Beyond Euclidean Distance for Error Measurement in Pedestrian Indoor Location
Germán Martín Mendoza-Silva, Joaquín Torres-Sospedra, Francesco Potortì, Member, IEEE, Adriano Moreira, Stefan Knauth, Rafael Berkvens, Member, IEEE, and Joaquín Huerta

Abstract—Indoor positioning systems (IPSs) suffer from a lack of standard evaluation procedures enabling credible comparisons: this is one of the main challenges hindering their widespread market adoption. Traditionally, accuracy evaluation is based on positioning errors defined as the Euclidean distance between the true positions and the estimated positions. While Euclidean is simple, it ignores obstacles and floor transitions. In this article, we describe procedures that a positioning error defined as the length of the pedestrian path that connects the estimated position to the true position. The procedures apply pathfinding on floor maps using visibility graphs (VGs) or navigational meshes (NMs) for vector maps and fast marching (FM) for raster maps. Multifloor and multibuilding paths use the information on vertical in-building communication ways and outdoor paths. The proposed measurement procedures are applied to position estimates provided by the IPSs that participated in the EvAAL-ETRI 2015 competition. Procedures are compared in terms of pedestrian path realism, indoor model complexity, path computation time, and error magnitudes. The VGs algorithm computes shortest distance paths; NMs produce very similar paths with significantly shorter computation time; and FM computes longer, more natural-looking paths at the expense of longer computation time and memory size. The 75th percentile of the measured error differs among the methods from 2.2 to 3.7 m across the evaluation sets.

Index Terms—Error measurement, indoor pathfinding, indoor positioning system (IPS) evaluation, Wi-Fi fingerprinting.

Manuscript received April 12, 2020; accepted August 11, 2020. Date of publication September 4, 2020; date of current version November 17, 2020. The work of Germán Martín Mendoza-Silva was supported by Universitat Jaume I under Grant PREDOC/2016/55. The work of Joaquín Torres-Sospedra was supported by the Ministerio de Ciencia, Innovación y Universidades (INSIGNIA) under Grant PTQ2018-009981. The Associate Editor coordinating the review process was Dr. Martti Kirkko-Jaakkola. (Corresponding author: Joaquín Torres-Sospedra.)

Germán Martín Mendoza-Silva and Joaquín Huerta are with the Institute of New Imaging Technologies, Universitat Jaume I, 12071 Castellón de la Plana, Spain (e-mail: gmendoza@uji.es; huerta@uji.es).

Joaquín Torres-Sospedra is with UBIK Geospatial Solutions S.L., 12006 Castellón de la Plana, Spain (e-mail: jtorres@ubi.es).

Francesco Potortì is with the National Research Council—Information Science and Technologies Institute (CNR-ISTI), 56124 Pisa, Italy (e-mail: potorti@isti.cnr.it).

Adriano Moreira is with the Algoritmi Research Centre, University of Minho, 4704-553 Braga, Portugal (e-mail: adriano.moreira@algortimi.uminho.pt).

Stefan Knauth is with the Faculty for Computer Sciences, Mathematics and Geomatics, HFT Stuttgart—University of Applied Sciences, 70174 Stuttgart, Germany (e-mail: stefan.knauth@hft-stuttgart.de).

Rafael Berkvens is with the IDLab, Faculty of Applied Engineering, University of Antwerp—imec, 2000 Antwerp, Belgium (e-mail: rafael.berkvens@uantwerpen.be).

Digital Object Identifier 10.1109/TIM.2020.3021514

I. INTRODUCTION

Performance of indoor location-based services is strictly linked to the accuracy of the underlying indoor positioning systems (IPSs) [1]. IPSs exhibit errors in the range from a few centimeters for technologies, such as UWB [2] or ultrasound [3], to the more commonly used pedestrian IPSs that exhibit errors of several meters [4]. The latter is typically based on Wi-Fi, BLE, and magnetic field signatures and often combined with pedestrian dead reckoning. Generally speaking, the environment influences the behavior and the accuracy of an IPS, as IPSs commonly rely on signal measurements that are heavily affected by the indoor environment characteristics [5]. In the following, we will implicitly make reference to a person or a robot that navigates across an indoor environment.

Several IPS evaluation criteria are possible and significant, for example, cost, computational demand, and privacy [1]. However, the prime criterion is accuracy, which is some form of statistics based on positioning errors [1], [6]. The positioning error for an IPS, used in a single floor, is commonly measured as the Euclidean distance between the ground-truth position and the estimated position [6]–[9]. The Euclidean distance is easy and fast to compute and arguably the most significant error definition when Line-of-Sight (LoS) exists between the true and estimated positions. This is the usual case for small errors (within centimeters) or when the target scenarios are free from relevant obstacles. However, errors observed for the most-often-used IPSs are within a few meters, so LoS being impaired by walls or ceilings is not an uncommon occurrence.

Our preliminary work [10] proposed to define the positioning error as the length of the path that a pedestrian could follow between an IPS-estimated position and the true position. The 2-D paths were determined using visibility graphs (VGs) from floor plans in vector format. That preliminary work highlighted divergences between the proposed error measurement and the Euclidean distance measurement that affect the perceived accuracy of an IPS. That perceived accuracy is important for tuning and comparisons among IPSs. However, the number of alternative routes can become large due to the many degrees of freedom [11], especially if we consider a complex multifloor scenario with multiple endpoints. As the number of elevators, stairs, and building entrances increases, finding the optimal route becomes more complex [12]. This article technically extends our preliminary work by: 1) considering
two new pathfinding methods—navigational meshes (NMs) and fast marching (FM); 2) performing an evaluation on a multibuilding multistory scenario with outdoor navigation; and 3) using position estimates obtained by participants in the IPIN 2015 competition. The implementation of pathfinding methods, measurement procedures, and analyses presented in this article are available\(^1\) with an Apache-2.0 license. In summary, this article contributes to advance the state of art in IPS evaluation, with the following specific contributions:

1) description of five error measurement procedures for IPSs, using either vector or raster floor map information to compute the walking distance between two points;

2) comparison of the proposed measurement procedures and the Euclidean distance approach (with floor and building misidentification penalties) in terms of pedestrian path realism, indoor model complexity, path computation time, and error magnitude.

II. POSITIONING ERROR IN INDOOR POSITIONING SYSTEMS

Unlike the global navigation satellite systems (GNSSs) used for positioning in most open outdoor environments, current IPSs must be tuned for each targeted indoor scenario [1]. One reason is that many IPSs rely on radio signals, whose propagation is strongly affected by the specifics of each different building [5]. IPSs relying on inertial navigation are strongly affected by the building layout. No solution currently dominates the IPSs market because those that deliver high accuracy also have known drawbacks. As far as high-accuracy systems are concerned, IPSs based on UWB and ultrasound require hardware deployments whose cost compromises their scalability, while vision-based IPSs force the users to keep their device in a fixed position in order to get a view of the environment [1]. IPSs based on Wi-Fi, BLE, and magnetic field signatures, often complemented with inertial sensors’ readings, are the most frequently used in commercial solutions and the scientific literature for pedestrian navigation [1], [5]. These IPSs, however, have typical error ranges within a few meters. Errors of a few meters in open outdoor fields may not be significant. However, obstacles in complex indoor environments are separated by distances similar in magnitude to those errors which, therefore, have a much higher impact on the quality of an indoor navigation system.

The problem is not new and has received some attention in the literature. Pulkkinen and Verwijnen [13] discusses standardized baselines and metrics. Adler et al. [14] makes a survey of experimental evaluation criteria adopted in the IPIN papers, where different evaluation metrics were considered. Liu and Schneider [15] proposed a novel way to compute the similarity between moving objects, based on geographic and semantic components of the movements. de la Osa et al. [16] deals with the lack of a predominant solution for defining the ground truth when comparing indoor position estimates. Anagnostopoulos et al. [17] defined a novel dynamic evaluation procedure with predefined geometrical paths (see [18]) to capture real-life usage of the scenarios, as well as other

---

\(^1\)With DOI: 10.5281/zenodo.3741390
A. Visibility Graphs and Navigation Meshes

The VG and NM methods are based on vector maps and require knowing the information about the navigable space of the target environment, which we call free space. The free space is formed of all indoor positions where the positioning subject can be found without colliding with obstacles. The free space is represented in this work using a polygon with holes $E$, neither of which are necessarily convex. Section IV describes how to obtain free space representation from floor plans. When computing paths using VG or NM, we define the forbidden space as the space along the perimeter of obstacles that a path must avoid so that a real-world object of given size does not collide with the obstacles.

The free space is constructed as illustrated in Fig. 2. The limits of the environment (polygon boundaries) are grown inward, and the inner obstacles (polygon holes) are grown outwards by a quantity that depends on the physical size of the subject. Once the forbidden space is removed from the free space, thus obtaining a new polygon $F$, the subject can be considered having null size as far as path computation is regarded when using VG or NM. This work used the CAD/CAM technique known as polygon offsetting [27] to compute the forbidden space.

Given a set $S$ of disjoint polygonal obstacles, the VG of $S$ has a node for every vertex of $S$ and a visibility arc connecting any two nodes in LoS of each other; nodes are in LoS when they can be joined by a segment that does not collide with any edge of $S$. In this work, $S$ is composed of the boundary and the holes of $F$. Algorithms for the efficient construction of VGs already exist [23]. The shortest collision-free path between two points is composed of arcs of the VG, and once nodes and visibility arcs for the start and destination points are added [23]. Pathfinding with VG requires setting arc weights to the Euclidean distances between each pair of nodes and using a pathfinding algorithm, e.g., Dijkstra’s [28] or $A^*$ [29]. The paths found using VG usually have hard turns, which are not ideal for describing people’s movement. Also, VGs typically have a large number of arcs, which increases the computational cost of pathfinding. For example, the simple environment of Fig. 2 resulted in over 50 visibility arcs. For the experiments in this work, we used the graph-based MATLAB’s implementation of Dijkstra’s algorithm [30]. Also, we created a MATLAB implementation for VG construction that performs brute-force testing of arc eligibility. The determination of a VG requires several geometrical operations, being the most used operations the point-in-polygon and ray- or segment-to-environment intersection.

A NM [31], [32] subdivides $F$ into a set of convex polygons. The convex polygons represent spaces where movement between two points of the polygon’s boundary is possible without a collision. A constrained triangulation could be used as an NM. For example, a Delaunay triangulation [33] created from the vertices of $F$, for which the triangles outside the environment are removed, is an NM. However, an NM based solely on a triangulation normally has too many unnecessary divisions of the space. Several approaches exist for NM construction [34]. A MATLAB implementation for NM was created for the experiments in this work. It builds a constrained Delaunay triangulation and iteratively combines adjacent polygons to remove nonessential edges, thus obtaining new convex polygons [32]. The output is a graph representing the obtained polygons. A path is computed using Dijkstra’s [30], and it is later straightened using LoS testing to remove unnecessary intermediate points [31].

Fig. 3 presents an example of the resulting graphs for a building (TI) of the scenario used in later sections. The graph obtained by NM has far fewer arcs than that produced by VG. The obtained path is not guaranteed to be the shortest one because Dijkstra’s finds the shortest route between polygons, without any visibility information, and because the subsequent straightening step uses visibility information only to remove unnecessary intermediate points.

For complex environments, pathfinding for VG is considerably slower than for NM, even accounting for the straightening step (see results in Section V). For both VG and NM, finding the route involves creating a new graph by adding the endpoints to the static floor graph and adding new arcs for each endpoint. In VG, new arcs are added that connect each endpoint to those vertices of $F$ that are visible from the endpoint. In NM, new arcs are added that connect each endpoint to vertices of the polygon containing it. If an endpoint lies outside the valid areas, i.e., outside the boundaries or inside a hole of $F$, a correction step is required before pathfinding. The correction step moves an endpoint to the
cells and, thus, nontraversable obstacles. Fig. 4(a) presents \( \tau \) in large areas. The value of speed, which avoids to force paths toward the center zone, depends on image resolution. \( \tau \) was empirically determined and set to 30 to avoid speed reduction for cells representing positions farther than 2 m from obstacles.

In (1), \( \tau \) is a distance threshold for speed reduction. All cells farther than \( \tau \) from obstacles have the same (maximum) speed, which avoids to force paths toward the center zone in large areas. The value of \( \tau \) depends on image resolution. For example, for the images later mentioned in Section V, \( \tau \) was empirically determined and set to 30 to avoid speed reduction for cells representing positions farther than 2 m from obstacles. The 0.01 value was added to avoid zero-velocity cells and, thus, nontraversable obstacles. Fig. 4(a) presents an example speed map, where the darker the shade the lower the speed. Obstacles and areas outside of the target environment (nonvalid areas) are represented with the darkest shade. Fig. 4(b) depicts the output from applying FM over the speed map from Fig. 4(a). Lighter shades correspond to points reached sooner by the wavefront originating from the source point (blue dot). The paths shown in Fig. 4 show that FM does not require endpoint corrections. In fact, since nonvalid areas have minimum speed values, they are actively avoided. FM does not require offsetting either (although it can be optionally applied) because cells close to obstacles have low-speed values and are avoided.

The computation cost for building a path with FM depends on the size of the input image, which is related to the size of the environment and its complexity. Image resolution should be balanced so that the image retains all details relevant for pathfinding while not adding a useless computation burden.

The VG method was chosen because it produces the shortest paths. NM is a computationally lighter alternative to VG that somehow relaxes the requirement of finding the shortest path. FM relaxes that requirement even further. We provide the source code of our proposed methods and evaluation framework, to enable comparison with other methods.\(^2\) Alternative methods for finding the shortest path, such as predefined waypoints [24], the direct application of Dijkstra’s to raster maps [24], or quadtrees [38], may be considered in future extensions of our procedures.

IV. IPSs ERROR MEASUREMENT PROCEDURES

The solution to finding a route across several floors and buildings builds upon the 2-D pathfinding methods introduced in Section III. In the following, two variants are considered here named single model (SM) and endpoints expansions (EES). Both require static knowledge of the length of connections between any two pair of accesses to each floor. Here and in the following, a static information can be precomputed, as it is dependent on map information only, while dynamic information also depends on the endpoints. In the most common case, interfloor connections include vertical connections between adjacent floors, and interbuilding connections are outdoor ground-level routes between building accesses, but more complex configurations may exist; for example, vertical connections between nonadjacent floors, horizontal connections above ground level between buildings, and more: we neglect these cases in the subsequent description without loss of generality.

The SM variant connects the individual graphs produced for each floor by the VG or NM methods using arcs that represent the interfloor and interbuilding connections. This variant produces a single 3-D graph, which is the classic representation used for navigation services in indoor environments [39]. Being based on graphs, it cannot be used with FM pathfinding.

The EES variant is not based on graphs and can be used with any of the three pathfinding methods. A complete path between source and destination is composed of an ordered sequence

\[^2\]With DOI: 10.5281/zenodo.3741390
Fig. 5. Shortest distance path between two endpoints located in distinct buildings and floors, computed using the EE variant with NM pathfinding.

Algorithm 1 Path Determination for the SM Variant

**Input:** Environment Data Representation, Endpoint pairs

**Output:** Paths between endpoint pairs

1. Floor model creation
2. Single model integration
3. **for each pair of endpoints do**
   4. Endpoint correction
   5. Path computation
4. **end**

Algorithm 2 Path Determination for the EE Variant

**Input:** Environment Data Representation, Endpoint pairs

**Output:** Paths between endpoint pairs

1. Floor model creation
2. **for each pair of endpoints do**
   3. Endpoint expansion
   4. Endpoint correction
   5. Path computation
   6. Endpoint contraction
3. **end**

of path stretches that generally include interfloor stretches, interbuilding stretches, and intrafloor stretches computed via 2-D pathfinding. All the stretches are statically known apart from the extreme ones (first and last). Each extreme stretch is chosen from the set of intrafloor paths connecting an endpoint to each access of the floor where the endpoint lies, meaning that sets of intrafloor paths are dynamically computed.

Fig. 5 represents a complete path where each stretch is either:

1) **Vertical Blue Segment:** Statically known interfloor connection.
2) **Horizontal Blue Segment:** Statically known interbuilding connection.
3) **Green Segment:** A statically computed intrafloor stretch.
4) **Red Segment:** A dynamically computed intrafloor extreme stretch.

In order to build a path, the EE variant chooses a sequence of stretches using an always-forward strategy, that is, floor changes are done only if they represent a floorwise or buildingwise advancement toward the destination. EE finds a set of possible paths, which includes the shortest path. The number of possible paths, thus, identified is large and exponentially related to the number of interfloor and interbuilding connections. In the experimental implementation, only a subset of possible routes is computed, in order to shorten computation time.

Fig. 6 illustrates an example for steps 3–6 from Algorithm 2. Legend for the black circles is: **e:** estimate; **g:** ground truth; **c:** correction, **s_1,** **s_2,** **t_1,** and **t_2:** floor entrances; and **b_1** and **b_2:** building doors. **xy** represents the computed path between points **x** and **y**.

Algorithms 1 and 2 are pseudocode representations of SM and EE, respectively. Both algorithms take the map and a set of endpoint pairs as input, each pair being the real and estimated positions from the use-case presented in Section V. The steps are as follows.

1) **Floor Model Creation:** Loading of the environment data, i.e., the polygons or the image of each floor and the vertical and interbuilding connection information. The output is the 2-D graphs for the VG and NM methods and the speed transformation for the FM method.

2) **Single Model Integration:** Linking together the graphs created for each floor by the NM or VG methods. For each interfloor or interbuilding connection, new arcs are created: for VG, they are visibility arcs that irradiate from the access node; and for NM, they are segments that connect the access node to the vertices of the containing polygon.

3) **Endpoint Expansion:** For each pair of endpoints, the pair is removed. Always-forward sets are computed. For each always-forward set, a new endpoint pair is added for each of the two extreme stretches.

4) **Endpoint Correction:** Correction of endpoints lying outside the environment to the closest point inside the valid areas. FM does not require this step. Corrections are tracked and later included in the error magnitude.

5) **Path Computation:** Use one of the three 2-D pathfinding methods to compute a path stretch for each endpoint pair not associated with a static precomputed stretch length.

6) **Endpoints Contraction:** Obtain the sequences of stretches connecting the original endpoint pairs using the information computed or stored for each path stretch.

Fig. 6 illustrates an example for steps 3–6 from Algorithm 2. Step 3 produces basic information for creating the paths between endpoints. Step 4 adds a correction if required. Step 5 uses or computes path stretches to obtain actual paths between the endpoints. Finally, step 6 selects the minimum distance path among those obtained in step 5.
The outputs of Algorithms 1 and 2 are the paths that connect the input endpoint pairs. The application of Algorithm 1 or 2 with any of the three 2-D pathfinding methods is hereinafter called a measurement procedure. Five measurement procedures are proposed.

1) VG-SM: Uses Algorithm 1 with VG for 2-D pathfinding.
2) VG-EE: Uses Algorithm 2 with VG for 2-D pathfinding.
3) NM-SM: Uses Algorithm 1 with NM for 2-D pathfinding.
4) NM-EE: Uses Algorithm 2 with NM for 2-D pathfinding.
5) FM-EE: Uses Algorithm 2 with FM for 2-D pathfinding.

Procedures VG-SM and NM-SM produce shortest distance paths, while VG-EE, NM-EE, and FM-EE produce several alternative paths for each pair of endpoints lying on distinct floors or buildings. While measurements other than the shortest distance are generally useful for pedestrian navigation, only shortest-distance is considered in the following. Examples of significant paths different from the shortest are those that take into account the subject familiarity with the target environment, including those that contain the closest floor or building exit from an endpoint and those that leave the origin building through the exit that is closest to the destination building.

V. USE CASE: THE IPIN 2015 TRACK 3 COMPETITION

The error measurement procedures were applied to position estimates provided by IPSs, trained, and evaluated on data from the UJIIndoorLoc data set [40]. UJIIndoorLoc includes publicly available training and validation sets and a test set that is kept secret by the data set curators. The test set was used in the IPIN 2015 (EvAAL-ETRI) Track 3 competition [41] to evaluate IPSs based on Wi-Fi fingerprinting. This article applies the proposed error measurement procedure to the 20716 estimates provided by the four teams (5179 estimates each team) that participated in that competition (“RTLSUM” [42], “HFTS” [43], “ICSL” [44], and “MOSAIC” [45]).

A. Preparation of Environment Information

The UJIIndoorLoc’s data were collected in three university buildings. The environment data used in the measurement procedures included a 2-D depiction of each building’s floor that represented the building’s boundaries and the inner structural obstacles. The representation of a floor was either a set of polygons (vector format) or an image (raster format). The actual floors are connected through stairs or elevators, and buildings communicate through outdoor paths between their entrance doors. Environment information was obtained from accurate CAD and GIS data [46]. Fig. 7 shows the free space representation for the first floor of each of the three buildings where doors and similar obstacles are removed.

Polygon simplification was performed on the vector format to reduce the number of vertices and edges, which is particularly relevant for curves, such as pillars, which were reduced from 40 to 8 vertices each. Offset (as described in Section III) for VG and NM was set to 0.2 m, which accounts for (half) the average width of a person while also avoiding blockage of small entrances. Raster images had a cell (pixel) size of 0.1 m.

Besides, the line thickness was chosen to represent thin but continuous representations of obstacles’ edges.

The interfloor (vertical) and interbuilding (outdoor) communication ways were represented as static information in the form of triplets containing two endpoint tags and the distance between endpoints. That allows, for instance in FM-based approaches, to split pathfinding into independent steps by creating a layer for each floor and representing the stairs and elevators as static transit nodes between the independent layers [12]. The compiled data for vertical and outdoor ways considered any vertical way between two adjacent floors from the same building and one route for any pair of doors from two buildings. Automatic methods for topology extraction or feature identification from floor plans may reduce the effort of map data preparation [47], [48]. However, we manually compiled the data relative to vertical and outdoor ways in order to ensure the accuracy of distance computation. Table I presents the numbers of vertical ways along with other building information. Note that this procedure does not prevent the use of simple maps, even obtained from sketches, as long as they represent closed buildings with well-defined entrance points.

B. Experimental Results

Results presented here were obtained from experiments carried out on a PC with Intel Core i7-8700 CPU @ 3.2 GHz, 16 GB of RAM memory, running MATLAB R2019a on MX Linux 18.2 Continuum. The procedures were implemented in MATLAB, favoring correctness over efficiency. CPU times were measured only once, so they are only intended to give a grasp of relative computation times.

Tables II and III present the measured times for steps of Algorithms 1 and 2, respectively. In the tables, $\Sigma$ is the total step time, while $\mu$ and $\sigma$ are the mean and standard deviation of a set of times. The times for EE and endpoints correction are the mean per endpoint pair, while for paths computation are the mean per intrafloor computed path.
Although the time spent for floor model creation is not negligible, its importance is limited because it takes less than 3 min to complete for all procedures. Furthermore, the graphs or images resulting from this step can be stored and later reused by EE or SM for path computation. Times for EE and endpoints correction are small compared with the times employed in the paths computation, but they are not negligible. Their $\sigma$ values were close to zero and are not reported.

Floor and building misidentifications increased the number of paths to compute in the EE variants. Considering all teams, the total path processing time for the NM and VG methods was about eight times larger in EE variants than in SM variants. In contrast, the increase is 26 times if only the MOSAIC team is considered. EE is also demanding in terms of memory, mainly because the path information is kept for later analysis.

The times for the paths computation step are the most important measures presented in Tables II and III. Times for NM-based procedures are small and thus affordable for evaluation purposes, which is typically an offline procedure, even for large evaluation sets. Times for VG-based procedures are about seven times larger than those based on NM although those times should not be a concern in most scenarios for VG-based procedures. In contrast, FM-EE needs more than 2 s on average to run a single path computation. In our experiment, FM-EE took more than five days to compute the paths for the original evaluation pairs. Given its computational cost, the FM-based procedure is suited for small evaluation sets when the need for a realistically smoothed path is prevalent over reducing the computation burden.

The SM integration step is performed only once. NM-SM runs significantly faster than VG-SM in this step because NM produces floor graphs much less complex than VG, as discussed in Section III. This effect is seen for the endpoints correction and paths computation steps, which are smaller in Table II than in Table III.

Table IV compares the complexity of the map structures: graphs for VG and NM and raster for FM. VG and NM use the same number of nodes, yet the number of arcs of VG is almost nine times that of NM. FM computes the paths from images of the same size as the input maps, which is over six million pixels in our experiment.

| Procedure | Floor model creation $\Sigma (s)$ | Single model integration $\mu (ms)$ | Endpoints correction $\mu (ms)$ | Paths computation $\mu + \sigma (s)$ |
|-----------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| VG-SM     | 52                               | 32                               | 0.1                              | 0.21 ± 0.01                     |
| NM-SM     | 122                              | 1                                | 0.1                              | 0.02 ± 0.00                     |

Table II

**Empirical Computation Times (in Seconds) for Steps of Algorithm 1**

| Procedure | Floor model creation $\Sigma (s)$ | Endpoints expansion $\mu (ms)$ | Endpoints correction $\mu (ms)$ | Paths computation $\mu + \sigma (s)$ |
|-----------|----------------------------------|--------------------------------|----------------------------------|----------------------------------|
| VG-EE     | 52                               | 7.4                            | 2.4                              | 0.23 ± 0.05                     |
| NM-EE     | 122                              | 8.8                            | 2.6                              | 0.03 ± 0.02                     |
| FM        | 1                                | 7.4                            | NA                              | 2.51 ± 0.11                     |

Table III

**Empirical Computation Times for Steps of Algorithm 2**

While important, evaluation time matters only for time affordability, and thus, it is a secondary aspect of the proposed procedures. The main aspect is the error magnitudes as measured by the procedures. To set those magnitudes in the context of IPSs’ evaluation, Fig. 8 presents comparisons between the errors as measured by the proposed procedures and those measured by the EvAAL procedure. The EvAAL error is the sum of the 2-D Euclidean distance $r$ between the estimate $(x_e, y_e)$ and its ground truth $(x_g, y_g)$, plus penalties for floor difference and building inequality [8]. In [41], the penalties were 4 and 50 m, respectively.

$$e = \| (x_e, y_e) | (x_g, y_g) \| + 4 | f_e - f_g | + \begin{cases} 0, & b_e = b_g \\ 50, & b_e \neq b_g \end{cases} (2)$$

The usage of the EvAAL procedure with the floor and building penalties from (2) is hereinafter called EFP04 procedure. Results from the VG-SM and VG-EE procedures are exactly the same, as are results from NM-SM and NM-EE. Therefore, for simplicity, error measures are shown only for the EE variant, without further distinction. In Fig. 8, positions with both correct floor and building identification, which are indicated by gray and green dots, lie near the diagonal, meaning that their EFP04 error magnitudes are similar to those of the proposed procedures. Green dots are position estimates that were corrected because they lied outside the valid evaluation area; in these cases, error magnitude includes the correction length. Fig. 8(a) contains no green dots because FM does not use estimate correction, while Fig. 8(d) does not either because the MOSAIC team correctly placed all estimates inside the valid area.

Blue dots represent samples with floor (but not building) misidentification. The ICSL team had the lowest floor detection rate. HTFS had the best floor detection rate, and in fact, it has fewer blue dots. From a pedestrian perspective, the EFP04 procedure always underestimated the error when the position was estimated in the right building but on the wrong floor. The 4-m floor penalty of the EFP04 procedure was approximately equal to the floor height, which should be considered a lower bound to the length of the path walked by the subject to change floors.

Red dots, only shown in Fig. 8(d), represent building misidentifications, which are the cases with the largest errors, only present in results from the MOSAIC team. The 50-m building penalty of the EFP04 procedure was too large for most cases. Also, that penalty was far too small for a few other cases that required changing floors at the two buildings, as previously shown in Fig. 5. In general, compared with pathfinding-based alternatives, the EFP04 procedure mostly underestimated the magnitude of the error for the evaluated environment and estimation sets.
Fig. 8. Examples of error measurement differences. Green and gray dots represent cases for which floor and building were correctly identified, with or without applying corrections, respectively. Blue and red dots represent the cases of the floor and building misidentification, respectively. (a) RTLSUM. (b) ICSL. (c) HTFS. (d) MOSAIC.

Fig. 9. Measures of the proposed procedures for the MOSAIC team: (a) VG vs NM and (b) NM vs FM.

Fig. 9 [but also Fig. 8(d)] helps on path comparison among the four compared procedures, for the MOSAIC team. The charts help confirming that: 1) the EFP04 procedure produces error values lower than the other produces, apart from the case of building misidentification; 2) while the VG method computes the shortest path, NM provides a close approximation; and 3) while the FM method produces the longest path for most cases, in a few cases, it may produce a path shorter than that of NM.

Fig. 10 presents the CDFs of measures as provided by the four procedures already explored and by three additional ones: the E3D, which is addressed in some works [6], [49]; and the EvAAL procedure using the same 50-m building penalty as EFP04; and no floor penalty (EFP00) or 15-m floor penalty (EFP15). The EFP15 procedure has been used in on-site Tracks (1 and 2) of the IPIN competitions [8]. E3D, EFP00, and EFP04 provide percentile values that are notably similar for all teams but ICSL, the one with the smallest floor hit ratio. Thus, the comparisons between EFP04 and the proposed method also apply to E3D and EFP00.

The 75th percentile values of EFP15 and VG are close for all teams. The difference increases for percentiles above 75th: it stays below 3 m up to the 95th for all teams but MOSAIC. For the same reason, the HFTS team, which has the best floor detection rate, moves up to the second place with both VG and FM. The MOSAIC team ranks third when using the VG- and NM-based procedures. The MOSAIC team had a few building detection errors, each of them bringing a 50-m penalty whose weight is higher in EFP04 than in EFP15, where the effect of the floor penalty is more relevant. This explains why MOSAIC, with its good floor detection rate, ranks better with EFP15 (second place) than EPF04 (fourth place).

Table V reports the accuracy of each team as measured by EPS04, EPS15, and the three proposed procedures using the 75th percentile metric. The 75th percentile is part of the EvAAL evaluation framework and is used in IPIN competitions [21], [22]. Notice, in Table V, that EPF04, EPS15, VF, and FM produce different rankings. While RTLSUM keeps the first place in all cases, ICSL moves from the second to the fourth place with EPS15, VG, or NM, and to the third place when using FM, because EFP04 is more lenient of floor misdetection than EPS15, VG, or FM. For the same reason, the HFTS team, which has the best floor detection rate, moves up to the second place with both VG and FM. The MOSAIC team ranks third when using the VG- and NM-based procedures. The MOSAIC team had a few building detection errors, each of them bringing a 50-m penalty whose weight is higher in EFP04 than in EFP15, where the effect of the floor penalty is more relevant. This explains why MOSAIC, with its good floor detection rate, ranks better with EFP15 (second place) than EPF04 (fourth place).

FM builds path always longer than VG and apparently longer than NM too; the difference is most notable when misidentifying a building, which explains MOSAIC getting the lowest rank with FM.

The similarity of values in Table V for VG- and NM-based procedures requires an explanation. Fig. 11 shows how example paths produced by VG and NM are very similar. The reason is that NM uses a path straightening procedure using LoS testing: while this method does not produce the all percentiles values. In comparison with FM, VG is easier to compute and understand, the latter being a fundamental aspect when comparing systems. Thus, the VG-based procedure is recommended for all cases where high reliability is needed or high quantile metrics are involved.

| Team  | EFF04 (m) | EFF15 (m) | VG (m) | NM (m) | FM (m) | Floor (%) | Building (%) |
|-------|-----------|-----------|--------|--------|--------|-----------|-------------|
| RTLSUM| 8.44      | 8.48      | 8.96   | 8.96   | 10.49  | 93.74     | 100.00      |
| ICSL  | 10.87     | 12.75     | 15.26  | 15.26  | 14.54  | 86.93     | 100.00      |
| HTFS  | 11.61     | 12.52     | 12.57  | 12.57  | 14.08  | 96.25     | 100.00      |
| MOSAIC| 12.12     | 12.41     | 12.62  | 12.62  | 14.83  | 93.86     | 98.65       |

Table V

Teams’ Evaluation Results

The similarity of values in Table V for VG- and NM-based procedures requires an explanation. Fig. 11 shows how example paths produced by VG and NM are very similar. The reason is that NM uses a path straightening procedure using LoS testing: while this method does not produce the all percentiles values. In comparison with FM, VG is easier to compute and understand, the latter being a fundamental aspect when comparing systems. Thus, the VG-based procedure is recommended for all cases where high reliability is needed or high quantile metrics are involved.

Table V reports the accuracy of each team as measured by EPS04, EPS15, and the three proposed procedures using the 75th percentile metric. The 75th percentile is part of the EvAAL evaluation framework and is used in IPIN competitions [21], [22]. Notice, in Table V, that EPF04, EPS15, VF, and FM produce different rankings. While RTLSUM keeps the first place in all cases, ICSL moves from the second to the fourth place with EPS15, VG, or NM, and to the third place when using FM, because EFP04 is more lenient of floor misdetection than EPS15, VG, or FM. For the same reason, the HFTS team, which has the best floor detection rate, moves up to the second place with both VG and FM. The MOSAIC team ranks third when using the VG- and NM-based procedures. The MOSAIC team had a few building detection errors, each of them bringing a 50-m penalty whose weight is higher in EFP04 than in EFP15, where the effect of the floor penalty is more relevant. This explains why MOSAIC, with its good floor detection rate, ranks better with EFP15 (second place) than EPF04 (fourth place).
shortest path in all cases as VG does, its results are usually very close. FM actively avoids edges, thus creating smooth but sinuous paths that divert from the shortest distance paths found by the VG method or their good approximation provided by the NM method. Also, the NM method is less computing-intensive. SM variants are much less resource-demanding than EE variants and, thus, recommended unless the FM method is required.

Correct identification of building and floor is of utmost importance for an IPS. The proposed measurement procedures avoid overpenalization or underpenalization for floor and building misidentifications, from a pedestrian perspective, given that they do not require the usage of compromise penalty values for heterogeneous environments. They do require the usage of a well-defined offset parameter, which is required for VG and NM, but optionally applicable to FM as preprocessing.

The proposed measurement procedures focused on error measurement in indoor environments. The indoor character influenced aspects of procedures, such as the endpoint correction. In general, a positioning system that has a building (map) information should validate the position estimations to avoid nonaccessible areas, such as obstacles or inaccessible areas. If the estimates should strictly lie indoors, then outside areas can be considered nonaccessible, and the correction procedure should be applied. Systems that provide seamless indoor and outdoor localization will consider outdoor areas as accessible.

VI. CONCLUSION

Comparing IPSs is a complex task and involves many metrics. We have arguably moved a step forward in the direction of improving the usefulness of the most important metric, that is, positioning accuracy. In the case of a person or robot, most accuracy measures currently used are some statistics on positioning errors along the path, where the positioning error is defined as the Euclidean distance between an estimated position and the corresponding correct (ground truth) position. Euclidean distance is a good choice for approximating the cost of making a bad estimate: it is simple to compute and explain, its mathematical properties are well known, and it does not require to be tuned using free parameters.

The ISO/IEC 18305:2016 standard [6] uses E3D, while the IPIN competitions use 2-D Euclidean distance with floor penalty (EFP). Floor penalty accounts for the cost of bad positioning estimate that is perceived by a person or robot in the case of floor detection errors but adds a parameter \( X \), which needs to be tuned to the environment on the basis of experience. The walking distance proposed in this article is an improvement over both 3-D and EFP in that it accounts for the cost of floor detection errors and additionally for the cost of going around 2-D obstacles, such as walls, which may be significant in office-like environments. In comparison with EFP, it removes the need of a floor penalty parameter, which is a compromise value for heterogeneous environments found in a nonalgorithmic way; it instead uses an offset parameter that is based on the size of the target and the minimal entrance (e.g., door) size in a well-defined way. In the experiments, the offset value was 0.2 m, which accounted for the average person width and avoided entrance blockage.

The procedures described in this article are tailored for pedestrian paths determination. They were developed considering floors and buildings as entities with well-defined boundaries and known entrance points. Thus, they are not appropriate for use cases, such as UAVs (e.g., drones), or boundary-less environment representations.

The source code for all the procedures is available with an Apache-2.0 license and can be extended to include other pathfinding methods. As an example, the EE variants can be used in further analyses that are not limited to the shortest path.

The positioning error defined in this article will be experimented in the next editions of the IPIN competition and
possibly used as an optional modification to the EvAAL framework [8].

REFERENCES

[1] A. Basiri et al., “Indoor location based services challenges, requirements and usability of current solutions,” Comput. Sci. Rev., vol. 24, pp. 1–12, May 2017.

[2] A. R. Jiménez Ruiz and F. Seco Granja, “Comparing ubisense, BeSpoon, and DecaWave UWB location systems: Indoor performance analysis,” IEEE Trans. Instrum. Meas., vol. 66, no. 8, pp. 2106–2117, Aug. 2017.

[3] J. Aparicio, F. J. Álvarez, L. Hernandez, and S.Holm, “A review of techniques for ultrasonic indoor localization systems,” The J. Acoust. Soc. Amer., vol. 145, no. 3, p. 1884, 2019.

[4] P. Bahl and V. Padmanabhan, “Radar: An in-building RF-based user location and tracking system,” in Proc. IEEE INFOCOM Conf. Comput. Commun., 19th Annu. Joint Conf. Comput. Commun. Societies, Mar. 2000, pp. 775–784.

[5] P. Davidson and R. Piche, “A survey of selected indoor positioning methods for smartphones,” IEEE Commun. Surveys Tuts., vol. 19, no. 2, pp. 1113–1136, 3rd Quart., 2017.

[6] Information Technology—Real Time Location Systems—Test and Evaluation of Localization and Tracking Systems, Standard ISO/IEC 18305:2016, ISO Central Secretary, 2016.

[7] F. Lemic, A. Behboudi, V. Handziski, and A. Wolisz, “Experimental decomposition of the performance of fingerprinting-based localization algorithms,” in Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), Oct. 2014, pp. 355–364.

[8] F. Potortì, P. Barsocchi, M. Girolami, J. Torres-Sospedra, and R. Montoliu, “Evaluating indoor localization solutions in large environments through competitive benchmarking: The EvAAL-ETRI competition,” in Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), Oct. 2015, pp. 1–10.

[9] D. Lyberopoulos and J. Liu, “The Microsoft indoor localization competition: Experiences and lessons learned,” IEEE Signal Process. Mag., vol. 34, no. 5, pp. 125–140, Sep. 2017.

[10] G. M. Mendoza-Silva, J. Torres-Sospedra, and J. Huerta, “A more realistic error distance calculation for indoor positioning systems accuracy evaluation,” in Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), Sep. 2017, pp. 1–8.

[11] S. Feld, “Scoring of alternative routes using implicit building topologies,” in Proc. Sci. Inf. Conf. (SAI), Jul. 2015, pp. 329–336.

[12] Y.-H. Lin, Y.-S. Liu, G. Gao, X.-G. Han, C.-Y. Lai, and M. Gu, “The IFC-based path planning for 3D indoor spaces,” Adv. Eng. Informat., vol. 27, no. 2, pp. 189–205, Apr. 2013.

[13] T. Pulkkinen and J. Verwijnen, “Evaluating indoor positioning errors,” in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Oct. 2015, pp. 167–169.

[14] S. Adler, S. Schmitt, K. Wolter, and M. Kyas, “A survey of experimental methods for smartphones,” in Proc. 4th Int. Conf. Ubiquitous Comp., pp. 1–10, 2013.

[15] J. Torres-Sospedra et al., “A comparative study of navigation meshes,” in Proc. 9th Int. Conf. Motion Games MIG, 2016, pp. 91–100.

[16] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, no. 4, pp. 1591–1595, 1996.

[17] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” IEEE Robot. Autom. Mag., vol. 20, no. 4, pp. 111–120, Dec. 2013.

[18] D.-J. Kroon. (2009). Accurate Fast Marching: Multistencils Second Order Fast Marching. Accessed: Apr. 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/accurate-fast-marching.html

[19] J. R. Finkel and J. L. Bentley, “Quad trees a data structure for retrieval on composite keys,” Acta Inf., vol. 4, no. 1, pp. 1–9, 1974.

[20] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, no. 4, pp. 1591–1595, 1996.

[21] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” IEEE Robot. Autom. Mag., vol. 20, no. 4, pp. 111–120, Dec. 2013.

[22] D.-J. Kroon. (2009). Accurate Fast Marching: Multistencils Second Order Fast Marching. Accessed: Apr. 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/accurate-fast-marching.html

[23] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, no. 4, pp. 1591–1595, 1996.

[24] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” IEEE Robot. Autom. Mag., vol. 20, no. 4, pp. 111–120, Dec. 2013.

[25] D.-J. Kroon. (2009). Accurate Fast Marching: Multistencils Second Order Fast Marching. Accessed: Apr. 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/accurate-fast-marching.html

[26] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, no. 4, pp. 1591–1595, 1996.

[27] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” IEEE Robot. Autom. Mag., vol. 20, no. 4, pp. 111–120, Dec. 2013.

[28] D.-J. Kroon. (2009). Accurate Fast Marching: Multistencils Second Order Fast Marching. Accessed: Apr. 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/accurate-fast-marching.html

[29] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, no. 4, pp. 1591–1595, 1996.

[30] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” IEEE Robot. Autom. Mag., vol. 20, no. 4, pp. 111–120, Dec. 2013.

[31] D.-J. Kroon. (2009). Accurate Fast Marching: Multistencils Second Order Fast Marching. Accessed: Apr. 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/accurate-fast-marching.html

[32] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, no. 4, pp. 1591–1595, 1996.

[33] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” IEEE Robot. Autom. Mag., vol. 20, no. 4, pp. 111–120, Dec. 2013.

[34] D.-J. Kroon. (2009). Accurate Fast Marching: Multistencils Second Order Fast Marching. Accessed: Apr. 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/accurate-fast-marching.html

[35] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, no. 4, pp. 1591–1595, 1996.

[36] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” IEEE Robot. Autom. Mag., vol. 20, no. 4, pp. 111–120, Dec. 2013.

[37] D.-J. Kroon. (2009). Accurate Fast Marching: Multistencils Second Order Fast Marching. Accessed: Apr. 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/accurate-fast-marching.html

[38] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” Proc. Nat. Acad. Sci. USA, vol. 93, no. 4, pp. 1591–1595, 1996.

[39] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” IEEE Robot. Autom. Mag., vol. 20, no. 4, pp. 111–120, Dec. 2013.

[40] D.-J. Kroon. (2009). Accurate Fast Marching: Multistencils Second Order Fast Marching. Accessed: Apr. 2020. [Online]. Available: https://www.mathworks.com/help/matlab/ref/accurate-fast-marching.html
based indoor positioning, indoor navigation, machine learning, and GIS applications.

S. Choi, J. Yoo, and H. J. Kim, “Machine learning for indoor localization: Deep learning and semi-supervised learning,” in Proc. USB Site 6th Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), 2015.

R. Berkvens, M. Weyn, and H. Peremans, “Localization performance quantification by conditional entropy,” in Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), Oct. 2015, pp. 1–7.

M. Benedito-Bordonau, D. Gargallo, J. Avariento, A. Sanchis, M. Gould, and J. Huerta, “UII Smart Campus: Un ejemplo de integracion de recursos en la Universitat Jaume I de Castello,” in Proc. IV JJIDE, Toledo, Spain, 2013, pp. 417–426.

Q. Xiong et al., “Free multi-floor indoor space extraction from complex 3D building models,” Earth Sci. Informat., vol. 10, no. 1, pp. 69–83, Mar. 2017.

Y. Pang, C. Zhang, L. Zhou, B. Lin, and G. Lv, “Extracting indoor space information in complex building environments,” ISPRS Int. J. Geo-Inf., vol. 7, no. 8, p. 321, Aug. 2018.

D. Lymberopoulos, J. Liu, Y. Zhang, P. Dutta, X. Yang, and A. Rowe, Microsoft Indoor Localization Competition 2016. Accessed: Apr. 2020.

[Online]. Available: http://research.microsoft.com/en-us/events/msindoordlocompetition2016/default.aspx

Germán Martín Mendoza-Silva received the bachelor’s degree in computer science from the University of Oriente, Santiago de Cuba, Cuba, in 2005, the M.Sc. degree in geospatial technologies from the University of Münster (WWU), Münster, Germany, the Universidade NOVA de Lisboa (UNL), Lisbon, Portugal, and Universitat Jaume I (UJI), Castellón de la Plana, Spain, in 2015, and the Ph.D. degree in informatics from UJI in 2020.

He is a Researcher with UJI focused on WLAN-based indoor positioning, indoor navigation, machine learning, and GIS applications.

Joaquín Torres-Sospedra received the Ph.D. degree in ensembles of neural networks and machine learning from Universitat Jaume I, Castellón de la Plana, Spain, in 2011.

Since January 2020, he has been the Scientific Coordinator with UBIK Geospatial Solutions, Castellón de la Plana. He has authored more than 120 articles in journals and conferences. His current research interests include indoor positioning solutions based on Wi-Fi & BLE, machine learning, and evaluation.

Dr. Torres-Sospedra is the Chair of the IPIN International Standards Committee and the IPIN Off-Site Competition.

Adriano Moreira received the Licenciatura degree in electronics and telecommunications engineering and the Ph.D. degree in electrical engineering from the University of Aveiro, Aveiro, Portugal, in 1989 and 1997, respectively.

He is an Associate Professor, with Habilitation, with the University of Minho, Braga, Portugal, where he is also a Researcher with the Algorithmi Research Centre. He co-founded the Computer Communications and Pervasive Media Research Group, University of Minho, where he is also the Director of the MAP-tele doctoral program in Telecommunications. In the past few years, he has participated in many research projects funded by national and EU programs. He is the author of several scientific publications in conferences and journals and the author of one patent in the area of computational geometry. His research activities have been taking place within the ubicom@uminho research subgroup, which has been focusing on the creation of technologies for smart places.

Dr. Moreira, together with his colleagues, won the first prize on the off-site track of the EvAAL-ETRI Indoor Localization Competition (IPIN 2015 and 2017) and the second prize of the corresponding competition in 2016.

Stefan Knauth received the Diploma degree in physics from Goethe-Universität, Frankfurt, Germany, in 1996, and the Ph.D. degree in solid-state physics from Leipzig University, Leipzig, Germany, in 2001.

From 2005 to 2008, he was a Senior Research Assistant and the Deputy Head of the CEESAR Centre (now iHomeLab), Lucerne University of Applied Sciences, Lucerne, Switzerland. His research was focused on ambient-assisted living and smart energy. His earlier stations include a 3GPP System Engineer with Siemens ICM/ICN, Leipzig, and a System Engineer for ISS research payloads in cooperation with ESA and Kayser-Threde (now OHB System AG), Munich, Germany. Since 2008, he has been a Full Professor of computer science with HFT Stuttgart—University of Applied Sciences, Stuttgart, Germany. His research interests include indoor positioning, the Internet of Things (IoT), low resource embedded systems, building automation, and mobile applications.

Rafael Berkvens (Member, IEEE) received the M.Eng. and Ph.D. degrees from the University of Antwerp, Antwerp, Belgium, in 2012 and 2017, respectively.

He is a Senior Researcher with the IDLab, University of Antwerp, in collaboration with imec, Leuven, Belgium. He is involved in or supervisor of several national and international research projects. His main research interests include localization and tracking based on wireless communication, low-power wireless sensor networks, the Internet of Things, and context awareness.

Joaquín Huerta is currently a Full Professor with the Department of Information Systems, Universitat Jaume I, Castellón de la Plana, Spain, where he teaches several courses related to GIS and Internet technologies. He is also the Head of the GEOTEC Research Group and the Director of the Erasmus Mundus Master of Science in Geospatial Technologies Degree Program, run jointly by the University of Münster, Münster, Germany, and the Universidade NOVA de Lisboa, Lisbon, Portugal. He has been the principal investigator of several research projects, including EU projects such as A-WEAR, GEO-C, and EUROGEOSS. In addition to academic activities, he is a Founding Member of UBIK Geospatial Solutions, Castellón de la Plana. His current research interests include indoor positioning, smart cities, mobile and web GIS applications, and augmented reality.

Francesco Potortì (Member, IEEE) is a Senior Researcher with the National Research Council—Information Science and Technologies Institute (CRN-ISTI), Pisa, Italy, where he has been working since 1989. He is a coauthor of over 80 peer-reviewed scientific articles. He is also a supervisor of the use of software licenses in some European projects. He encourages the use of free software licenses in the research community. His research interests include communications protocols, especially satellite and terrestrial wireless; Internet technology, especially TCP/IP on wireless channels; and simulation of communications systems; currently, they include received signal strength (RSS)-based indoor localization, interoperability, and evaluation of indoor localization systems.

Dr. Potortì has been a member of the IPIN Steering Board since 2019. He was the Organizer of the 2011–2013 EvAAL Competitions. He defined the EvAAL Framework, which is the basis for the competitions hosted by IPIN, the Indoor Positioning and Indoor Navigation International Conference. He was the Organizer of the IPIN Competitions from 2014 to 2017. He was the Chairman of the tenth edition of the IPIN Conference and the associated sixth edition of the IPIN Competition in 2019.

Dr. Moreira, together with his colleagues, won the first prize on the off-site track of the EvAAL-ETRI Indoor Localization Competition (IPIN 2015 and 2017) and the second prize of the corresponding competition in 2016.

[44] S. Choi, J. Yoo, and H. J. Kim, “Machine learning for indoor localization: Deep learning and semi-supervised learning,” in Proc. USB Site 6th Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), 2015.

[45] R. Berkvens, M. Weyn, and H. Peremans, “Localization performance quantification by conditional entropy,” in Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), Oct. 2015, pp. 1–7.

[46] M. Benedito-Bordonau, D. Gargallo, J. Avariento, A. Sanchis, M. Gould, and J. Huerta, “UII Smart Campus: Un ejemplo de integracion de recursos en la Universitat Jaume I de Castello,” in Proc. IV JJIDE, Toledo, Spain, 2013, pp. 417–426.

[47] Q. Xiong et al., “Free multi-floor indoor space extraction from complex 3D building models,” Earth Sci. Informat., vol. 10, no. 1, pp. 69–83, Mar. 2017.

[48] Y. Pang, C. Zhang, L. Zhou, B. Lin, and G. Lv, “Extracting indoor space information in complex building environments,” ISPRS Int. J. Geo-Inf., vol. 7, no. 8, p. 321, Aug. 2018.

[49] D. Lymberopoulos, J. Liu, Y. Zhang, P. Dutta, X. Yang, and A. Rowe, Microsoft Indoor Localization Competition 2016. Accessed: Apr. 2020.

[Online]. Available: http://research.microsoft.com/en-us/events/msindoordlocompetition2016/default.aspx