A More Realistic Error Distance Calculation for Indoor Positioning Systems Accuracy Evaluation

Germán M. Mendoza-Silva, Joaquín Torres-Sospedra, Joaquín Huerta
Institute of New Imaging Technologies, Universitat Jaume I
Avda. Vicente Sos Baynat S/N, Castellón, Spain
{ gmendoza, jtorres, huerta }@uji.es

Abstract—The accuracy of indoor positioning systems is commonly computed as a metric based on the Euclidean distance from estimated locations to actual locations. This paper suggests that positioning error distances should be computed as the lengths of the paths that a person may follow when going from wrongly estimated positions to the real positions. The paper proposes a method that calculates the paths from floor plan and obstacles information using the visibility graphs and offsetting techniques, which are commonly used in robotics and CAD/CAM for navigation and manufacturing, respectively. Demonstration of the proposed method was done using a WiFi fingerprinting method based on kNN for pedestrian navigation. Comparisons between our proposed distance and the simple Euclidean distance have shown that the error distances are underestimated and that the differences between the two distances cannot be accurately represented by a fixed quantity in the context of an Indoor Positioning System (IPS) deployed in a library building. We consider that our proposed positioning error distance is more in line with the subjective error perceived by IPS users.

I. INTRODUCTION

It is accepted that indoor positioning and indoor navigation are hot research topics and that they are valuable for industry and LBS applications. Accuracy evaluation is fundamental for any new method or algorithm in the technology world, and in particular for indoor positioning and navigation. It commonly includes the definition of a metric upon which the evaluation is performed. For indoor positioning systems, the de facto standard has been the usage of metrics based on the Euclidean distance between estimated and actual positions, although other definitions of accuracy have been addressed [1]. The error calculation is not only used for ranking methods’ accuracy, but also for tuning parametrized algorithms [2]. The reasons to support the use of the Euclidean distance include:

1) it is the logical choice for 2D environments with (almost) no obstacles,
2) it is intuitive and easy to implement,
3) it is computationally fast, and
4) it requires no information about the testing environment.

The research works reviewed for this paper’s study agreed in the usage of the Euclidean distance as the basis for an evaluation metrics. They are mainly related to reviews and/or evaluation of methods for indoor positioning and range from early papers in this topic like RADAR [3] to recent indoor localization reviews like [4]. The Euclidean distance is a reasonable choice for most evaluations, mainly in early evaluation and simple environment layouts.

However, the Euclidean distance subtracts a significant realism to comprehensive evaluation and fails to recognize certain situations that may lead to a negative subjective evaluation by users of, for instance, pedestrian navigation. This is the case of room, floor or, even, building misidentification.

To improve accuracy’s determination, this paper proposes to use floor plans and obstacles information from the testing environment to calculate walkable paths between actual and estimated locations, which provide a more realistic estimation of distances from a person perspective (or navigable paths for robotic-based positioning). From a polygonal representation of a floor plan and its obstacles, the offsetting technique and visibility-graphs-based path finding technique (mainly used in CAD/CAM and robotics, respectively) are combined with knowledge of inter-floor traversing ways to determine collision-free paths. The lengths of the collision-free paths are then used as more realistic distances between the actual and the estimated locations in a multi-floor IPS evaluation.

To test the proposed method, a WiFi fingerprint database was collected in two floors of a university library building. Over this database, a classical kNN fingerprinting method was applied. The ground truth from the database and the location estimations provided by the kNN were used to compute distances using the Euclidean method and the method proposed in this paper. The resulting distances were compared in order to find relations between them and possible effects on the used fingerprinting method. The comparison results led to further analysis of the suitability of the proposed method for IPS evaluation. Specifically, in this paper we:

1) propose a method for calculating realistic distances between actual and estimated locations,
2) show that the positioning error is underestimated in some environments and adding fixed penalty quantities might not be realistic enough, and
3) provide guidelines on which situations to use the distance calculation method proposed in this paper.

The remaining sections of this paper are: Section II addresses positioning error distance calculation. Section III explains the proposed distance calculation method. Section IV and Section V presents the experimental work and a discussion on its results, respectively. Finally, Section VI exposes conclusions drawn from this paper’s work and suggest its possible continuation directions.
II. ERROR DISTANCE CALCULATION IN INDOOR POSITIONING SYSTEMS EVALUATION

In indoor positioning research works, it is common to assume that the error distances are calculated using the Euclidean distance, although many papers do not explicitly state its usage. Examples of the latter case are indoor positioning research papers that provide general overviews [5], review wireless technologies [4], or focus on WiFi [6], inertial sensors [7], BLE [8], RFID [9], and ultrasound [10]. Nevertheless, other research works have properly defined accuracy in terms of the Euclidean distance [11], [12], described how it is calculated [13], or even differentiated between accuracy and precision [1].

Research works where a realistic distance definition is most significant are those addressing IPS benchmarking platforms and competitions. An example of benchmarking platform is EVARILOS [14], which provides several accuracy metrics: point-related accuracy metrics and a room accuracy metric. The point-related metrics are the mean, standard deviation, minimum and maximum values of the error distance calculated using the 2D or 3D Euclidean distance. The room accuracy metric is the rate of success at identifying a room.

Examples of IPS competitions are the Microsoft (2014-2016), EvAAL (2011-2013) and IPIN (2014-2016) competitions. The Microsoft competitions [15], [16], [17] have used the 2D and/or the 3D Euclidean distance in a local coordinate system to calculate the positioning error. Each year, the evaluation environment included rooms and halls on one floor of a building. The team reporting the lowest averaged error have won the price.

The EvAAL 2011 to 2013 competitions [18], [19], [20] considered as accuracy metrics (1) the ratio (fraction of the total time) of success on locating the user inside areas of interest and (2) the third quartile of the errors (Euclidean distance) between the reference and location estimations along predefined paths.

The 2014 IPIN competition used the 75 percentile error CDF as the score for accuracy ranking [21], [22]. The 2015 and 2016 IPIN competitions used the EvAAL evaluation framework. For both competitions, the 2D positioning error distance was calculated using the geodesic distance between actual and estimated locations, and penalties were added for floor error and building error. The final error score was the third quartile of the (penalized) errors.

The 2015 and 2016 IPIN competitions acknowledged the importance of building and floor misidentification, which is realistically addressed in this paper. Figure 1 exemplifies why building, floor, and room identification matters in indoor localization, in terms of the effort that a person would require for going to right location after being misled by an incorrect position estimation. The depicted cases are as follows.

1) Room misidentification, represented by red squares. In this case, a very short error might be reported for two locations, but the wall between them is not considered in the Euclidean distance based positioning error. Although

2) Floor misidentification, shown by green diamonds. These cases, though less common than room misidentification, should be considered, as seen in [22]. A wrong floor identification implies for a person to take the stairs or an elevator to go to the right floor.

3) Building misidentification, portrayed by blue circles. In this case, despite the distance is not large, due to the buildings’ entrance disposition, a person would need to walk a huge distance.

The misidentification cases presented in Figure 1, coupled with the fact that indoor spaces commonly contain obstacles, lead to the questions of (1) whether the Euclidean distance should be used in some environments to calculate a positioning error metric, and (2) whether adding fixed penalty quantities is enough for making the error distance more realistic. Additionally, a misidentification rate summarizes the precision of the algorithm but omits interesting information, e.g., the distance between the estimated and real rooms is not provided.

From the perspective of a person using an IPS for localization, the subjective error in positioning would be the distance of the path that the person needs to follow between the real and the IPS-estimated locations. The estimation of such paths needs contextual information, most likely floor plans of the testing environment, but also knowledge about the environment. Contextual and map information has been used previously for estimation corrections in techniques known as waypoint corrections and map matching [23] or in more heuristic-like ways [24].

In our work, we use floor plan and obstacles information to determine paths from the locations estimated by an IPS to the actual, target location. The paths calculation uses the offsetting and visibility graphs techniques, which are well-known in CAD/CAM and robotics. Explicit knowledge about the location of floor traversing ways (stairs and elevators) have been also used.
III. DISTANCE CALCULATION USING MAP INFORMATION

The proposed error distance calculation method uses an existing methodology in robot navigation. The methodology harnesses the notions of forbidden (configuration) space and visibility graphs, which are used to compute a 2D collision-free path based on the polygonal representation of the environment’s obstacles and limits.

In the proposed method, the paths between two endpoints, which may lie in different building’s floors, are determined as follows. The information of obstacles and the walls that limit the target environment for each floor is obtained from floor plans. Given the diameter of the subject of localization (e.g., a person or a robot) and given a floor’s obstacles, the space where the subject cannot be (the forbidden space) can be determined. Each floor’s forbidden space is then used to compute a visibility graph, i.e., a graph connecting vertices that have direct vision. The visibility graph is in turn used to compute in-floor collision-free paths. If the target endpoints lie in different floors, the floors’ exits/entrances are used as new endpoints for the in-floor paths computation. The floors’ exits/entrances, as well as the cost (distance) of the inter-floor displacements, are determined from information on the ways that link the building’s floors.

In the study presented in this paper, (1) each floor’s environment is a planar 2D region with polygonal, fixed obstacles, and (2) the obstacles are open sets, i.e., the subject of localization is allowed to touch them. The above restrictions are reasonable for positioning error calculation. The following two subsections explain the concepts of forbidden configuration space and visibility graphs.

A. Forbidden Space

Considering the subject of localization as a point simplifies the collision-free path computation. This simplification is achieved by determining the forbidden space, i.e., the areas in the environment where the subject would collide with the obstacles. In this paper, the subject’s shape is considered to be a disk $D_r$ with radius $r$. The set of the environment’s obstacles is denoted as $S = \{P_1, ..., P_t\}$ and the forbidden space as $C_{forb}(D_r, S)$. The forbidden space can be characterized by individual components calculated from the obstacles, as in eq.1:

$$C_{forb}(D_r, S) = \bigcup_{p \in S} C_{obs}(D_r, P) : P \subseteq S,$$

For calculating the sets $C_{obs}(D_r, P)$, the CAD/CAM technique known as polygon offsetting [25] has been used, which is adequate for outer offsets and inner offsets (insets). Outer offsets are used to determine the forbidden space of obstacles inside the environment; while inner offsets are used to determine the forbidden space of the environment’s boundary. A polygon’s outer and inner offsets, $O_{out}$ and $O_{in}$ respectively, are defined as:

$$O_{out}(P, D_r) = \{ p \in \mathbb{R}^2 : d(p, P) = r \},$$

$$O_{in}(P, D_r) = \{ p \in A : d(p, P) = r, A \subseteq P \},$$

where $d(p, P)$ is the minimum distance from the point $p$ to the edges of the polygon $P$ and $A \subseteq P$ indicates that all the points in $A$ lie inside $P$. If the walls that bound the environment are represented by a polygon, denoted as $P_{walls}$, and the obstacles in the environment are represented by other polygons, denoted as $P_i \in S$, we then calculate a forbidden space that includes the outer walls as:

$$C_{forb}(D_r, P_{walls} \cup S) = O_{in}(P_{walls}, D_r) \cup \bigcup_{P_i \in S} O_{out}(P_i, D_r),$$

In this equation, $D_r$ is the offsetting radius.

B. Visibility Graphs

The shortest collision-free path between two endpoints and among a set $S$ of disjoint polygonal obstacles can be computed using visibility graphs. In the visibility graph of $S$, denoted as $G_{vis}(S)$, the nodes are vertices from $S$ and two nodes $v$ and $w$ are connected by an arc if it does not collide with any obstacle’s edge. In order to consider the subject as a point and to take into account the environment’s boundary, the $G_{vis}(C_{forb}(D_r, P_{walls} \cup S))$ is computed instead. In addition to the vertices of $C_{forb}(D_r, P_{walls} \cup S)$, the target path endpoints can also be considered as an input for computing the visibility graph.

The shortest collision-free path between two points is composed by arcs of the visibility graph from the forbidden space and from the path endpoints [26]. An efficient visibility graph construction algorithm is described in [27], [26]. If the Euclidean distance is considered as the weight of arcs of the visibility graph (including the path endpoints), the shortest path can be computed using a graph searching algorithm, e.g., Dijkstra’s [28] or $A^*$ [29].
Figure 3 shows two examples of a subject (point) and obstacles for which the forbidden space, the visibility graph, and the shortest path between two endpoints have been determined. Notice that the paths are composed of vertices from the forbidden space and they create a visibility chain, i.e., in the path, each vertex "sees" the previous vertex and the following vertex. The usage in (b) of an offsetting radius larger than the one used in (a) caused the forbidden spaces of two obstacles to collide. Notice that two colliding forbidden spaces have been merged into one polygon, as the forbidden spaces should be disjoint. Figure 3 shows that the offsetting radius, i.e., the considered subject's diameter, influences the path determination.

C. Error Distance Calculation Method

The polygon offsetting method from [25] and its implementation from [30] support the calculation of the forbidden space for a non-convex polygonal representation of a floor plan and its obstacles. The information given by the forbidden space can be directly used to calculate the visibility graph, using the methods presented in [27], [31] and the implementation from [26]. The latter enables to pre-compute the visibility graph for a forbidden space. The pre-computed visibility graph does not include the target path's endpoints. The endpoints and their connecting arcs can be added later for path calculation.

As the proposed distance calculation method is destined to IPS evaluation, the endpoint corresponding to the estimated location may be located inside the forbidden space region or outside the target environment. Therefore, we have added a correction step before performing the path calculation. In general, if a path’s endpoint lies inside an obstacle’s forbidden region, the endpoint is moved to the point on the external edges of forbidden region which is at the smallest distance from the endpoint. Similarly, if the endpoint is outside the target environment or inside the limits’ forbidden region, it is moved to point on the internal edges of limits’ forbidden region which is at the smallest distance from the endpoint. We denote the stretch spanning from the original endpoints \( p_{\text{start}} \) and \( p_{\text{end}} \) to their corrections as \( p_{\text{start}}' \) and \( p_{\text{end}}' \), respectively. Notice that \( p_{\text{start}}' \) and \( p_{\text{end}}' \) may have zero length.

1) Real and estimated position in the same floor: For each \( p_{\text{start}} \) and \( p_{\text{end}} \) (or their corrections), the pre-computed visibility graph’s vertices visible to the endpoint are determined. The visible vertices determination is the same used for computing the visibility graph. Then, arcs that connect \( p_{\text{start}} \) or \( p_{\text{end}} \) to its respective visible vertices are added (along with the endpoint) to the pre-compute the visibility graph. Once the visibility graph has the new vertices and arcs, an \( A^* \) search is performed on the visibility graph using the Euclidean distance as heuristic function.

The steps already explained in this subsection allow the calculation of 2D collision-free path among obstacles in an enclosed space. It then remains to address multi-floor and multi-building contexts.

2) Real and estimated position in different floors: For multi-floor contexts, and before path determination, the inter-floor traversing ways (and their entrances) are identified. These ways are commonly stairs, ramps, and elevators, which are usually narrow passages. The paths that people generally follow when traversing one of them are very similar in shape and thus in distance. Therefore, we keep their information apart from the floors’ polygonal representation. A fixed weight is given to each of these ways. The weight is estimated as the horizontal and vertical distance for the case of stairs and ramps, and just the vertical distance in the case of elevators. The information kept for each inter-floor traversing way is stored as a pair of locations (the way’s entrances) with the way’s length. This study has not automatically gathered inter-floor ways’ information from floor plans because (1) gathering the vertical displacements or connections is a complex automatic process, and (2) in the context of IPS evaluation, inter-floor traversing ways’ information needs to be gathered only once.

The first step when determining the path between two endpoints located on different floors is choosing an inter-floor way in the floor of \( p_{\text{start}} \). The path calculation method determines paths through all inter-floor ways that have entrances on the floor of \( p_{\text{start}} \) and that conduct to the floor of \( p_{\text{end}} \). From all these paths, the method outputs three routes which cover possible people’s choices regarding inter-floor way’s entrance/exit:

1) The shortest path: the best possible scenario, where the user takes the stairs or elevator and whose resulting path between \( p_{\text{start}} \) and \( p_{\text{end}} \) is the shortest possible one.
2) The longest path: the worst possible scenario, where the user takes the stair or elevator and whose resulting path between \( p_{\text{start}} \) and \( p_{\text{end}} \) is the longest possible one.
3) The closest-exit path: a realistic scenario, where the user decides to take the closest stair or elevator. The path length is between the shortest and longest possible ones.

Figure 4 presents the three considered paths determined between two endpoints in different floors. The path traversing an inter-floor way is denoted as \( p_l \). In this figure, the upper “X” mark represents the starting point, which is moved (corrected) to environment’s boundary. Notice how the path resulting from
choosing the closest inter-floor exit is not necessarily the best path. The figure also shows, for the case of the longest path, how paths are composed of stretches. In this example, \( p_{c_{end}} \) has zero length.

The path calculation between two endpoints considering a particular inter-floor way’s entrance uses the 2D path computation previously presented. The method determines \( p_{f_{start}} \), which is the path from \( p_{start} \) (or its correction) to a selected inter-floor entrance. The method also computes \( p_{f_{end}} \), which is the path from the exit in the destination floor of the selected inter-floor way to \( p_{end} \) (or its correction). Finally, the path from \( p_{start} \) to \( p_{end} \), or more properly, from the estimated location to the actual location, is the following concatenation:

\[
path = p_{c_{start}} \sim p_{f_{start}} \sim pw \sim p_{f_{end}} \sim p_{c_{end}},
\]

where \( \sim \) represents a path concatenation operator. Given \( A \) and \( B \) ordered collections of points, if \( C = A \sim B \), then \( C \) contains all the points in \( A \) and all the points in \( B \) (but the first one) in that order.

![Fig. 4. Finding a path in two floors of a building. The “X” marks show the path’s endpoints. The red, green and orange trajectories show the longest, shortest and closest exit paths, respectively. For the longest path, the different stretches that compose the path have been tagged.](image1)

3) Real and estimated position in different buildings:
Path computation for multi-building contexts lies beyond the scope of our study because it requires more information than building floor plans and obstacles. Additionally, considering multi-building contexts would increase the complexity of the method, in terms of the required information and the method’s implementation. To compute paths for multi-building contexts, pedestrian ways or street network information is required. Before computing the paths, the entrances/exits (called herein after building’s doors) must be determined. To determine the path between an estimated location in building \( A \) and the actual location in building \( B \), outdoor paths between all pairs of doors \( (d_{Ai}, d_{Bj}) \) have to be determined, being \( d_{Ai} \) the \( i^{th} \) door of building \( A \) and \( d_{Bj} \) the \( j^{th} \) door of building \( B \). In turn, the best paths between the estimated location and \( d_{Ai} \), and between \( d_{Bj} \) and the actual location, have to be computed using the method explained in the previous subsection.

IV. EXPERIMENTS

This section presents the application of the proposed distance calculation method for the evaluation of kNN fingerprinting over a WiFi fingerprints database. Fingerprinting is a popular approach in indoor positioning, mainly for smartphone based indoor localization. It has been used as the core localization method [3] and as a baseline method for other methods’ evaluation [22]. Given a set of data points and a query point in an n-dimensional space, and two parameters specified by the value of \( k \) and a distance metric on the previous space, the kNN method finds the \( k \) data points that are closest to the query point. The kNN can be extended to handle multi-floor environments using a voting procedure, i.e., the floor value that appears most often in the \( k \) selected points is the estimation’s floor.

The rest of this sections describes the used WiFi fingerprint database, how the proposed distance calculation method has been applied to the evaluation of kNN fingerprinting-based IPS, and a comparison of the positioning error distances determined using the new method and the Euclidean distance.

A. Experimental Setup

The fingerprint dataset used in the experiments was collected in two consecutive floors of a university’s library building, in an area among bookshelves. Both floors have an almost identical layout and are communicated by two elevators and three stairways. For the experiments, the floors’ target areas were those closest to the fingerprints collection area and to the inter-floor traversing ways. Figure 5 shows the representation of target area once it has been prepared for paths computation.

![Fig. 5. Experiments target area. Green circles show entrances of inter-floor traversing ways (depicted by blue arrows) over the representation of each floor.](image2)

![Fig. 6. A floor’s plan and close-up of the fingerprint collection area. In the close-up, colored diamonds represent the positions of the fingerprint groups: blue are groups 1 and 5; green is group 2; fuchsia is group 3; and brown is group 4.](image3)
The fingerprint collection was organized into five groups, which are differentiated by colors in Figure 6. Apart from the distribution of the groups, the figure also presents one of the original floor plans. The fingerprints from groups 1, 2, and 3 were collected facing the “Up” and “Down” directions, while those from groups 4 and 5 were collected facing the “Right” and “Left” directions. Per each triplet location-direction-floor, six fingerprints were collected. The wireless networks detected in less than 5 percent of the samples were removed in order to reduce the data dimensionality and discard ephemeral access points. The number of networks was reduced from 200 to 62.

Groups 1 and 5 were used to compose a training set, and groups 2, 3 and 4 to compose a test set.

The parameters for the kNN fingerprinting method were:

- The Euclidean distance as metric to compare fingerprints
- $k$ values ranging from 1 to 41

### B. Experimental Results

Figure 7 shows the mean positioning error using the Euclidean distance and the proposed distances, for several $k$ values. For the Euclidean method, the floor misidentification penalty value described in [24] (15 m) was added to the positioning error. In the proposed method, the shortest and longest paths -for those cases where the floor was wrongly estimated- are considered. The results for the closest inter-floor way’s entrance paths are not shown because in the experiments the closest entrance paths coincided with the shortest paths. An excerpt of this results is shown in Table I for a detailed comparison among systems reporting similar results.

---

**TABLE I**

<table>
<thead>
<tr>
<th>$k$-value</th>
<th>EUC</th>
<th>SPA</th>
<th>LPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>2.4949</td>
<td>2.7417</td>
<td>2.7976</td>
</tr>
<tr>
<td>13</td>
<td>2.4825</td>
<td>2.7271</td>
<td>2.7691</td>
</tr>
<tr>
<td>15</td>
<td>2.4843</td>
<td>2.7157</td>
<td>2.7576</td>
</tr>
<tr>
<td>17</td>
<td>2.4839</td>
<td>2.7177</td>
<td>2.7596</td>
</tr>
</tbody>
</table>

---

Further analyses were performed to deeper explore the relation between the shortest path length and the Euclidean distance using the $k$ value of 11. This value has been selected since it provided a low mean positioning error (below 2.5 m), it is not so high according to many other previous works given the number of fingerprints per reference point (12), and shows some interesting findings. The scatter plot shown in the Figure 8 relates Euclidean distances and shortest path length positioning errors. In the figure, the green color marks the cases where there are full line-of-sight (FLOS) situations between actual locations and estimated locations and the estimation did not need to be corrected. The red color is used for the cases with partial line-of-sight (PLOS), where the estimations needed correction (they were inside obstacles) and those corrections had line-of-sight with the actual location. The remaining cases, i.e., those requiring a multi-stretch collision-free path (NLOS), are represented in blue color. Table II shows percentages corresponding to value ranges of the difference between the Euclidean distance and the best path length.

---

**TABLE II**

<table>
<thead>
<tr>
<th>Situation</th>
<th>Interval</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLOS</td>
<td>[0,0]</td>
<td>41.00%</td>
</tr>
<tr>
<td>PLOS</td>
<td>(0.0,0.5]</td>
<td>15.50%</td>
</tr>
<tr>
<td>NLOS</td>
<td>(0.0,0.5]</td>
<td>26.93%</td>
</tr>
<tr>
<td></td>
<td>(0.5,1.0]</td>
<td>8.33%</td>
</tr>
<tr>
<td></td>
<td>(1.0,3.06]</td>
<td>8.23%</td>
</tr>
</tbody>
</table>

According to the statistics for the NLOS cases (43.50% of the cases), the average difference between the euclidean distance and the proposed one (shortest path) is 0.50 m with an standard deviation of 0.57, which denotes a high variability. Notice that despite there are 56.50% of samples with full or partial line-of-sight, there still are more than 8% samples reporting a difference higher than 1 m for the NLOS cases.
V. Discussion

This section discusses the results presented in Section IV and provides insights into the usage of our proposed method.

A. Discussion of Experimental Results

The results show that the Euclidean distance underestimates the positioning error distance when the actual and the estimated locations are on the same floor. Figure 8 shows that there might be a linear relation between the Euclidean distance and the proposed distance. In many cases, there were many FLOS cases between the actual and the estimated locations (green values), in which the proposed distance was equivalent to the Euclidean distance. Also, there were many CLOS cases (red values), and the Euclidean distance was almost equivalent to the proposed distance. This full and partial line-of-sight represented 56.5% of total cases. The most important conclusion from the figure is that for the NLOS cases (blue values), the linear relation is less clear. In 8.23% of NLOS cases, the difference is higher than 1 m. For the NLOS cases, the distance provided by the proposed method is always higher than the Euclidean distance if the floor is correctly estimated. This demonstrates that the traditional error in positioning underestimates the real error in positioning in many cases.

Figure 8 also shows that adding a penalty term (e.g., wrong floor estimation) to the Euclidean distance does not create a fair alternative to the actual distance from a subjective user’s perspective. The highest error for the Euclidean distance plus a penalty term was 16.65 m (1.65 m of in-floor distance and 15 m of penalty), but for the distance computed with the proposed method was 13.59 m considering the shortest possible path. However, there was a case reporting an error of 16.18 m (Euclidean distance + penalty) which had a realistic distance of 15.18 m with the proposed method. Therefore, the worst case considering the Euclidean distance does not necessarily correspond to the worst case considering the proposed distance.

In spite of having many cases in which the Euclidean distance and the proposed distance are equivalent, there are many cases in which they are not correlated. Comparing two independent Indoor Positioning Systems might report different conclusions depending on the metric used to compare them. As shown in Table I, the kNN algorithm with a $k$ value of 13 reported better results than the kNN algorithm with a $k$ value of 15 based on the Euclidean distance results. However, the results obtained with a $k$ value of 15 were better based on the proposed metric. This was just an example of how selecting a metric makes the difference in a comparative study.

B. Suggestion on the Usage of the Proposed Distance Method

The previous subsection presented the benefits of using the proposed method for gaining a more realistic positioning error distance estimation. This subsection presents practical drawbacks of the proposed method in comparison to the Euclidean method. Suggestions on the contexts where it should be used are also provided.

Although there are libraries that notably eases the method implementation [26], [30], the simplicity of the Euclidean distance method implementation is unmatched. The proposed method expects a polygonal representation of the floor plans and obstacles. If they are available through a design or GIS software, the polygons, expressed as a collection of ordered vertices, can be directly obtained. If they are available as images, a previous processing that may include, e.g., polygon filling and edge/contours detection techniques is required. The Euclidean method does not require environment information. Regarding computational cost, the Euclidean method has a running time of $O(n)$, where $n$ is the number of estimation points. In the case of the proposed method, for a precomputed visibility graph, the time complexity of finding the paths is bounded by $O(nv^2)$, where $v$ is the number of vertices representing the polygonal environment. Therefore, the size of the target environment does not influence the running time, but the complexity of its polygonal representation does.

Given the proposed method’s drawbacks in relation to the Euclidean method, its usage is preferred in:

- Indoor positioning benchmarking platforms and competitions, where the effort of setting the method up can benefit numerous evaluations. In particular, when the best indoor positioning systems are reporting very similar accuracy.
- When the building, floor or room detection rates report many significant errors. For example, in the IPIN 2015 competition [22] using the UJIIndoorLoc database [32], the competitors’ floor detection rates were between 86.93% and 96.25%, and one of them had a building detection rate of 98.65%. In those cases, the floor and building penalties were 4 m. and 50 m., which are not in-line to the perceived error by the user.

VI. Conclusions and Future Work

This paper has presented a new approach for the positioning error calculation. The approach computes the path that a person may follow when going from the location estimated by an IPS to the actual location. The proposed method uses the polygonal representation of floor plans and obstacles and inter-floor traversing ways information as inputs. The paths determination uses the offsetting technique and the path-finding technique based on visibility-graphs in combination with a heuristic for inter-floor ways usage. The performed experiments, apart from demonstrating the feasibility of the proposed method usage, showed that (1) the Euclidean distance underestimates the actual distance from an IPS user perspective (i.e., the errors reported by an IPS are higher than expected under the subjective user’s point of view), (2) the actual distance cannot be represented by adding a fixed amount to Euclidean distance, and (3) the penalty scheme used in some research works for floor misidentification is a reasonable choice but it might not be a fair representation of the actual error distance. Although the complexity of using the Euclidean distance as error distance in a positioning metric has no rival, the proposed one is more in-line to the subjective error perceived by users of pedestrian navigation.
Further studies may improve this paper’s work in several directions. They may cover building misidentification cases and contexts with lower success ratios of room, floor, and building detection than the one presented in this paper. They may also detail a method that takes floor plans given as images and obtains polygonal representations with a small number of vertices. Moreover, we will study how to represent the floor change in those buildings with special features. We will also determine whether it is fair for an IPS evaluation to include the change in those buildings with special features. We will also.

ACKNOWLEDGMENT

Germán M. Mendoza-Silva gratefully acknowledges funding from grant PREDOC/2016/55 by Universitat Jaume I.

REFERENCES


