Multi-floor pedestrian navigation service based on a hybrid indoor positioning system

Wiem Fekih Hassen*[*†], Faiza Najjar*, Mohamed Ali Nabi*, Sofien Kriaa*, Harald Kosch[‡], Lionel Brunie[†]
*LISI Lab, INSAT, Carthage University, 1080 Tunis, Tunisia *ENSI, Campus University of Manouba, 2010, Tunisia
[†]LIRIS Lab, INSA Lyon, 69100 Lyon, France [‡]University of Passau, Passau, Germany
Email: {wiem,lionel}.{fekih-hassen,brunie}@insa-lyon.fr, {faiza,mohamedali,sofien}.{najjar,nabi,kriaa}@ensi-uma.tn
Email: Harald.Kosch@uni-passau.de

***Abstract*—With the advent of wearable and smartphone technologies, the need for pedestrian navigation services has increased over the last decade. Outdoor, the GPS remains currently the navigation standard. However, indoor navigation is in a development stage and suffers from many limitations, mainly related to the used positioning technologies and techniques and to the problem of finding the path from the origin to the destination. Previous studies have been limited to either positioning services or navigation services within a single floor building. The focus of this work is to develop a multi-floor pedestrian navigation service. This prototype is composed of two main modules. First, a positioning module, is conceived to compute the mobile unit (MU) position using the pedestrian dead reckoning (PDR) and the received signal strength indicator (RSSI) fingerprinting positioning techniques. Second, a navigation module, is designed to identify the rooms' numbers inside a building and to generate the graph for the whole building. The performance of our prototype is evaluated in the building IT Zentrum/International House on the campus of the University of Passau in Germany.**

***Index Terms*—Pedestrian navigation service, PDR, RSSI fingerprinting, rooms' numbers, automatic graph, multi-floor path finding.**

I. Introduction

In the past decade, considerable attention has been paid to indoor navigation service in guiding and tracking people in unfamiliar buildings. In the mobile context, this service is composed of three main phases: positioning, route search and route guidance and tracking. *Positioning* is the core component of navigation, it estimates the current position and optionally, the orientation and the speed of a mobile unit (e.g., smartphone, tablet, etc.) using positioning technologies (e.g., GPS, WiFi, etc.) and techniques (e.g., proximity, trilateration, etc.) [1]–[3]. *Route search* consists of finding the shortest path from the starting position to a given destination based on an indoor/outdoor map and a routing algorithm. *Route guidance* of a MU involves the display of navigation instructions in the form of text (geometric or symbolic), images or voice along a given path, whereas the *tracking* is the comparison between the current position and the planned path.

Recent developed prototypes have mainly focused on one of the two first phases of the navigation service and specifically on the positioning phase [4]–[6]. In order to estimate the MU position within a single-floor building, these works have adopted different positioning technologies and techniques that were based on the WiFi trilateration, the WiFi fingerprinting or the PDR [4]–[6]. The PDR technique was mainly integrated throughout the navigation process, since displaying the path information from the origin to the destination. The principle of the PDR consists of picking up data from the MU sensors to estimate the MU position. As the MU sensors are already running in the background and used by pre-installed applications in the MU, the integration of the PDR technique in a positioning system, even for long-term, does not consume more battery energy.

Most pedestrian navigation prototypes developed so far were designed to meet the need of a single floor navigation service [4]–[7]. Whereas, people tend to loose orientation easier within multi-floor buildings due especially to vertical travel.

In this paper, we propose a multi-floor pedestrian navigation service that operates inside a specific building and displays the following features: *(i)* the MU position based on the WiFi fingerprinting and the PDR positioning techniques, *(ii)* the path to reach the destination and *(iii)* the rooms' numbers and the list of the required landmarks and points of interest (PoI).

The key contributions of this paper are summarized as follows:

1) We developed a hybrid positioning system that computes the MU position inside a building.
2) We proposed an automatic multi-floor indoor graph construction algorithm that generates the graph for the whole building.
3) We provided an automatic rooms identification algorithm that assigns each room within a specific building with a number based on the adopted numbering pattern.

This paper is divided into five sections. In section II, we give some background information. We start by presenting the positioning technologies and techniques then, we outline the most adopted numbering pattern in the current buildings and finally, we describe the graph creation methods. In section III, we devote some related works. In section IV, we discuss in detail the proposed model. In section V, we show the experimental results of our proposal. Finally, we conclude in section VI.

II. Background

This section starts with a brief overview of the current positioning technologies and techniques, followed by a description of the most applied numbering patterns and ends by an introduction of the most common graph generation methods and path planning algorithms.

A. Positioning technologies and techniques

1) Positioning technologies: recently, more and more positioning technologies are available to find the MU's position everywhere. These technologies can be divided into two main categories [1]–[3]. First, the RF based technologies, this group contains mainly the satellite-based positioning technologies, known as GNSS, the WLAN which incorporates the WiFi, the Bluetooth low energy (BLE) and the radio frequency identification (RFID). Second, the MU's sensors based technologies, in this group, data are extracted from the MU sensors to estimate the MU position.

2) Positioning techniques: various positioning techniques were used to estimate the MU position in indoor environments [1]–[3] and can be grouped into radio frequency (RF) (i.e., proximity, RSSI fingerprinting, trilateration) and MU sensors (i.e., PDR) based techniques. In the following, we briefly introduce the two most frequently applied positioning techniques in the literature: the RSSI fingerprinting and the PDR [1]–[7].

- *RSSI fingerprinting*, this technique is commonly used in the WiFi technology and is composed of two main steps:
  - *Training/offline step*: consists of storing the different RSSI values, collected in some special positions (i.e. reference point (RP)) into a database, called radiomap.
  - *Matching/online step*: in this step the MU position is estimated, in real time. Many supervised learning algorithms were employed to execute this step, including the k-nearest neighbors (kNN), the weighted kNN (WkNN), the support vector machine (SVM) and the neural network algorithms [7]. The choice of the appropriate algorithm depends on the size of the radiomap.
- *PDR*, this technique extracts data from smartphone sensors (e.g., accelerometer, gyroscope, magnetometer, etc.) in order to estimate the current MU position (latitude and longitude) based on its previous one, taking into account the step length and the MU heading.

B. Numbering patterns

There are four foremost numbering patterns to identify the different rooms of a building namely, one side, N-shaped curve, clock wise and odd/even.

1) One side Numbering: it is the simplest numbering pattern inside buildings containing rooms and connectors such as corridors, elevators, stairs and escalators. Its principle consists of locating rooms only in one side of a corridor, and room numbers increase incrementally through the corridor from the entrance [8].

2) N-shaped curve numbering: in the N-shaped curve numbering pattern, rooms are located on both sides of the corridors. Room numbers increase incrementally from the entrance first, on the left side, then on the right one, in N shaped pattern [8].

3) Clock wise numbering: in the clock wise numbering pattern, rooms are on both sides of the corridor and the room numbers should flow in ascending order in clock wise movement from the main entrance. Numbering starts from the left side of the corridor and continue increasing in the opposite direction on the right side [8].

4) Odd/even numbering: in the odd/even numbering patterns, rooms are on the both sides of the corridor. Rooms' numbers should flow in ascending order in a zigzag pattern through the corridor from the main entrance that odd numbers are located on the left side and even numbers are located on the right side [8].

C. Graph creation

When computing the shortest route from the origin to the destination, the indoor environment is mainly represented by a navigation graph which is capable of applying the shortest routing algorithm. A navigation graph is created on the basis of floors maps. It can be defined as a graph to support navigational guidance and it consists of a set of nodes and edges (or lines). A node represents a decision point in a building such as doors and turning point in a corridor and it is usually given by its coordinates. Each edge links two different nodes and it provides information about possible path. The graph generation methods have been widely studied in the literature and the two most common ones are: the weighted indoor routing graph and the hierarchical graphs [9], [10].

1) Weighted Indoor Routing Graph (WIRG): WIRG is proposed by Goetz and Zipf in 2011 [9]. Its principle consists of creating graphs on the basis of the following steps:

- For each corridor a centerline is calculated and added to the graph. All the centerlines are connected with each other.
- Each door is presented by a node in the routing graph. Doors are projected orthogonally onto the corresponding corridor and nodes in the same room are connected pairwise with each other via edges.

For multi-floor building, the final graph is created in two stages. First, for each floor, a graph is created based on the previous steps. Second, vertical connections are added to the final graph according to the available stairs and/or elevators.

2) Hierarchical graphs: This approach consists of creating a base graph by extracting information from a floor plan [10]. It is based on the concept of polygons and links between them. A polygon represents region with physical barrier that can be a geometric object or a group of objects. It may also be used to hide certain un-needed details or to group objects having the some common features.

D. Path planning algorithms

Path planning or routing is the core process of the navigation service that consists of finding possible paths from a source to a destination based on user's needs and requirements, such as the shortest, the fastest or the cheapest path. The implementation of a path planning algorithm depends on the

location models (maps), provided from different location-sensing technologies and, more precisely, the graph of the test area.
Several solutions are proposed to find the optimal path and can be subdivided into two main groups: static algorithms and dynamic algorithms.

- *Static routing algorithms* such as Dijkstra, A* and Bellman-Ford set up the optimal path between a source and a destination without any controlling mechanism if the user deviates from the planned path [11].
- *Dynamic routing algorithms* such as D* (Dynamic A*) and D* Lite construct an optimal path from the source to the destination, using a controlling mechanism through the navigation process; if the traveled path differ from the recommended one, then re-construct a new path from the current position of the user to the final destination [11].

E. Points of interests (PoI) and Landmarks

Pedestrian navigation services are developed to provide different environmental features to help the user to easily reach the desired destination. The most common features are pathway, PoIs and landmarks [12].

1) Points of interests: can be defined as special services inside the building, which depends on both the building and the person context. As instance of PoIs inside an academic institutions (School, universities, etc), we can cite classrooms, laboratories, professor offices and kitchens. These rooms are especially helpful and interesting to users unfamiliar with the environment.

2) Landmarks: in terms of pedestrian navigation service, landmarks are natural, built or culturally prominent and unique environmental features that serve as references points through a navigation path. As examples of landmarks inside a building we can mention elevator, coffee, beverage and food distributors.

III. Related work

Recently, pedestrian navigation services have been discussed extensively in the literature, but few are the prototypes that have been focused on both the positioning and the navigation services at the same time.
In [13] an indoor navigation system, called Travi-Navi, which integrated the WiFi fingerprinting and the PDR techniques was presented. The developed system was based on the augmented reality technique that consists of the real time image captured by the MU camera. Travi-Navi has been limited to guiding the user inside a single floor building without any interest on displaying the MU position. Nevertheless, the use of all of these MU sensors throughout the navigation process drain rapidly the MU battery.
In [14] a continuous indoor/outdoor navigation service, named ioNavi, was developed. It was made up of a crowdsourcing approach that consists of collecting the walking traces in order to guide the user from the subway station to her destination. The walking traces were calculated using the PDR technique. The ioNavi user was asked to specify her destination before submitting her navigation request, whereas the initial position is computed based on the WiFi fingerprinting and on the cell towers proximity positioning techniques. Just after getting off the train, the user was guided to the stairs or the elevator indicated in the planned path.
In [15] the architecture of a multi-floor indoor pedestrian navigation system, was designed. It consisted of two main modules namely, positioning and route search. The MU position throughout the navigation process was computed using the magnetic fingerprinting technique and the path was calculated by applying the ant colony optimization (ACO) algorithm. The performance of each module was evaluated separately and a complete navigation prototype has not yet been developed: the adopted positioning technique was tested experimentally inside a university building, whereas, the ACO algorithm was theoretically assessed.

IV. Proposed Model

The aim of this section is to propose a multi-floor pedestrian navigation service. We start by describing a usage example and exposing the architecture of the proposed service. Then, we will detail our hybrid indoor positioning system. After that, we will outline the multi-floor indoor graph construction algorithm and we will finish by presenting the extended Odd/Even identification algorithm.

A. A usage example

In this part, we describe a usage scenario of our navigation service tested inside a three-floor building on the campus of the University of Passau in Germany. Alice, a PhD student, completely unfamiliar with the building, has to attend, first, a conference arranged on the ground floor and then, a meeting on the second floor of the same building. When Alice arrives at the main entrance door, she turns on the navigation application and she specifies her first destination and subsequently the path is displayed. After attending the conference, Alice wants to drink a coffee. So, she has only to launch a query and the list of coffee distributors located in the ground floor, will be displayed on the screen of her smartphone. Thereafter, she has to attend the meeting. Hence, she turns on again the navigation application and she specifies her current position and her destination. If, the initial position is not well-defined therefore, either Alice can display the number of the different rooms of the present floor and she can then determine her position, or the application can automatically estimate the smartphone position if Alice is located in a specific area on the second floor of this building.

B. General architecture

The general architecture of our application is highlighted in Fig. 1. It presents an indoor pedestrian navigation service, composed of two main modules: the positioning module to estimate the MU position and the navigation module to find the shortest path between the origin and the destination points. The purpose of the positioning module is to estimate the MU

position using different positioning techniques and technologies. In our system, we adopted the WiFi and the MU sensors positioning technologies, as they are available everywhere and they do not require any additional infrastructure.
The aim of the navigation module, specifically designed for indoor navigation, is to create the indoor multi-floor map(s) of the considered building, to automatically generate the graph of each floor in order to find the path between an origin and a destination and finally, to display the rooms' numbers and the list of PoIs and landmarks required by the user.

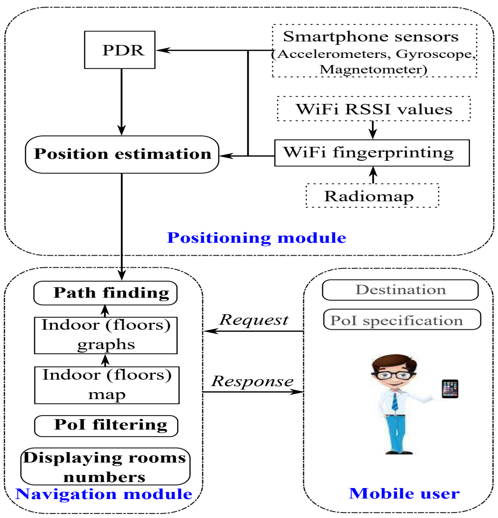


Fig. 1. General system architecture

The implementation of our navigation service is carried out with the following assumptions:

- An entrance/exit node emphasizes on stairs or elevator.
- A building is a structure that has at least two floors and two entrances/exits located at different sides of the building.
- The architecture of floors are nearby the same in a given context (i.e. academic buildings such as universities and hospitals)
- The width of a corridor is greater than or equal to 2 m.

C. Hybrid indoor positioning system

In this section, we present our proposed indoor positioning service. We start by describing the problem to overcome, then we outline the preliminary study and we finish by exposing the principle.

1) Problem statement: the proposed hybrid indoor positioning system is based on both the PDR and the WiFi fingerprinting positioning techniques. The WiFi fingerprinting technique estimates the MU position by comparing the online recorded RSSI values to the previously stored RSSI values in the radiomap. The major drawbacks of this technique is the dependency of its accuracy on the recorded RSSI values and the increase of the MU battery consumption. On the other hand, the PDR technique estimates the relative MU's position based on the previous position and a reference point as a starting point. The major limit of this technique is the accumulation of the large error rates after some short time, specifically, if no suitable update with an absolute position is available.

2) Preliminary Study: Fig. 2 depicts the preliminary experiment results of the implementation the PDR positioning techniques. The details of the experimental part are described in section V. We note that the large error of the PDR is caused when the MU is rotated by a specific angle. In order to determine this value, we opted for the *Sensor Kinetics* application to investigate the MU rotation sensor. We followed the same path as in Fig. 2 and we realized that from an angle of rotation equal to **45**° the PDR positioning technique begins to accumulate errors. We carried out some experiments to assess the impact of the two adopted positioning techniques on the MU battery consumption. We found that the Wifi fingerprinting consumes 6% of the MU battery, while the PDR consumes only 3%.

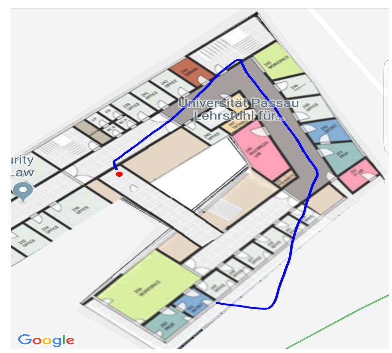


Fig. 2. Preliminary study of the application of the PDR positioning technique

3) Principle: the principle of the proposed service is depicted in Fig. 3 and consists of the following steps: at the initiation of the application, the MU position is estimated using the WiFi fingerprinting and once a step is detected, the PDR is initiated. The correction process is triggered after each **5 seconds** and if the MU is rotated by an angle equal to or greater than **45**°. The selection of the value **5 seconds** is based on the experimental results revealed in [16], regarding an indoor navigation service.

D. Our proposed multi-floor indoor graph construction

We propose a new floor map generation inspired from the WIRG graphs for the following twofold reasons:

- All map objects should be taken into account to create a versatile graph to represent floor map and then to use it in the navigation process.
- Only the geometry of the object being designed is required. Additional information related to the object structure is not needed.

In order to create the multi-floor indoor graph construction, we adopt the following assumptions: a floor contains n corridors and each corridor is represented by its four extremities points namely, top left, top right, bottom left and bottom right.

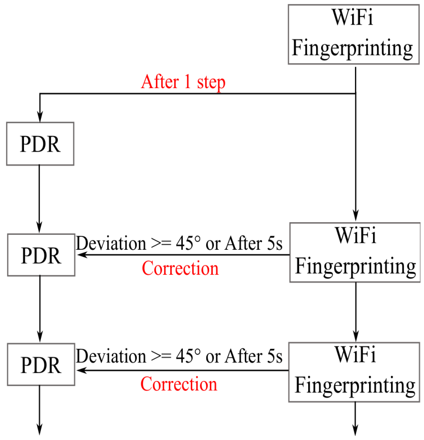


Fig. 3. Hybrid indoor positioning system

The different steps of the multi-floor indoor graph construction is described in Fig. 12. The graph for the whole building is generated based on two algorithms: Algorithm 1 and Algorithm 2. Algorithm 1 provides a graph for each floor (or level) separately and Algorithm 2 adds links between the different floors.

The principle of Algorithm 1 consists of adding a centerline in the middle of each corridor that will be subsequently divided into two main parts (left and right). This process is repeated, recursively, while, the width of each part still more than 2 m. Finally, edges are added between real nodes and virtual nodes (virtual nodes are the orthogonal projection of real nodes onto the centerlines).

Algorithm 1 also investigates the connection between two adjacent corridors. This connection is made either on adding a virtual node on the intersection point or on designing an edge that links the last virtual node of one corridor to the start virtual node of the second one.

The interest of creating a graph with multiple lines in a wide corridor mainly consists of making pedestrian navigation service easier and faster in the processing. If, for example, a user is moving inside a ground floor corridor (for example the width is equal to 8 m). At some point, she wants to leave the building and since the exit door is situated at the end of the corridor on the right side then, the shortest path to follow is to continue walking on the right side of the corridor not at the middle, as a classical graph suggests.

The principle of Algorithm 2 is based on selecting the entrance/exit nodes of each floors and then, adding edges between nodes that belong to the same stair or elevator.

E. Extended Odd/Even identification algorithm

Usually, when the user launches her navigation request, she often specifies the room number of her destination (e.g. office, conference room, seminar room, etc.). However, room numbers identification in each building is mostly done manually. Therefore, it is interesting and useful for indoor navigation to integrate this step to use it in different computations of the navigation process.

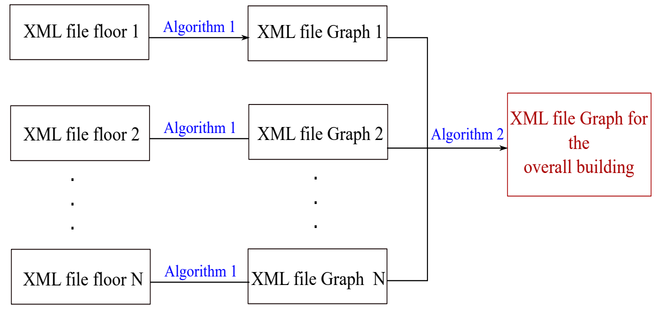


Fig. 4. Steps of multi-floor indoor graph construction

Algorithm 1 One floor indoor graph construction

Input: $XMLfile$ generated from JOSM and specifies first, for each real node (N) the *id*, the *longitude*, the *latitude*, the *position*, the *type* and the *corridor number (i)* and second, the *corridor dimensions* (the width l_i and length L_i).
Output: $XMLfile$ that adds virtual nodes (VN) and edges (E) **and** displays the final graph *G(N_i,E_i)* on the map.
Parse N data points from the XML file
for each corridor i in the XML file **do**
  / k is a real integer, it takes as initial value the width of the corridor and then the width of parts */*
  k $\longleftarrow l_i$
  while k $\geq$ 2 m **do**
    AddCenterLine(k,L_i)
    DividePart(k,L_i,"right")
    DividePart(k,L_i,"left")
    VN_i = AddVirtualNodes(N_i,k)
    E_i = AddEdges(N_i, VN_i)
    k $\longleftarrow k/2$
  end while
end for
for each adjacent corridor j to corridor i **do**
  if IntersectionLines(corridor$_i$,corridor$_j$) == true **then**
    / IntersectionNode is the intersection point between corridor$_i$ and corridor$_j$ */*
    VN_i = AddVirtualNodes(IntersectionNode, l_i)
  else
    E_i = AddEdges(VN_i,VN_{i+1})
  end if
end for

In [8] the room numbering system defined within sixty two buildings was investigated. The four numbering patterns, introduced in section II-B, were used and **45.16%** of the considered buildings have adopted the odd/even model, making it one of the most used. Moreover, in the current building, it is often that both the one side and the odd/even rooms numbering models are applied at the same floor.

In this paragraph, we detail the principle of our proposed algorithm called, extended Odd/Even. It consists of applying either

Algorithm 2 Vertical indoor graph construction

Input: *N XML files* generated from JOSM and contain the graph G_i*(N,E)* for each floor *i*, where *N* is the set of virtual and real nodes and *E* is the set of edges.
Output: *XML file* that contains the graph of the overall building (including the vertical links).
for each floor i in the building **do**
  Parse EntranceExit nodes from the XML file of the floor i
  for each EntranceExit node j **do**
    if $i >= 2$ **then**
      newEdge(EntranceExitMap(EntranceExitNumber), IdNode)
      / EntranceExitMap is a map data structure stores data in key/value with unique keys (the key here is EntranceExitNumber) */*
      / All doors in different floor of the same stairs/elevator have the same EntranceExitNumber */*
    end if
    EntranceExitMap(EntranceExitNumber) = IdNode
    / Update the value of the key EntranceExitNumber on the map data structure */*
  end for
end for

the one side or the odd/even numbering patterns, considering the front of the room, its type, its position and its dimension. The front gives information about the room situated on the opposite side of the same corridor. It takes as value the type of this room or the value "null".
The type of the room can be:

- *Small room* such as office, server room and kitchen.
- *Big room* such as laboratory, conference room and amphitheater.
- *Parent and child rooms*: are joined by a shared door and they have only one corridor entry. The two rooms have the same number and an alpha suffix is added to differentiate between the parent room and the child room.

The extended odd/even algorithm takes as inputs the XML file generated from Java Open Street Map (JOSM) of the floor, the *id*, the *longitude* and the *latitude* of each room. It starts by identifying the front and the type of the room. Then, it applies the odd/even or the one side numbering patterns. Finally, it adds the letter *"a"* or *"b"* to the room number if it is a parent room or a child room, respectively.

V. Evaluation

In this section, we experimentally evaluate the performance of our navigation service. We start by specifying the indoor work environment, then, we highlight the steps implementation and the experimental results of the positioning and the navigation modules.

A. Indoor Work Environment

To evaluate the performance of our all proposals in a real indoor environment, we consider the building IT Zentrum/International House on the campus of the University of Passau in Germany. It consists of a three floors building, and it was built in the shape of the mathematical symbol "$\surd$", as illustrated in Fig 5. To evaluate our proposal, we adopted for the smartphone Samsung Galaxy S6 Edge with Android operating system 5.0 Lollipop.

Fig. 5. IT-Zentrum/International House

B. Positioning: Experimental results

We adopted for the RSSI fingerprinting positioning technique, only in a specific area of the second floor of the ITZ building as we want to concentrate our evaluation on a small geographical area, pinpointed in Fig 6 (i.e., area 1, area 2 and area 3). In the following, we detail the steps of implementing the two phases of this technique.

Fig. 6. Areas of application of the WiFi RSSI fingerprinting positioning technique

1) Radiomap construction: the first step in the radiomap construction is the specification of the number and the positions of the references points (RPs), that depend on the dimension of the considered test area. In fact, in order to increase the accuracy of the positioning service, it is recommended to maximize the number of RPs. However, with a high number of RPs, the database becomes bigger and therefore the final response time of the positioning service is increased.

The main goal of our work is to develop an Android application that guides the user to her final destination. We assume that if the user is standing in front of the destination door, then she arrives at her goal. On the basis of this proposal, we apply the RSSI fingerprinting positioning technique only on corridors and not inside rooms. Corridors in the second floor of the ITZ building are not too large and their width is equal to *3 m*. Two types of RPs are considered: RPs near to walls and RPs located in the middle of the corridor. In the first type, RPs are *0.5 m* apart from the left and the right walls of the corridor and they are *2 m* apart from each other, as illustrated in Fig. 7. In the second type, a RP represents the intersection point of four distinct RPs, as shown in Fig 7.

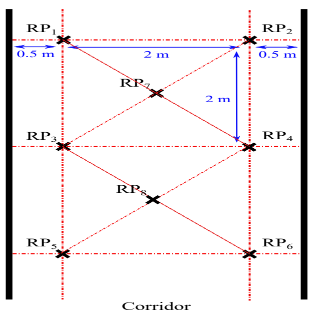


Fig. 7. RPs positions

In fact, the training phase needs recording the RSSI for each RP in the building, which is time consuming, expensive and labor-intensive. To overcome such drawback, we have implemented an Android application that ensures the automatic construction of the radiomap. Its principal consists of the following steps:

1) Enter the latitude and longitude of the already fixed RPs into the radiomap.
2) For each RP, record the 10 strongest WiFi RSSI values. The stored value is the mean of four values measured in four directions (north, east, south and west).
3) Use the stored information to create a SQL file that contains the SQLite queries to be used by the navigation application.

In our test area, we have considered *124 RPs* and we have spent two days to construct the radiomap.

2) Map matching algorithm: the second phase of the RSSI fingerprinting technique is to estimate the MU position, in real time, using a map matching algorithm. As we applied the RSSI fingerprinting positioning technique in a small area of the building, thereby our radimap is not too large. Then, we opted for the kNN algorithm. For the parameter k, we tested the WiFi fingerprinting positioning technique for two different values of k: k=3 and k=5. As highlighted in table I, kNN algorithm is more accurate for the special case *k=5*. Hence, we retain the value k=5 throughout the use of the WiFi fingerprinting technique.

3) Results of the WiFi RSSI fingerprinting, the PDR and the proposed positioning techniques: the WiFi RSSI fingerprinting positioning technique is applied in three specific areas, as depicted in Fig. 6. In the area 1, the MU can capture nearly fifty seven WiFi APs and the WiFi signals are medium to weak (the RSSI values vary between *-62 dBm* and *-88 dBm*). Inside the area 2, the WiFi signals are significantly stronger (we captures nearby twenty seven WiFi APs and the RSSI values vary between *-38 dBm* and *-66 dBm*. Whereas, inside the area 3, the MU captures nearby twenty WiFi APs and the RSSI values vary between *-47 dBm* and *-73 dBm*.

Fig. 8 shows the real trajectory adopted to test our hybrid positioning system and to compare it with the traditional PDR and the Wifi fingerprinting techniques. As revealed in Fig. 12, our proposal improves significantly the result obtained by the PDR and achieves a result very close to the WiFi fingerprinting. Throughout these trajectories, we recorded the values of 40 different estimated positions and we calculated the mean position error. The final result is summarized in Table I. From this we deduce that our proposed positioning system achieved an accuracy about ***2 m*** without causing noteworthy battery power depletion.

Fig. 8. Real trajectory

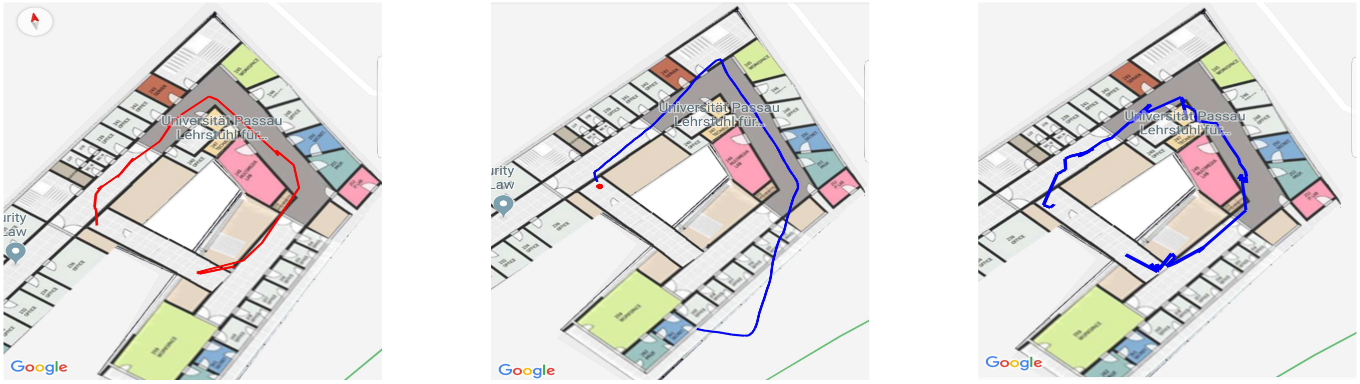
centering

TABLE I
MEAN POSITION ERROR AND BATTERY CONSUMPTION OF EACH POSITIONING TECHNIQUE

	WiFi Fingerprinting		**PDR**	**Proposed Solution (WiFi +PDR)**
	K=3	*K=5*		
Mean Position Error (in m)	1.97	1.32	4.31	1.96
Battery consumption (in %)	4	6	2	3

C. Navigation: Experimental results

1) Numbering pattern identification: the developer of the application has to specify, first, the numbering pattern adopted in the corresponding building and based on the appropriate algorithm, the number of each room is added in the XML file of the map building and displayed to the end user. The numbering pattern used in the ITZ building to identify the different rooms is the extended odd/even algorithm. Fig. 10 outlines an example of the application of this algorithm on



(a) WiFi fingerprinting (b) PDR (c) Hybrid positioning system

Fig. 9. Trajectories of the different considered positioning techniques

the first floor. The rooms' numbers are displayed on the map only upon request from the user.

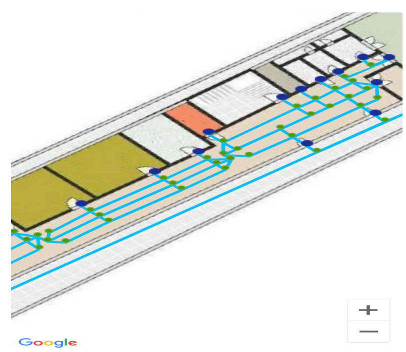
Fig. 10. Extended odd/even numbering pattern illustration

2) Indoor graph construction: once all the rooms in the building are identified, the next step consists of creating the graph of each floor. In order to achieve the goal, we applied the one floor indoor graph construction algorithm. Fig. 11 gives two examples of graphs: the first graph is generated through a corridor with a width equal to 3 m (see Fig. 11(a)) on the second floor and the second graph is created through a large corridor (see Fig. 11(b)) on the ground floor, where the blue circles represent the real nodes and the green circles indicates the virtual nodes.

3) Multi-floor path finding: Fig. 12 represents the result of applying the Dijkstra algorithm between two different points. In fact, the user is located on the ground floor (near to room number *"005"*) and she wants to reach a meeting room (room number *"222"*) on the second floor. First, the user launches her request by specifying the origin and the destination points, as indicated in Fig. 12(a). Then, the application displays the result to the user between the origin point and the exit door of the ground floor (see Fig. 12(b)). Finally, the path between the exit door of the second floor and the destination point is displayed, as can be seen in Fig. 12(c).

4) Landmarks and PoIs displaying: the display of the lists of landmarks and PoIs are treated in the same manner. Fig. 13 gives an example of PoI, it displays the location of the laboratory rooms.

(a) Graph generated through corridors with a width equal to 3 m

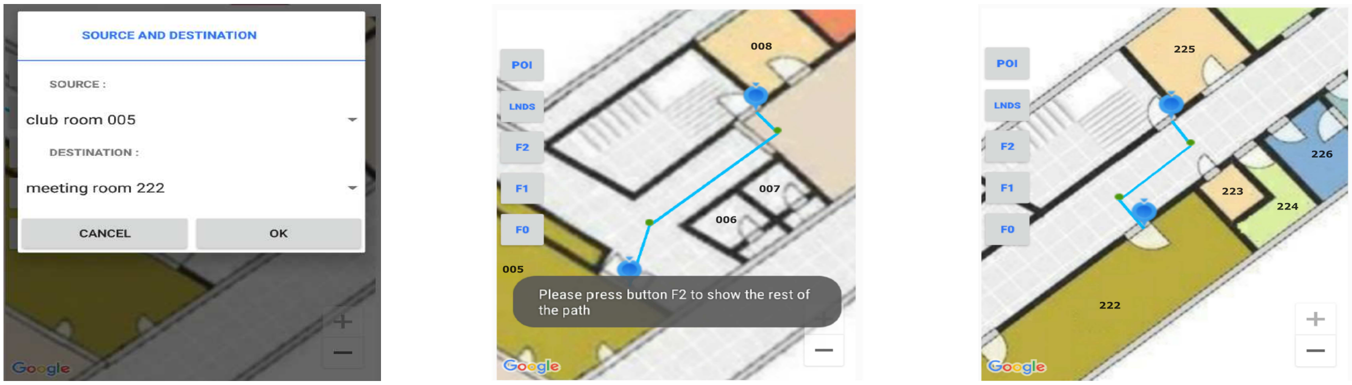


(b) Graph generated through a large corridor

Fig. 11. Graph generation

VI. Conclusion

In this paper, we have outlined a pedestrian navigation service inside a multi-floor building. The contribution of this work is threefold. First, we proposed a hybrid positioning system that estimates the MU position on the basis of the PDR



(a) Launch the navigation request

(b) Display the path between the origin point and the exit door (ground floor)

(c) Display the path between the entry door and the destination door (second floor)

Fig. 12. Path finding between an origin and a destination points located on different floors (The buttons F_0, F_1 and F_2 are added to display the maps of the ground floor, the first floor and the second floor, respectively and the two buttons PoI and LNDS are designed to display the list of PoIs and landmarks.)

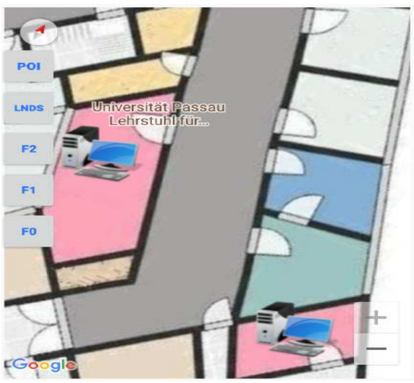


Fig. 13. Example of PoIs displaying

and the WiFi fingerprinting techniques. Second, we aimed at the automatic generation of an indoor graph in order to find the shortest path between an origin and a destination points. Third, we proposed an alternative of the classical odd/even numbering pattern to meet the requirements of recent buildings. This algorithm provides the identification of all rooms based on their location; if they are located on one side or on both sides of the corridor. We evaluated our prototype inside the International House building on the campus of the University of Passau in Germany. In our future work, we intended to concentrate to enhance our proposed hybrid positioning system to estimate the MU position anywhere in the building.

References

[1] Wiem Fekih Hassen, Faiza Najjar, Lionel Brunie, Harald Kosch, and Yahya Slimani. Smart PDR integration for ubiquitous pedestrian navigation service. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*, pages 1558–1563. IEEE, 2017.

[2] Wiem Fekih Hassen and Faiza Najjar. A positioning Handoff Decision Algorithm for Ubiquitous Pedestrian Navigation Systems. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pages 487–494. IEEE, 2016.

[3] Wiem Fekih Hassen and Faiza Najjar. A positioning technology switch algorithm for ubiquitous pedestrian navigation systems. In *Embedded and Ubiquitous Computing (EUC), 2015 IEEE 13th International Conference on*, pages 124–131. IEEE, 2015.

[4] Xianghong Li, Dongyan Wei, Qifeng Lai, Ying Xu, and Hong Yuan. Smartphone-based integrated PDR/GPS/Bluetooth pedestrian location. *Advances in Space Research*, 59(3):877–887, 2017.

[5] Massimo Ficco, Francesco Palmieri, and Aniello Castiglione. Hybrid indoor and outdoor location services for new generation mobile terminals. *Personal and ubiquitous computing*, 18(2):271–285, 2014.

[6] Jiantong Cheng, Ling Yang, Yong Li, and Weihua Zhang. Seamless outdoor/indoor navigation with WIFI/GPS aided low cost Inertial Navigation System. *Physical Communication*, 13:31–43, 2014.

[7] Qian Lu, Xuewen Liao, Shulin Xu, and Wei Zhu. A hybrid indoor positioning algorithm based on WiFi fingerprinting and pedestrian dead reckoning. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on*, pages 1–6. IEEE, 2016.

[8] Ryo Nitami, Akimasa Suzuki, and Yoshitoshi Murata. Development of a pedestrian navigation system without additional infrastructures. In *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, pages 203–209. IEEE, 2014.

[9] Marcus Goetz and Alexander Zipf. Using crowdsourced geodata for agent-based indoor evacuation simulations. *ISPRS International Journal of Geo-Information*, 1(2):186–208, 2012.

[10] Edgar-Philipp Stoffel, Korbinian Schoder, and Hans Jürgen Ohlbach. Applying hierarchical graphs to pedestrian indoor navigation. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 54. ACM, 2008.

[11] Jung-Hao Peng, I-Hsum Li, Yi-Hsing Chien, Chen-Chien Hsu, and Wei-Yen Wang. Multi-robot path planning based on improved d* lite algorithm. In *Networking, Sensing and Control (ICNSC), 2015 IEEE 12th International Conference on*, pages 350–353. IEEE, 2015.

[12] Anahid Basiri, Pouria Amirian, Adam Winstanley, Stuart Marsh, Terry Moore, and Guillaume Gales. Seamless pedestrian positioning and navigation using landmarks. *The Journal of Navigation*, 69(1):24–40, 2016.

[13] Yuanqing Zheng, Guobin Shen, Liqun Li, Chunshui Zhao, Mo Li, and Feng Zhao. Travi-navi: Self-deployable indoor navigation system. *IEEE/ACM Transactions on Networking*, 25(5):2655–2669, 2017.

[14] Xiaoqiang Teng, Deke Guo, Yulan Guo, Xiaolei Zhou, Zeliu Ding, and Zhong Liu. IONavi: an indoor-outdoor navigation service via mobile crowdsensing. *ACM Transactions on Sensor Networks (TOSN)*, 13(2):12, 2017.

[15] Kaixu Liu, Gianmario Motta, Tianyi Ma, and Tao Guo. Multi-floor indoor navigation with geomagnetic field positioning and ant colony optimization algorithm. In *Service-Oriented System Engineering (SOSE), 2016 IEEE Symposium on*, pages 314–323. IEEE, 2016.

[16] Kazuya Miyazaki, Masahiro Mochizuki, Kazuya Murao, and Nobuhiko Nishio. Cross-assistive approach for pdr and wi-fi positioning. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 777–786. ACM, 2014.