taken in the same order, due to the fact that they use time series to predict location. In Table 2, average error is shown in meters, obtained during the estimation of the position for the different classifiers. The classifier with the best results was the Bayesian network.

Fig. 7 shows the box diagram with the errors obtained in meters, for the different techniques. From the graph we can see that the Bayesian network and KStar are the algorithms with the best results.

To determine if the difference between the different classifiers is significant, we proceeded to perform a Mann Whitney. First, the

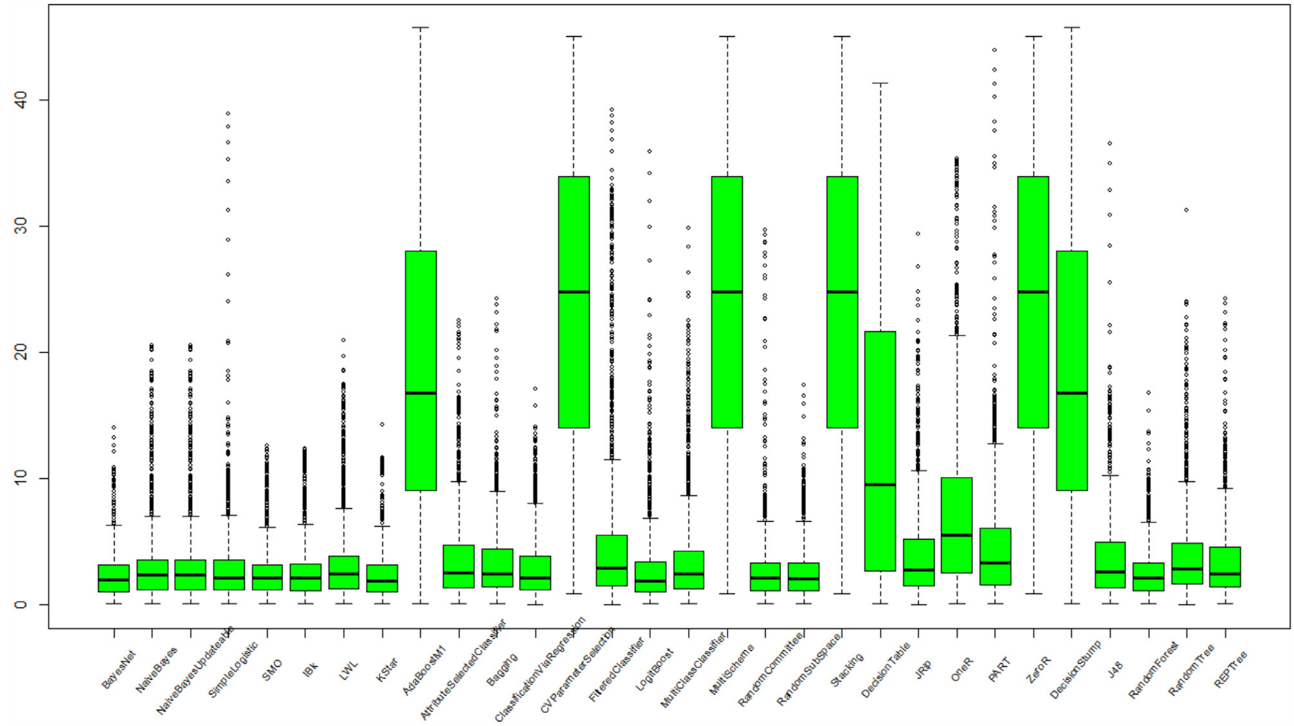


Fig. 7. Diagram of boxes with the error in meters for different techniques.

Table 2
Average error in meters, obtained by the different classifiers.

Classifier	Error	Classifier	Error
BayesNet	2.398849	MultiClassClassifier	3.665971
NaiveBayes	3.025434	MultiScheme	23.73343
NaiveBayesUpdateable	3.025434	RandomCommittee	2.806345
SimpleLogistic	2.974045	RandomSubSpace	2.656918
SMO	2.633645	Stacking	23.73343
IBk	2.659693	DecisionTable	13.13979
LWL	3.471543	JRip	4.040351
KStar	2.437436	OneR	7.443464
AdaBoostM1	18.95733	PART	4.705549
AttributeSelectedClassifier	3.679663	ZeroR	23.73343
Bagging	3.397746	DecisionStump	18.95733
ClassificationViaRegression	2.943992	J48	3.727805
CVParameterSelection	23.73343	RandomForest	2.549873
FilteredClassifier	5.380433	RandomTree	3.972994
LogitBoost	3.057921	REPTree	3.472747

Table 3
Average error in meters, obtained for the different classifiers.

Classifier	Error	Classifier	Error
NaiveBayes	1.92E−07	MultiScheme	0
NaiveBayesUpdateable	1.92E−07	RandomCommittee	0.01662141
SimpleLogistic	0.00036881	RandomSubSpace	0.0100968
SMO	0.0176855	Stacking	0
IBk	0.02355766	DecisionTable	1.06E−220
LWL	2.60E−13	JRip	5.84E−33
KStar	0.6363632	OneR	9.79E−185
AdaBoostM1	0	PART	6.54E−51
AttributeSelectedClassifier	2.12E−22	ZeroR	0
Bagging	4.05E−21	DecisionStump	0
ClassificationViaRegression	7.19E−06	J48	1.04E−22
CVParameterSelection	0	RandomForest	0.0678743
FilteredClassifier	4.06E−45	RandomTree	6.63E−41
LogitBoost	0.08278222	REPTree	7.87E−20
MultiClassClassifier	1.72E−16		

Bayesian network was compared with the rest of the classifiers; H0 the median values were equal and when H1 the median values were different. H0 was only accepted with KStar and was rejected with the other classifiers. Then H1 was changed and it was defined to be the lower value of the median. Therefore, when evaluating the Bayesian Network and the rest of the classifiers, it was not possible to accept H0, so H1 was accepted indicating that the median error of the Bayesian network was considered to be inferior to the rest of the classifiers. The result of the test can be seen in Table 3. Although the result for KStar cannot be considered statistically inferior, we can read from Table 1 that the mean error was lower.

The semi-automatic calibration system simplifies the tasks of calibrating and updating information in the environment, thus simplifying the tasks of deploying the system. In addition, the system allows to incorporate beacons for their calibration and subsequent elimination. However, if continuous calibration is desired, it is necessary to leave the iBeacon deployed. Continuous calibration facilitates the maintenance of the system since it is not necessary to carry out this process in any specific way.

With the use of the triangulation system, which calculates probabilities, it was possible to reduce error, in comparison to techniques like signpost. Signpost is limited to giving the discrete positions in the plane, associated with the points where the iBeacons are located. Similarly, due to the existence of obstacles, the use of fingerprint allowed the system to be more stable; it also allow to attain adaptable precision, depending on the number of iBeacons used and their deployment was at low-cost.

In areas where more precision will be required, it is only necessary to introduce more iBeacons. Thanks to continuous calibration, it is not necessary to carry out the whole fingerprint process manually; this simplifies the adaptation of the system.

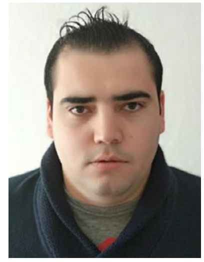
In future work, we will propose the use of the vehicle's odometry, compass and accelerometers, in order to combine location with Wi-Fi signal levels. Odometry has still not been addressed, due to the fact that the motors used in the vehicle's construction have to be modified, since they are not equipped with the necessary encoders for odometry. Compass and accelerometers have not been used either, because sensors for their measurement are not installed.

Acknowledgments

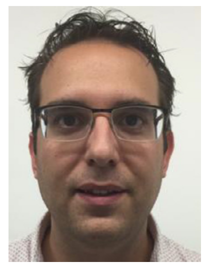
This work has been supported by project Diseño y desarrollo de un carro autónomo de transporte de mercancías para la asistencia a personas con movilidad reducida en centros comerciales. Project co-financed with Junta de Castilla y León y Fondo Europeo de Desarrollo Regional (FEDER) funds. Fundación General de la Universidad de Salamanca PC_TCUE15-17_F2_027.

References

[1] T. Nishimura, K. Koji, Y. Nishida, H. Mizoguchi, Development of a nursing care support system that seamlessly monitors both bedside and indoor locations, Procedia Manuf. 3 (2015) 4906–4913.
[2] L. Calderoni, M. Ferrara, A. Franco, D. Maio, Indoor localization in a hospital environment using random forest classifiers, Expert Syst. Appl. 42 (1) (2015) 125–134.
[3] P.E. Lopez-de-Teruel, F.J. Garcia, O. Canovas, R. Gonzalez, J.A. Carrasco, Human behavior monitoring using a passive indoor positioning system: a case study in a SME, Procedia Comput. Sci. 110 (2017) 182–189.
[4] D. Gu, K.-S. Chen, Design and performance evaluation of wiimote-based two-dimensional indoor localization systems for indoor mobile robot control, Measurement 66 (2015) 95–108.
[5] G. Ernesto Martín, L. Galilea, J. Luis, M. Meca, F. Javier, S.M. David, E.Z. Felipe, P.P. Luis, Infrared sensor system for mobile-robot positioning in intelligent spaces, Sensors 11 (5) (2011) 5416–5438.
[6] R.M. Alkan, M.H. Saka, İ.M. Ozulu, V. İlçi, Kinematic precise point positioning using GPS and GLONASS measurements in marine environments, Measurement 109 (2017) 36–43.
[7] M. Juan, C. Javier Bajo, Y. de Paz, D.I. Tapia, Intelligent environment for monitoring alzheimer patients, agent technology for health care, Decis. Support Syst. 44 (2) (2008) 382–396 ISSN 0167-9236, doi:10.1016/j.dss.2007.04.008.
[8] S. Seol, E.-K. Lee, W. Kim, Indoor mobile object tracking using RFID, Futur. Gener. Comput. Syst. 76 (2017) 443–451.
[9] S.H. Fang, C.H. Wang, T.Y. Huang, C.H. Yang, Y.S. Chen, An enhanced ZigBee indoor positioning system with an ensemble approach, in: IEEE Commun. Lett, 16, 2012, pp. 564–567. April.
[10] G. Villarrubia, J. Bajo, J.F. De Paz, J.M. Corchado, Monitoring and detection platform to prevent anomalous situations in home care, Sensors 14 (6) (2014) 9900–9921.
[11] J.F. de Paz, D.I. Tapia, R.S. Alonso, C. Pinzón, J. Bajo, J.M. Corchado, Mitigation of the ground reflection effect in real-time locating systems based on wireless sensor networks by using artificial neural networks, Knowl. Inf. Syst. 34 (1) (2013) 193–217.
[12] J. Bajo, J.F. De Paz, G. Villarrubia, J.M. Corchado, Self-organizing architecture for information fusion in distributed sensor networks, Int. J. Distrib. Sens. Netw. 11 (3) (2015).
[13] G. de Blasio, A. Quesada-Arencibia, C.R. García, J.M. Molina-Gil, C. Caballero-Gil, Study on an indoor positioning system for harsh environments based on Wi-Fi and bluetooth low energy, Sensors 17 (6) (2017) 1299.
[14] M. Neges, C. Koch, M. König, M. Abramovici, Combining visual natural markers and IMU for improved AR based indoor navigation, Adv. Eng. Inf., 31, 2017, pp. 18–31.
[15] G.C. La Delfa, S. Monteleone, V. Catania, J.F. De Paz, J. Bajo, Performance analysis of visualmarkers for indoor navigation systems, Front. IT EE 17 (8) (2016) 730–740.
[16] X. Huang, S. Guo, Y. Wu, Y. Yang, A fine-grained indoor fingerprinting localization based on magnetic field strength and channel state information, Pervasive Mob. Comput. 41 (2017) 150–165 ISSN 1574-1192.
[17] N.Ul Hassan, A. Naeem, M.A. Pasha, T. Jadoon, C. Yuen, Indoor positioning using visible LED lights: a survey, ACM Comput. Surv. 48 (2) (2015).
[18] P. Barsocchi, M.G.C.A. Cimino, E. Ferro, A. Lazzeri, F. Palumbo, G. Vaglini, Monitoring elderly behavior via indoor position-based stigmergy, Pervasive Mob. Comput., 23, 2015, pp. 26–42. ISSN 1574-1192.
[19] Z. Li, T. Braun, D. Dimitrova, A passive wifi source localization system based on fine-grained power-based trilateration, in: Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015.
[20] S. He, G. Chan, Wi-fi fingerprint-based indoor positioning: recent advances and comparisons, IEEE Commun. Surv. Tut. 18 (1) (2016).
[21] V. José Luis Carrera, Z. Zhao, T. Braun, Z. Li, A. Neto, A real-time robust indoor tracking system in smartphones, Comput. Commun. 117 (2017).



André Sales Mendes PhD Student on Computer Engineer. He is a researcher at the BISITE Research Group (http://bisite.usal.es). He has graduated in Computer Engineering at the University of Salamanca. In the same university he also has obtained a master's degree in Intelligent Systems. During the past years I have won several awards and scholarships related to innovation and artificial intelligence.



Gabriel Villarrubia González PhD Student on Computer Engineer. Currently, he is Assistant Professor at the Department of Computer Science at the University of Salamanca and member of the BISITE Research group. He obtained a Technical Engineering in Management of Computer in 2009 at the Pontifical University of Salamanca, then an Engineering in Computer Sciences degree in 2010 at the same university and postgraduate in Intelligent Systems at the University of Salamanca in 2011. He is co-author of more than 50 papers published in recognized journal, workshops and symposiums, most of them related to Intelligent Systems, Ambient Intelligence, and Distributed Systems. He has been member of the organizing and scientific committee of several international symposiums such as FUSION, PAAMS, MIS4TEL, ISAMI, PACBB.

Javier Caridad Hernández He is finishing his degree in Physical Sciences. He has developed his professional career developing sensors and electronic devices for industry and IoT applications.

Daniel Hernández de la Iglesia (PhD candidate) He has finished his studies in Technical Engineering in Computer Systems at the University of Salamanca (2013), and he has a degree in Computer Engineering from the same University (2014). He completed his studies with the Master in Intelligent Systems of the University of Salamanca (2015). He is currently a member of the computer and automatic department as a PhD student.

Juan Francisco De Paz (PhD.) Received a PhD in Computer Science from the University of Salamanca (Spain) in 2010. He is Assistant Professor at the University of Salamanca and researcher at the BISITE research group (). He obtained a Technical Engineering in Systems Computer Sciences degree in 2003, an Engineering in Computer Sciences degree in 2005 at the University of Salamanca and Statistic degree in 2007 in the same University. He has been co-author of published papers in several journals, workshops and symposiums.