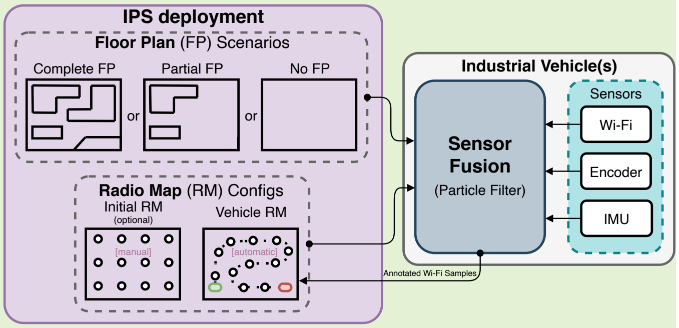
Real-World Deployment of Low-Cost Indoor Positioning Systems for Industrial Applications

Ivo Silva, Cristiano Pendão and Adriano Moreira *Member, IEEE*

***Abstract*—The deployment of an Indoor Position System (IPS) in the real-world raised many challenges, such as installation of infrastructure, the calibration process or modelling of the building's floor plan. For Wi-Fi-based IPSs, deployments often require a laborious and time-consuming site survey to build a Radio Map (RM), which tends to become outdated over time due to several factors. In this paper, we evaluate different deployment methods of a Wi-Fi-based IPS in an industrial environment. The proposed solution works in scenarios with different space restrictions and automatically builds a RM using industrial vehicles in operation. Localization and tracking of industrial vehicles, equipped with low-cost sensors, is achieved with a particle filter, which combines Wi-Fi measurements with heading and displacement data. This allows to automatically annotate and add new samples to a RM, named vehicle Radio Map (vRM), without human intervention. In industrial environments, vRMs can be used with Wi-Fi fingerprinting to locate human operators, industrial vehicles, or other assets, allowing to improve logistics, monitoring of operations, and safety of operators. Experiments in an industrial building show that the proposed solution is capable of automatically building a high-quality vRM in different scenarios, i.e., considering a complete floor plan, a partial floor plan or without a floor plan. Obtained results revealed that vRMs can be used in Wi-Fi fingerprinting with better accuracy than a traditional RM. Sub-meter accuracies were obtained for an industrial vehicle prototype after deployment in a real building.**



***Index Terms*—collaborative positioning, indoor positioning, industrial vehicles, particle filter, positioning system deployment, radio map, sensor data fusion, simultaneous localization and mapping (SLAM), Wi-Fi fingerprinting.**

I. Introduction

Research papers about Indoor Position System (IPS) usually present a general description of the solution and technologies but lack an analysis of the requirements and the deployment effort to set up the system, e.g., the time and cost to set up, whether it requires the installation of infrastructure, or if a calibration process is necessary. Moreover, most systems are only evaluated in the laboratory or in very small spaces. In industrial environments there are other challenges regarding the deployment of an IPS, for instance, there are spaces that often change the indoor layout due to reconfiguration of production lines, and industrial spaces whose floor plan is not available or is not detailed enough.

Localization in indoor environments, where GPS is not useful due to high attenuation of GPS signals caused by the building, can be supported by several technologies, i.e., Wi-Fi [1, 2], Ultra-wide band (UWB) [3], Bluetooth [4, 5], RFID [6], among others. Over the years, the research community [7, 8, 9] and industry[1] have shown interest in Wi-Fi technology because it explores Access Points (APs) that are already installed in the buildings, thus the deployment of Wi-Fi-based IPSs is cost-effective, in comparison to other technologies that require installation of additional infrastructure.

Wi-Fi fingerprinting [1] is a well-known positioning technique that provides absolute position estimates, with mean errors between 2 and 6 m [2]. It comprises an offline and an online phase. In the offline phase (calibration), a site survey is performed with the collection of Wi-Fi Samples (WSs) (signal strength measurements) at known positions (Reference Points (RPs)), resulting in a Radio Map (RM) of the surveyed area. In the online phase, a WS collected at an unknown position is compared against WSs of the RM, and a position estimate is obtained. Different metrics (e.g. Manhattan, Euclidean or cosine distance) have been used for matching between the test WS and the WSs in the RM, and positioning algorithms such as the the k-Nearest Neighbour (kNN) have been used to estimate a position.

Building a RM for Wi-Fi fingerprinting is a laborious and

Manuscript received xx December 2019; revised xx April 2020; accepted xx xxxx 2020. Date of publication xx xxxx 2020; date of current version xx xxxx 2020.

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020 and the PhD fellowship PD/BD/137401/2018.

I. Silva, C. Pendão, and A. Moreira are with the Algoritmi Research Centre, University of Minho, 4800-058 Guimarães, Portugal (email: {ivo, cpendao}@dsi.uminho.pt, adriano.moreira@algoritmi.uminho.pt). (Corresponding author: Ivo Silva)

[1] IPS at Ford (Almussafes, Spain): https://www.europapress.es/motor/sector-00644/noticia-ford-almussafes-desarrolla-sistema-geolocalizacion-incrementar-eficiencia-20210128151431.html

time-consuming task because it requires the mapping of RPs and the collection of WSs at each RP. The complexity of this task increases proportionally to the size of the building and the grid size that defines the distance between neighbor RPs. For instance, if the grid size is denser, more RPs are surveyed which means that there will be a higher cost and the accuracy is expected to increase, hence there is usually a trade-off between cost and accuracy when deciding the grid size of the RM. Once finished, the RM represents a snapshot of the building's radio environment at that particular time. With time, the RM becomes outdated due to many reasons, such as propagation effects, changes in the indoor layout of the building, moving furniture, or the addition or removal of APs. Therefore, the RM needs to be periodically updated, leading to additional costs in terms of time and human resources.

Several methods have been explored to build RMs: manual - site survey which takes a long time and requires human resources; interpolation - using sniffers that collect WSs in a few locations, then using an interpolation technique to estimate APs' signal levels in all RPs [10, 11, 12, 13]; Simultaneous Localization and Mapping (SLAM) - a system that locates itself whilst collecting Signals of OPportunity (SOP) to map the radio environment [14, 15, 16].

Usually, the requirements and deployment configurations of Indoor Position Systems (IPSs) are static and strict, namely: deployment of infrastructure, e.g., UWB which is expensive and in some scenarios impossible to deploy; calibration (or site survey), necessary in fingerprinting-based methods (e.g. Wi-Fi or Bluetooth) that depend on a RM that needs to be created and periodically updated; floor plan information, which is frequently used in IPSs especially the ones based on Particle Filters (PFs), however no floor plan or only a partial floor plan (low detail) may be available in some scenarios. Moreover, in many environments, floor plans frequently change due to alterations in the indoor layout making previous floor plan models outdated. Since most IPSs have strict deployment configurations they do not adapt well and do not work properly if the requirements cannot be ensured.

To address this, we present a new method for industrial positioning that is adaptable to different environments and deployment configurations. The system is able to operate with a complete Floor Plan (cFP), a partial Floor Plan (pFP) or with no Floor Plan (nFP) at all. Moreover, it can operate with an initial Radio Map (iRM), when it is possible to perform a site survey, or without an iRM. This solution explores industrial vehicles in operation to reduce deployment effort in industrial environments. Industrial vehicles have an important role in the day-to-day tasks of industrial environments, mostly in the transportation of raw materials and finished goods. In all deployment configurations, the system operates in a SLAM approach. By localizing vehicles as they operate, it is possible to collect WSs that can be added to the RM allowing to map the radio environment automatically. We refer to iRM as a manually constructed RM based on a site survey and refer to vehicle Radio Map (vRM) as the set of WSs that are autonomously annotated by vehicles in operation. This approach can also be used to build BLE radio maps. A Particle Filter (PF) is used to fuse data from vehicles equipped with low-cost sensors: Wi-Fi and motion sensors (wheel encoder and Inertial Measurement Unit (IMU)). The PF tracks the vehicles in operation and decides whether each estimated position is reliable or not so that the latest WS can be annotated (assign a position to the sample) and added to the vRM. With this approach, one can construct and maintain RMs without additional effort or costs. In addition, the PF allows easy integration of floor plans, supporting scenarios with cFP, pFP or nFP. This is possible because the PF updates particles' weights using Wi-Fi similarity to preserve particles that are closer to the true position. The main contributions of this paper are:

- A low effort deployment and maintenance IPS that is versatile for scenarios with different requirements and deployment configurations;
- A SLAM approach for the automatic construction and maintenance of Wi-Fi RMs in industrial environments using industrial vehicles equipped with low-cost sensors;
- A novel method to update particles' weights based on the similarity of WSs, using the weighted kNN algorithm, which allows to improve vehicle tracking.

II. Related Work

In past years, some research works proposed different approaches to optimize IPS deployment. He *et al.* [8] proposed a solution to optimize the number of APs and their positions to achieve better positioning accuracy. In [17], a solution is presented for automatic construction of RMs based on Received Signal Strength (RSS) (Wi-Fi, Bluetooth Low Energy (BLE), and RFID) to reduce deployment effort. The solution in [18] combines the automatic RM construction with AP selection to optimize the deployment of the IPS.

Some works explore mobile robots to perform SLAM of its surroundings [14, 15, 16]. SLAM systems are usually equipped with sensors to map the building, i.e., sonar, LiDAR or camera to detect indoor features such as walls, obstacles, or other elements. With the ability to locate themselves while collecting SOP, they may be used for creating the RM of the building, however, they are expensive due to the sensors they use, and usually require complex configuration. In [15] a mobile robot equipped with sonars, a wheel encoder, and a depth camera is capable of creating a RM by locating itself with a two-stage process. Initially, it performs obstacle-avoidance-based navigation and odometry-based correction for bearing angles. Then, it uses depth-camera images (RGB-D) for SLAM. Experiments with Wi-Fi fingerprinting showed a mean error of 5.2 m, in a corridor 54 m long. In [16], a robot equipped with several sensors (LiDAR, magnetometer, light sensor, smartphone (Wi-Fi)) generates an indoor grid map while simultaneously collecting signals from Wi-Fi interfaces and other sensors, allowing to construct and keep the RM up-to-date. The LiDAR allows to map the space using an Improved Maximum Likelihood Estimated (IMLE) scan-matching algorithm, while collecting data from other sensors, including Wi-Fi interfaces. Since WSs are obtained while the robot is moving, signal strength values of WSs collected within a 1 m radius of a reference point are averaged.

The SLAM approach described in [19] explores smartphones to collaboratively localise and construct the RM of the building. The system merges tracks from different users, performs loop closure detection (when users pass by a previously visited location), and optimizes a graph to generate accurate RM. Two Pedestrian Dead Reckoning (PDR) methods have been employed, one based on a step counter (using the smartphone's IMU) and another based on Project Tango (smartphone with additional sensors for high tracking accuracy). The advantages of this approach are that it does not require any knowledge of the map and locations of the APs. Experimental results in an area of 130x70 m have shown a positioning accuracy of 0.6 m and 4.76 m for Tango-based PDR and step counter-based PDR, respectively.

The idea of collaborative RMs where users independently contribute with WSs to build a joint RM has been explored in [20, 21, 22, 23]. There are different approaches to collaborative RMs: (i) users explicitly and manually report their position to the system while collecting WSs; (ii) users collect data passively (e.g. Wi-Fi and motion data), which can be used to estimate PDR trajectories and, subsequently, annotate the collected WSs; (iii) the system uses an opportunistic sensing approach and, from time to time, prompts the user to report whether they are in the estimated position. Since these solutions depend on user participation, they raise some challenges, for instance, users may choose not to share their information or indicate a wrong position and, they may not visit all areas of the building (leaving those areas unmapped).

Kim *et al.* [20] proposed an autonomous solution for the construction of RMs as well as the localization of users. Inertial data is automatically collected by users on their daily routine, without needing to survey the entire building. The system progressively builds the RM as more users contribute with data. The position of users is obtained by merging PDR with Wi-Fi fingerprinting. The average positioning error obtained in experiments is 6.9 m for a building with 60x66 m.

Both works in [21, 22] use Wi-Fi fingerprinting systems that take user feedback to improve RMs. These systems start with a baseline Wi-Fi fingerprinting solution which already has an existing RM, but users contribute with new WSs collected automatically by a smartphone. These systems also use users' feedback to build the RM. Taking the users input has its shortcomings, for instance, the system can be hindered by malicious actors that feed the system with bad data. The systems in [21, 22] were designed for pedestrians, hence they depend on the motion model of a pedestrian, which cannot be applied to industrial vehicles. In the proposed approach in this paper, a proper motion model is considered for the vehicles, which are tracked with motion sensors (an encoder and an IMU) for improved accuracy.

A different approach to build RMs is explored in [24], where a motion dynamics model is combined with Gaussian Process Latent Variable Models (GP-LVM) to reconstruct the RM from a set of received signal strength values. Gaussian processes are used to determine the model of each AP present in the building, allowing to interpolate the RM of the building. A user collects data to be used as training data for the GP-LVM algorithm, which then builds the RM of the environment. Performed experiments using the estimated RM achieved a mean error of 3.97 m.

Other alternatives eliminate the need to build and maintain the RM through interpolation (estimation of signal strength values) based on Radial Basis Functions (RBF) [10], universal Kriging [25], Voronoi tesselation [11], Inverse Distance Weighted (IDW) [12], or Log-Distance Path Loss (LDPL) [13]. The main drawback of these solutions is that they provide approximations of signal strength levels (that do not account for signal effects like reflection, refraction and multipath). Conversely, the WSs of collaborative RMs are collected on-site, therefore, they are more reliable as they accurately represent the radio environment at the time they were collected. A combination of SLAM and propagation models is used in [26] to automatically build the RM of an industrial space. Initially, a 3D force-directed graph is built from WSs collected by fixed Wi-Fi anchors. Then, the graph estimates APs' positions which allows to model the radio environment using a mobile unit equipped with Wi-Fi and motion sensors. As a result, the RM is automatically built without calibration or knowledge about the space. Since this method estimates APs' positions, it is prone to errors that can affect the positioning accuracy of the RM. In addition, it requires the installation of Wi-Fi anchors, which can be challenging due to the number of beacons and the locations where they have to be installed.

In comparison with related works, the distinctive features of this approach include a novel solution versatile for different deployment configurations in an industrial environment. This solution explores industrial vehicles in operation to automatically build and maintain the RM.

Since WSs are annotated by vehicles on-site, vRMs are better than interpolated RMs and better than solutions that depend on user feedback, because the PF automatically annotates WSs without human intervention. This approach is cheaper than relying on mobile robots for the creation and maintenance of RMs since it uses existing vehicles that do not disturb or affect operations and do not require expensive sensors such as LiDAR. Moreover, this solution is less complex and lower-cost to deploy than solutions that depend on the installation of dedicated infrastructure (Wi-Fi anchors or UWB), because it explores the already available Wi-Fi infrastructure.

III. Approach

The efficient deployment of an IPS is possible when it is simple to set up and configure and, if possible, without human intervention. In this paper, we explore industrial vehicles in operation to optimize the deployment of an IPS. Industrial vehicles have an important role in the day-to-day tasks of factory plants, mostly in the transportation of materials, and material handling. Enabling the localization and tracking of these vehicles allows to build the RM and to improve logistics processes and better manage the operations.

In order to achieve accurate tracking of industrial vehicles, it is necessary to adopt a sensor fusion technique that combines data from sensors. In this paper, we consider that vehicles are equipped with a limited set of low-cost sensors, i.e., Wi-Fi interfaces (to measure RSS values), an IMU to measure

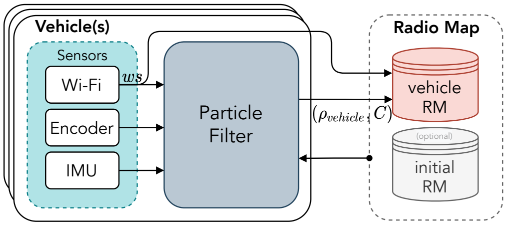


Fig. 1: Architecture of the proposed solution.

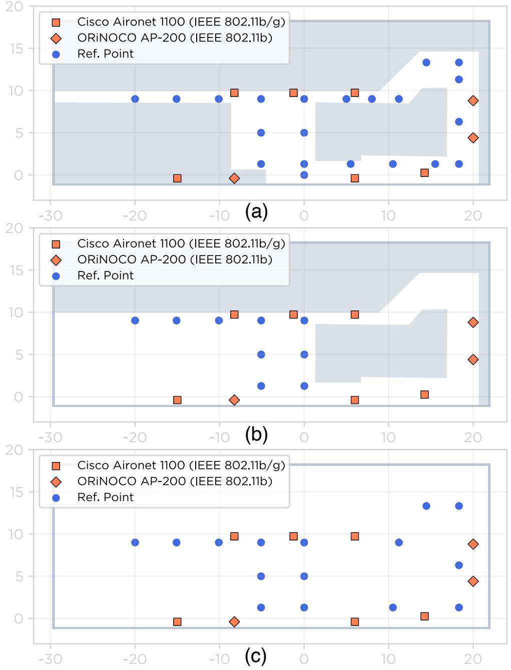

Cisco Aironet 1100 (IEEE 802.11b/g)
ORiNOCO AP-200 (IEEE 802.11b)
Ref. Point
20
15
10
5
0
-30
-20
-10
0
10
20
(a)
(b)
(c)

Fig. 2: Considered floor plan scenarios: (a) complete FP (cFP); (b) partial FP (pFP); and, (c) no FP (nFP).

the heading (orientation), and a wheel encoder to measure the displacement (travelled distance). This section describes the approach used for sensor fusion as well as the process in which vehicles are explored to build and maintain RMs automatically.

A. Architecture

Fig. 1 shows the architecture of the proposed IPS. It comprises one or multiple industrial vehicles with Wi-Fi and motion sensors. Each vehicle includes a PF, used to perform sensor fusion of data from all sensors, producing position estimates and corresponding confidence values. PF localization is adequate for sensor fusion in cases where there are non-linear system dynamics and arbitrary noise distributions. The advantages of PFs are that they are simple to implement, allow easy integration of the building's floor plan, and are good at dealing with sensors' noise.

The RM is used by the PF in the particles' weighting phase. The Wi-Fi RM is composed of the vRM and the iRM (optional). The vRM contains all WSs annotated by vehicles and the iRM contains WSs previously collected that serve as a starting point for the system.

The confidence is a metric based on the dispersion of particles, that determines whether the latest PF position estimate is reliable or not. When the confidence is high, it allows to build and maintain the vRM by assigning a position estimate to the WSs collected by vehicles. The following sections describe the PF processes and how the confidence metric is calculated.

B. Scenarios and Radio Map Configurations

The characteristics of industrial environments can vary significantly, depending on the type of business, e.g., it can be a simple storage warehouse or a complex space with several production lines. The navigable areas, where industrial vehicles operate are important as they can be represented in a floor plan to assist in indoor tracking and localization. Some industrial spaces have detailed floor plans available, but there are other spaces where a large part is an open area and only a partial floor plan is available, and there are also cases where the floor plan is not available.

Therefore, three possible scenarios were considered (Fig. 2): **cFP** - complete floor plan of the space is available; **pFP** - partial floor plan of the space is available (some areas are mapped in the floor plan, but there is a large open space); **nFP** - no floor plan is available, large open area.

The proposed PF depends on a RM because particles' weights are updated based on Wi-Fi information and the RM is used as a reference. Without RM, it is not possible to update particles' weights, therefore the PF works as a pure Dead Reckoning (DR) solution using only the motion sensors. In this scenario, the Vehicle Dead Reckoning (VDR) trajectory containing drift is corrected by detecting collisions of particles in walls or obstacles, defined in the floor plan.

When the RM is available it can be just the vRM, or a full Radio Map (fRM), composed of an iRM and a vRM. The iRM is based on a manual site survey, and the vRM is composed of WSs that are annotated while the vehicle operates. The proposed solution may operate using the following RM configurations: **VDR** - without iRM and vRM, pure dead reckoning; **vRM** - without iRM, but with vRM; **fRM** - full radio map composed of the iRM and the vRM. Since building the iRM is time-consuming and difficult to accomplish in large scenarios, the iRM used in experiments is a sparse RM (with a few samples) whose RPs are represented in blue in Fig. 2.

C. Particle Filter - Indoor Positioning System

The PF solution capable of locating and tracking industrial vehicles is based on our previous work [27]. This PF, however, has several differences from the previous version, namely: (i) it works in different scenarios, with and without the floor plan of the building; (ii) it uses a new method to update particles' weights using weighted kNN to estimate WSs similarity at the position of each particle, and dynamically adjusting parameters; (iii) it works with or without an iRM; (iv) new experiments were conducted to validate this solution. In this work, we focus particularly on the ability to continuously track the vehicle while collecting WSs to build and maintain a vRM.

The PF works as follows. First, particles are created around the vehicle's initial position. Second, as new sensor readings are received, particles' states are updated considering noise in the sensors. Third, particle's weights are updated based on WSs similarities, by comparing a new received WS to the WSs in the RM (note that the RM is dynamic and evolves over time because it comprises samples of the vRM). Fourth, over time, some particles disperse due to the noise added to the sensors' readings, and when their weights become low, they are removed in the resampling process. In scenarios that consider a floor plan, particles that hit walls or obstacles are assigned a zero weight. In the resampling step, the particles with lower weights are removed and the particles with higher weights are duplicated to keep the number of particles constant. The continuous updating of particles' weights and the resampling process allow the estimated position to remain near the true position, minimizing the cumulative drift in the heading. A reliability metric, named confidence, allows to determine when to add annotated WSs to the vRM.

The PF consists of a set of M particles defined as $p = (w, x, y, h, ho)$, where w represents the weight of the particle, x and y represent the position coordinates of the particle, h represents the heading, and ho represents the heading offset used to improve heading estimates.

1) Initialization: In industrial environments, vehicles are usually parked in known positions (e.g. charging stations or dedicated parking spots) and start their operation from a known pose (position ρ_{ini} and heading h_{ini}). This information is used to initialize particles around the vehicle's position. Particles are uniformly distributed within a radius r of a given initial position ρ_{ini}, have an initial heading defined as h_{ini}, and an initial heading offset (ho) given by $U(0, n_{ho})$. As a result, each particle has a unique position and heading offset.

2) Particles' movement model: The movement model of the particles follows a DR approach, as follows:

$$\begin{aligned} h &= \theta + n_\theta + ho \\ x &= x_{-1} + (l + n_l) * \cos(h) \\ y &= y_{-1} + (l + n_l) * \sin(h) \end{aligned} \tag{1}$$

where h, x, y represent respectively, the heading and position coordinates of the particle. (x_{-1}, y_{-1}) refer to the previous position of the particle, l represents the latest displacement sample, θ represents the latest heading sample, and ho represents the particle's heading offset. Both n_l and n_θ represent zero-mean Gaussian distributed random variables necessary to model the noise of the encoder and IMU, respectively.

3) Updating Particles Weights: In this process, higher weights are assigned to particles that are closer to the true position and lower weights are assigned to particles that are probably further from the true position. Particles' weights are updated by comparing the similarity between the latest WS collected by the vehicle and the RM samples. Then, the weighted kNN is applied to estimate the normalized similarity at the position of each particle. Finally, the normalized similarity is converted into a weight value assigned to the particle.

A WS (reading from a Wi-Fi interface) is defined as the set of n RSSI values of APs:

$$ws = \{RSSI_1, ..., RSSI_n\} \tag{2}$$

The obtained WS is compared against the RM samples using a distance function (Manhattan distance). The radio map, $RM = \{(\rho_0, ws_0), ..., (\rho_m, ws_m)\}$ represents a set of m WSs, where each ws is associated to the position ρ where it was collected. The Manhattan distance between a WS and a RM sample is given by:

$$d_M(ws, rm) = \sum_{i=1}^{n} \left| RSSI_{ws}^i - RSSI_{rm}^i \right| \tag{3}$$

where n represents the number of detected APs, ws represents the WS and rm defines the WS of the RM.

The distance is then converted into a normalized similarity:

$$s_n = 1 - \left(\frac{d_M - min(D)}{max(D) - min(D)} \right) \tag{4}$$

where D represents the set of all distances between the WS and RM samples. Higher similarities are expected in locations around the vehicle's true position.

Then, the normalized similarity at the position of each particle is obtained using the weighted kNN algorithm, as described in Algorithm 1. First, the distances (Euclidean) between the particle and all RM samples are calculated. Second, these distances are organized in ascending order. Third, the weight of the particle is computed using the inverse distance weighted method, which assigns higher weights to shorter distances and lower weights to larger distances. Fourth, the estimated similarity is obtained using the weighted average of the k_{uw} WSs closer to the particle position.

Algorithm 1 Estimate particle normalized similarity.

```
Input
    $k_{uw}$ - parameter of the weighted $k$NN algorithm
    $p$ - particle
    $WS_p$ - set of Wi-Fi samples positions
    $WS_s$ - set of Wi-Fi samples normalized similarities
Output
    $\hat{s_n}$ - estimated similarity at particle's position
procedure ESTIMATE PARTICLE NORM. SIMILARITY
    $D = d_{Man.}(p, WS_p)$ //set of Manhattan distances between the particle and each Wi-Fi sample
    $D \leftarrow sort(D)$
    $\varpi \leftarrow \{\}$ //initialize set of weights
    for $d \in D$ do
        $\varpi \leftarrow \varpi \cup \{\frac{1}{d^2}\}$ //convert distance into weight and add it to the set of weights
    $\hat{s_n} = \frac{\sum_{i=1}^{k_{uw}} \varpi^i \times WS_s^i}{\sum_{i=1}^{k_{uw}} \varpi^i}$ //norm. similarity using w-$k$NN
    return $\hat{s_n}$
```

Fig. 3 shows an example of the estimated similarities at the positions of particles using this approach. The 5 particles refer to the state of the PF at a given moment. The 3 RM samples represent the normalized similarity after the collection of a new WS. The estimated similarity of each particle using the algorithm above is shown at the position of each particle, as expected, each WS of the RM contributes to the estimated

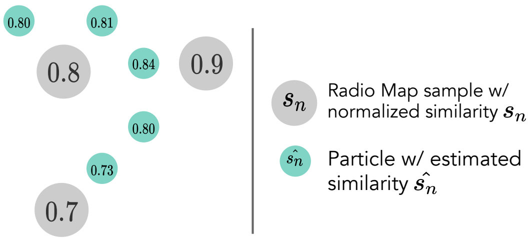

0.80
0.81
0.8
0.84
0.9
0.80
0.73
0.7
s_n Radio Map sample w/ normalized similarity s_n
$\hat{s_n}$ Particle w/ estimated similarity $\hat{s_n}$

Fig. 3: Estimated similarity of each particle.

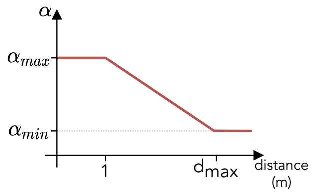


Fig. 4: Dynamic α based on the distance between the particle and the nearest WS.

similarity, being that the nearest WSs have more influence over the estimated value. The k_{uw} parameter defines the number of nearest WSs considered in the algorithm.

Once the particle's estimated normalized similarity is obtained, its weight is updated:

$$w = \alpha \times (1 - \hat{s_n}) + (1 - \alpha) \times w_{-1}, \quad \alpha \in [0, 1] \tag{5}$$

where w_{-1} represents the previous weight of the particle and α is a parameter that defines how rapidly the particle's weight varies based on the latest WS (see Section III-C.4 for further details about α). The last step in the process to update particles' weights comprises the normalization of weights between 0 and 1.

4) Dynamic Alpha: The α parameter allows to define how much the particle's weight is influenced by the latest WS. The α value is based on the distance between the particle and the nearest RM sample, being that when the distance is small, α is larger and vice versa. The rationale is that particles far from any RM sample are in positions not yet mapped and, therefore, their weights should not be updated significantly.

Fig. 4 shows how α is obtained based on the distance (d) between the particle and the nearest RM sample. When the particle is close to the nearest RM sample ($d \leq 1\ m$), α is assigned α_{max}. For $1\ m < d < d_{max}$, a linear function is considered where α decreases as d increases. When $d \geq d_{max}$ of the nearest WS of the RM, α is assigned the α_{min} value.

5) Resample: Resampling is a process in which particles with lower weights are removed, and particles with higher weights are randomly selected to be copied. Lower weight particles tend to be in areas where it is unlikely to be the true position. A threshold value, w_{th}, is used to determine which particles are removed. The remaining subset of particles, with weights higher than w_{th}, are randomly selected and copied to keep the number of particles constant. We adopted the multinomial resampling approach [28], where the particles are selected using their weight as the probability of being selected. Each resampled particle has a new heading offset, defined as:

$$ho' = ho + n_{ho} \tag{6}$$

where ho represents the particle's original heading offset and n_{ho} represents a zero-mean Gaussian distributed random variable. Particles with higher weights are the ones that better represent the pose of the vehicle, hence, their heading offset is used in resampled particles, to which is added the n_{ho} component to represent the noise distribution of the vehicle's heading. This process allows to improve heading estimates.

D. Estimate vehicle's position and heading

The estimated pose of the vehicle is given by the weighted average of the positions and headings of all particles:

$$\rho_{vehicle}(x, y) = \frac{\sum_{i=1}^{M} p_i(x, y) \times p_i.w}{\sum_{i=1}^{M} p_i.w} \tag{7}$$

and,

$$h_{vehicle} = \tan^{-1}\left(\frac{\sum_{i=1}^{M} \sin(p_i.h) \times p_i.w}{\sum_{i=1}^{M} \cos(p_i.h) \times p_i.w}\right) \tag{8}$$

E. Confidence in position estimates

Most IPSs, even the most reliable ones, do not provide a measure that defines whether the estimated position is accurate or not. A confidence measure is especially important for localization and mapping because if the IPS is able to self-assess whether its provided estimate is reliable, it can decide whether to use the estimated position for mapping.

In a PF, a higher dispersion of particles occurs when there is higher uncertainty, hence the cluster of particles is more dispersed. The higher the dispersion, the lower the confidence, hence the confidence can be used as a parameter to determine whether new WSs should be annotated and added to the vRM.

As described above, particles are initialized around the initial position, where confidence is high, hence WSs can be annotated during this initial period. As time goes by and the vehicle starts moving, particles start to disperse due to the noise in the sensor's readings. In situations when the confidence is low, WSs should not be annotated, as it is probable that a larger error is associated with those WSs.

Higher confidence is observed after resampling, due to the removal of particles with lower weights. This metric also benefits from scenarios that use the floor plan, which allow removing particles that hit walls or obstacles.

The dispersion of particles is defined as:

$$\vartheta = \frac{1}{N}\sum_{i=1}^{N} d_E(\rho_{vehicle}, p_i) \times w_i \tag{9}$$

where $\rho_{vehicle}$ is the PF estimated position, p_i and w_i represent the ith particle position and weight, respectively, and d_E represents the Euclidean distance between $\rho_{vehicle}$ and p_i. The confidence is defined as a function of dispersion:

$$C(\vartheta) = \begin{cases} -0.25\vartheta + 1, & 0 \leq \vartheta < 4 \\ 0, & \vartheta \geq 4 \end{cases} \tag{10}$$

where ϑ represents the dispersion of particles in meters.

When the dispersion of particles is larger than 4 m, there is high uncertainty in the PF estimated position, hence the confidence is zero. In preliminary experiments, we found that the dispersion of particles depends on the noise added to heading and displacement and the floor plan configuration. The larger the noise added to heading or displacement samples the larger is the dispersion. In addition, in scenarios where the floor plan is not available, particles tend to disperse more in comparison with scenarios where a partial or complete floor plan is available. The 4 m dispersion was chosen because when the uncertainty in the estimated position is high, there is a higher probability for having a large error associated, and in this application, it is crucial to annotate WSs with the lowest possible positioning error. When the dispersion is lower than 4 m, the confidence has a linear relation with dispersion, the lower the dispersion, the higher the confidence in the position estimate. When confidence is higher than a given threshold C_{th}, new WSs can be annotated and added to the vRM.

F. Vehicle Radio Map

Vehicles contribute to build and maintain the vRM by adding annotated WSs to the vRM whenever the confidence of the position estimates is high enough. The vRM is different from a standard RM because WSs were collected automatically and have an associated error, as opposed to a manually created RM where WSs have zero error.

Upon receiving a new WS, the vehicle checks if the confidence is high enough before deciding to add it to the vRM. When the confidence is equal or greater than C_{th}, the latest position estimate and confidence are assigned to the WS, creating a vehicle WS, defined as:

$$vws = (\rho_{vehicle}, ws, C) \tag{11}$$

where ws represents the signal strength values of detected APs, $\rho_{vehicle}$ represents the current position of the vehicle and C represents the current confidence.

The quality of annotated WSs depends on the positioning performance of the PF. Higher precision and accuracy lead to a better vRM. Annotated WSs are only added to the vRM after 120 s since they were collected to prevent recent WSs from being used to update particle's weights because it would cause a dragging effect where particles closer to the more recent WSs would have a higher similarity.

G. Wi-Fi Fingerprinting

VRMs obtained with industrial vehicles may be used for plain Wi-Fi fingerprinting-based positioning in other applications. In industrial environments Wi-Fi fingerprinting can be used for several purposes: improving logistics operations; safety reasons (locate vehicles or people); tracking materials along the supply chain (either raw materials as they move from the warehouse to production lines or finished products from production lines to the warehouse). The k-Nearest-Neighbor estimator is one of the most used techniques to estimate a position in Wi-Fi fingerprinting. It computes the centroid of the k locations of the most similar WSs of the RM, as follows:

$$\rho_{wifi} = \frac{\sum_{i=1}^{k} \rho_i(x, y)}{k} \tag{12}$$

where ρ_i represents the position of a WS.

IV. Real-world Experiments

Conducted experiments included three phases. First, the development of a mobile unit equipped with several sensors. Second, the setup at the testing space with the mapping of RPs and the collection of WSs for the iRM used in some experiments. Third, conducting the experiments.

A. Setup and Mobile Unit Prototype

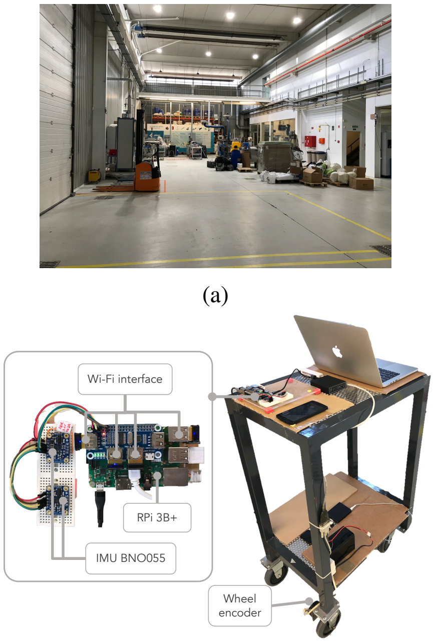
Real-world experiments were conducted at the PIEP building located at the Azurém Campus of the University of Minho. PIEP (shown in Fig. 5a) measures approximately 50x20 m and is an industrial building for research on plastic polymers. It is similar to a factory plant in many aspects since it has large open spaces, narrow corridors, and heavy machinery (injection molding machines). The iRM, used in some experiments (fRM config), is a sparse radio map with ≈5 m between RPs (blue points in Fig. 2). A site survey was conducted to build the iRM with the collection of 40 WSs per RP.

Test data was collected using a mobile unit, a manually pushed trolley to emulate an industrial vehicle. The mobile unit, represented in Fig. 5b, is equipped with the positioning module, composed of: a Raspberry Pi (RPi) Model 3B; 4x external Wi-Fi interfaces (Edimax EW7811-Un); 2x IMUs (Adafruit BNO055) to measure the heading; and, an absolute encoder (US Digital A2) to measure the displacement. The RPi is used as a computer to collect data from sensors. Using multiple Wi-Fi interfaces leads to improvements in positioning performance [29], because there is a low correlation between signals from distinct Wi-Fi interfaces, therefore signals can be merged into one sample. We have adopted this method to improve the accuracy of the proposed solution.

Total retail cost of our prototype is 386€: RPi (35€), encoder (255€), IMUs (2x 30€), Wi-Fi interfaces (4x 9€).

B. Low-cost IMU

In previous work [27], we used industrial-grade IMUs (Xsens Mti-300 AHRS) for obtaining the heading of the vehicles. Although these sensors are reliable and less prone to drift and magnetic perturbations, they are very expensive, making the IMU the most expensive part of the positioning module. Using lower-cost IMUs allows to drastically reduce the cost of the positioning module, but as a consequence, these sensors are noisier and less accurate than industrial-grade IMUs. In this work, we have used the Adafruit BNO055 IMU as a substitute for the industrial-grade IMUs used in our previous works. This sensor was chosen for several reasons. First, it is supported by the RPi, therefore it can be easily integrated into this solution. Second, it is cheap and easily found in online stores. Third, it performs in-chip data fusion capable of producing absolute orientation in quaternion or Euler angle formats at 100 Hz. Fourth, it


(a)

(b)

Fig. 5: Experiments: (a) PIEP building; (b) Mobile unit prototype with installed sensors.

TABLE I: Particle filter parameters.

Param	M	r	n_l	n_θ	n_{ho}	
Value	3000	1 m	N(0, 0.012 m)	N(0, 1°)	N(0, 8°)	
Param	k_{uw}	α_{min}	α_{max}	d_{max}	w_{th}	C_{th}
Value	3000	0.05	0.3	2 m	0.2	0.75

allows to collect raw measurements from an accelerometer, gyroscope and magnetometer which can be used in a sensor fusion algorithm to estimate the sensor's orientation. Fifth, the research community has revealed an increasing interest in this sensor for positioning and navigation, especially in recent years as shown by a search in the Scopus database with the query `"(ALL(bno055) AND TITLE-ABS-KEY( positio* OR navigat* ))"`. Overall the Adafruit BNO055 IMU is a great sensor for research due to it being low-cost and simple to work with.

We tested a Madgwick filter [30] implementation to perform sensor fusion from raw measurements, however, our results were no better than the ones provided by the integrated filter, which had less noise and drift than the results achieved with the Madgwick filter, therefore we opted to use the integrated filter of the IMU sensor in our solution. Two IMUs were installed in the mobile unit because in the future we intend to explore sensor fusion methods to combine data from both sensors to improve heading estimates.

C. Particle Filter parameters

Table I lists the parameters used in performed experiments. The same parameters were used in all experiments in order to assess the versatility of the PF with the same parameters in different floor plan scenarios and RM configurations.

D. Results

The mobile unit was manually pushed in five trajectories at a speed of ≈1 m/s, to construct the vRM. Test data was collected from the mobile unit's sensors and served as input for the proposed PF. All annotated WSs were accumulated in the vRM and used from one trajectory to the next one. Therefore, the vRM used in the last trajectory includes all WSs that were annotated in previous trajectories.

The positioning error reported in this section was obtained by comparing estimated positions with ground truth data collected by a video camera mounted in the mobile unit. The video camera recorded tags affixed to the floor (in known positions) when the mobile unit passes by them. The error metric is the Euclidean distance between the position estimates and the ground truth.

The PF was run for 9 experiments considering the combinations of 3 scenarios (cFP, pFP, and nFP) with 3 RM configurations (VDR, vRM, and fRM). The PF was run three times for each trajectory, and the results include position estimates of all runs. The positioning results of performed experiments are reported in Table II. As expected, the scenario with better positioning results is the one that uses the complete Floor Plan (cFP). In this scenario, the floor plan allows to remove particles that hit walls or obstacles allowing to improve estimates. In addition, this scenario is the one that better performs with VDR, showing that an iRM is not necessary to achieve good results. The scenario with the partial Floor Plan (pFP) shows gradual improvements in the positioning accuracy as a richer RM is used. As expected, the worst results are observed with VDR, then the configuration with vRM has improvements below the P99th, showing that this scenario benefits from WSs that are added to the RM. This is particularly important in open areas that are not mapped in the floor plan, where drift in the heading can significantly affect the estimated trajectory. The scenario with no Floor Plan (nFP) is the most challenging, showing large errors with VDR. This scenario has significant improvements in the vRM configuration because of the annotated WSs. To annotate WSs with higher accuracy, it is necessary an iRM, because it allows to keep particles within the neighborhood of RPs which minimizes the maximum error.

TABLE II: Positioning results for different scenarios and RM configurations (in meters).

	cFP			**pFP**			**nFP**		
	VDR	**vRM**	**fRM**	**VDR**	**vRM**	**fRM**	**VDR**	**vRM**	**fRM**
Mean	0.88	0.88	**0.67**	1.45	0.90	**0.62**	3.92	1.91	**0.63**
P$_{75th}$	1.17	1.20	**0.88**	1.96	1.11	**0.88**	6.03	2.70	**0.87**
P$_{99th}$	2.61	2.38	**1.71**	3.81	2.76	**1.54**	10.92	6.06	**1.64**
Max	2.89	4.27	**2.34**	3.84	3.90	**2.12**	24.70	7.56	**3.53**

cFP – complete FP; **pFP** – partial FP; **nFP** – no FP; **VDR** – vehicle DR; **vRM** – vehicle RM; **fRM** – full RM.

All three scenarios (cFP, pFP and nFP) achieve the best results with the fRM configuration because it allows to update particles' weights right from the beginning (particles initialization), which consequently allows to minimize large errors and improve the overall accuracy of the system. Since the overall performance of the PF is improved, annotated WSs have a lower error associated. The scenario that benefits the most from using the fRM configuration is the nFP scenario because, without the floor plan, the only reference data used to correct the drift is the iRM and the vRM.

Figs. 6a and 6b depict the estimated paths of the first trajectory considering the nFP scenario. A clear improvement is observed in the fRM estimated trajectory. The PF corrects most of the drift in comparison to the vRM trajectory. Figs. 6c and 6d show annotated WSs of the same trajectory, demonstrating that the positioning error of the vast majority of annotated samples is below 1 m. Annotated WSs do not cover the whole trajectory since there were periods during which confidence was lower than C_{th}. Less WSs are annotated when considering only the vRM (Fig. 6c) because the PF does not have any reference data that helps in removing particles. When the fRM is provided (Fig. 6d) it allows to annotate more samples, because the iRM has a significant impact on the update weights process allowing the PF to have high confidence during longer periods. Figs. 7a and 7b show the confidence vs error of the same trajectory. In both cases, it is observed that a higher density of position estimates is observed when $C > 0.6$ typically with errors lower than 2 m (red areas). When the confidence is lower than that, larger errors are observed, especially in the vRM configuration because the PF does not have reference data, and is not efficient in removing particles. When the iRM is provided (fRM config), confidence is high most of the time, with the exception of a few cases.

The CDF curves in Fig. 8 show that the best performance is achieved in fRM configurations, followed by the combinations pFP+vRM, cFP+vRM and cFP+VDR which have similar results. For these configurations, over 90% of all position estimates are below 2 m, therefore they are suitable for building and maintaining vRMs. The remaining configurations have larger errors, and should not be used for this application. Gradually worse results are observed for the combinations pFP+VDR, nFP+vRM and nFP+VDR, respectively. These results show that the best tracking of industrial vehicles is achieved when in fRM configurations, independently of the scenario. In case the iRM is not available, indoor tracking is possible in the cFP and pFP scenarios.

1) Computational Complexity - Optimization of Annotated Wi-Fi samples: In previous experiments we used a large value for k_{uw} to consider all RM samples when updating weights. This includes all annotated WSs and the iRM samples, in cases where the iRM was used. As the number of annotated WSs increases, the number of samples used in the update weights process also increases, leading to a computational complexity of $O(n \times \log n)$. Since the number of particles in the PF remains constant, the computational complexity depends on k_{uw}, therefore it should be a constant value. When we used low values for k_{uw} (e.g. $k_{uw} = 3, 5, 10$), results were worse, therefore it is important to use an adequate k_{uw} value and to determine which WSs should be selected. In our experiments considering all WSs (i.e. large k_{uw}), results were better, showing that it is necessary to use WSs uniformly dispersed through space.

In order to keep the computational complexity constant, we implemented simple random sampling to limit the number of

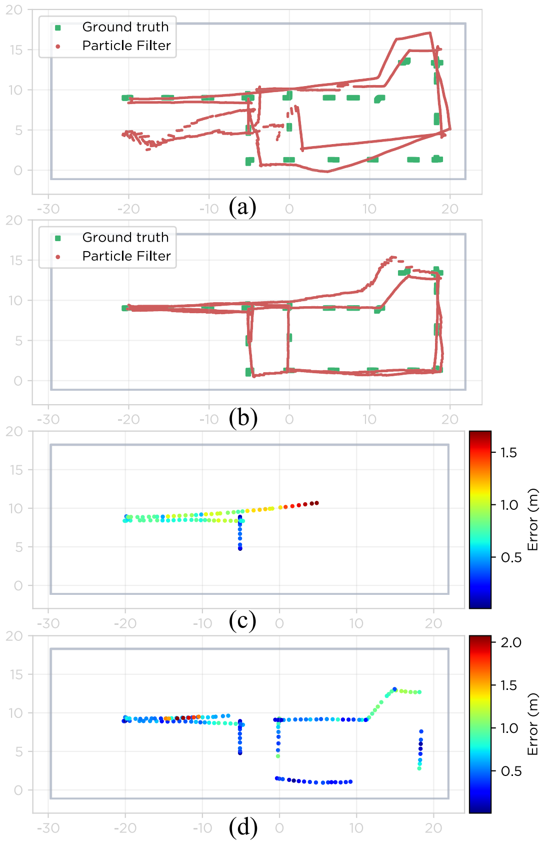


Fig. 6: Trajectory 1 for the no Floor Plan (nFP) scenario. Estimated trajectory: (a) vRM; (b) fRM. Annotated WSs: (c) vRM; (d) fRM.

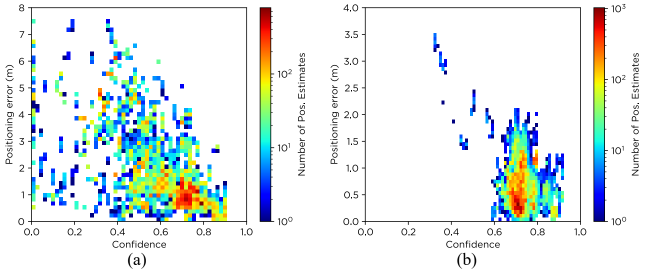


Fig. 7: Confidence vs positioning error of trajectory 1 for the no Floor Plan (nFP): (a) vRM; (b) fRM.

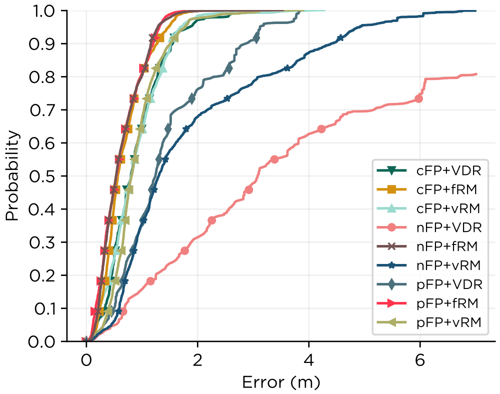


Fig. 8: CDF curves of all scenarios and RM combinations.

WSs when updating particles' weights. With this approach, every time the particle's weights are updated, a limited number of WSs are sampled from the RM. WSs are selected based on a uniform distribution, therefore each WSs has the same probability of being selected resulting in a set of WSs that are uniformly dispersed through space. The k_{uw} used in these experiments was the same as the number of WSs selected. Table III shows the results of the nFP+fRM scenario considering different numbers of WSs. A clear improvement is observed as the number of WSs increases.

By limiting the number of WSs to 200 or higher it is possible to achieve similar results to the ones that consider all WSs, as reported in Table II. When the number of selected WSs is larger than 200, results seem to stabilize, the main difference is observed in the maximum error which tends to decrease as the number of WSs increases. From 200 to 300 WSs, a major improvement is observed in the maximum error. From 300 to 400, no significant improvements are observed. Although the maximum error keeps improving when the number of WSs is larger than 200, the P99th is stabilized meaning that the maximum error can be attributed to a rare occurrence, therefore, it seems that the cost-benefit between the computational effort and the positioning performance is best when 200 WSs are selected for updating particles' weights. This shows that the computational complexity will not be a hindrance to the positioning system.

A density of 200 $WS/1000\ m^2 = 0.2\ WS/m^2$ is enough to ensure good positioning results because optimal positioning results were achieved with 200 WSs for a building with an area of about 1000 m^2 (50x20 m). A similar performance is expected in larger buildings as long as the same density of WSs is ensured in the process to update particles' weights.

TABLE III: Positioning results of nFP+fRM scenario using simple random sampling to limit number of WSs.

	No. of Wi-Fi Samples						
	25	**50**	**100**	**150**	**200**	**300**	**400**
Mean	2.68	2.03	0.89	0.71	**0.63**	**0.63**	**0.63**
P$_{75th}$	4.15	3.05	1.02	0.99	0.89	0.87	**0.86**
P$_{99th}$	10.43	5.87	4.49	2.53	1.71	**1.66**	1.69
Max	15.98	8.07	6.23	5.89	4.37	3.74	**3.62**

2) Vehicle Radio Map vs Traditional Radio Map:

As previously mentioned, Wi-Fi RMs can be explored for many applications including the localization of people or assets in industrial environments. Therefore, vRMs can be of the utmost importance to easily support plain Wi-Fi fingerprinting applications. The only requirement is that a Wi-Fi-enabled device is used to allow indoor localization.

To evaluate how the vRM performs in a Wi-Fi fingerprinting scenario, we compared the positioning performance between the iRM and the vRM obtained with plain Wi-Fi fingerprinting using kNN with $k = 5$. In this experiment, the set of WSs of trajectory 5 was considered as test data. As vRM, we considered the WSs that were annotated in the other 4 trajectories, for the configurations cFP+vRM, pFP+vRM and nFP+vRM which are the ones where annotated WSs are added to the vRM and an iRM is not provided. Considered vRMs are composed of a total of 1401, 1216, and 247 WSs in cFP+vRM, pFP+vRM and nFP+vRM configurations, respectively. The iRM is composed of 840 WSs, with 40 WSs collected at each of the 21 RPs, depicted in Fig. 2.

Table IV presents the plain Wi-Fi fingerprinting results using the vRM of each scenario. As expected, the results of cFP and pFP are better than the results of the nFP, because they have more WSs and have lower positioning error associated. Results of cFP and pFP are also better than the ones achieved with the iRM. The results achieved with the iRM are slightly worse than the ones achieved with the vRMs in cFP and pFP scenarios, which is expected because the vRMs have more samples with a lower distance between neighboring samples. Worse results are observed in the nFP scenario because there is an area of the building without any annotated WSs. In this configuration (nFP+vRM), there is no floor plan to assist in removing particles and there is no iRM, therefore it is not able to annotate WSs in some areas because there is not enough confidence in position estimates. As previously stated, in order to properly construct a vRM in a scenario without a floor plan, it is necessary to use the iRM. For that configuration (nFP+fRM), Wi-Fi fingerprinting results are on par with the ones achieved with the other configurations with a mean error of 1.42 m, P99th of 5.54 m and a maximum error of 6.28 m.

TABLE IV: Plain Wi-Fi fingerprinting results (in meters) of each scenario in comparison to an iRM.

	cFP+vRM	**pFP+vRM**	**nFP+vRM**	**iRM (grid 5x5m)**
Mean	1.32	1.24	6.66	1.50
P$_{75th}$	1.69	1.59	8.34	2.24
P$_{99th}$	3.57	4.18	33.93	5.40
Max	4.98	6.04	34.31	7.51

cFP – complete FP; **pFP** – partial FP; **nFP** – no FP; **vRM** – vehicle RM; **iRM** – initial RM.

3) Comparison with Similar Systems:

In Table V a comparison is made between our approach and other solutions for the construction and maintenance of RMs. It compares the techniques for localization, the sensors used, if the system depends on user's feedback, whether a RM is obtained from a site survey or from interpolation, and the performance of Wi-Fi fingerprinting achieved.

The proposed system achieved the best Wi-Fi fingerprinting results with the automatically built vRMs. The advantage of using dedicated sensors (i.e. IMU and wheel encoder) to track the vehicle's movement is that it leads to lower error in annotated WSs, which consequently leads to better Wi-Fi fingerprinting results. Using Wi-Fi fingerprinting, the proposed solution achieved 1.32 m, 1.24 m and 1.42 m mean error in cFP+vRM, pFP+vRM and nFP+fRM scenarios, respectively. In [16] an unmanned ground vehicle is used to map the space and collect SOP from Wi-Fi, magnetometer and light sensors. It achieved a mean error of 1.89 m with Wi-Fi fingerprinting using the automatically build RM. This solution is more expensive than the proposed one, as it requires several sensors, including a LiDAR. The advantages of this solution are that it does not depend on the initial pose of the device and the floor plan, since it uses the LiDAR sensor to generate the map of the space. In [15], the mobile robot equipped with

TABLE V: Comparison of similar systems for the construction of RMs.

Solution	**Technology**	**Sensor(s) / Device(s)**	**UF***	**Type**	**Initial RM**	**Mean Error (m)**
Proposed solution	Particle Filter	IMU, wheel encoder, Wi-Fi	No	Site survey	No No Yes	1.32 (cFP+vRM) 1.24 (pFP+vRM) 1.43 (nFP+fRM)
[16]	IMLE	LiDAR, magnetometer, light sensor, Wi-Fi	No	Site survey	No	1.89
[15]	Visual Odometry + Particle Filter	Kinect camera, sonars, wheel encoder	No	Site survey	No	5.2
[24]	GP-LVM	Laptop (Wi-Fi)	No	Interpolation	No	3.97
[22]	Wi-Fi fingerprinting	Smartphone (Wi-Fi)	Yes	Site survey	Yes	Between 2 and 4 m
[20]	PDR	Smartphone (Wi-Fi+IMU)	No	Site survey	No	6.9

***UF** – User feedback; **cFP**: complete FP; **pFP** – partial FP; **nFP** – no FP; **vRM** – vehicle RM; **fRM** – full RM.

sensors is capable of locating itself using DR and QR codes which are observed by the camera, thus it does not need an iRM. As a disadvantage, it explores several sensors such as sonars, a wheel encoder, and a camera that are more expensive than the ones used in our system. In [24] Gaussian process latent variable models are applied to estimate the locations of WSs collected by a person walking through a building. This process allows to estimate signal strength levels in the building similar to the ones obtained in an iRM. Systems that depend on users' feedback [21, 22], are susceptible to malicious users that can feed the system with wrong information hindering performance and they also require the iRM. Our solution does not depend on users' feedback since it self-assesses whether the estimated position is reliable enough in order to annotate a WS, besides, it does depend on a RM to function, but it is basic with low density. Without any training (calibration) cost and only relying on PDR, the system in [20] depends on users' daily routine to build RMs, achieving an average error of 6.9 m with Wi-Fi fingerprinting.

V. RM construction and Maintenance - effort

As previously mentioned, the construction and maintenance of RMs is probably the main disadvantage of Wi-Fi-based indoor positioning solutions since it requires time and effort to collect WSs. There are some techniques for optimizing or automating the RM construction process (presented in Section II), however many systems still use manually collected RMs. In this section we compare the time and effort it takes to manually construct a building's RM to the time it takes to construct the RM using the solution presented in this paper, considering the building used in experiments as an example.

The first step in the construction of a RM consists of mapping RPs by measuring and determining the coordinates of each RP. 21 RPs were mapped at the PIEP building for a grid of ≈5 m between neighbor RPs. Then, the surveying process comprises the collection of WSs at the position of each RP. It is important to highlight that many indoor positioning systems based on RMs use denser grids with many more RPs, therefore they take a longer time to complete.

We have spent a total of 5 man/hours to map RPs at PIEP, and 0.6 man/hours to survey the building. The site survey process considered 21 $RPs * 80\ s = 28\ min$, with the collection of 40 WSs per RP (time interval between consecutive WS is around 2 s), additional time is necessary to move from the one RP to the next. The total time spent on the manual construction of the iRM at the PIEP building is 5.6 man/hours, although it is important to note that the duration of these tasks is proportional to the building size because larger buildings require more RPs.

In the performed experiments, it took only 53 minutes (accumulated time of trajectories) to obtain the vRM of the building with zero effort since it explores the vehicles that are operating in the factory to map the space. Contrarily to RMs that have to be manually updated, this solution automatically keeps the RM up to date without additional effort.

VI. Conclusions and Future Work

In this paper we addressed the problem of deploying an IPS in a real industrial environment. The main challenges are related to the construction of the RM and the diversity of industrial spaces that can have different floor plan scenarios. The proposed solution is suitable for different scenarios, i.e., considering a complete Floor Plan (cFP), a partial Floor Plan (pFP) or no Floor Plan (nFP); and allows to automatically build the RM using industrial vehicles equipped with low-cost sensors. The IPS deployment effort is significantly reduced because this solution does not depend on any calibration, installation of infrastructure, or human effort to build the RM. Localization/tracking of vehicles is enabled by a PF that merges Wi-Fi with motion sensors data, allowing vehicles to annotate and add WSs to the vRM based on a reliability metric.

Experiments in an industrial building showed that, with accurate tracking provided by the PF, industrial vehicles can successfully build vRMs in different floor plan scenarios. Wi-Fi fingerprinting results revealed that vRMs perform better than a traditional RM (grid of ≈5 m). VRMs can be used to locate people and improve safety, essential in an industrial context where vehicles and people interact frequently, as well as to locate assets, enhance operations and logistics processes by tracking vehicles. For instance, tracking vehicles allows to gather data that can be used to optimize routes and assets' locations to improve efficiency.

As future work, we intend to conduct experiments with real industrial vehicles and explore ways to optimize vRMs. In a matter of hours, a large number of WSs are annotated and added to the RM, therefore, it is necessary to optimize the RM. Also as future work, we plan to use two low-cost IMUs simultaneously. Sensor fusion techniques may be applied in raw measurements from IMUs to minimize drift and magnetic perturbations that affect the estimated heading. These sensors may also be used to estimate displacement through the integration of accelerometer data. If displacement estimates obtained from low-cost IMUs are reliable, it would allow to improve the displacement estimates or track the vehicle with just the IMU, eliminating the need for the encoder.

References

[1] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM 2000. Conf. Comput. Commun. Ninet. Annu. Jt. Conf. IEEE Comput. Commun. Soc. (Cat. No.00CH37064)*, vol. 2, 2000, pp. 775–784. arXiv: 1106.0222.

[2] S. He and S.-H. G. Chan, "Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[3] Z. Yin, X. Jiang, Z. Yang, *et al.*, "WUB-IP: A high-precision UWB positioning scheme for indoor multiuser applications," *IEEE Syst. J.*, vol. 13, no. 1, pp. 279–288, 2019.

[4] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, Nov. 2015.

[5] H. Zou, Z. Chen, H. Jiang, *et al.*, "Accurate indoor localization and tracking using mobile phone inertial sensors, WiFi and iBeacon," *4th IEEE Int. Symp. Inert. Sensors Syst. Inert. 2017 - Proc.*, no. 1, pp. 1–4, 2017.

[6] A. Koch and A. Zell, "RFID-enabled location fingerprinting based on similarity models from probabilistic similarity measures," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2016-June, pp. 4557–4563, 2016.

[7] Y. Tao and L. Zhao, "A novel system for WiFi radio map automatic adaptation and indoor positioning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10 683–10 692, 2018.

[8] Ying He, Weixiao Meng, Lin Ma, *et al.*, "Rapid deployment of APs in WLAN indoor positioning system," in *2011 6th Int. ICST Conf. Commun. Netw. China*, IEEE, Aug. 2011, pp. 268–273.

[9] P. Mpeis, T. Roussel, M. Kumar, *et al.*, "The Anyplace 4.0 IoT Localization Architecture," in *2020 21st IEEE Int. Conf. Mob. Data Manag.*, vol. 2020-June, IEEE, Jun. 2020, pp. 218–225.

[10] H. Meng, F. Yuan, T. Yan, *et al.*, "Indoor Positioning of RBF Neural Network Based on Improved Fast Clustering Algorithm Combined With LM Algorithm," *IEEE Access*, vol. 7, pp. 5932–5945, 2019.

[11] M. Lee and D. Han, "Voronoi tessellation based interpolation method for Wi-Fi radio map construction," *IEEE Commun. Lett.*, vol. 16, no. 3, pp. 404–407, 2012.

[12] J. Talvitie, M. Renfors, and E. S. Lohan, "Distance-Based Interpolation and Extrapolation Methods for RSS-Based Localization With Indoor Wireless Signals," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1340–1353, Apr. 2015.

[13] C. Luo, L. Cheng, M. C. Chan, *et al.*, "Pallas: Self-Bootstrapping Fine-Grained Passive Indoor Localization Using WiFi Monitors," *IEEE Trans. Mob. Comput.*, vol. 16, no. 2, pp. 466–481, 2017.

[14] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proc. IROS '91IEEE/RSJ Int. Work. Intell. Robot. Syst. '91*, vol. No. 91TH03, IEEE, 1991, pp. 1442–1447.

[15] P. Mirowski, R. Palaniappan, and T. K. Ho, "Depth camera SLAM on a low-cost WiFi mapping robot," *2012 IEEE Conf. Technol. Pract. Robot Appl. TePRA 2012*, vol. 1, no. 908, pp. 1–6, 2012.

[16] J. Tang, Y. Chen, L. Chen, *et al.*, "Fast fingerprint database maintenance for indoor positioning based on UGV SLAM," *Sensors (Switzerland)*, vol. 15, no. 3, pp. 5311–5330, 2015.

[17] C. Esposito and M. Ficco, "Deployment of RSS-Based Indoor Positioning Systems," *Int. J. Wirel. Inf. Networks*, vol. 18, no. 4, pp. 224–242, Dec. 2011.

[18] M. Ficco, C. Esposito, and A. Napolitano, "Calibrating Indoor Positioning Systems with Low Efforts," *IEEE Trans. Mob. Comput.*, vol. 13, no. 4, pp. 737–751, Apr. 2014.

[19] R. Liu, S. H. Marakkalage, M. Padmal, *et al.*, "Collaborative SLAM Based on WiFi Fingerprint Similarity and Motion Information," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1826–1840, 2020. arXiv: 2001.02759.

[20] Y. Kim, Y. Chon, and H. Cha, "Smartphone-based collaborative and autonomous radio fingerprinting," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 1, pp. 112–122, 2012.

[21] A. Moreira and F. Meneses, "Where@UM - Dependable organic radio maps," *2015 Int. Conf. Indoor Position. Indoor Navig. IPIN 2015*, no. October, pp. 1–9, 2015.

[22] Y. Luo, Y. P. Chen, and O. Hoeber, "Wi-Fi-based indoor positioning using human-centric collaborative feedback," *IEEE Int. Conf. Commun.*, 2011.

[23] Y. Ye and B. Wang, "RMapCS: Radio Map Construction From Crowdsourced Samples for Indoor Localization," *IEEE Access*, vol. 6, pp. 24 224–24 238, 2018.

[24] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. 20th Int. Jt. Conf. Artifical Intell.*, 2007, pp. 2480–2485.

[25] S. Kram, C. Nickel, J. Seitz, *et al.*, "Spatial interpolation of Wi-Fi RSS fingerprints using model-based universal kriging," in *2017 Sens. Data Fusion Trends, Solut. Appl.*, vol. 2017-Decem, IEEE, Oct. 2017, pp. 1–6.

[26] C. Pendao and A. Moreira, "FastGraph Enhanced: High Accuracy Automatic Indoor Navigation and Mapping," *IEEE Trans. Mob. Comput.*, vol. 20, no. 3, pp. 1027–1045, Mar. 2021.

[27] I. Silva, A. Moreira, M. J. Nicolau, *et al.*, "Floor Plan-free Particle Filter for Indoor Positioning of Industrial Vehicles," in *ICL-GNSS Int. Conf. Localization GNSS*, Tampere, Finland, 2020.

[28] J.-A. Fernández-Madrigal and J. L. Blanco Claraco, *Simultaneous Localization and Mapping for Mobile Robots*, 2. IGI Global, Ed., ser. Advances in Computational Intelligence and Robotics. IGI Global, 2013, ch. Appendix B, pp. 407–411.

[29] A. Moreira, I. Silva, F. Meneses, *et al.*, "Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning," in *2017 Int. Conf. Indoor Position. Indoor Navig.*, Sapporo, Japan: IEEE, Sep. 2017, pp. 1–8.

[30] S. O. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," University of Bristol, Tech. Rep., 2010.

Ivo Silva received the M.Sc. degree in telecommunications and informatics engineering from the University of Minho, Guimaraes, Portugal, in 2016, where he is currently pursuing a PhD degree in telecommunications focused on indoor positioning of industrial vehicles based on Wi-Fi. He is a Researcher with the Algoritmi Research Centre and an Invited Assistant Professor with the University of Minho. His research interests are in indoor positioning and navigation, mobile computing, and smart devices.

Cristiano Pendão received the M.Sc. in Telecommunications and Informatics Engineering in 2012 from the University of Minho, Portugal and the PhD degree in the fields of Telecommunications/Computer Science Engineering from the Universities of Minho, Aveiro and Porto in 2019. He is a Professor at the School of Engineering, University of Minho, and a researcher at the Algoritmi Research Centre. His research interests are in positioning and navigation systems, computer vision, and mobile computing.

Adriano Moreira is an Associate Professor, with Habilitation, at the University of Minho, and a researcher at the Algoritmi Research Centre. He received a degree in Electronics and Telecommunications Eng. and a PhD degree in Electrical Eng., respectively in 1989 and 1997, from the University of Aveiro. He co-founded the Computer Communications and Pervasive Media research group, and is the Director of the MAP-tele doctoral program in Telecommunications. His research activities have been taking place within the ubicomp@uminho research sub-group, which has been focusing on the creation of technologies for smart places. He participated in many research projects funded by national and EU programs. He is the author of several scientific publications in conferences and journals, and one patent in the area of computational geometry. Together with his colleagues, won the 1st prize on the off-site track of the EvAAL-ETRI Indoor Localization Competition (IPIN 2015 and 2017) and the 2nd prize of the corresponding competition in 2016.