Human Motion Likelihood Representation Map-Aided PDR Particle Filter

Mohamed Anis Ghaoui, Bastien Vincke, Roger Reynaud

Human Motion Likelihood Representation Map-aided PDR Particle Filter

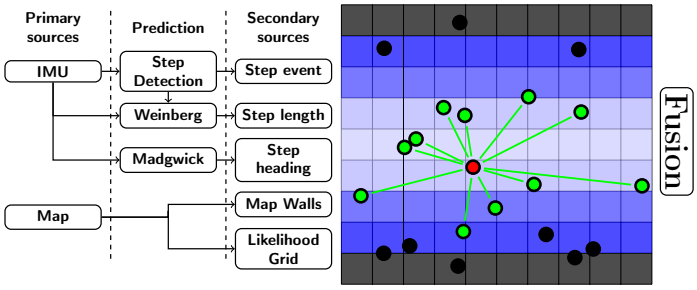
Mohamed Anis Ghaoui*, Bastien Vincke, and Roger Reynaud ‡

15-sept-2022

Abstract

Indoor localization systems are seeing increasing demand. Those for pedestrians are receiving a particular focus. Some of these systems leverage Inertial Measurement Unit (IMU) data collected from a device worn by the pedestrian. The IMU data are used to predict and estimate the pedestrian's location. This paper proposes a system based on a Pedestrian Dead Reckoning (PDR) and Particle Filter (PF) with human motion likelihood grid and floor map filtering. We set an evaluation method by creating pedestrian ground truth landmarks and by measuring statistical properties at these landmarks allowing the comparison to similar techniques. The algorithms, implementation, landmarks, and data used for the experiments of this paper are available as free Open Source.

Keywords: Indoor Localization, Pedestrian Dead Reckoning, Particle Filter, Floor Map, Human Motion, Likelihood Grid.



1 Introduction

Localization is one of the most desired services that has been developing over the last decades due to the increasing demand of Location-Based Services [1, 2]. Outdoor localization has reached a satisfying level of maturity and consumer-ready solutions exist, thanks to satellite-based solutions like Global Navigation Satellite System. Indoor localization techniques see efforts to enhance their capabilities in terms of accuracy, uncertainty, and price. Since satellite signals cannot be received in indoor environments, other ways to locate a device were sought. There are numerous techniques that deploy an infrastructure using various technologies such as WiFi [3], Bluetooth [4], UWB [3], RFID [5] or vision-based [6]. These techniques can have prohibitive deployment and energy costs and are often considered intrusive [7]. Others worked on self-contained techniques that use embedded sensors when using wearable embedded sensors [8].

A phone is the typical consumer device that is ubiquitous and always carries an Inertial Measurement Unit (IMU) which enables a self-contained navigation technique: Inertial Navigation System (INS). INSs are attractive because IMUs are reliable, cheap, omnipresent, have a high data rate and low power.

Pedestrian Dead-Reckoning (PDR) seems a promising technique. It leverages accelerometer, gyroscope, and magnetometer data to reconstruct the displacement of each step of a pedestrian. However, in an indoor environment, the magnetometer data is often considered partially or totally unusable [9], making the PDR a relative positioning system to its starting position.

The Particle Filter (PF) models the dynamics of the localization problem by representing the hypotheses space with individual particles and by adding randomness during particles' propagation. This randomness is related to the IMU sensors noises. The PF is particularly interesting in the case of indoor localization because of the ability to fuse a new type of information [10]: the indoor map. Indoor maps are mathematical objects that represent some geometrical and/or semantic information to describe the indoor environment. They can be represented in various forms and levels of details such as floor maps, grids, and spatial graphs [11]. In this paper, we pick up on the works of [12], analyze it and propose an enhancement. We are particularly interested in leveraging the human motion likelihood extracted from a map in the form of a grid. We gradually explain the steps that led to the final result and the impact of this behavior described by the grid. This grid is different from an occupancy grid used in probabilistic robotic localization. Our grid is a mapping of the likelihood of a step performed by the human where each particle of the PF will draw its weight from. While occupancy grids aim to represent the environment by an evenly spaced field random variable and to indicate the probability of a cell being occupied.

This paper is organized as follows: Section II presents the main contributions of this paper. Section III summarizes the state of the art related to our problem and discusses the works of [12]. Section IV sets the mathematical foundations of the problem. Section V is a description of the system. Section VI presents the experiments. Finally, Section VII analyzes the obtained results and provides a discussion surrounding them.

*mohamed-anis.ghaoui@universite-paris-saclay.fr

†SATIE—CNRS UMR 8029, Univ. Paris-Saclay ,91405 Orsay CEDEX, France

‡Manuscript received DATE; revised DATE.

2 Contributions

This paper presents the following contributions:

- An extension of [12] via a robust particle filtering based on floor map usage, and on a particle weighting via a human motion likelihood representation of the indoor environment.
- An evaluation method of the proposed system in terms of accuracy, robustness, and integrity.
- A dataset with IMU (accelerometers and gyroscopes), landmarks data, and the floor map of the building.
- A Free Open Source implementation of the framework.[1]

3 Related works

3.1 Embedded Sensors Indoor localization

Murata [13] identified six main challenges related to indoor localization which are often overlooked. We noted that researchers worked on each point with different approaches. Localization needs to be continuous and accurate in order to offer reliable guidance. Their preliminary tests consider that two meters accuracy is the limit for their indoor application. Probabilistic approaches are designed to consider the sensors' noise but often incorporate correction techniques that may produce a jump in position estimation which is highly undesirable [14].

The techniques relying on understanding human motion [15] need to be flexible in order to adapt to the individual's distinctive walking pattern. Such adaptivity is far from reach, and thus motion models are approximated. It is often mentioned that humans have multiple motion patterns among: forward, backward, sidestep, turn, ... and each of them presents specific sensor signatures. That is why human motion recognition is problematic in itself [16, 17]. In this paper, the user is considered to be a forward walker.

3.2 Pedestrian Dead-Reckoning and Particle Filter

Pedestrian Dead-Reckoning (PDR) [18] is a technique that combines data from gyroscopes, accelerometers, and magnetometers to reconstruct a human walking motion. There are two main variations [19]: the strap-down approach and the step-and-heading approach. The latter interests us because it sequentially predicts the current pedestrian position by computing the displacement relative to the previous position. This is done by predicting PDR's three main components: step detection, step length, body heading.

[20] deployed an adaptative step detection and counting using accelerometers and gyroscopes signals. By placing an adaptive low-pass filter and a peaks/valleys detection in the signals. All thresholds used in their work are computed adaptively.

[1] `https://github.com/anisghaoui/humolire`

Both [12] and [21] built a PDR-PF system leveraging IMUs data: accelerators and gyroscopes, and additionally the magnetometer for [21]. Even though their prediction functions differ, they produce the same observation data: step length and step heading. To build on the work of [12], we will need to use the same or equivalent prediction functions presented in section 5.1. For instance, both *Lu* [12] and *Zhao* [21] used the Madgwick filter [22] for angle prediction, which is known to be fast and robust to sensor error. It also deduces the absolute roll and pitch angles from the vertical acceleration, leaving only the yaw relative to its starting angle if the magnetometer is not used. [12] used double accelerometer integration for step length prediction. While [21] said: "The human gait is modelled by an inverted pendulum of a knee-less biped on the sagittal plane". [23] explains the inverted pendulum model even further which is very similar to the Weinberg approximation [24, 25]. We will use the Madgwick filter for heading prediction without magnetometer data and Weinberg Method for step prediction.

Due to drift phenomena associated with noisy gyroscope data, localization with PDR becomes erroneous. Thus, the addition of a PF on top is often done as seen in the IPIN 2018 [26]. This is understandable to leverage the discrete prediction of the PDR associated with the discrete multi-hypothesis capabilities that the PF performs. Previously cited papers do this to achieve better localization [27–29] or to use of a kind of map [12, 21].

The PF is especially interesting since [12, 21, 27–29] all differed in the number of particles used for their PF, respectively 256, 500, 1000, 1000, 1000 particles. This indicates that, for what appears to be the same problem, either not every approach is very efficient with its particles use, or the problem dealt with has a hidden layer of complexity.

Researchers have tried machine learning by feeding IMU data to their models and by training them to perform a robust double integral. In their previous work, [30] used a linear model to perform a robust integration. But in [31], they trained a long short-term memory neural network to perform the robust double integration. The aforementioned neural network named RoNIN was trained on the RoNIN dataset, the most significant IMU dataset (40 hours of records). They then compared their results to the well-known Oxford model IONet [32] trained on the OxIOD IMU dataset [33]. We will also compare our system to RoNIN, because for the same inputs, our results are clearly better as shown in Fig. 15.

3.3 Map Usage

According to [11], maps come in different types, embedded pieces of information, and levels of detail: floor-plan, Building Information Modelling, point cloud, textured mesh, and even hand-drawn. [34] and [35] used incomplete or inaccurate prior knowledge, i.e. maps, to obtain better results in their applications. [28] used a floor plan map to eliminate candidate particles that land behind an impassable wall.

[36] considered the indoor environment as a graph where every corridor is an edge and intersection is a vertex. Then, they tracked the gyroscope for 90° rotation signature and

updated their position in the graph by observing the walked distance.

For estimation, we will stick to the usage of a map-assisted PF. Both [12, 21] used the floor map of the building where the experiment was conducted. They applied particles wall-filtering via the floor map where an updated particle that goes into an inaccessible area would be eliminated; else, it survives. The major flaw in their systems is that their particles weights are binary. Thus, they only use the PF as a multi-hypotheses propagator rather than a complete probabilistic estimation filter. This is sensible since the wall-filtering is already efficient at distinguishing particles that perform an impossible physical movement. However, this also means that anything that is not an impossible physical location is plausible. Which, in our view, is flawed. We believe that humans in an indoor space tend to move according to a motion likelihood related to the indoor environment configuration. We chose to represent this behavior as a grid in section 5.2.2.

4 Problem statement and Methodology

4.1 Mathematical Model

Both [12] and [21] processed the floor transition via barometer tracking. In this paper, we will consider working only on a single floor. This means locating on a 2D plan the phone that a pedestrian is carrying in terms of position (x, y) and heading θ. Thus, the device is represented by the state vector $\vec{P_t}$ **at each step** of timestamp t:

$$\vec{P_t} = [x_t \quad y_t \quad \theta_t]^T \quad | \quad \vec{P_t} \in \mathbb{R}^3 \tag{1}$$

A data collection system on the phone is needed to be able to acquire the accelerometer and gyroscope data. From this data, we produce an observation vector $\vec{U_t}$ described in section 5.1:

$$\vec{U_t} = [dx_t \quad dy_t \quad d\theta_t]^T \quad | \quad \vec{U_t} \in \mathbb{R}^3 \tag{2}$$

Where dx_t, dy_t are the displacements along the x and y axes, respectively, and $d\theta_t$ is the change in heading between two consecutive steps. Then, the displacement of the pedestrian is deduced by observing the accelerometer and gyroscope data at each step via the following equations:

$$dx_t = SL_t \times \cos(\theta_t) \qquad dy_t = SL_t \times \sin(\theta_t) \tag{3}$$

Where SL_t is the step length at step index t. θ and SL are obtained via the predictions explained in Section 5.1. The result is then accumulated over the previous position to compute the new position:

$$\vec{P_t} = \vec{P}_{t-1} + \vec{U_t} \tag{4}$$

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} dx_t \\ dy_t \\ d\theta_t \end{bmatrix} \tag{5}$$

with $x_0 = X_0,\ y_0 = Y_0,\ \theta_0 = \Theta_0$ as initial conditions.

The predictions of the PDR are then sequentially fed to the PF to estimate the next position. The PF uses multiple particles representing possible trajectories that the pedestrian could have stepped into [37]. Each particle is represented by a state vector and weight: $\vec{P}_t^i, w_t^i$ (ith particle at t time index). The PF adds a noise vector $\vec{N}_i^t$ to both the displacement (dx_t^i, dy_t^i) and differential heading $d\theta_t^i$. Equation (5) becomes:

$$\vec{P}_t^i = \vec{P}_{t-1}^i + \vec{U}_t^i + \vec{N}_t^i \tag{6}$$

$$with\ \vec{N}_t^i = \begin{bmatrix} \delta_t^i \cos(\theta_t^i) & \delta_t^i \sin(\theta_t^i) & \alpha_t^i \end{bmatrix}^T \tag{7}$$

Where δ_t^i and α_t^i are the independent random variables respectively drawn from Gaussian distributions $\mathcal{N}_\delta$ and $\mathcal{N}_\alpha$ with parameters $\mu_\delta, \sigma_\delta$ and $\mu_\alpha, \sigma_\alpha$ defining them respectively (their empirical values are given in section 7.1). Replacing equations (3) and (5) in equation (6) develops into equation (8).

$$\begin{cases} \begin{bmatrix} x_t^i \\ y_t^i \\ \theta_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1}^i \\ y_{t-1}^i \\ \theta_{t-1}^i \end{bmatrix} + \begin{bmatrix} SL_t^i \times \cos(\theta_t^i) \\ SL_t^i \times \sin(\theta_t^i) \\ d\theta_t^i \end{bmatrix} + \begin{bmatrix} \delta_t^i \times \cos(\theta_t^i) \\ \delta_t^i \times \sin(\theta_t^i) \\ \alpha_t^i \end{bmatrix} \\ w_t^i = w_{t-1}^i \times \mathcal{L}(x_t^i, y_t^i) \end{cases} \tag{8}$$

A weight is applied to each particle's state vector. The weight allows to filter some particles according to their likelihood $\mathcal{L}$ which is a function of the particle's location is: (x_t^i, y_t^i).

We compute the particles' weighed average $\bar{P}_t$ as specified in equation (9) to choose the pedestrian's new position. This presents an edge case where the $\bar{P}_t$ may land in a forbidden area.

$$\bar{P}_t = \left[\bar{x}_t = \sum_i^N w_t^i \times x_t^i, \quad \bar{y}_t = \sum_i^N w_t^i \times y_t^i, \quad \bar{\theta}_t = \sum_i^N w_t^i \times \theta_t^i \right]^T \tag{9}$$

In the Fig. 1, we noticed that $\bar{P}_t$ (green particle) might land inside/behind the wall, which is physically impossible. To counter this edge case, after computing $\bar{P}_t$, we choose the closest valid particle among all particles. This ensures that $\bar{P}_t$ is valid without having to perform a domain validity analysis. To our experience, this method does not degrade the estimation's performance and is deemed necessary.

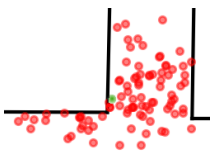


Figure 1: Particles around the corner.

This will produce the successive positions that a pedestrian walked for each step. It effectively means that our system is **Event Triggered** by the steps of the user. Once the positions are obtained, we can compute metrics to evaluate the impact of different parameters.

4.2 Coordinate Systems

We define three coordinate systems: Global (GCS), User (UCS), and Device (DCS) according to Fig. 2. We make assumptions about the pedestrian verticality to be the same as the building they are in (z_g and z_u are the same). So, only

a translation $T_{u\to g}$ and rotation $R_{u\to g}$ around the heading θ are needed to express the change of system. The DCS is presented relative to the user carrying the device. We suppose the user to hold the device horizontally as explained in section 6. This implies that we suppose both $\psi \ll$ and $\phi \ll$ meaning that only a translation $T_{d\to u}$ (corresponding the user's arm) and rotation $R_{d\to u}$ around α (gyroscope z-axis) are needed to express the change of system.

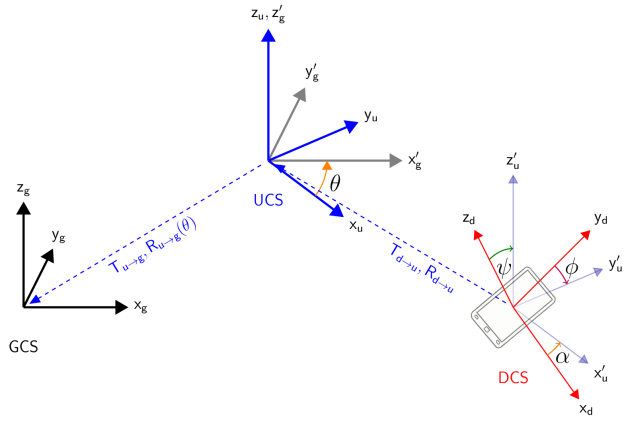


Figure 2: Coordinate systems changes.

5 System description

The system is split into two main parts: the prediction and estimation functions. The former are a set of tools that transform the raw synchronized accelerometer and gyroscope data into step events t and observation vector $\vec{U}_t$. The estimation functions are the set of tools that leverage the prediction result by deploying probabilistic mechanisms. The proposed system is offline processing. The processing time of our system is elaborated in section 7.5.

5.1 Prediction

5.1.1 Step detection

Step detection techniques have been studied and tested [20]. Zero-crossing peak valley detection is a technique where the centered acceleration norm is computed from the 3-axis accelerometer A_{norm}. Then, it is filtered via a low-pass second-order Butterworth filter with a cut-off frequency of 2 Hz. Then, it finds the moments where A_{norm} crosses 0, respectively, from negative/positive to positive/negative, noted t_i^+, t_i^- where $i \in \mathbb{N}$. Then, find the peaks/valleys in each interval such as:

$$t_{peak} = \underset{t}{argmax}\ A_{norm}(t) \quad | \quad t \in [t_i^+, t_i^-] \tag{10}$$

$$t_{valley} = \underset{t}{argmin}\ A_{norm}(t) \quad | \quad t \in [t_i^-, t_{i+1}^+] \tag{11}$$

Since each step is defined as a peak then a valley, every pair (t_{peak}, t_{valley}) defines a step interval. Finally, only peaks/valleys above/below a manually fixed threshold are kept.

5.1.2 Step length

A well-known method is *Weinberg*'s [24] which considers the step length SL to be proportional to a function of the difference between the peak and valley accelerations:

$$SL = K \times \sqrt[4]{a_{peak} - a_{valley}} \quad | \quad K \in \mathbb{R}^+ \tag{12}$$

Where K is the Weinberg gain proportional to the length of the pedestrian's leg.

5.1.3 Heading

The heading is predicted by the *Madgwick* filter [22] that uses gyroscope and accelerometer data to estimate the attitude's quaternion at each sample. We convert the quaternions to Euler angles. Moreover, using the coordinate system changes explained in section 4.2, we only keep the heading value, θ.

The *Madgwick* filter returns a continuous array of angles. So, we subsample it by only picking the steps events from the continuous heading prediction. Then, it is differentiated to obtain the change in heading $d\theta_t$.

$$d\theta_t = \theta_t - \theta_{t-1} \tag{13}$$

It is important to note that we consider choosing the heading of a step closer to when the foot landed (closer to t_{valley}).

5.2 Estimation

5.2.1 Particle Filter

A PF mainly consists of 4 processes [37] : **Initialization, Prediction, Update and Resampling**. We will see and discuss these below.

Initialization We select a starting region by randomly normally generating particles around the starting position and a heading that is the starting heading with some noise added to it. So, there is already some incertitude around the pedestrian's starting position for robustness's sake. Each particle weighs $1/N$.

Prediction The PF predicts the state of particles every time a step is detected via the state model from equation (8). It receives two informations from the prediction stage (Section 5.1), the step length SL and the differential heading $d\theta_t$. Every particle computes its new heading and position with angle and radius additive Gaussian noises according to equation (8).

Update In the case of indoor localization, providing a floor map allows the introduction of filtering as [12] and [28] did. The updated particles are declared valid if they do not cross a wall. This is a straightforward constraint rule. Nevertheless, it is the most accurate mapping of the physical behavior of the pedestrian by the PF. It is often referred to as **Map Aiding**. The number of particles after this step is lower than the optimal number that the PF is designed for. A floor map of the experiment building is shown in Fig. 3.

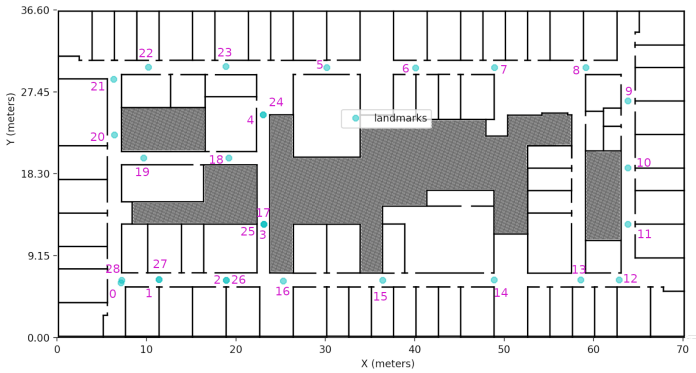


Figure 3: A floor-plan of the experiment building: grayed out areas are forbidden. User walks along the landmarks from 0 to 28.

Resampling It removes lighter-weight particles and duplicates the heavier-weight ones to avoid what is known as PF degeneracy. This process is deemed of the utmost importance [38,39] and has been well studied [39–42]. On the other hand, resampling too often presents the risk of obstructing the hypotheses space because the heaviest particles will cluster together and will shadow every other particle (a black-hole-like effect). This is why *Kong* [43] and *Pham* [44] each introduced a criterion to know when resampling is needed. After implementing and testing different resampling algorithms and criteria, we noticed that the behavior of PF was not enhanced nor degraded due to already using *Filtering* and *Regeneration*. We thus abandon resampling in favor of the following one.

Regeneration Although it was not formally named in [12] and [28], it consists of regenerating the missing particles to have the entirety of particles best represent the updated hypotheses space. The optimality zone of the PF is sensitive to the particles count that is deemed necessary for the PF to be able to correctly represent the hypotheses space (see section 7.1). Since *Filtering* deleted some particles, the PF is left with $\tilde{N}$ particles, such as $\tilde{N} \leq N$. We need to regenerate the $N - \tilde{N}$ missing particles. The way of regenerating these missing particles affects how the PF paves the hypotheses space and what it "believes" to be the more likely position. If all particles had the same weight, the particles would regenerate according to a majority vote. But, we propose in section 5.2.2 a method to weigh our particles and to introduce *a priori likelihood*.

The association of *Map Filtering* and *Regeneration* emulates a resampling where the particles landing in inaccessible zones (e.g., behind walls) are eliminated, and only the valid particles will have offspring.

Subset Regeneration: We studied more formally the approach [12] implemented in their open-source code. In order to reach N particles, it considers that the new M particles need to be generated from M randomly selected subsets of $\hat{N}$ particles defined as $\hat{N} = ss \times \tilde{N}$ where $ss \in]0, 1]$ is the subset proportional size. Fig. 4 shows a visualization of algorithm 1.

Our contribution is to randomly draw new particles from

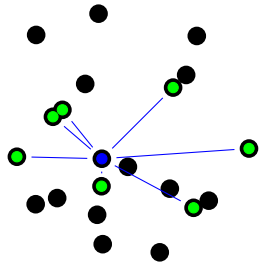


Figure 4: $\hat{N}$ particles subset regeneration are green.

Algorithm 1: Particles subset regeneration function.

```
   Input: particles, subset size
   Output: valid particles
1  mean particles = ∅
2  candidates = ∅
3  for i ← 1 to N − Ñ by 1 do
       // no repetition in selection
4      subset = random pick (particles, subset size)
       // average θ, x , y and weight
5      mean particles.append(average(subset))
       // generate the new particle from the subset's
          statistical properties
6      N_θ, N_x, N_y, N_w = statistical properties(subset)
7      candidates.append(particle(N_θ, N_x, N_y, N_w))
   // check if the candidates do not cross walls
8  valid = filter(candidates, mean particles)
9  return valid
```

the `statistical properties` (mean and variance) of the `subset` and to verify that the new proposed particles do not cross a wall of the map. This acts as a random re-sampling based on the paved hypotheses space where each randomly selected subset is a pavement from where the new particle is picked. It emulates a re-sampling algorithm but bases itself on both statistical properties of the particles and physical properties of the indoor environment. After our tests, we empirically selected $ss = 5\%$ and preferably $\tilde{N} \times ss \leq 25$.

5.2.2 Human Motion Likelihood grid

Our final contribution is the creation of a grid that represents the human motion likelihood. To represent the human motion inside a building on a floor map, we set an *a priori* about the likelihood during motion associated with the different indoor dispositions and configurations. We suppose that a human is more *likely* to move in the center of a corridor rather than hugging the wall, walk through the center of a door rather than the sides.

We convert the floor map of length L and width W to a grid: $G \in \mathbb{R}^{l \times w}$ of length $l = L/scale$ and width $w = W/scale$, with each cell $g \in G \mid g \in [0, 1]$. In our case, $scale = 5\ cm$. So the grid is composed of $5 \times 5\ cm^2$ cells. Then, we compute the distance from each cell to the closest wall.

The resulting distances map is converted into a likelihood grid via a membership function f defined as $f : \mathbb{N}^{l \times w} \rightarrow [0, 1]^{l \times w}$. The choice of the mapping functions f, shown in Fig. 5 as an example, affects how the grid constrains the particles during their propagation. In every case, the

function f creates a centering constraint as it pushes the particles into the center of corridor. It has 2 parameters: **min range**, that determines the end of the forbidden area; and **max range**, that determines the end of the transition area.

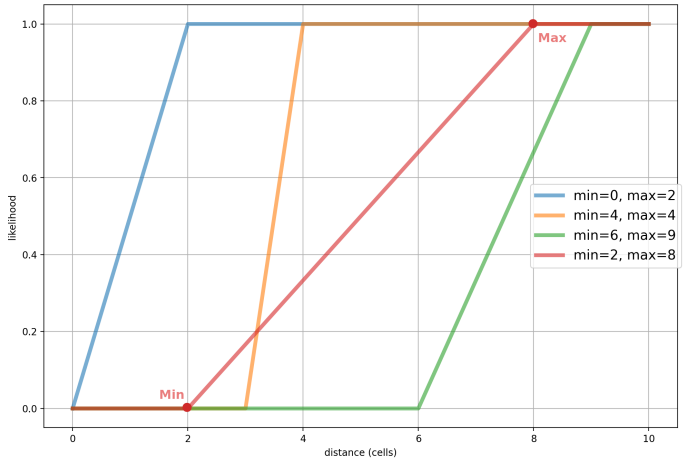


Figure 5: Example of distance to likelihood functions $f(d) = \mathcal{L}$. .

The resulting grid is shown in Fig. 6 and represents our *a priori* on human motion likelihood. A zoomed n example of a corridor grid is given in Fig. 7a. Black cells are walls with an associated likelihood of 0. Fairer cells have a higher likelihood of a step occurring on them and thus associate a more important weight to the particles landing on them.

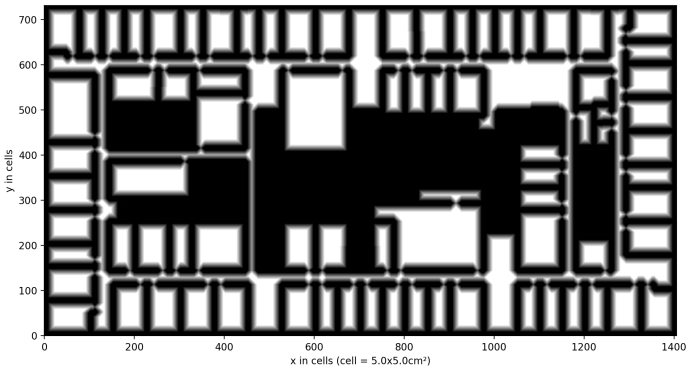
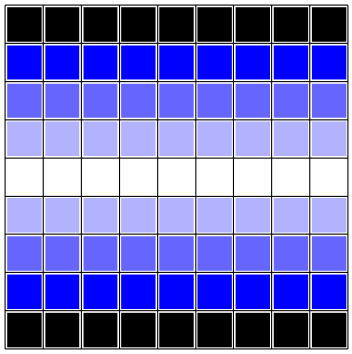


Figure 6: Likelihood grid built upon the floor map in Fig. 3: darker areas are less likely.

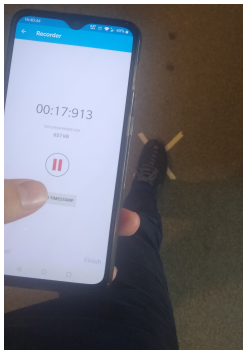
6 Setup and experiment

In order to test different configurations of the system's parameters, we need to set up an experiment that reflects a realistic human motion in indoor space. We use a OnePlus 6T phone to acquire `LSM6DS3` IMU data (accelerometer + gyroscope. The summary of this IMU's characteristics[2] in Table 1 shows that the IMU uses 16 signed bits to code

[2] `https://eu.mouser.com/datasheet/2/389/dm00133076-1798402.pdf`



(a) Example of corridor grid.



(b) Walking over a cross and pressing the timestamp button.

Figure 7: Left: grid concept. Right: experiment setup.

its data. The Linear acceleration measurement range is selected because we noticed that the accelerometer data never surpassed 1.5g. Also, human rotation didn't exceed $180°/s$. So, an Angular rate measurement range of $\pm 250°/s$ seams reasonable.

Parameter	values	unit
Linear acceleration measurement range	± 2	g
Linear acceleration sensitivity at± 2g	0.061	mg/LSB
Angular rate measurement range	± 250	$°/s$
Angular rate sensitivity at ± 250 $°/s$	8.75	m$°/s$/LSB

Table 1: Summary of `LSM6DS3` IMU.

The user holds the phone horizontally, the USB port towards their chest, and the back camera looks at the ground. This ensures that the coordinate system does not change during the experiment and considers Android's Z rotation axis as the pedestrian heading θ.

To produce relevant metrics to measure the performance, we need to set up a pedestrian ground truth. This ground truth must have a satisfying spatial and temporal accuracy and uncertainty. So, we put landmarks in the form of crosses on the ground and ask the pedestrian to step on them and simultaneously press a timestamp button on the phone. This procedure is illustrated in Fig. 7b. We use a high-precision laser telemeter to locate the crosses relative to the building's walls. If synchronization is done correctly, the moment the user presses the button should correspond to when they step on the cross. This reduces the ground truth effort into a synchronization problem. Experiments showed that this evaluation method was satisfactory as we can count how many timestamps the user produced and invalidate the run if any are missing. We also have a worst case error of a step around the cross compared to when the user pressed the timestamp button (the user pressed the button a step before, while on, or after the cross).

While looking for an acquisition tool, we settled for the INRIA-Tyrex team's open-source app: SensLogs [45]. The app allows for a plethora of sensor acquisition and different sampling frequencies. We select the maximum frequency our phone allows: 400 Hz. The temporal synchronization

is guaranteed because all the sensors are timestamped with the same clock source. Thanks to this, the tedious work of synchronization is abstracted, and we have an all-ready solution.

We define a trajectory to have various indoor cases as shown in Fig. 3. The trajectories ought not to loop around a single place many times as this biases when performing map aiding. This happens if the first loop is correctly performed, then the others are quasi-automatically correct, which makes a system looks more permanent than it is.

We previously mentioned that our system had many parameters. However, two of them are particular because they are specifically related to the user holding the phone: the Weinberg gain from section 5.1.2 and the acceleration threshold from section 5.1.1. We tested our system's invariance to its user by having four different users performing the same trajectory. Then, we fit the aforementioned two parameters to each user.

To measure our system's performance at each user-made timestamp, we compute the **Euclidean** distance from the average weighed position $\bar{P}_t$ to the landmark l_t for landmarks count, K:

$$E = \sum_{t=1}^{K} \frac{E_t}{K} \mid E_t = \sqrt{\left(l_t - \bar{P}_t\right)^2} \tag{14}$$

And the **Mahalanobis** distance of the landmark l_t to the N particles' positions $\mathbf{P}_t \in \mathbb{R}^{N\times 2}$ with weights $\mathbf{W}_t \in \mathbb{R}^N$ and weighed average $\bar{P}_t$ presented in equation (9).

$$M = \sum_{t=1}^{K} \frac{M_t}{K} \mid M_t = (l_t - \bar{P}_t) \times \mathbf{S}_t^{-1} \times (l_t - \bar{P}_t)^T \tag{15}$$

With $\mathbf{S} \in \mathbb{R}^{2\times 2}$ being the covariance matrix of the particles where each particle has two components: $P_t = (x_t, y_t)$:

$$\mathbf{S}_t = \mathbb{E}\left[\left(\mathbf{W}_t^T\mathbf{P}_t - \bar{P}_t\right) \times \left(\mathbf{W}_t^T\mathbf{P}_t - \bar{P}_t\right)^T\right] \tag{16}$$

It is important to notice that the weighted average position $\bar{P}_t$ differs from the one presented in section 4.1. It is not the closest valid particle but the statistically weighted average of the particles. This makes our measure take into account the worst-case scenario for the PF.

The parametric study must be done over several runs over the same data due to the random nature of the Monte Carlo sequence. Then, these runs are averaged to obtain the curves shown below. We performed an average over ten runs. The success rate of a trajectory is defined as $^{completed\ runs}/_{total\ runs}$ with a run considered *completed* if it has as many user timestamps as landmarks there are, and a return to initial position error defined as $E_{final} \leq 2.5m$ inspired by [21]. An example of a run is given in Fig. 8.

7 Results, analysis, and discussions

We are interested in showing how the integrity of the system evolves and the impact of the human motion likelihood grid that we conceptualized in section 5.2.2. The available data and code can recreate the results and cover how to obtain them in more details. The first step is to determine Weinberg's gain and acceleration threshold experimentally. This is simply done by making users walk a measured distance, count their steps, and then empirically reverse engineer each parameter for each user.

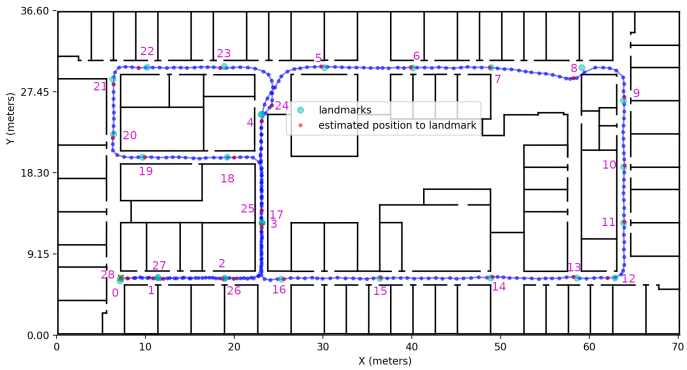


Figure 8: An estimated trajectory of a run. Red dots are the steps when the user pressed on the timestamp in Fig. 7b.

7.1 Optimal particles count

In order to discuss results, we first need to determine how many particles our system requires to model the hypotheses space correctly. For that, we proceed to compute both average E and M for the same trajectory, same data, and same user for a given number of particles: N. The Fig. 9 shows that the system converges for at least 200 particles. Counts beyond this number do not significantly change either metric but increase the necessary computing resources. We will be using 200 particles for the remaining experiments.

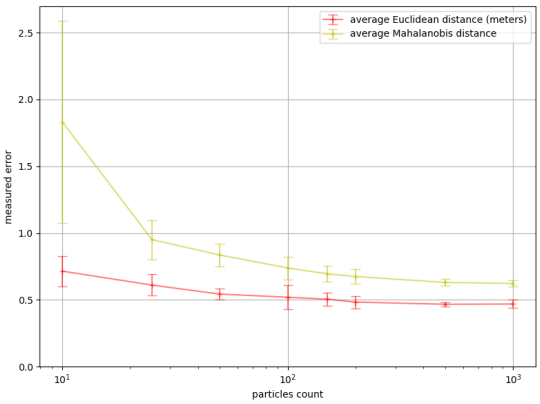


Figure 9: Measured distances as a function of particles count.

We also give the particles' propagation noise used in equation (7) in table 2.

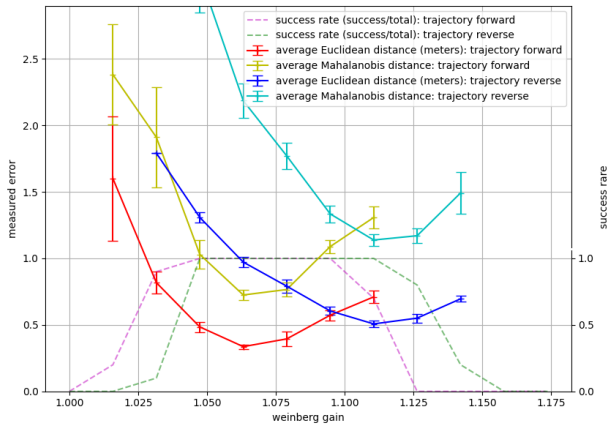


Figure 10: Measured distances as a function of Weinberg gain.

Gaussian distribution $\mathcal{N}$	mean (μ)	standard deviation (σ)
Step noise δ [meters]	0	0.15
Angle noise α [rad]	0	$\pi/45$

Table 2: PF's propagation noise distributions.

7.2 Weinberg's functional interval

Starting from the empirical Weinberg's gain value from section 7, we figure out the optimal value for a user. The Fig. 10 shows that both E and M (red and yellow curves) reach a minimum value close to the empirical Weinberg gain's value. Since this value is never supposed to change for a user, we make them walk the exact trajectory but in reverse order to obtain the blue and cyan curves. We also compute the success rate of each trajectory to understand the impact of this parameter.

We first notice that the two trajectories do not have optimal Weinberg gain value. Nevertheless, an interval exists where the success rates overlap and keep the errors to a minimal value. Table 3 shows the comparison between the empirically found and statistically computed Weinberg gain's value. This implies that the step noise δ introduced in equation (6) corrects the errors in step length predictions.

User 1's Weinberg gain	empirical	statistical
forward	1.07	$[1.05, 1.09]$
reverse	1.07	$[1.05, 1.110]$

Table 3: Weinberg gain: empirical vs statistical values.

7.3 Grid effect and integrity

By measuring the signed error in the position of all particles relative to a landmark: e_l and its standard deviation σ_{e_l}, we can observe the effect of the grid on the distribution of particles throughout the whole experiment. The errors in X and Y, e_{x_l} and e_{y_x} are respectively surrounded by $\pm 3\sigma_{e_x}$ and $\pm 3\sigma_{e_y}$. We compare the performance of the system for the same data and parameters with a scenario where the **likelihood grid** is **enabled** and **disabled** on an average of 10 runs each time. This is equivalent to a comparison between [12] and our system.

In order to do this comparison, we need to find the optimal values of the two parameters **min range** and **max range** from section 5.2.2. To find the optimal values of both parameters, we perform a parametric study on the grid. The goal is to maximise the score of the grid version vs no-grid version, defined as:

$$score = \frac{1}{K}\sum_{t=1}^{K} \sigma_{e_{l_t}\ no-grid} - \sigma_{e_{l_t}\ grid} \tag{17}$$

Equation (17) is analogous to computing the area difference of incertitude ellipsoids at each key-point between the no grid and grid version in Fig. 14 and 13. In addition, equation (17) allows a balance of the centering constraint from section 5.2.2. We make sure that the system keeps its **integrity** by ensuring that the ground truth is still bounded by the error i.e. that the 0 y-axis is included in the purple corridor for Fig. 11. This is expressed as:

$$score_t = \begin{cases} 0 & if \quad (e_t + \sigma_t) \times (e_t - \sigma_t) \geq 0 \\ score_t & else \end{cases} ; \tag{18}$$

We compute the score for each combination of the two parameters according to these two equations. We obtain the heat map presented in Fig. 12. We first notice that, if **min** > **max**, the system cannot work because the forbidden area includes the allowed area, and all the particles receive a weight of 0. Secondly, the blue area shows that the score was at its maximum for those corresponding values of parameters. Thirdly, We notice that beyond a min value of 14 cells (70 cm), the forbidden area from a wall is too big and cuts the path for the particles. This is explainable because, in our path, some corridors are 150 cm wide. This means that we would have a forbidden area of 70 cm on each side with an error due to the discretization of the map into a grid. We will pick: **min range** = 11 and **max range** = 13 (one of the darkest blue cells in the Fig. 12).

In Fig. 11, the 0 on y-axis is always included between $\pm 3\sigma$ for both version. This means that in 99.7% of cases, the system includes the ground truth position inside the uncertainty ellipsoid. This expresses its integrity, i.e., how faithfully it can always contain the correct positive in its hypotheses space.

Another aspect to note is the impact of the usage of the grid. We do notice that enabling the grid (with both its parameters, see section 5.2.2) makes the area imprisoned between $\pm 3\sigma$ narrower than without the likelihood grid. This means that the likelihood grid, in that configuration, compresses the particles to the center of the corridor, which means that, on average more particles are closer to the correct position (landmark).

If we compare the ellipsoid of the particles' distribution at each landmark for an enabled and disabled grid as shown in Fig. 14. We notice that the ellipsoids are constrained by the walls and rarely go through them when the grid is enabled. This shows that the grid has an ameliorative effect

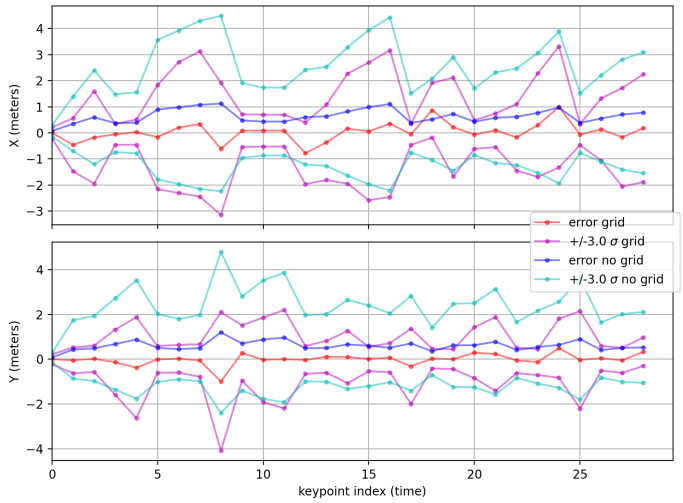


Figure 11: Measured signed errors on each key-points.

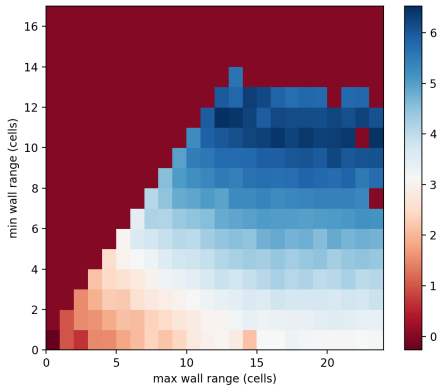


Figure 12: Score heat map for both grid parameters.

on the localization for these indoor configurations. The colors of the particles indicate their weights. In the case of the disabled grid, all particles have the same weight (color) $1/N$.

The system's behavior is interesting at some particular points that we encounter in the building during the walk, namely: 8, 12, and 24. By observing Fig. 8, 11 and 14, we notice that landmark 8 comes after a walk of $\approx 10m$ in a semi-open space. Thus, non-constrained particles tend to spread and uniformly pave the hypotheses spaces. Landmark 12 shows an interesting compression of the particles' ellipsoid. It comes after a 90° turn with some semi-open space, yet the grid version achieves an important compression and has no particles in the adjacent rooms like landmark 9 in Fig. 13. Landmark 24 is an edge case as it combines both difficulties of 8 and 12. It comes to a semi-open hyafter a 90° turn. Particles easily spread to fill the space. Then, they get funneled into the tight corridor. This makes the ellipsoid spread to its maximum and then fold back into the corridor. All of these physical phenomenons are correctly displayed by spikes in error e_t and std σ_t on both axis X and Y in the Fig. 11.

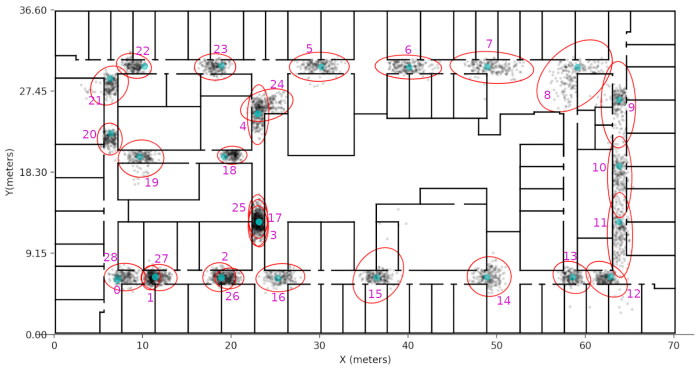


Figure 13: Incertitude ellipsoids with likelihood grid disabled.

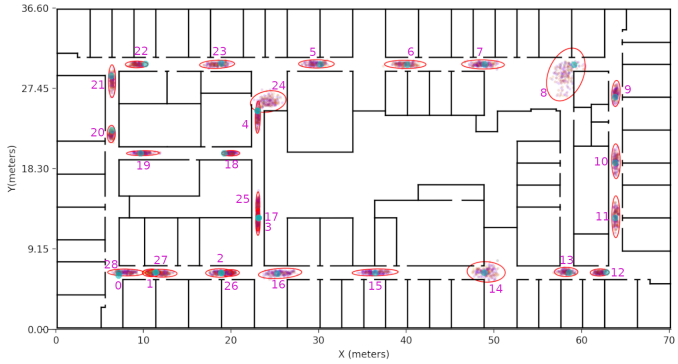


Figure 14: Incertitude ellipsoids with likelihood grid enabled.

7.4 Discussions

7.4.1 Map Filtering and Aiding

[12] deployed a backtracking procedure to ensure that the particle's life was coherent and did not cross walls over the last 20 steps. This seems redundant to us as our map filtering, presented in section 5.2.1, guarantees by construction that a particle dies if it hits a wall. The comparison to [12]'s system is made by comparing the presence and absence of our grid in section 7.3. In Fig. 15, we show the comparison of 3 methods: a fully deterministic PDR (no wall filtering), the prediction of the already trained RoNIN [31] and finally, our system. The resulting trajectory shows that map filtering is mandatory for indoor environments because it maps the physical displacement more faithfully than sheer statistical methods. It is also important to note that RoNIN relies on Android's orientation (Game Rotation Vector) which itself leverages the faulty indoor magnetometer data. To further show the difference between the 3 approaches, Table 4 gives the computed RMSE for each system. We also resolved some edge cases like the weighted average of particles in section 4.1.

System	RMSE[m]
Ours	0.811
RoNIN	7.534
deterministic PDR	4.671

Table 4: RMSE of comparison of Fig.15.

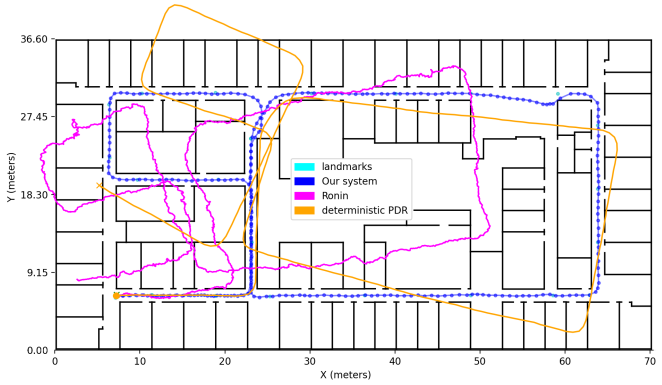


Figure 15: Comparison between 3 different approaches: Our system, RoNIN [31] and a deterministic PDR on the same trajectory.

7.4.2 Human Motion Likelihood Grid

At first, there seemed to be no enhancement in terms of accuracy. However, we showed that the integrity of the system and its ability to map the hypotheses space in the indoor environment more accurately are enhanced. Nevertheless, although the users were told to "walk as they like" and "follow the path of the crosses", they will tend to walk in segments from one landmark to another. Which makes for corridor walks to seem more accurate. However, this isn't true for landmarks 8, 12, 16 and 24 in Fig. 14. We notice that those configurations differ from simple corridors. However, the grid reinforces the system's integrity by preventing the propagation of deceptive particles in the previous steps, making the localization at those difficult spots more reliable. Nevertheless, this comes at some cost of flexibility. The Fig. 12 showed that there is a balance in how constraining the grid is and how relaxed it is. Thus, the choice of the likelihood function presented in section 5.2.2 has a key role.

Our hypothesis remains valid as long as we suppose a single pedestrian is walking alone in a building. It begs how this likelihood evolves if more pedestrians were walking either in the same, opposite direction, or even at a stop.

7.5 Processing Time

Our system is offline processing on CPU: Intel i5-9500(6)@4.400GHz. The floor map from Fig.3 is composed of nearly 300 segments that represent walls. The grid is made of 5x5 cm^2 cells for a total of $1400 \times 720 \approx 10^6$ cells (see Fig.7a), with each cell containing a floating value. This grid is built only once per floor. It is then a read-only memory object.

The particles count was set to be 200 particles in Section7.1. The implementation is done on Python with an average step processing speed of 4.5 steps/s. With the knowledge that a human can walk between 1 to 3 steps a second, the current offline processing is satisfactory even before optimization.

Optimization can be done in several ways in future studies. The most straightforward one is to change the implementation language from an interpreted one to a compiled one (`C++, C#, Java/Kotlin` for Android). This change will reduce memory usage and execution time.

8 Conclusion

This paper presents an indoor localization system based on a phone that the pedestrian carries. We have based our work on [12]. Then, we have detailed the prediction and estimation functions that we set up. Then, we have introduced the human motion likelihood representation as a grid in the PF. This grid allowed to weigh the particles. We have also defined an experiment and evaluation method with reliable ground truth for the pedestrian's indoor movement. We have proceeded to the analysis of the results of the experiments.

Furthermore, we have given a qualitative appreciation for the metrics we have defined. These metrics showed that the integrity of our system is more important with the enabled grid. We compared our system to a variant of [12] and to [31].

In the future, we wish to experiment on multi-agent representation of different behaviors. More semantic elements will be added into the map to enhance indoor localization's integrity and robustness.

9 Data Transparency and Experiment Reproducibility

We declare that the used data is from real-world subjects and was collected with anonymity and integrity. The building's map is delivered as a numerical object with faithful measured dimensions of a frequently busy physical building. Any sensitive pieces of information that are not required for the experiments have been censored without compromising the integrity of the data or its ground truth.

References

[1] F. Zafari, A. Gkelias, and K. K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3):2568–2599, 2019.

[2] Fuqiang Gu, Xuke Hu, Milad Ramezani, Debaditya Acharya, Kourosh Khoshelham, Shahrokh Valaee, and Jianga Shang. Indoor localization improved by spatial context—a survey. *ACM Comput. Surv.*, 52(3), July 2019.

[3] Giuseppe Caso, Mai Le, Luca De Nardis, and Maria-Gabriella Di Benedetto. Performance comparison of wifi and uwb fingerprinting indoor positioning systems. *Technologies*, 6(1):14, Jan 2018.

[4] R. Faragher and R. Harle. Location fingerprinting with bluetooth low energy beacons. *IEEE Journal on Selected Areas in Communications*, 33(11):2418–2428, 2015.

[5] A. Motroni, P. Nepa, V. Magnago, A. Buffi, B. Tellini, D. Fontanelli, and D. Macii. Sar-based indoor localization of uhf-rfid tags via mobile robot. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2018.

[6] Anca Morar, Alin Moldoveanu, Irina Mocanu, Florica Moldoveanu, Ion Emilian Radoi, Victor Asavei, Alexandru Gradinaru, and Alex Butean. A comprehensive survey of indoor localization methods based on computer vision. *Sensors*, 20(9):2641, 2020.

[7] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios. Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Communications Surveys Tutorials*, 19(2):1327–1346, 2017.

[8] Xinyu Hou and Jeroen Bergmann. Pedestrian dead reckoning with wearable sensors: A systematic review. *IEEE Sensors Journal*, 21(1):143–152, 2020.

[9] Guanghui Hu, Weizhi Zhang, Hong Wan, and Xinxin Li. Improving the heading accuracy in indoor pedestrian navigation based on a decision tree and kalman filter. *Sensors*, 20(6):1578, Mar 2020.

[10] Qinglin Tian, Kevin I-Kai Wang, and Zoran Salcic. A low-cost ins and uwb fusion pedestrian tracking system. *IEEE Sensors Journal*, 19(10):3733–3740, 2019.

[11] Jorge Chen and Keith C. Clarke. Indoor cartography. *Cartography and Geographic Information Science*, 47(2):95–109, 2020.

[12] Chuanhua Lu, Hideaki Uchiyama, Diego Thomas, Atsushi Shimada, and Rin-ichiro Taniguchi. Indoor positioning system based on chest-mounted imu. *Sensors*, 19(2):420, Jan 2019.

[13] M. Murata, D. Ahmetovic, D. Sato, H. Takagi, K. M. Kitani, and C. Asakawa. Smartphone-based indoor localization for blind navigation across building complexes. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10, 2018.

[14] Masayuki Murata, Dragan Ahmetovic, Daisuke Sato, Hironobu Takagi, Kris M. Kitani, and Chieko Asakawa. Smartphone-based localization for blind navigation in building-scale indoor environments. *Pervasive and Mobile Computing*, 57:14 – 32, 2019.

[15] Richard Baker, Fabien Leboeuf, Julie Reay, and Morgan Sangeux. *The Conventional Gait Model : The Success and Limitations*, pages 1–19. 05 2017.

[16] Fuquan Zhang, Tsu-Yang Wu, Jeng-Shyang Pan, Gangyi Ding, and Zuoyong Li. Human motion recognition based on svm in vr art media interaction environment. *Human-centric Computing and Information Sciences*, 9(1):1–15, 2019.

[17] Jiayi Lin, Chengming Zou, Long Lan, Shanzhi Gu, and Xinshang An. Deep heading estimation for pedestrian dead reckoning. *Journal of Physics: Conference Series*, 1656:012009, sep 2020.

[18] Hui-Huang Hsu, Jung-Kuei Chang, Wei-Jan Peng, Timothy K Shih, Tun-Wen Pai, and Ka Lok Man. Indoor localization and navigation using smartphone sensory data. *Annals of Operations Research*, 265(2):187–204, 2018.

[19] Zhian Deng, Weijian Si, Zhiyu Qu, Xin Liu, and Zhenyu Na. Heading estimation fusing inertial sensors and landmarks for indoor navigation using a smartphone in the pocket. *EURASIP Journal on Wireless Communications and Networking*, 2017(1):160, 2017.

[20] Maan Khedr and Nasser El-Sheimy. A smartphone step counter using imu and magnetometer for navigation and health monitoring applications. *Sensors*, 17(11):2573, 2017.

[21] Hongyu Zhao, Wanli Cheng, Ning Yang, Sen Qiu, Zhelong Wang, and Jianjun Wang. Smartphone-based 3d indoor pedestrian positioning through multi-modal data fusion. *Sensors*, 19(20):4554, Oct 2019.

[22] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7, 2011.

[23] Tri-Nhut Do, Ran Liu, Chau Yuen, Meng Zhang, and U-Xuan Tan. Personal dead reckoning using imu mounted on upper torso and inverted pendulum model. *IEEE Sensors Journal*, 16(21):7600–7608, 2016.

[24] Harvey Weinberg. Using the adxl202 in pedometer and personal navigation applications. *Analog Devices AN-602 application note*, 2(2):1–6, 2002.

[25] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 37–42, 2009.

[26] V. Renaudin et al. Evaluating indoor positioning systems in a shopping mall: The lessons learned from the ipin 2018 competition. *IEEE Access*, 7:148594–148628, 2019.

[27] Q. Liang, L. Wang, Y. Li, and M. Liu. Indoor mapping and localization for pedestrians using opportunistic sensing with smartphones. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1649–1656, 2018.

[28] Chunyang Yu, Naser El-Sheimy, Haiyu Lan, and Zhenbo Liu. Map-based indoor pedestrian navigation using an auxiliary particle filter. *Micromachines*, 8(7):225, Jul 2017.

[29] Fang-Shii Ning and Yu-Chun Chen. Combining a modified particle filter method and indoor magnetic fingerprint map to assist pedestrian dead reckoning for indoor positioning and navigation. *Sensors*, 20(1):185, Dec 2019.

[30] Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[31] Hang Yan, Sachini Herath, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. `http://ronin.cs.sfu.ca/`, 2019.

[32] Changhao Chen, Chris Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

[33] Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, and Niki Trigoni. Oxiod: The dataset for deep inertial odometry. `http://deepio.cs.ox.ac.uk/`, 2018.

[34] Matteo Luperto, Michele Antonazzi, Francesco Amigoni, and N. Alberto Borghese. Robot exploration of indoor environments using incomplete and inaccurate prior knowledge. *Robotics and Autonomous Systems*, 133:103622, 2020.

[35] Ory Walker, Fernando Vanegas, and Felipe Gonzalez. A framework for multi-agent uav exploration and target-finding in gps-denied and partially observable environments. *Sensors*, 20(17):4739, Aug 2020.

[36] X. Zhou and M. Franklin. Indoor route and location inference using smartphone imu sensors. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–7, 2020.

[37] Marine Jouin, Rafael Gouriveau, Daniel Hissel, Marie-Cécile Péra, and Noureddine Zerhouni. Particle filter-based prognostics: Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, 72-73:2–31, 2016.

[38] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

[39] M. Bolic, P. M. Djuric, and Sangjin Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions on Signal Processing*, 53(7):2442–2450, 2005.

[40] V. Elvira, J. Míguez, and P. M. Djurić. Adapting the number of particles in sequential monte carlo methods through an online scheme for convergence assessment. *IEEE Transactions on Signal Processing*, 65(7):1781–1794, 2017.

[41] J. D. Hol, T. B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82, 2006.

[42] Lawrence M. Murray, Anthony Lee, and Pierre E. Jacob. Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, 25(3):789–805, 2016.

[43] Augustine Kong, Jun S. Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

[44] D Pham, Karim Dahia, and Christian Musso. A kalman-particle kernel filter and its application to terrain navigation. *Proceedings 6th ICIF*, page 24, 2003.

[45] Thibaud Michel, Pierre Genevès, Hassen Fourati, and Nabil Layaïda. Attitude Estimation for Indoor Navigation and Augmented Reality with Smartphones. *Pervasive and Mobile Computing*, 46:96–121, Mar 2018.

Mohamed Anis GHAOUI

(M.Sc. 2020) is currently a Ph.D candidate at Paris-Saclay University. His main work revolves around Indoor Localization for pedestrians via signal processing and data fusion of embedded sensors data. He also works on electronics for Human-Machine Interaction and Simulation. He is in the MOSS (Methods and Systems for Signal Processing) team of SATIE labs, France since 2020.

Bastien VINCKE

received the Ph.D. degrees in from the Paris-Sud University (now Paris-Saclay) Orsay, France, 2012. In 2013, he joined the department of Autonomous System, Institute of Fundamental Electronics, Paris-Saclay University as an Assistant Professor. he is now in Paris-Saclay University, in the MOSS (Methods and Systems for Signal Processing) team of SATIE labs. His current research interests include Algorithm Architecture Adequacy, SLAM, and, more generally, embedded systems.

Roger Reynaud

is a Pr. in Paris-Saclay University. His research topics deal with: data fusion using probabilistic and belief theories; obstacle detection, tracking, localization; Algorithms Prototyping with "Algorithm Architecture Adequacy" methodology; and Design of quasi-autonomous systems with an ap-

proach centered on intrinsic autonomy augmented by external assistance. His teaching focuses on instrumentation for physical measurements, signal processing, robotics, and data fusion.