Beyond Euclidean distance for error measurement in pedestrian indoor location

Beyond Euclidean Distance for Error Measurement in Pedestrian Indoor Location

Germán Martín Mendoza-Silva, Joaquín Torres-Sospedra, Francesco Potortì, Adriano Moreira, Stefan Knauth, Rafael Berkvens and Joaquín Huerta

***Abstract*—Indoor Positioning Systems 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 paper, we describe procedures that measure 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 or Navigational Meshes for vector maps, and Fast Marching for raster maps. Multi-floor and multi-building paths use information on vertical in-building communication ways and outdoor paths. The proposed measurement procedures are applied to position estimates provided by the Indoor Positioning Systems 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 Visibility Graphs algorithm computes shortest distance paths; Navigational Meshes produces very similar paths with significantly shorter computation time; Fast Marching 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 m to 3.7 m across the evaluation sets.**

***Index Terms*—Indoor Positioning System Evaluation, Error Measurement, Indoor Pathfinding, WiFi Fingerprinting.**

Manuscript received March 17, 2020; accepted May 1, 2020. Date of publication May 18, 2020; date of current version June 9, 2020. G. M. Mendoza-Silva gratefully acknowledges funding from grant PREDOC/2016/55 by Universitat Jaume I. J. Torres-Sospedra gratefully acknowledge funding from Ministerio de Ciencia, Innovación y Universidades (INSIGNIA, PTQ2018-009981). The Associate Editor coordinating the review process was Dr. XXXXXX XXXXX. (Corresponding author: Joaquín Torres-Sospedra.)

G.M. Mendoza-Silva and J. Huerta are with the Institute of New Imaging Technologies, Universitat Jaume I, Avda. Vicente Sos Baynat S/N, Castellón, Spain (e-mail: gmendoza@uji.es and huerta@uji.es).

J. Torres-Sospedra is with UBIK Geospatial Solutions S.L., Castellón, Spain (e-mail: jtorres@uji.es).

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

A. Moreira is with the Algoritmi Research Centre, University of Minho, Portugal (e-mail: adriano.moreira@algoritmi.uminho.pt).

S. Knauth is with the Faculty for Computer Sciences, Mathematics and Geomatics, HFT Stuttgart – University of Applied Sciences, Schellingstr. 24, Stuttgart, Germany (e-mail: stefan.knauth@hft-stuttgart.de).

R. Berkvens is with IDLab - Faculty of Applied Engineering, University of Antwerp - imec, Sint-Pietersvliet 7, 2000 Antwerp, Belgium (e-mail: rafael.berkvens@uantwerpen.be).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier no. 10.1109/TIM.2020.xxxxxxx

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 like UWB [2] or Ultrasound [3], to the more commonly used pedestrian IPSs that exhibit errors of several meters [4]. The latter are 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, 7, 8, 9]. The Euclidean distance is easy and fast to compute, and arguably the most significant error definition when Line-of-Sight (LoS) exists between 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 2D paths were determined using Visibility Graphs (VG) 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 multi-building multi-floor scenario with multiple endpoints. As the number of elevators, stairs and building entrances increases, finding the optimal route becomes more complex [12]. The current manuscript technically extends our preliminary work by (i) considering two new pathfinding methods –Navigational Meshes (NM) and Fast Marching (FM)–; (ii) performing an evaluation on a multi-building multi-story scenario with out-

door navigation; and (iii) using position estimates obtained by participants in the IPIN 2015 competition. The implementation of pathfinding methods, measurement procedures and analyses presented in this manuscript are available [1] with an Apache-2.0 license. In summary, this manuscript contributes to advance the state of art in IPS evaluation, with the following specific contributions:

- description of five error measurement procedures for IPSs, using either vector or raster floor map information to compute the walking distance between two points;
- 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 (GNSS) 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 WiFi, 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 *et al.* [13] discusses about 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 *et al.* [15] proposes a novel way to compute 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] defines a novel dynamic evaluation procedure with predefined geometrical paths (see [18]) to capture real-life usage of the scenarios, as well as other features such as ease of deployment and cost-efficiency. Even if discussion about evaluation strategies is open, the distance between the position estimate and the real position –i.e. the positioning error– still relies on the Euclidean distance.

[1]With DOI: 10.5281/zenodo.3741390

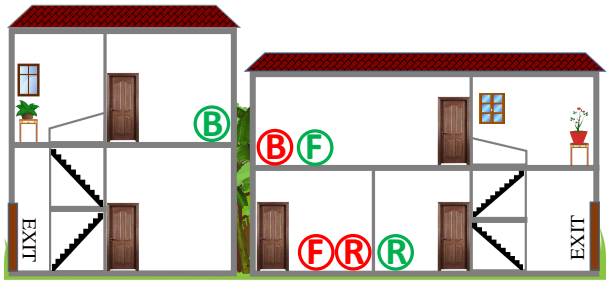


Fig. 1. Error underestimation from a pedestrian perspective when using 2D/3D Euclidean distance. The red and green colors of encircled signs indicate estimated and actual positions, respectively. The "B", "F" and "R" signs highlight identification failure cases for building, floor, and room, respectively.

The ISO/IEC 18305:2016 International Standard [6] provides an evaluation methodology for IPSs. The standard defines the magnitude of an error as the 3D (or 2D if only the horizontal components are considered) Euclidean distance (l_2 norm) between the actual and the estimated position. Despite the existence of the standard, the evaluation of IPSs is still an open challenge [19, 20]. In [19], the authors provide a critical reading of the standard that highlights the complexity of error measurement in indoor environments different from single-floor open spaces. They describe three alternatives for error measurement:

1) The 3D Euclidean distance defined in the standard. It is widely used in IPSs literature, it is very easy to compute, but it over- or under-estimates the importance of floor or building detection errors from a pedestrian perspective.
2) The 2D Euclidean distance with floor and building *penalties*, as used in several IPSs competitions [21, 22]. It is also very easy to compute, and the over- or under-estimation of the importance of a correct floor or building detection is less significant.
3) The pedestrian path length, as we proposed in [10]. It is not easy to compute as it requires the application of pathfinding methods to floor map information, but it can provide the distance that a pedestrian would walk.

Figure 1 depicts some cases where the Euclidean distance may significantly divert from error as may be perceived by a pedestrian or robot. Also, it shows that finding the proper floor penalty may be challenging in a heterogeneous environment. We acknowledge that precise floor map information may not exist for a given scenario. However, its availability would be valuable not only for evaluation, but also for radio map creation and map matching [5].

III. 2D Pathfinding

Pathfinding is an important topic not only for robotics and computer games [23, 24], but also to improve the accuracy of indoor positioning [25] or to support indoor navigation [26]. This section describes the pathfinding methods used for measuring the IPSs error: Visibility Graphs (VG), Navigation Meshes (NM) and Fast Marching (FM).

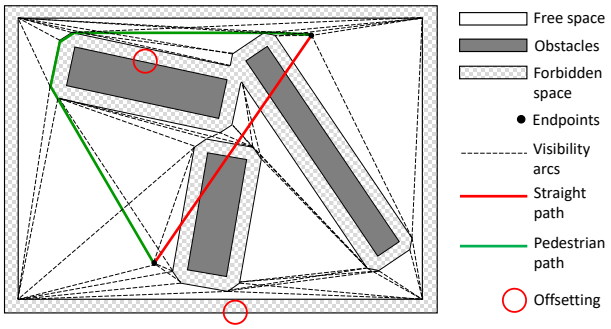


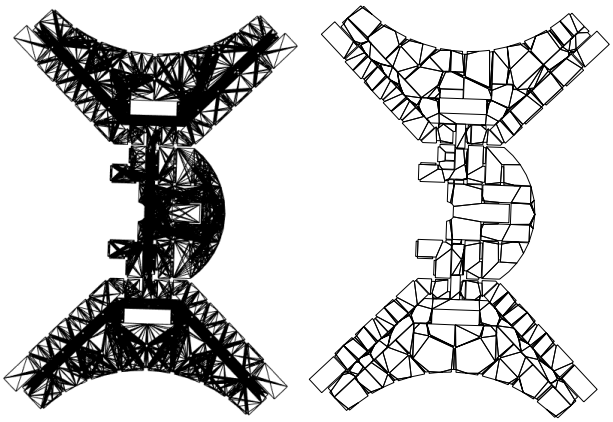
Fig. 2. Offsetting and VG path examples. The radius of the red circles matches the size of the subject.

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 the 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 Figure 2. The limits of the environment (polygon boundaries) are grown inwards 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, obtaining thus 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 ***Visibility Graph*** 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 by arcs of the VG, 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 like e.g., Dijkstra's [28] or A^* [29]. The paths found using VG usually have hard turns, which are not ideal for describing people movement. Also, VGs typically have a large number of arcs, which increases the computational cost of pathfinding. For example, the simple environment of Figure 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.



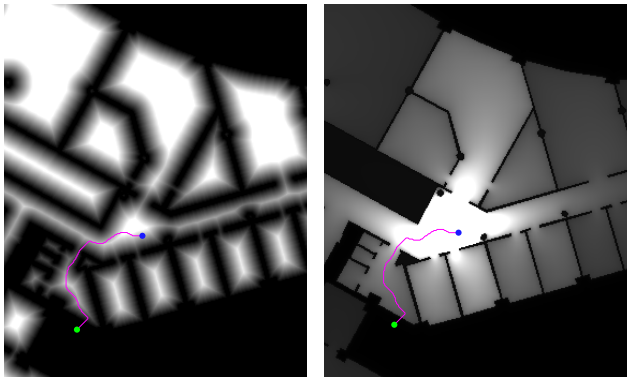
(a) Visibility Graph (b) Navigation Mesh

Fig. 3. Pathway representations for the TI building, lowest floor.

A ***Navigation Mesh*** [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].

Figure 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



(a) Speed transformation (b) FM image output

Fig. 4. FM examples. Blue and green dots represent origin and destination points, respectively. Magenta lines show the paths determined applying FM.

pathfinding. The correction step moves an endpoint to the closest point in a valid area. Such point is determined by finding the point on an edge of F which is the closest to the endpoint, and by translating the former inward or outward.

B. Fast Marching

Fast marching [35] is a method that, given a domain and propagation speeds, finds the time needed for a wave front to reach each point in the domain from a given source point. Unlike VG and NM, which are applied to vector maps, FM is most easily applied to raster images. In this case the domain is the set of image cells (pixels) and each cell is assigned a propagation speed value. The algorithm uses the known discrete solution to the Eikonal equation to update the cell values and thus find the time when the front crosses a point. For pathfinding, the speed in cells representing obstacles is set to zero. As the algorithm produces no local minimum [36], a path between source and destination points is determined following the maximum gradient direction from the destination point. The implementation used in the experiments presented in this paper is based on MultiStencils [37], an improvement on the original algorithm that produces smoother paths by solving the Eikonal equation through several stencils.

In this work, the *speed map* is created from a distance transformation applied to the binarized (e.g., black and white) map raster. The result is an image where the value of a cell is its distance to the nearest cell in the original image that corresponds to an obstacle. The distance value d of each cell is then transformed into a (non-zero) speed value v using:

$$v = 0.01 + \begin{cases} d & d < \tau \\ \tau & d \geq \tau \end{cases} \quad (1)$$

In Equation 1, τ is a *distance threshold* for speed reduction. All cells farther than τ from obstacles have the same (maximum) speed, which avoids to force paths towards the center zone in large areas. The value of τ depends on image resolution.For example, for the images later mentioned in Section V, τ 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 non-traversable obstacles. Figure 4a presents an example speed map, where the darker the shade the lower the speed. Obstacles and areas outside of the target environment (non-valid areas) are represented with the darkest shade. Figure 4b depicts the output from applying FM over the speed map from Figure 4a. Lighter shades correspond to points reached sooner by the wave front originating from the source point (blue dot). The paths shown in Figure 4 show that FM does not require endpoint corrections. In fact, since non-valid areas have the 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 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 2D pathfinding methods introduced in Section III. In the following, two variants are considered here named *Single Model* (SM) and *Endpoints Expansions* (EE). 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 pre-computed, as it is dependent on map information only, while *dynamic* information also depends on the endpoints. In the most common case, *inter-floor* connections include vertical connections between adjacent floors and *inter-building* connections are outdoor ground-level routes between building accesses, but more complex configurations may exist; for example, vertical connections between non-adjacent 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 inter-floor and inter-building connections. This variant produces a single 3D 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 ***EE 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 of

[2]With DOI: 10.5281/zenodo.3741390

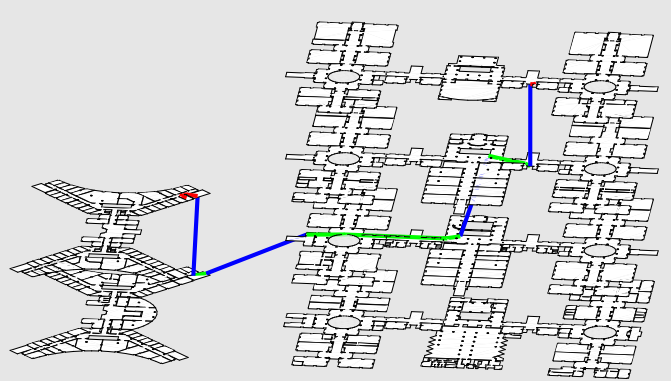


Fig. 5. Shortest distance path between two endpoints located in distinct buildings and floors, computed using the EE variant with NM pathfinding.

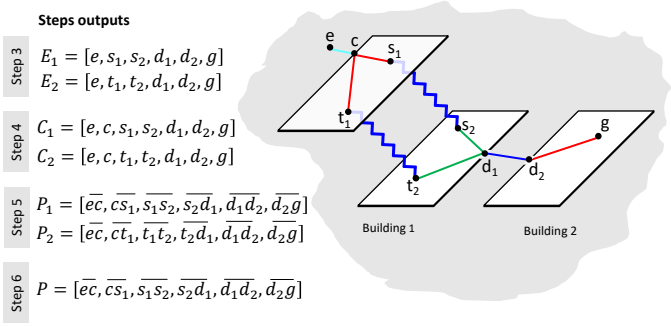


Fig. 6. Example of outputs for steps 3 to 6 from Algorithm 2. Legend for the black circles is e: estimate; g: ground truth; c: correction, s_1, s_2, t_1 and t_2: floor entrances; b_1 and b_2: building doors. $\overline{xy}$ represents the computed path between points x and y.

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

Figure 5 represents a complete path where each stretch is either:

- a vertical blue segment: statically known inter-floor connection;
- a horizontal blue segment: statically known inter-building connection;
- a green segment: a statically computed intra-floor stretch;
- a red segment: a dynamically computed intra-floor extreme stretch.

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
6 **end**

Algorithm 1: Path determination for the SM 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
7 **end**

Algorithm 2: Path determination for the EE variant.

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 floor-wise or building-wise advancement towards 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 inter-floor and inter-building connections. In the experimental implementation, only a subset of possible routes is computed, in order to shorten computation time.

Algorithm 1 and algorithm 2 are pseudo-code 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:

- *Floor model creation*: Loading of the environment data, i.e., the polygons or the image of each floor and the vertical and inter-building connection information. The output is the 2D graphs for the VG and NM methods, and the speed transformation for the FM method.
- *Single Model Integration*: Linking together the graphs created for each floor by the NM or VG methods. For each inter-floor or inter-building connection, new arcs are created: for VG, they are visibility arcs that irradiate from the access node; for NM, they are segments that connect the access node to the vertices of the containing polygon.
- *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.
- *Endpoint correction*: Correction of endpoints lying outside the environment to the closest point inside the valid areas. FM does not requires this step. Corrections are tracked and later included in the error magnitude.
- *Path computation*: Use one of the three 2D pathfinding methods to compute a path stretch for each endpoint pair not associated with a static pre-computed stretch length.
- *Endpoints contraction*: Obtain the sequences of stretches connecting the original endpoint pairs using the information computed or stored for each path stretch.

Figure 6 illustrates an example for steps 3–6 from Algorithm 2. Step 3 produces the 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 Algorithm 1 and Algorithm 2 are the paths that connect the input endpoint pairs. The application of Algorithm 1 or Algorithm 2 with any of the three 2D pathfinding methods is hereinafter called a *measurement procedure*. Five measurement procedures are proposed:

- *VG-SM*: Uses Algorithm 1 with VG for 2D pathfinding.
- *VG-EE*: Uses Algorithm 2 with VG for 2D pathfinding.
- *NM-SM*: Uses Algorithm 1 with NM for 2D pathfinding.
- *NM-EE*: Uses Algorithm 2 with NM for 2D pathfinding.
- *FM-EE*: Uses Algorithm 2 with FM for 2D 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 which 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 dataset [40]. UJIIndoorLoc includes publicly-available training and validation sets and a test set that is kept secret by the dataset curators. The test set was used in the IPIN 2015 (EvAAL-ETRI) Track 3 competition [41] to evaluate IPSs based on WiFi fingerprinting. This paper applies the proposed error measurement procedure to the 20,716 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 was collected in three university buildings. The environment data used in the measurement procedures included a 2D 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]. Figure 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.

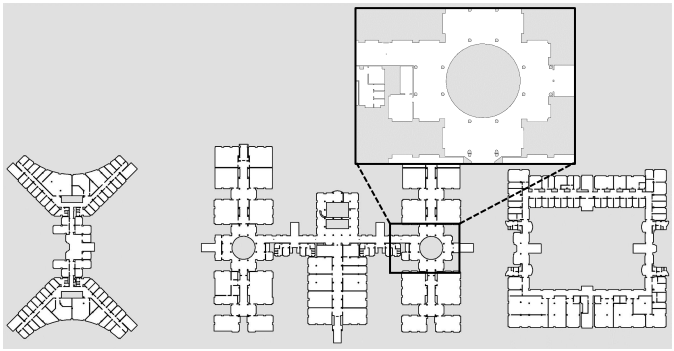


Fig. 7. Geometric data of buildings, representing the free space in white color. From left to right, the buildings are identified as TI, TD and TC.

TABLE I
General characteristics of environment's buildings.

Building	Area (m^2)	Floors	Vert. Comm.	Build. Doors
TI	15,600	4	18	7
TD	37,150	4	37	8
TC	27,100	5	24	4

The inter-floor (vertical) and inter-building (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 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.

Table II and Table III present the measured times for steps of Algorithm 1 and Algorithm 2, respectively. In the tables, Σ is the total step time, while μ and σ are the mean and

TABLE II
EMPIRICAL COMPUTATION TIMES (IN SECONDS) FOR STEPS OF ALGORITHM 1.

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

TABLE III
EMPIRICAL COMPUTATION TIMES FOR STEPS OF ALGORITHM 2.

Procedure	Floor model creation Σ (s)	Endpoints expansion μ(ms)	Endpoints correction μ (ms)	Paths computation $\mu \pm \sigma$ (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

standard deviation of a set of times. The times for *Endpoints expansion* and *Endpoints correction* are the mean per endpoint pair, while for *Paths computation* are the mean per intra-floor 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 *Endpoints expansion* and *Endpoints correction* are small compared to the times employed in the *Paths computation*, but they are not negligible. Their σ 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 Table II and Table 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 *Single model 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 are smaller in Table II than in Table III.

TABLE IV
COMPLEXITY OF THE ENVIRONMENT REPRESENTATION FOR ALL FLOORS.

VG	2.3×10^4 nodes – 3.2×10^5 arcs
NM	2.3×10^4 nodes – 3.6×10^4 arcs
FM	13 floor images of $2,971 \times 2,101$ pixels

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 6 million pixels in our experiment.

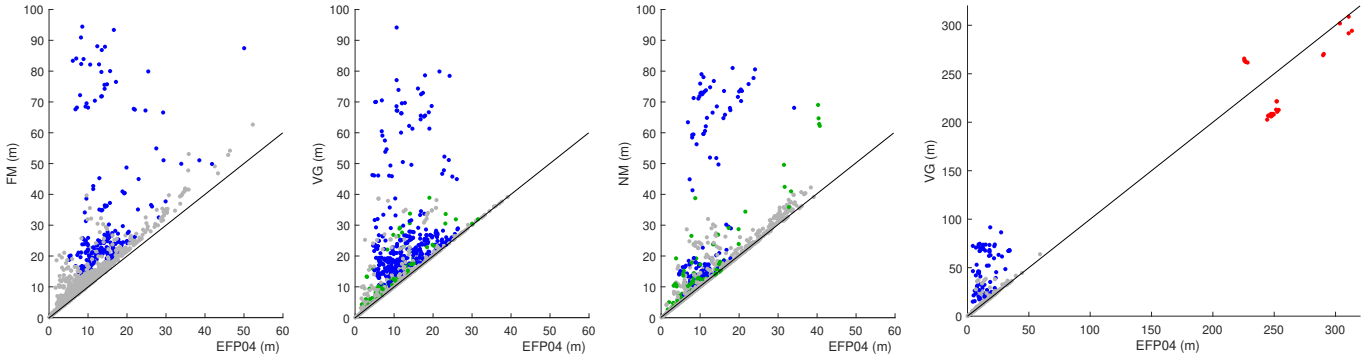
While important, evaluation time matters only for time affordability, and thus it is a secondary aspect of the proposed procedures. The main aspect is the error magnitudes as measured by the procedures. To set those magnitudes in the context of IPSs' evaluation, Figure 8 presents comparisons between the errors as measured by the proposed procedures and those measured by the EvAAL procedure. The EvAAL error is the sum of the 2D 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 m and 50 m, respectively:

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

The usage of the EvAAL procedure with the floor and building penalties from Equation 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 Figure 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. Figure 8a contains no green dots because FM does not use estimate correction, while Figure 8d 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 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.

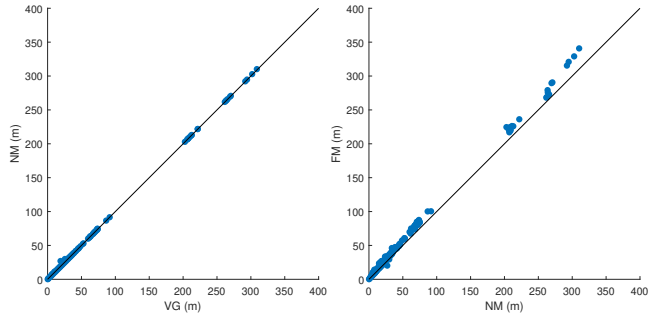
Red dots, only shown in Figure 8d, represent building misidentifications, which are the cases with the largest errors, only present in results from the MOSAIC team. The 50 m building penalty of the EFP04 procedure was too large for most cases. Also, that penalty was far too small for a few of



(a) RTLSUM (b) ICSL (c) HTFS (d) MOSAIC

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 cases of floor and building misidentification, respectively.

other cases that required changing floors at the two buildings, as previously shown by Figure 5. In general, when compared to pathfinding-based alternatives, the EFP04 procedure mostly underestimated the magnitude of the error for the evaluated environment and estimation sets.



(a) (b)

Fig. 9. Measures of the proposed procedures for the MOSAIC team.

Figure 9 (but also Figure 8d) 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.

Figure 10 presents the CDFs of measures as provided by the four procedures already explored and by three additional ones: the 3D Euclidean distance (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.

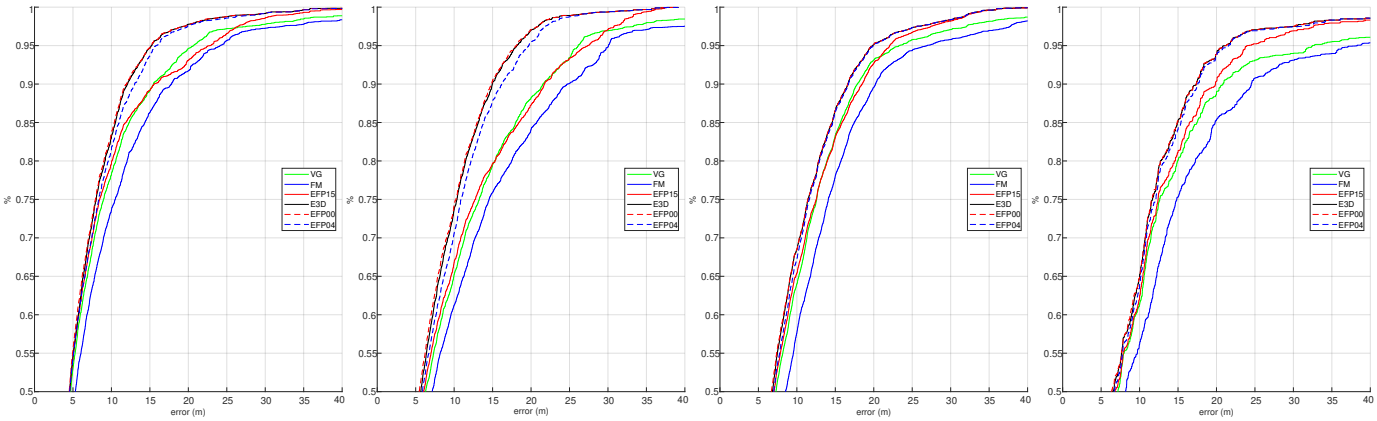
The 75th percentile values of EFP15 and VG are close for all teams. The difference increases for percentiles above 75th: it stays below 3 m up to the 95th for all teams but MOSAIC. Thus, EFP15 measures may arguably be a good mathematically simple approximation to VG-based measures for the percentiles 75th or lower. The FM-based procedure, which computes paths more realistic and thus longer than the shortest ones, provides the largest measures as seen across 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 a high reliability is needed or high quantile metrics are involved.

TABLE V
TEAMS EVALUATION RESULTS.

Team	EFP04 (m)	EFP15 (m)	VG (m)	NM (m)	FM (m)	Floor (%)	Building (%)
RTLSUM	8.34	8.48	8.96	8.96	10.49	93.74	100.00
ICSL	10.87	12.75	13.26	13.26	14.54	86.93	100.00
HFTS	11.61	12.52	12.57	12.57	14.09	96.25	100.00
MOSAIC	12.12	12.41	12.62	12.62	14.83	93.86	98.65

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 EFP04, 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



(a) RTLSUM (b) ICSL (c) HTFS (d) MOSAIC

Fig. 10. CDFs comparison of error magnitudes. Besides the explored measures, the charts also include those produced by the 3D Euclidean distance (E3D) and by the EvAAL procedure using floor penalties of zero (EFP00) and fifteen (EFP15).

(a) VG (b) NM (c) FM

Fig. 11. Examples of computed paths using the proposed pathfinding methods.

misidentifying a building, which explains MOSAIC getting the lowest rank with FM.

The similarity of values in Table V for VG-based and NM-based procedures requires an explanation. Figure 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 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 over- or under-penalization 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 pre-processing.

The proposed measurement procedures focused on error measurement in indoor environments. The indoor character influenced aspects of procedures, like the endpoint correction. In general, a positioning system that has building (map) information should validate the position estimations to avoid non-accessible areas like, obstacles or inaccessible areas. If the estimates should strictly lie indoors, then outside areas can be considered non-accessible 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, it does not require to be tuned using free parameters.

The ISO/IEC 18305:2016 standard [6] uses 3D Euclidean distance, while the IPIN competitions use 2D Euclidean distance with floor penalty (EFPX). 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 paper is an improvement over both 3D and EFPX in that it accounts for the cost of floor detection errors and additionally for the cost of going around 2D obstacles like walls, which may be significant in office-like environments. In comparison with EFPX, it removes the need of a floor penalty parameter, which is a compromise value for heterogeneous environments found in a non-algorithmic 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 average person width and avoided entrance blockage.

The procedures described in the paper 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 like 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 paper 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, E. S. Lohan, T. Moore, *et al.*, "Indoor location based services challenges, requirements and usability of current solutions," *Computer Science Review*, vol. 24, pp. 1–12, 2017.

[2] A. R. Jiménez Ruiz and F. Seco Granja, "Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 2106–2117, 2017.

[3] J. Aparicio, F. J. Álvarez, Á. Hernández, *et al.*, "A review of techniques for ultrasonic indoor localization systems," *The Journal of the Acoustical Society of America*, vol. 145, no. 3, pp. 1884–1884, 2019.

[4] P. Bahl and V. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proc. IEEE INFOCOM 2000 Conf. Computer Communications. Nineteenth Annual Joint Conf. IEEE Computer and Communications Societies*, IEEE, 2000.

[5] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1347–1370, 2017.

[6] ISO Central Secretary, "Information technology – real time locating systems – test and evaluation of localization and tracking systems," en, Standard ISO/IEC 18305:2016, 2016.

[7] F. Lemic, A. Behboodi, V. Handziski, *et al.*, "Experimental decomposition of the performance of fingerprinting-based localization algorithms," in *2014 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 355–364.

[8] F. Potortì, P. Barsocchi, M. Girolami, *et al.*, "Evaluating indoor localization solutions in large environments through competitive benchmarking: The evaal-etri competition," in *2015 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Oct. 2015.

[9] D. Lymberopoulos and J. Liu, "The microsoft indoor localization competition: Experiences and lessons learned," *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 125–140, 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 *2017 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2017, pp. 1–8.

[11] S. Feld, "Scoring of alternative routes using implicit building topologies," in *2015 Science and Information Conference (SAI)*, 2015, pp. 329–336.

[12] Y.-H. Lin, Y.-S. Liu, G. Gao, *et al.*, "The ifc-based path planning for 3d indoor spaces," *Advanced Engineering Informatics*, vol. 27, no. 2, pp. 189–205, 2013.

[13] T. Pulkkinen and J. Verwijnen, "Evaluating indoor positioning errors," in *2015 Int. Conf. Information and Communication Technology Convergence (ICTC)*, 2015, pp. 167–169.

[14] S. Adler, S. Schmitt, K. Wolter, *et al.*, "A survey of experimental evaluation in indoor localization research," in *2015 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2015.

[15] H. Liu and M. Schneider, "Similarity measurement of moving object trajectories," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoStreaming*, ser. IWGS '12, Redondo Beach, California: Association for Computing Machinery, 2012, pp. 19–22.

[16] C. M. de la Osa, G. G. Anagnostopoulos, M. Togneri, *et al.*, "Positioning evaluation and ground truth definition for real life use cases," in *2016 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2016.

[17] G. G. Anagnostopoulos, C. M. de la Osa, T. Nunes, *et al.*, "Practical evaluation and tuning methodology for indoor positioning systems," in *2016 Fourth Int. Conf. Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, 2016, pp. 130–139.

[18] T. Schwartz, C. Stahl, A. Mahr, *et al.*, "Using natural footstep-accurate traces for indoor positioning evaluation," in *2012 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2012, pp. 1–8.

[19] F. Potortì, A. Crivello, P. Barsocchi, *et al.*, "Evaluation of indoor localisation systems: Comments on the iso/iec 18305 standard," in *2018 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2018.

[20] D. B. Ahmed, L. E. Díez, E. M. Diaz, *et al.*, "A survey on test and evaluation methodologies of pedestrian localization systems," *IEEE Sens. J.*, vol. 20, no. 1, pp. 1–1, Jan. 2020.

[21] F. Potortì, A. Crivello, and F. Palumbo, "11 - the evaal evaluation framework and the ipin competitions," in *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*, ser. Intelligent Data-Centric Systems, J. Conesa, A. Pérez-Navarro, J. Torres-Sospedra, *et al.*, Eds., Academic Press, 2019, pp. 209–224.

[22] V. Renaudin, M. Ortiz, and J. Perul, et al., "Evaluating Indoor Positioning Systems in a Shopping Mall: The Lessons Learned From the IPIN 2018 Competition," *IEEE Access*, vol. 7, pp. 148 594–148 628, 2019.

[23] J. Latombe, *Robot motion planning*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 1991.

[24] R. Graham, H. McCabe, and S. Sheridan, "Pathfinding in computer games," *The ITB Journal*, vol. 4, no. 2, p. 6, 2003.

[25] H. Liu, B. Chen, P. Tseng, *et al.*, "Map-aware indoor area estimation with shortest path based on rss fingerprinting," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1–5.

[26] P.-y. Gilliéron, D. Büchel, I. Spassov, *et al.*, "Indoor navigation performance analysis," in *In Proceedings of the European Navigation Conference GNSS 2004*, 2004.

[27] X. Chen and S. McMains, "Polygon offsetting by computing winding numbers," in *ASME 2005 Int. Design Engineering Technical Conf. and Computers and Information in Engineering Conf.*, American Society of Mechanical Engineers, 2005, pp. 565–575.

[28] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[29] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[30] MathWorks®, *shortestpath method, in MATLAB® R2017b*, 2017.

[31] G. Snook, "Simplified 3d movement and pathfinding using navigation meshes," *Game programming gems*, vol. 1, no. 1, pp. 288–304, 2000.

[32] P. Tozour and I. Austin, "Building a near-optimal navigation mesh," *AI game programming wisdom*, vol. 1, pp. 298–304, 2002.

[33] B. Delaunay *et al.*, "Sur la sphere vide," *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, vol. 7, no. 793-800, pp. 1–2, 1934.

[34] W. Van Toll, R. Triesscheijn, M. Kallmann, *et al.*, "A comparative study of navigation meshes," in *Proc. 9th Int. Conf. Motion in Games*, ACM, 2016, pp. 91–100.

[35] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

[36] A. Valero-Gomez, J. V. Gomez, S. Garrido, *et al.*, "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, *Accurate fast marching: Multistencils second order fast marching*, https://www.mathworks.com/matlabcentral/fileexchange/24531-accurate-fast-marching, Accessed: Apr. 2020, 2009.

[38] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta informatica*, vol. 4, no. 1, pp. 1–9, 1974.

[39] Esri, *Arcgis indoors*, https://pro.arcgis.com/en/pro-app/help/data/indoors/get-started-with-arcgis-indoors.htm, Accessed: Apr. 2020.

[40] J. Torres-Sospedra, R. Montoliu, A. Martinez-Uso, *et al.*, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Int. Conf. Indoor Positioning and Indoor Navigation*, 2014.

[41] J. Torres-Sospedra, A. Moreira, S. Knauth, *et al.*, "A realistic evaluation of indoor positioning systems based on Wi-Fi fingerprinting: The 2015 EvAAL–ETRI competition," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 2, pp. 263–279, 2017.

[42] A. Moreira, M. J. Nicolau, F. Meneses, *et al.*, "Wi-fi fingerprinting in the real world - RTLS@UM at the EvAAL competition," in *Proc. 6th Conf. Indoor Positioning and Indoor Navigation*, 2015.

[43] S. Knauth, M. Storz, H. Dastageeri, *et al.*, "Fingerprint calibrated centroid and scalar product correlation rssi positioning in large environments," in *Proc. 6th Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2015.

[44] S. Choi, J. Yoo, and H. J. Kim, "Machine learning for indoor localization: Deep learning and semi-supervised learning," in *USB On-*

Site Proc. 6th Int. Conf. Indoor Positioning and Indoor Navigation (IPIN), 2015.

[45] R. Berkvens, M. Weyn, and H. Peremans, "Localization Performance Quantification by Conditional Entropy," in *Proc. 6th Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2015.

[46] M. Benedito-Bordonau, D. Gargallo, J. Avariento, *et al.*, "UJI Smart Campus: Un ejemplo de integracion de recursos en la Universitat Jaume I de Castello," in *IV JIIDE*, Toledo, Spain, 2013, pp. 417–426.

[47] Q. Xiong, Q. Zhu, Z. Du, *et al.*, "Free multi-floor indoor space extraction from complex 3d building models," *Earth Science Informatics*, vol. 10, no. 1, pp. 69–83, 2017.

[48] Y. Pang, C. Zhang, L. Zhou, *et al.*, "Extracting indoor space information in complex building environments," *ISPRS Int. Journal of Geo-Information*, vol. 7, no. 8, p. 321, 2018.

[49] D. Lymberopoulos, J. Liu, Y. Zhang, *et al.*, *Microsoft Indoor Localization Competition 2016*, Available at http://research.microsoft.com/en-us/events/msindoorloccompetition2016/default.aspx Accessed: Apr. 2020.

Germán Martín Mendoza-Silva has a Bachelor in Computer Science from the University of Oriente, Cuba in 2005; a Msc. in Geospatial Technologies from WWU (Germany), UNL (Portugal) and UJI (Spain) in 2015; and a PhD in Informatics from the UJI (Spain) in 2020. His research interests are on WLAN-based indoor positioning, indoor navigation, machine learning and GIS applications.

Joaquín Torres-Sospedra received his PhD about Ensembles of Neural Networks and Machine Learning from Universitat Jaume in 2011 I. Since January 2020 he is the Scientific Coordinator of UBIK Geospatial Solutions. 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. He is the chair of the IPIN International Standards Committee and IPIN off-site Competition.

Francesco Potortì is a senior researcher at the CNR-ISTI institute in Pisa, Italy, where he has worked since 1989. Organizer of the 2011-2013 EvAAL competitions. Defined the EvAAL framework, which is the basis for the competitions hosted by IPIN, the Indoor Positioning and Indoor Navigation international conference. Organizer of the IPIN competitions from 2014 to 2017. Chairman of the tenth edition of the IPIN conference and the associated sixth edition of the IPIN competition in 2019. Member of the IPIN steering board since 2019. His research interests included communications protocols, specifically satellite and terrestrial wireless; Internet technology, specifically TCP on wireless channels; simulation of communications systems. Currently they include RSS-based indoor localization, interoperability and evaluation of indoor localization systems. Co-author of over 80 peer-reviewed scientific papers. Supervisor of the use of software licenses in some European projects. He encourages the use of free software licenses in the research community.

Adriano Moreira is an Associate Professor, with Habilitation, at University of Minho, and a researcher at the Algoritmi Research Centre. He received the "Licenciatura" degree in Electronics and Telecommunications Engineering and the PhD degree in Electrical Engineering, respectively in 1989 and 1997, from the University of Aveiro. He co-founded the Computer Communications and Pervasive Media research group, and is the Director of the MAP-tele doctoral program in Telecommunications. His research activities have been taking place within the ubicom@uminho research sub-group, which has been focusing in the creation of technologies for smart places. In the past few years he 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. Together with his colleagues, won the 1st prize on the off-site track of the EvAAL-ETRI Indoor Localization Competition (IPIN 2015 and 2017) and the 2nd prize of the corresponding competition in 2016.

Stefan Knauth received a Diploma in physics from Goethe-Universität, Frankfurt, Germany, in 1996 and a Ph.D. in solid state physics from Leipzig University, Leipzig, Germany, in 2001. Since 2008 he is full professor for computer science at Stuttgart University of Applied Sciences - HFT Stuttgart, Germany. His research interests include indoor positioning, IoT, low resource embedded systems, building automation and mobile applications. From 2005 to 2008, he was a Senior Research Assistant and Deputy Head of the CEESAR centre (now iHomeLab) at Lucerne University of Applied Sciences, Lucerne, Switzerland. Research was focused around ambient assisted living and smart energy. Earlier stations include 3GPP system engineer for Siemens ICM/ICN, and system engineer for ISS research payloads in cooperation with ESA and Kayser-Threde (now OHB System AG).

Rafael Berkvens is a senior researcher at IDLab, University of Antwerp, Belgium, in collaboration with imec. He received his M.Eng. (2012) and Ph.D. (2017) from the University of Antwerp. He is currently 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, internet of things, and context awareness.

Joaquín Huerta is a full professor at the Department of Information Systems from University Jaume I in Spain, where he teaches several courses related to GIS and Internet Technologies. His current research interests are indoor positioning, smart cities, mobile and web GIS applications and augmented reality. He is the head of the GEOTEC Research Group, Director of the Erasmus Mundus Master of Science in Geospatial Technologies degree program, run jointly with the universities of Münster and Nova de Lisboa. He is and has been the principal investigator of several research projects including EU projects as A-WEAR, GEO-C, EUROGEOSS. In addition to academic activities he is a founding member of UBIK Geospatial Solutions http://ubikgs.com