

Industrial Environment Multi-Sensor Dataset for Vehicle Indoor Tracking with Wi-Fi, Inertial and Odometry Data

Ivo Silva ^{1,*}, Cristiano Pendão ^{1,2,*}, Joaquín Torres-Sospedra ¹ and Adriano Moreira ¹

¹ Centro ALGORITMI, Universidade do Minho, Campus de Azurém, 4800-058 Guimarães, Portugal; jtorres@algoritmi.uminho.pt (J.T.-S.); adriano@dsi.uminho.pt (A.M.)

² Department of Engineering, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal

* Correspondence: ivo@dsi.uminho.pt (I.S.); cpendao@dsi.uminho.pt (C.P.)

Abstract: This paper describes a dataset collected in an industrial setting using a mobile unit resembling an industrial vehicle equipped with several sensors. Wi-Fi interfaces collect signals from available Access Points (APs), while motion sensors collect data regarding the mobile unit's movement (orientation and displacement). The distinctive features of this dataset include synchronous data collection from multiple sensors, such as Wi-Fi data acquired from multiple interfaces (including a radio map), orientation provided by two low-cost Inertial Measurement Unit (IMU) sensors, and displacement (travelled distance) measured by an absolute encoder attached to the mobile unit's wheel. Accurate ground-truth information was determined using a computer vision approach that recorded timestamps as the mobile unit passed through reference locations. We assessed the quality of the proposed dataset by applying baseline methods for dead reckoning and Wi-Fi fingerprinting. The average positioning error for simple dead reckoning, without using any other absolute positioning technique, is 8.25 m and 11.66 m for IMU1 and IMU2, respectively. The average positioning error for simple Wi-Fi fingerprinting is 2.19 m when combining the RSSI information from five Wi-Fi interfaces. This dataset contributes to the fields of Industry 4.0 and mobile sensing, providing researchers with a resource to develop, test, and evaluate indoor tracking solutions for industrial vehicles.

Keywords: Industry 4.0; datasets; fingerprinting; motion sensors; industrial vehicles; indoor tracking; indoor positioning; Wi-Fi; IMU; encoder



Citation: Silva, I.; Pendão, C.; Torres-Sospedra, J.; Moreira, A. Industrial Environment Multi-Sensor Dataset for Vehicle Indoor Tracking with Wi-Fi, Inertial and Odometry Data. *Data* **2023**, *8*, 157.

<https://doi.org/10.3390/data8100157>

Academic Editors: Giuseppe Ciaburro and Juanle Wang

Received: 10 July 2023

Revised: 12 October 2023

Accepted: 17 October 2023

Published: 23 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Industry 4.0 is a generation of manufacturing development driven by disruptive trends in numerous areas, including the Internet of Things (IoT) [1], deep learning [2], and edge computing [3], among others [4]. Industrial vehicles have an important role in manufacturing, participating in the transport of raw materials and finished goods as well as moving materials within factories. Factories benefit from tracking vehicles to monitor and control operations, improve logistics and safety, and also to enable inter-connectivity between vehicles, other machines and even human operators. Contrarily to outdoor environments where Global Positioning System (GPS) is used to locate vehicles [5], in indoor environments, satellite-based positioning systems are not reliable. Alternatively, indoor vehicles can be localised using Wi-Fi-based positioning systems [6]. Wi-Fi is ubiquitous, being present in most environments, including industrial buildings; hence, its infrastructure can be explored for localisation without additional costs.

Wi-Fi fingerprinting [7,8] is one of the most used positioning techniques based on Wi-Fi. It consists of two phases, the calibration (or offline) phase and the online phase. In the calibration phase, a radio map is built by collecting Wi-Fi samples in known locations, which are also known as reference points. In the online phase, an operational sample is compared against the radio map using a (dis)similarity function to find the radio map samples that are more similar to the operational sample. Then, a position estimate is obtained using

an algorithm (e.g., the k-Nearest Neighbour (k-NN)), based on the most similar radio map samples. Despite being low cost and simple to implement, Wi-Fi fingerprinting has a few drawbacks: namely, it requires the collection of the radio map, and it is prone to large errors [9], which occur due to numerous reasons such as changing the indoor layout, adding/removing APs, or propagation effects.

Although Wi-Fi may provide an absolute position, additional sensors are necessary to accurately track industrial vehicles. The vehicle's movement can be tracked using sensors that measure the heading (orientation) and the displacement (travelled distance). IMU sensors allow measuring the absolute heading of the vehicle, detecting which way it is going towards, and wheel encoders may be used as odometers to measure the travelled distance. The vehicle's trajectory based on motion sensors can be determined with dead reckoning. An initial position must be provided, and then upon receiving new observations, the trajectory is updated based on the previous position. This approach is prone to cumulative errors because the IMU sensor is affected by bias and noise, leading to drift in the heading. Therefore, the position should be corrected after some time to avoid large errors.

Combining Wi-Fi with motion sensor data mitigates the drawbacks of both Wi-Fi fingerprinting and dead reckoning [10]. For instance, Wi-Fi fingerprinting can provide the absolute position necessary for the initial position of dead reckoning and correct the drift in the estimated trajectory. In addition, large positioning errors may be reduced when combining Wi-Fi with motion sensor data. Kalman filters and particle filters are two of the most-known sensor fusion approaches to fuse Wi-Fi with motion sensors [6,11,12].

Testing and evaluation are crucial parts of the development of an Indoor Positioning System (IPS), usually requiring real-world experiments to collect data for validation and testing. When the objective of the IPS is to localise and track industrial vehicles equipped with multiple sensors, this process comprises several stages:

1. Preparing the setup where experiments are conducted:
 - (a) Defining the reference points for the radio map;
 - (b) Mapping the locations of the APs and reference points, measuring their positions with respect to the building's coordinate reference system.
2. Preparing the hardware and software:
 - (a) Physically connect all sensors to a computer;
 - (b) Develop sensor acquisition software;
 - (c) Test software collecting data from all sensors simultaneously.
3. Perform calibration by collecting the radio map at reference points;
4. Collect the sensor data and ground truth in multiple trajectories.

There are several public Wi-Fi datasets for Wi-Fi-based positioning systems, [13–19], which are collected in a variety of scenarios, including universities, office buildings, shopping malls, and industrial factory-like space. There are also hybrid datasets, namely, with Wi-Fi and Bluetooth Low Energy (BLE) data [20], with Wi-Fi, BLE, and Zigbee [21], with Wi-Fi, BLE, and magnetometer data [22], with BLE and IMU data [23], or even with Wi-Fi, BLE, cellular signal and multi-sensor data (magnetometer, accelerometer, gyroscope, barometer, and ambient light sensor) [24]. The International Conference on Indoor Positioning and Indoor Navigation (IPIN) provides their competitions' datasets, containing multi-sensor data along with ground truth [25–34], which are available at <https://ipin-conference.org/resources.html> (accessed on 7 July 2023). In 2021, Microsoft co-organised the Indoor Location Competition 2.0. The competition introduced a first-of-its-kind large-scale indoor location benchmark dataset [35], containing dense indoor signatures of Wi-Fi, an geomagnetic field, BLE iBeacons, and ground truth locations collected by smartphones from numerous buildings in Chinese cities. Despite the existence of these public datasets, none of them incorporate data gathered specifically in an industrial setting from multiple sensor types, including Wi-Fi. Public access datasets are usually collected at office buildings, laboratories or controlled scenarios. Therefore, they are not considering some relevant features present in industrial layouts, such as wide-open areas

with metallic machinery, high ceilings and constraints on where to place the Wi-Fi APs. Existing datasets on office buildings are useful for the research community but often include narrow large corridors, small offices and locations of APs that do not mimic a real industrial setup. Additionally, while datasets for mobile robots in indoor environments exist, encompassing data from laser scanners, cameras, and odometry [36,37], none of these datasets are tailored for indoor vehicles equipped with Wi-Fi interfaces and motion sensors to accurately track their movement.

In industrial settings, it is difficult to conduct real-world experiments in the production area where operators, robots and vehicles are moving in space; therefore, datasets are essential for the development, prototyping and validation of novel IPSs. This allows researchers to focus mostly on the development of the IPSs instead of needing to perform the above-mentioned stages to prepare the experiment's setup and collect sensor data.

In this paper, we present a dataset for vehicle tracking in indoor environments, which includes Wi-Fi and motion sensor data. We fully describe the dataset, how it was collected, and show examples of data usage.

To the best of our knowledge, there is no other open-source dataset with these characteristics: (1) data collection at an industrial building with large open spaces and heavy machinery; (2) Wi-Fi data from multiple synchronous Wi-Fi interfaces, which has been shown to improve positioning performance [38]; (3) orientation data from two low-cost IMUs, providing raw accelerometer, gyroscope, and magnetometer data as well as absolute orientation provided by integrated sensor fusion algorithms; (4) displacement data provided by an absolute encoder attached to the wheel; and (5) ground-truth data determined by a computer vision approach which automates the annotation process and improves accuracy by providing not only the position but also the orientation of the mobile unit. The objective of this paper is to share this indoor tracking dataset with the community, providing an additional resource for the development, evaluation and benchmarking of IPSs. This contribution is especially valuable for the research community focused on positioning solutions for Industry 4.0 and IoT. A potential use of this dataset is for the development of solutions for vehicle automation and mobile robots, as it contains data from sensors commonly found in these solutions. This dataset will also enable the research community to test their Machine/Deep Learning proposals for PDR [39,40], Wi-Fi fingerprinting [41,42] or a combination of both [43,44] with external datasets.

The remainder of this paper is organised as follows. A detailed description of the dataset is made in Section 2. The data collection approach is presented in Section 3, describing the building where data were collected and the software and hardware necessary to collect and store the data. An analysis of the data is made in Section 4, providing several details and statistics about the dataset. Examples of two possible data uses are provided in Section 5, using two baseline methods, namely, dead reckoning and Wi-Fi fingerprinting. Finally, the conclusions are presented in Section 6.

2. Data Description

The dataset is structured in several folders inside the data folder, whose contents are as follows:

- T1–T6 trajectory sub-folders—each sub-folder contains the data relative to a trajectory;
- RadioMap sub-folder—contains a set of Wi-Fi samples collected at known locations, also known as radio map, which can be used for Wi-Fi fingerprinting-based positioning systems;
- aps.csv—contains the information regarding the Wi-Fi APs detected in the building. APs whose IDs start with 2, e.g., 2XXX, are the ones whose position is not known. The positions of the other nine APs (IDs 0XXX and 1XXX) are included in this file;
- tags.csv—contains the information regarding the radio map reference points;
- floor_plan.jpg—the image of the building's floor plan, which can be used for visualisation purposes or sensor fusion approaches that explore the floor plan information (e.g., particle filters);

- `floor_plan_cal.jpg`—the image of the building’s floor plan with four calibration points marked, which can be helpful for plotting and using the floor plan information;
- `floor_plan_cal.csv`—contains coordinates of the calibration points, which are marked in the `floor_plan_cal.jpg` image.

2.1. Trajectory Files

Each trajectory sub-folder contains nine sub-sets of data (sub-files), as follows:

$$D_j = \{E, I_1, I_2, W_1, W_2, W_3, W_4, W_m, GT\} \quad (1)$$

where j defines the number of the trajectory with $j \in \{1, \dots, 6\}$. E represents the set of samples from the encoder sensor in the `enc.csv` file; I_1 and I_2 represent the set of samples from both IMU sensors in `imu_1.csv` and `imu_2.csv`, respectively; $W_1, W_2, W_3,$ and W_4 represent the set of samples from Wi-Fi interfaces in `rss_1.csv, rss_2.csv, rss_3.csv,` and `rss_4.csv`, respectively; W_m represents the set of averaged Received Signal Strength (RSS) values from all Wi-Fi interfaces in `rss_m.csv`; and GT represents the set of Ground Truth (GT) records in `ground_truth.csv`.

2.1.1. Encoder Data

Encoder samples in the `enc.csv` file are defined as follows:

$$e = \{t, d, \theta\} \quad (2)$$

where t represents the time in milliseconds, d represents the displacement since the previous sample (in meters), and θ defines the absolute angle at which the wheel is positioned at (value between 0 and 2π , in radians). An example of the encoder data from a CSV file is provided in Figure 1.

```
time,displacement,angle
0.000,0.001199913860746,2.672099084803318
0.021,0.001090830782496,2.689552377323262
0.041,0.001418080017245,2.712241657599188
0.061,0.001418080017245,2.734930937875114
0.081, ...
```

Figure 1. Example of encoder data file from trajectory T1.

2.1.2. IMU Data

IMU samples in the `imu_i.csv` files (Figure 2) are defined as follows:

$$imu = \{t, q_w, q_x, q_y, q_z, roll, pitch, yaw, gyr_x, gyr_y, gyr_z, acc_x, acc_y, acc_z, mag_x, mag_y, mag_z\} \quad (3)$$

where t represents the time in milliseconds, (q_w, q_x, q_y, q_z) represent the orientation in quaternion format, $(roll, pitch, yaw)$ represent the orientation in Euler angles (degrees), and $gyr, acc,$ and mag tri-axis values represent the raw data from the gyroscope, accelerometer, and magnetometer sensors, which measure the angular rate (rad/s), acceleration (m/s^2) and magnetic field (μT), respectively. An example of the IMU data from a CSV file is provided in Figure 2.

```
time,v,x,y,z,roll,pitch,yaw,gyro_x,gyro_y,gyro_z,acce_x,acce_y,acce_z,magn_x,magn_y,magn_z
0.038,0.999755859375,-0.0228271484375,0.003173828125,-0.0001220703125,-0.3125,2.625,90.0,0.0,0.0,0.0087266462599716,0.43,-0.5700000000000001,9.44,-84.6875,-2.5625,-18.5625
0.089,0.999755859375,-0.02313232421875,0.00152587890625,-0.0001220703125,-0.125,2.625,90.0,0.0054541539124822,0.0010908307824964,0.0098174770424681,0.43,-0.39,9.45,-83.5625,-3.25,-18.875
0.137,0.999755859375,-0.02313232421875,0.00128173828125,0.000244140625,-0.0625,2.625,90.0625,-0.0010908307824964,0.0032724923474893,0.0229074464324255,0.28,-0.4,9.62,-84.0,-2.875,-18.5625
0.192,0.999755859375,-0.02301025390625,0.0008544921875,0.0006103515625,-0.0625,2.625,90.125,0.0021816615649929,0.0021816615649929,0.0087266462599716,0.25,-0.36,9.35,-51.5625,-1.75,-10.5
0.241, ...
```

Figure 2. Example of IMU data file from T1.

2.1.3. Wi-Fi Data

Wi-Fi samples in the `rss_i.csv` files are defined as follows:

$$w = \{t, rss_1, \dots, rss_N\} \quad (4)$$

where t represents the time in milliseconds, and rss_i represents the RSS value of the i -th AP (in dBm), with a total of 27 APs. A default RSS of -120 dBm is assigned to the APs that were not detected in the Wi-Fi sample. An example of the Wi-Fi data from a CSV file is provided in Figure 3.

```
time,0001,0002,0003,1001,1002,1003,1004,1005,1006,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017
0.701,-50,-52,-28,-32,-41,-44,-37,-28,-32,-33,-33,-55,-52,-77,-120,-120,-120,-120,-120,-120,-24,-120,-120,-120,-120,-120,-120
2.319,-49,-56,-34,-27,-41,-41,-32,-25,-27,-32,-32,-53,-52,-80,-120,-120,-120,-120,-120,-120,-29,-120,-120,-120,-120,-120,-120
4.005,-49,-56,-39,-24,-40,-42,-35,-25,-25,-29,-30,-52,-52,-82,-120,-120,-120,-120,-120,-120,-27,-120,-120,-120,-120,-120,-120
5.614,-50,-50,-39,-27,-38,-42,-35,-22,-32,-32,-33,-52,-50,-82,-120,-120,-120,-120,-120,-120,-29,-120,-120,-120,-120,-120,-120
7.234,...
```

Figure 3. Example of Wi-Fi data file from T1.

2.1.4. Ground Truth Data

The location of the trajectory samples, also known as ground truth positions (GT) in the `ground_truth.csv` file, is defined as:

$$gt = \{t, tag, p_x, p_y, p_z, \phi\} \tag{5}$$

where t represents the time in milliseconds, tag is the identifier of the point, (p_x, p_y, p_z) are the Cartesian coordinates of the GT position, and ϕ is the GT orientation (in degrees). An example of the GT data from a CSV file is provided in Figure 4.

```
time,tag,x,y,z,orientation
0.033,42,-5.1746,5.1034,0.975,90.2836
0.066,42,-5.1746,5.1037,0.975,90.2836
0.100,42,-5.1741,5.1046,0.975,90.0000
0.133,42,-5.1736,5.1058,0.975,89.7149
0.166, ...
```

Figure 4. Example of ground truth data file from T1.

2.2. Radio Map

The radio map sub-folder includes the six sub-sets of data, as follows:

$$RM = \{W_1, W_2, W_3, W_4, W_m, RP\} \tag{6}$$

where W_i represents the set of samples from each Wi-Fi interface, W_m represents the set of averaged RSS values from all Wi-Fi interfaces, and RP represents the set of reference points where Wi-Fi samples were collected. Wi-Fi samples included in `rss_i.csv` files are defined by Equation (4), and the reference points included in `ref_pos.csv` file have the same format as GT points (Equation (5)).

The radio map is made of 40 Wi-Fi samples per reference point, with 10 Wi-Fi samples collected at each of four directions turning clockwise, as shown in Figure 5. The mobile unit was placed on the reference point position (RP), and then it was turned to face each one of the four directions while collecting Wi-Fi samples.

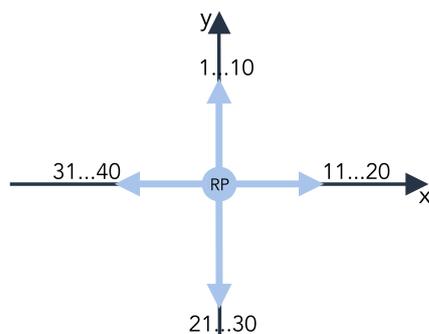


Figure 5. Mobile unit orientations while collecting Wi-Fi samples: 10 Wi-Fi samples were collected at each orientation for a total of 40 Wi-Fi samples at each reference point.

2.3. Supporting Software

In addition to the dataset, several Python scripts are provided in the code folder:

- `data_analysis.py`—main file that parses the dataset, performs a statistical analysis, and executes Wi-Fi fingerprinting and dead reckoning algorithms as examples of data usage;
- `plots.py`—contains code to generate all plots presented in this paper;
- `indoor_positioning.py`—includes Wi-Fi fingerprinting and dead reckoning algorithms as well as the methods to compute the positioning error;
- `config.py`—used as a configuration file that holds several values to configure, including the path to the dataset folder, the folder where results and plots are saved into and also Wi-Fi fingerprinting parameters, the k values and distance functions, that can be Manhattan (city-block) or Euclidean;
- `requirements.txt`—lists the required Python packages to run the code.

3. Data Collection Approach

The main architecture of the data collection system, depicted in Figure 6, represents each module and the software technology used to collect data from each sensor. Data from the Wi-Fi interfaces are obtained from a multi-threaded Java application that saves data from each interface into a separate Comma-Separated Values (CSV) file. Similarly, data from the IMUs are obtained from a Python application, running separately for each IMU sensor. This application is also multi-threaded to avoid blocking when it is writing to the files. Data from IMUs are exported into separate CSV files. Data from the encoder sensor are obtained from a Python application and exported into a CSV file. Ground truth data are obtained from a video that records the **gt!** tags, placed on the floor, when the mobile unit moves. A Python application processes the video to obtain the true position and heading of the mobile using, which are exported into a CSV file.

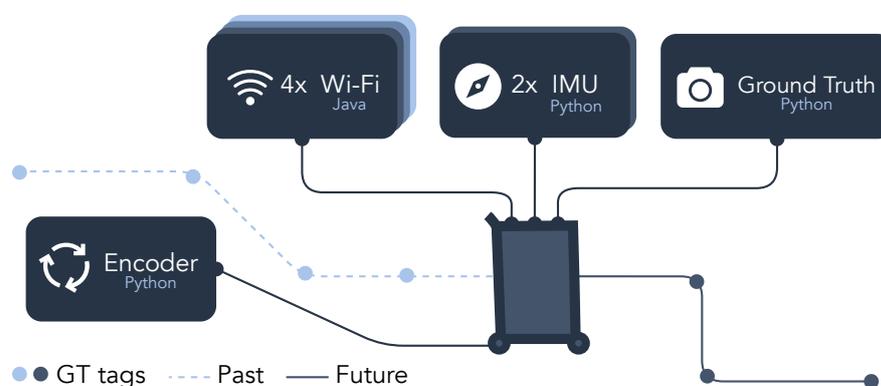


Figure 6. Data collection process: Wi-Fi data are collected from a Java application; data from the two IMUs were collected by a Python application; data from the encoder sensor were obtained from another Python application; ground truth data were collected by a video camera and processed by a Python application. Output data from all applications are exported to CSV files.

The sensor samples are synchronised in time because the data collection programs for each sensor are executed on the same computer. Consequently, when retrieving a new sensor record, these programs acquire the timestamp from the operating system, enabling them to share a common time reference.

The data collection took place at the PIEP building, which is part of the University of Minho's Azurém Campus. PIEP (Figure 7) is a Centre for Innovation in Polymer Engineering; hence, it has an open-space area quite similar to a factory plant with plastic extrusion machines. The building measures 20 m by 50 m and has a Wi-Fi infrastructure with several APs installed inside. All APs are emitting in the 2.4 GHz frequency band. Figure 8 shows the floor plan, the positions of APs and their IDs as well as the reference

points where Wi-Fi samples were collected to build the radio map. These reference points also mark the locations where GT data were collected. Grey areas in the floor plan represent obstacles or non-navigable areas.

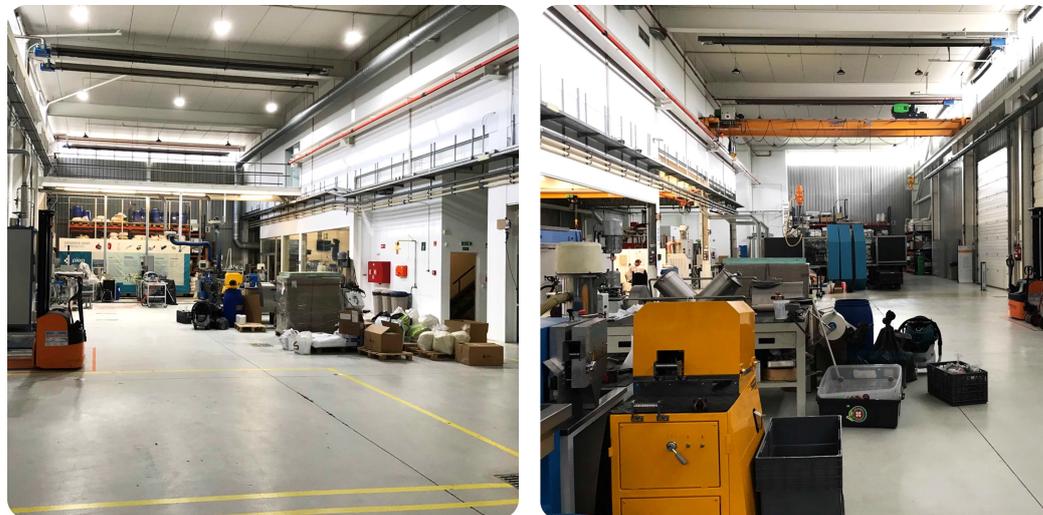


Figure 7. PIEP building (industrial space) at the University of Minho.

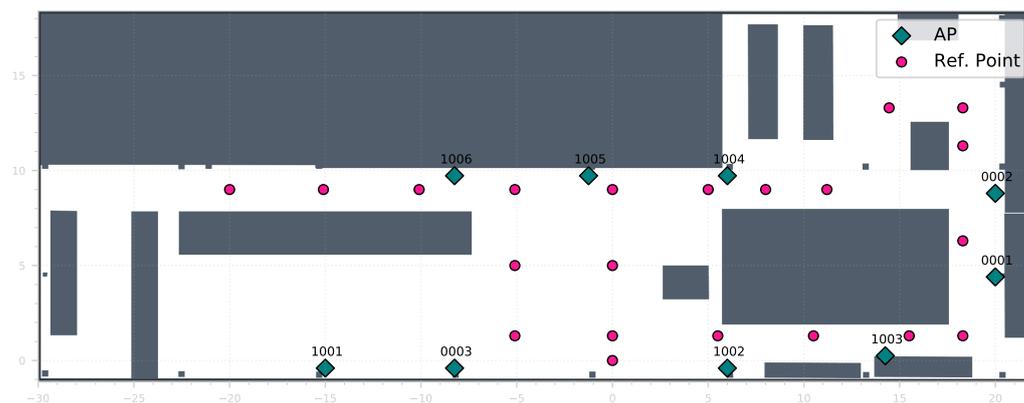


Figure 8. Floor plan of the PIEP building, including the deployed APs and radio map reference points.

Table 1 includes details about the deployed APs, namely, their ID, brand and model, and the transmission channel of each AP in the 2.4 GHz frequency band. These APs are the ones whose positions are known. Despite that, there are also other APs available in the building whose positions are not known.

Table 1. Configuration of deployed APs.

AP ID	Brand/Model	Channel	Frequency
0001	ORiNOCO AP200	1	2412 MHz
0002	ORiNOCO AP200	6	2432 MHz
0003	ORiNOCO AP200	9	2452 MHz
1001	Cisco Aironet 1100 series	1	2412 MHz
1002	Cisco Aironet 1100 series	6	2432 MHz
1003	Cisco Aironet 1100 series	11	2462 MHz
1004	Cisco Aironet 1100 series	9	2452 MHz
1005	Cisco Aironet 1100 series	3	2422 MHz
1006	Cisco Aironet 1100 series	7	2442 MHz

3.1. Mobile Unit

A manually pushed trolley is used to emulate an industrial vehicle equipped with several sensors. The trolley measures $40 \times 60 \times 97.5$ cm ($W \times D \times H$). Figure 9 represents the relative positions of the Raspberry Pi (RPI) and sensors with respect to the mobile unit. These devices/sensors are listed below:

- 1 \times Raspberry Pi 3B+: computer that runs programs to collect data from sensors;
- 1 \times Absolute Encoder (US Digital A2): Wheel-attached sensor at 62.5 mm height that measures the displacement;
- 2 \times IMU sensors (Adafruit BNO055): Low-cost IMU sensors providing absolute orientation and/or raw magnetometer, gyroscope, and accelerometer measurements;
- 4 \times Wi-Fi interfaces (Edimax EW7811-Un): external Wi-Fi interfaces compliant with IEEE 802.11 b/g/n (2.4 GHz frequency band).

The laptop is used to remotely control the data collection process by connecting to the RPi using the Secure Shell (SSH) protocol.

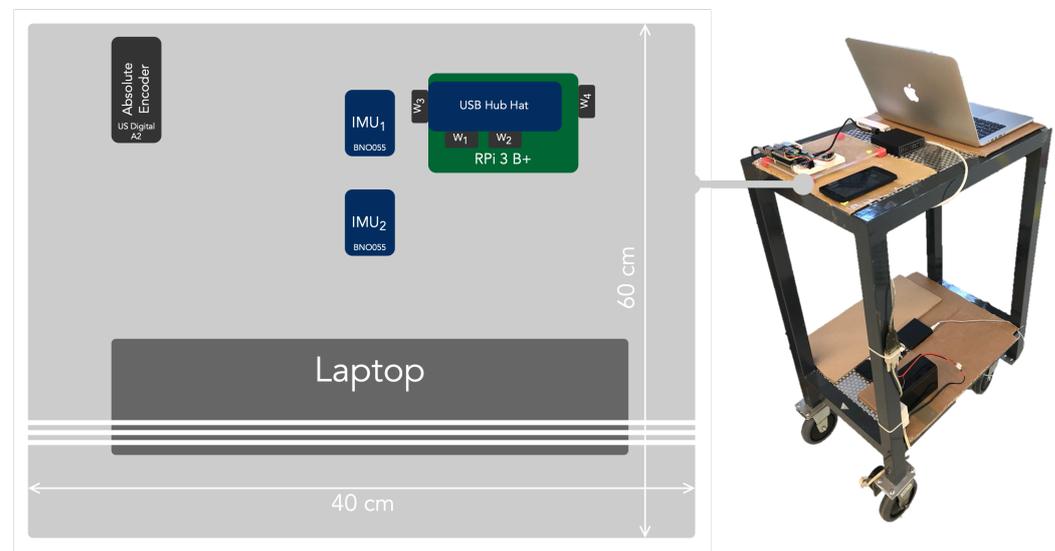


Figure 9. Sensors placement (position) on the mobile unit (depth is minimised to reduce image space).

3.2. Encoder

In this work, we opted to use a rotary encoder attached to one of the vehicle's wheels as an odometer to measure the displacement. Rotary encoders can be absolute or incremental. Absolute encoders allow reading the absolute angle at which the wheel is positioned whenever a new reading is obtained. Incremental encoders generate a series of pulses during movement, which can be converted into a velocity from the previous sample. Another difference between absolute and incremental encoders is that when it is first powered, the absolute encoder always reports a known position (wheel position), while the incremental encoder always starts from zero and acts as a counter for each pulse generated. We selected the US Digital A2 absolute encoder to measure the displacement as it is accurate, reliable, and it is simple to read and convert the measured angles into a distance value. The sample rate was configured to 50 Hz, i.e., an interval of 20 ms between consecutive readings.

Converting Absolute Encoder Angle into Displacement

Given that the absolute encoder reports the angle at which the wheel is positioned, at least two angle readings are necessary to determine the angle difference. Assuming that the wheel does less than one complete turn between two consecutive samples, the angle difference can be converted into a distance value as follows:

$$d = \frac{\angle AB}{360} \times 2\pi r \quad (7)$$

where d represents the measured displacement in the time interval between t_0 and t_1 , $\angle AB$ represents the angle (in degrees) measured in the same time interval, and r represents the wheel radius. The mobile unit's wheels have a radius of $r = 62.5$ mm.

3.3. IMU

We opted for low-cost sensors to evaluate their feasibility in real-world applications; therefore, we decided to include two IMU sensors in the mobile unit. We selected the Adafruit BNO055 IMU; since it has several fusion modes and provides the absolute position, it can be easily connected to the RPi, it is low-cost, and it is simple to work with. The IMU's RPi connection setup is shown in Figure 10. In order to connect two BNO055 sensors to the same RPi device, it is necessary to change the default I2C address in one of the IMUs. One device has the default I2C address, which is 0x28. And the device where the ADR pin is connected to 3.3 V has a default I2C address of 0x29.

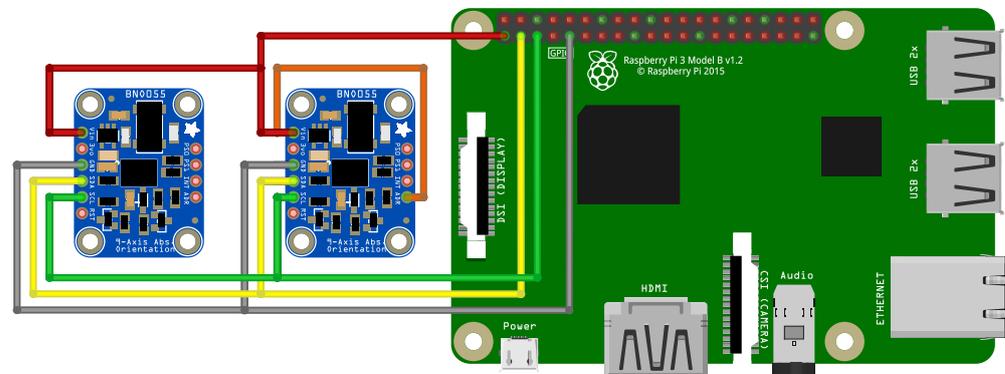


Figure 10. RPi GPIO connection setup to connect the IMUs enabling simultaneous data collection from both sensors.

Data collection was performed with a Python program composed of two threads with different purposes. One is responsible for obtaining the data from the sensor, and the other is responsible for writing the data into a CSV file. They implement a producer–consumer approach, where the thread that reads data from the sensor is the producer, and the thread that writes the data to the CSV file is the consumer. This approach reduces possible delays in the data collection process, which would be caused in case the process was synchronous with one thread, because writing to the file could block and cause delays in the data collection. Configuration and reading data from the sensor were permitted by the custom CircuitPython driver for the BNO055 sensor (https://github.com/adafruit/Adafruit_CircuitPython_BNO055 (accessed on 7 July 2023)).

The sensor was configured with the NDOF operation mode (<https://www.bosch-sensortec.com/products/smart-sensors/bno055/> (accessed on 7 July 2023)) that includes a sensor fusion algorithm where the fused absolute orientation data are calculated from the accelerometer, gyroscope and magnetometer. The absolute orientation is provided in quaternion and Euler angles formats. In addition to the absolute orientation, this operation mode also outputs raw data from the accelerometer, magnetometer (compass), and gyroscope.

The angle range for the rotation angles for roll, pitch, and yaw varies as follows:

- Pitch: -180° to 180° ;
- Roll: -90° to 90° (increasing with increasing inclination);
- Yaw: -180° to 180° (turning clockwise decreases value).

Although the sensor outputs the yaw ranging from 0° to 360° , we converted it into the -180° to 180° format. It is also relevant to mention that the yaw returned from the sensor increases the reported value when turning clockwise, but we inverted the yaw so that it

decreases when turning clockwise. We did this because by inverting the yaw, the angle varies according to the Cartesian referential (positive x -axis points to 0° , positive y -axis points to 90°).

No prior calibration is made before collecting data from the IMU sensor, because we assumed that the mobile unit emulates an industrial vehicle that initiates operation from a cold start without any sensor calibration. Despite that, the NDOF mode applies a fast magnetometer calibration, which results in a quick calibration of the magnetometer and higher output data accuracy.

The IMU sample rate was set to 20 Hz. Although the program was configured to obtain a new sample every 50 ms, there were times when the sensor took more than that time to return the orientation data; hence, in some cases, a small delay was observed.

IMU Angle Adjustment

Whenever starting the collection of data in a new trajectory, the selected IMU reports a yaw angle of 0° independently of the orientation of the mobile unit. In order for the reported yaw value to match the angle used in the Cartesian referential, an adjustment was made to yaw values, shifting the angle according to the initial orientation of the mobile unit.

$$yaw = yaw_r + yaw_i \quad (8)$$

where yaw represents the converted yaw (the absolute orientation), yaw_r represents the raw value obtained from the sensor, and yaw_i represents the initial angle at which the mobile unit started the trajectory. In cases where $yaw < -180^\circ$ or $yaw > 180^\circ$, an adjustment was made so that the yaw is within the range between -180° and 180° .

3.4. Wi-Fi

We used *four* Wi-Fi interfaces to collect signal strength data from multiple sources because signals from these interfaces are uncorrelated [38]. Hence, signal strength values from multiple interfaces can be averaged in a Wi-Fi sample to reduce noise.

The Wi-Fi sample rate is deeply dependent on hardware and software; in this particular case, an RPi with the Raspberry Pi OS (previously called Raspbian) was used. Raspberry Pi OS is a Debian-based Linux distribution; hence, to scan Wi-Fi, one may use Linux commands to obtain information from a Wi-Fi interface. The time it takes to perform a scan depends on the operating system and the list of channels to scan (2.4 GHz and/or 5 GHz band). Usually the Wi-Fi standard supported by the Wi-Fi interface specifies which channels are supported, e.g., IEEE 802.11 b/g/n support 2.4 GHz and IEEE 802.11 a/n/ac support 5 GHz.

The software used to collect Wi-Fi signals comprises a Java application that runs operating system commands to scan Wi-Fi interfaces. In this application, a main thread manages several other threads; each is responsible for scanning a specific Wi-Fi interface. The program runs cyclically as follows:

1. The main thread starts the scanning process by giving the start command to the scanning threads;
2. It waits for each scanning thread to finish scanning;
3. It gathers scans from all Wi-Fi interfaces;
4. When the last thread returns the scan information, the main thread assigns a timestamp to the Wi-Fi scans from all interfaces so that the same timestamp is shared.

This approach was adopted to keep data from Wi-Fi interfaces synchronised in time.

Threads dedicated to scanning Wi-Fi interfaces run the Linux command “`sudo iwlist wlan0 scanning`”, obtain the result of the command, and parse the relevant information (AP’s MAC, channel, Received Signal Strength Indicator (RSSI), link quality). After parsing the information, each thread returns the Wi-Fi scan data to the main thread.

3.5. Ground Truth

Ground truth (GT) is a crucial element of datasets as it serves two important purposes. Firstly, it can be used for training, such as fine-tuning algorithms. Secondly, it is essential for the validation and evaluation stages where it can be used to calculate the positioning error.

Various methods can be used to obtain GT, depending on the application and experimental scenario. Typically, the accuracy of the reference system is directly related to the accuracy that the positioning solution being tested is expected to achieve. It is desirable to have an order of magnitude of higher accuracy in the reference system. In indoor positioning, the reference system can be based on a higher-performing technology. For example, one can use an ultrawide-band (UWB) system as a reference to evaluate a Wi-Fi fingerprinting solution or perform experiments in a space with a specialised high-accuracy camera system to track the user position. However, the use of advanced solutions such as the ones described is often impossible due to the high costs and complexity involved and is not practical for large data collection campaigns (e.g., multiple buildings and floors).

A simpler solution can also be explored with good results. In Track 3 of the IPIN competition, the GetSensorData App is used to collect data. To obtain GT, users press a button when they are over a GT point, and the time instant is marked in a log file. This process is more suitable for collecting data in pedestrian applications. However, the accuracy of the GT depends on the user's precision in marking the exact instant when over the reference point, which can be challenging to do without stopping at the reference point.

The dataset presented in this paper can be used in indoor vehicle positioning and tracking solutions that can achieve relatively high accuracy through multi-sensory fusion (0.5 m or less). In a previous experiment, we obtained GT using a grid of GT tags and a camera mounted on the vehicle to record the path. In an offline stage, we manually annotated the time instant when the vehicle passed over each GT point. However, for longer data collection campaigns, the manual process can be significantly effortful and may lead to errors. One of our objectives for this dataset was to improve the quality of GT data. To achieve this, we adopted an approach based on computer vision to automate the process of obtaining GT and increase its accuracy.

First, a grid of GT points was established in the space. The distance between neighbour GT points was about 5 m with a total of 22 points. A GT reference point was mapped in the space as (0, 0) coordinate, and the remaining points were mapped in relation to the reference point using accurate laser distance meters (see Figure 8). All direction changes (turns) are performed over a GT point.

ArUco markers were used as GT tags, and a video camera pointed to the ground was mounted on the mobile unit. The camera position and reference point in the vehicle, as well as the visible area, were accurately calibrated. The video camera records the GT tags when the mobile unit passes by them.

The video feed obtained by the camera is processed using computer vision techniques in order to detect the ArUco markers, obtaining the tag ID as well as the pose of the vehicle in relation to the tag (Figure 11).

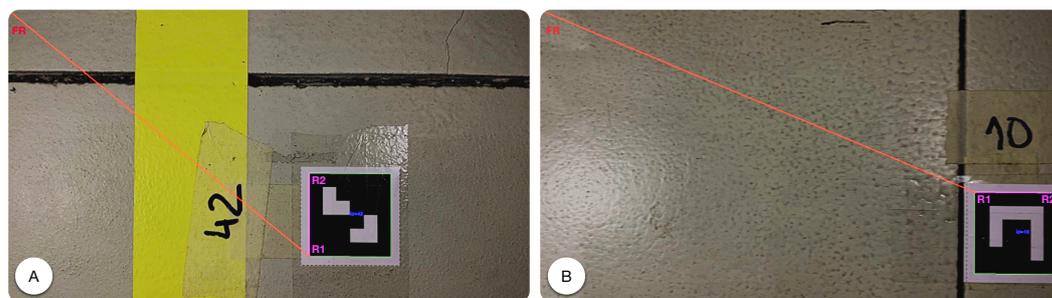


Figure 11. Ground truth using ArUco markers detection: (A) Tag 42, (B) Tag 10. FR represents the frame reference point; R1 and R2 represent two marker reference points used to compute the marker rotation.

When the marker is detected in a frame of the video, we start by identifying the marker corners. We used two marker corners as references ($R1$ and $R2$) to compute the marker rotation in the frame. Then, the image referential has to be converted to the real-world referential by mapping the visible area in the image (in pixels) to the real-world (meters). Then, the angle and displacement between the frame reference point (FR) and the marker reference point ($R1$) are computed. Finally, the current vehicle yaw can be computed as well as the distance between the vehicle reference point (VR) and the marker reference point ($R1$).

By automating this process, we are able to perform large-scale data collection surveys without manual labelling effort. When a GT tag is visible by the camera, we automatically obtain the 2D position (p_x, p_y) and the orientation (ϕ) ground truth with high accuracy. This allows calculating not only the error in the estimated position but also the error in the vehicle yaw (orientation). Between tags, one can also easily estimate ground-truth data, e.g., using interpolation techniques.

4. Data Analysis

Table 2 shows the number of samples collected by sensors, the time duration and the accumulated distance for each trajectory. The Wi-Fi samples count includes the samples from each Wi-Fi interface plus the merged Wi-Fi samples for each trajectory, hence the number of samples from each Wi-Fi interface and the merged sample is this value divided by five. The sample count for both IMUs is also provided. Despite having two programs collecting simultaneously with the same configured time between samples, IMU1 collected 15 samples more than IMU2 considering all trajectories. The time duration of each trajectory is the time in seconds that the trajectory took from the beginning until the end. The accumulated distance is the absolute distance obtained from the encoder sensor by summing the absolute value of all distance samples. The total duration from all trajectories sums up to almost 1 h of data (≈ 57 min.), and the total accumulated distance is over 1 km.

Table 2. Number of sensor samples, elapsed time and accumulated distance per trajectory.

	Number of Samples				Ground Truth	Time (s)	Acc. Dist (m)
	Wi-Fi	IMU1	IMU2	Encoder			
T1	1910	12,266	12,262	30,242	1400	617.621	215.357
T2	2090	13,392	13,387	33,008	1857	674.398	223.371
T3	2360	15,200	15,200	37,462	3553	765.564	229.434
T4	1100	7058	7056	17,404	2689	355.362	122.437
T5	945	6078	6078	15,000	1580	306.197	87.846
T6	2170	13,901	13,897	34,257	4068	700.210	193.858
Total	10,575	67,895	67,880	167,373	15,147	3419.352	1072.303

The mean time and standard deviation between consecutive samples are displayed in Table 3. Despite the existence of several Wi-Fi interfaces, the application that collects samples from them assigns the same timestamp for samples obtained from all interfaces; hence, only one column is necessary to represent the time in consecutive Wi-Fi samples. In contrast, samples from IMU sensors have distinct timestamps, so they are represented in two distinct columns. As expected, encoder and IMU samples have low time variation in two consecutive samples, thus having the expected sample rates of 50 Hz and 20 Hz, respectively, having a mean sampling period of 20 ms and 50 ms. Overall, encoder and IMU samples have a standard deviation of 1 ms and 5 ms, respectively. A higher standard deviation in IMU is probably due to the data provided by the integrated sensors (i.e., magnetometer, gyroscope, accelerometer), which take longer to process than encoder samples. Wi-Fi samples are dependent on the time it takes to scan the available channels in the 2.4 GHz frequency band, which is 1.614 s on average, with little variation since the standard deviation on all samples is 94 ms.

Ground-truth samples are the ones with higher variation between consecutive samples. This is due to a ground-truth sample being collected only when the camera on the mobile unit is able to 'see' a GT tag. A video was collected at 30 fps for each trajectory. Then, the computer vision algorithm processes each frame of the video, and when it detects a tag, it generates a new GT sample. For instance, if the mobile unit is stopped while a tag is visible by the camera and detected by the computer vision solution, it generates GT samples at a rate of 30 Hz. Therefore, when the mobile unit stops close to a GT tag, it is constantly detecting that tag; hence, it has more ground-truth samples. Conversely, when the mobile unit is constantly moving, there will be fewer GT samples. The speed of the vehicle influences the number of ground truth samples, since each tag will be less time visible to the camera when the vehicle moves at higher speeds. This is demonstrated by the number of GT samples obtained in trajectories T3 and T6, where despite T6 being shorter in duration and accumulated distance, it has more GT samples (4068) than T3 (3553). This is also shown by the significantly higher number of GT samples from trajectory T4 (2689 GT samples) in comparison to the number of GT samples from trajectories T1 (1400) and T2 (1857), both of which were longer in duration and accumulated distance.

Table 3. Sensor sample analysis: sampling period (time between consecutive samples from sensors, in seconds) (top); percentage of sampling period outliers per sensor (bottom).

	Wi-Fi	IMU1	IMU2	Encoder	Ground Truth
Sampling period (mean \pm standard deviation), in seconds					
T1	1.614 \pm 0.094	0.050 \pm 0.005	0.050 \pm 0.005	0.020 \pm 0.001	0.432 \pm 2.198
T2	1.611 \pm 0.027	0.050 \pm 0.005	0.050 \pm 0.005	0.020 \pm 0.001	0.339 \pm 1.853
T3	1.621 \pm 0.231	0.050 \pm 0.005	0.050 \pm 0.005	0.020 \pm 0.001	0.210 \pm 1.404
T4	1.614 \pm 0.030	0.050 \pm 0.004	0.050 \pm 0.004	0.020 \pm 0.001	0.131 \pm 1.081
T5	1.612 \pm 0.025	0.050 \pm 0.005	0.050 \pm 0.005	0.020 \pm 0.001	0.168 \pm 2.265
T6	1.610 \pm 0.030	0.050 \pm 0.005	0.050 \pm 0.005	0.020 \pm 0.001	0.162 \pm 1.380
All	1.614 \pm 0.094	0.050 \pm 0.005	0.050 \pm 0.005	0.020 \pm 0.001	0.432 \pm 2.198
Percentage of sampling period outliers (%)					
T1	0.262	1.394	1.240	1.462	N.A.
T2	0.478	1.128	1.382	1.594	N.A.
T3	0.212	1.329	1.263	1.564	N.A.
T4	0.455	1.332	1.361	1.028	N.A.
T5	0.529	1.283	1.267	1.467	N.A.
T6	0.000	1.331	1.180	1.124	N.A.
All	0.284	1.298	1.276	1.397	N.A.

N.A.—Non-applicable.

The percentage of outliers in each sampling period is displayed in the bottom part of Table 3. The sampling period is the time interval between two consecutive samples. The sampling period outliers are the sensor samples whose time since the previous sample was significantly higher than the other samples, meaning that they took longer to collect. The expression that defines the percentage of sampling period outliers is defined as follows:

$$P_{\text{out}} = \frac{N_o}{N_s} \times 100 \quad (9)$$

where N_o represents the number of sampling period outliers, and N_s represents the total number of sensor samples collected. The number of sampling period outliers is defined as follows:

$$N_o = \sum_{i=1}^{N_s} \begin{cases} 1 & \Delta t_i > \bar{x} + 3\sigma \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where Δt_i represents the time difference between sample i and the previous sample, \bar{x} represents the mean time between consecutive samples (mean sampling period), and σ represents the standard deviation sampling period. For a sensor sample to be considered an outlier, the time it took since the previous sample must be higher than the mean sampling period plus three times the standard deviation.

Overall, both IMUs and the encoder have a lower percentage of outlier samples, all being lower than 1.4%, when considering all sensor samples. Wi-Fi reports the lowest percentage of outlier samples with only 0.284%. This is justified, because the sampling period of Wi-Fi samples (overall 1.614 s) is much higher than that of the IMUs and the encoder, which are 50 ms and 20 ms, respectively. Due to the lower sampling period of the IMUs and encoder, more outlier samples are detected. Even a slight delay of a few milliseconds can cause these samples to be counted as outliers. In contrast, Wi-Fi samples have a higher delay tolerance because the standard deviation time between consecutive Wi-Fi samples is significantly larger than that for samples from the IMUs or the encoder. It is also relevant to mention that although obtaining fingerprints from the multiple Wi-Fi interfaces depends on the operating system to return the results, the percentage of outlier samples shows that the operating system is consistent in returning almost all samples without a significant delay. Outlier sampling period samples were not computed for ground truth. This metric does not apply to these types of data, because GT data are not continuously obtained, and it depends on the mobile unit, as previously mentioned. However, samples for Wi-Fi interfaces, the IMUs, and the encoder are continuously obtained at a specific sampling frequency.

5. Examples of Data Use

In this section, we provide two examples showing practical applications of the collected data. In the first example, dead reckoning is applied to determine the vehicles' trajectory over time using data from the encoder and IMU sensors. In the second example, Wi-Fi fingerprinting is used to determine the absolute position of the vehicle based on data obtained from the Wi-Fi interfaces.

5.1. Dead Reckoning

Dead reckoning is a tracking technique that requires an initial position and then determines the next position based on heading and displacement information. Although it produces accurate estimates in short periods, dead reckoning is prone to cumulative errors, which are mostly caused by sensor bias in the IMU that causes drift in the heading and magnetic perturbations that can significantly affect the heading. Consequently, in the long term, this approach can lead to large errors if the initial position is not re-set. To mitigate the accumulated error, dead reckoning may be combined with absolute positioning techniques which allow the user to re-set the initial position and reduce the accumulated error. In this paper, we use dead reckoning because it is a simple method to analyse the collected data and visualise the produced trajectories. It can serve as a baseline against which other researchers can compare their solutions to improve it and combine it with more complex solutions.

The estimated position by dead reckoning is given by the following:

$$p_i(x, y) = (x_{i-1} + \cos(yaw) \times d, y_{i-1} + \sin(yaw) \times d) \quad (11)$$

where $p_i(x, y)$ represents the latest position, $(x, y)_{i-1}$ represent the coordinates of the previous position, yaw represents the orientation provided by the IMU sensor, and d represents the displacement provided by the encoder.

Figure 12 represents the estimated dead reckoning trajectories with both IMUs. Visually analysing these plots, one can see that trajectories from both IMUs are affected by cumulative errors, namely sensor bias, drift, and also magnetic disturbances that cause sudden variations in the estimated heading, hence leading to distorted dead reckoning trajectories. One can also observe that with the exception of T5, where trajectories from both

IMUs are similar, in the remaining trajectories, it can be seen that the drift in the heading and the magnetic perturbations result in distinct trajectories. This shows that although these sensors share the same environment, the sensor bias of each sensor leads to a unique behavior in the estimated heading.

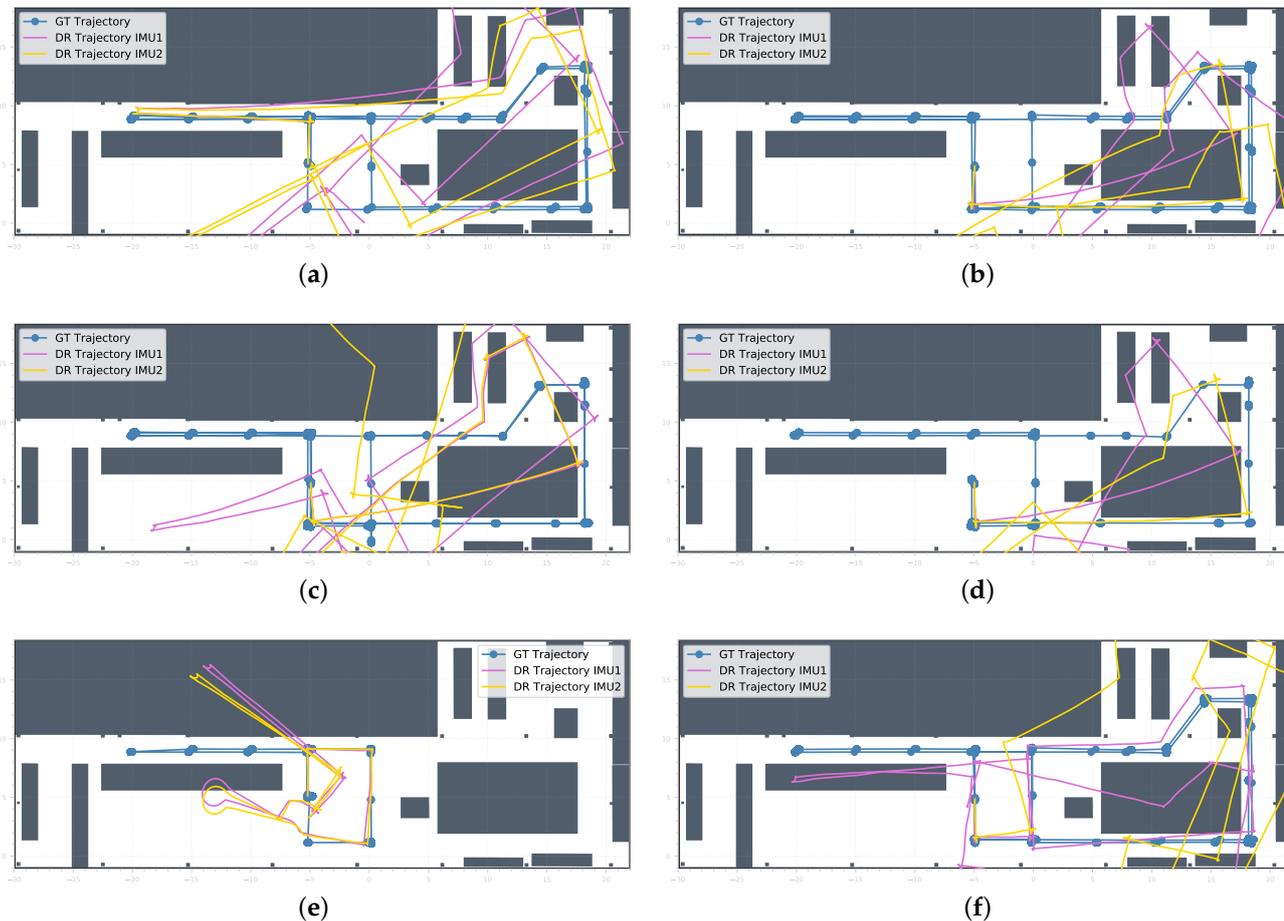


Figure 12. Estimated trajectories using dead reckoning: (a) T1, (b) T2, (c) T3, (d) T4, (e) T5, (f) T6. GT trajectory represents the ground-truth trajectory; DR Trajectory IMU1 and DR Trajectory IMU2, represented by the purple and yellow plot lines, define the dead reckoning trajectories obtained with IMU1 and IMU2, respectively.

The detailed positioning results are shown in Table 4. There are several trajectories in which the maximum error is quite large due to the accumulated error. Possible causes for this are associated with the industrial environment due to heavy machinery that affects the magnetometer, and since the IMUs are low cost, they are more prone to noise and magnetic disturbances. Overall, the IMU1 performs better than IMU2, having the best mean and 75th percentile error with 8.25 m and 9.57 m, respectively. However, it has a higher positioning error, with 42.12 m, than the 35.56 m obtained with IMU2.

Figure 13 shows the cumulative distribution functions (CDFs) of all trajectories for both IMUs, showing the probability (between 0 and 1) on the x -axis and the error (in meters) on the y -axis. These plots prove that the distinct trajectories, observed in Figure 12, lead to differences in the positioning performance on both IMUs. For instance, the trajectory with the best 75th percentile is T6 for IMU1 and T1 for IMU2. The best-performing trajectory, i.e., the one with the lowest maximum error, is T6 for IMU1 and T5 for IMU2. Conversely, the worst-performing trajectory, i.e., the one with the highest maximum error, is T2 for IMU1 and T6 for IMU2. Although these results seem suboptimal, this dataset has been used with a particle filter to combine dead reckoning with Wi-Fi fingerprinting, achieving a

mean error of 0.62 m [45], demonstrating that when combined with other techniques, dead reckoning enables accurate tracking of indoor vehicles.

Table 4. Dead reckoning results from both IMUs (in metres).

	T1	T2	T3	T4	T5	T6	Overall
IMU1							
Mean	6.19	20.22	8.82	12.02	3.73	2.26	8.25
P75th	12.63	28.91	9.19	13.34	9.54	2.59	9.57
P99th	15.80	42.12	20.44	34.07	9.57	5.90	42.12
Max	15.80	42.12	20.44	34.16	9.57	5.92	42.12
IMU2							
Mean	4.08	11.23	15.93	7.57	3.20	16.72	11.66
P75th	6.63	17.48	21.63	9.07	8.17	20.55	17.51
P99th	12.07	19.87	28.95	21.11	8.19	35.55	35.55
Max	12.07	19.87	28.95	21.17	8.20	35.55	35.56

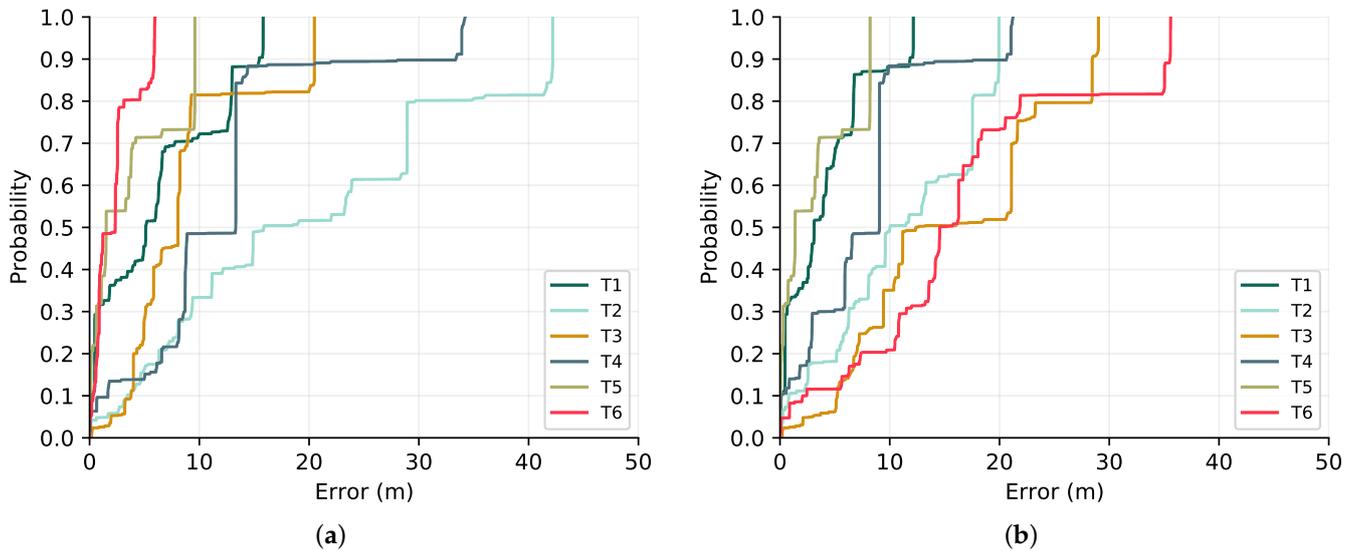


Figure 13. CDF of dead-reckoning trajectories obtained with (a) IMU1 and (b) IMU2.

5.2. Wi-Fi Fingerprinting

We computed the Wi-Fi fingerprinting positioning error for each track that was performed using single Wi-Fi interfaces and the merged data to compare how averaging Wi-Fi samples from multiple interfaces affects positioning results. We used $k = 3$ with the Manhattan (city-block) distance and evaluated the positioning error by determining the Euclidean distance between the ground truth and the estimated position. Since ground-truth positions are not synchronised with Wi-Fi samples, the positioning error was obtained only for the Wi-Fi samples that were closer in time for each of the ground-truth positions.

The Wi-Fi fingerprinting results of trajectory T3, obtained for each Wi-Fi interface and the interface with the averaged samples, are presented in Table 5. We opted for T3 to show these results because it is the longest trajectory, containing more Wi-Fi samples than the others.

The merged Wi-Fi data W_m has the best 99th percentile and maximum errors in comparison to results with single interfaces. Although the best mean and 75th percentile errors are achieved with Wi-Fi interface W_3 , averaging multiple Wi-Fi interfaces provides the best overall result, leading to improvements in the last percentile. Improvements are

significant in the maximum error reporting 8.04 m, with improvements ranging from 54% to 75% in comparison to single Wi-Fi interfaces.

Table 6 shows the positioning results of Wi-Fi fingerprinting for all trajectories, using the merged Wi-Fi data provided in W_m files. The last column aggregates position estimates from all trajectories with an overall mean error of 2.19 m, which outperforms other research works using Wi-Fi fingerprinting [9].

Table 5. Wi-Fi fingerprinting results of trajectory T3 using different Wi-Fi interfaces (in metres).

	W_1	W_2	W_3	W_4	W_m
Mean	3.88	3.22	2.71	5.12	3.30
P75th	6.28	4.96	3.56	8.02	4.96
P99th	12.52	8.18	12.67	17.80	8.02
Max	15.05	14.04	14.96	17.80	8.04

Table 6. Wi-Fi fingerprinting results using averaged Wi-Fi samples from all interfaces (in metres).

	T1	T2	T3	T4	T5	T6	Overall
Mean	2.00	1.84	3.30	1.24	2.68	1.88	2.19
P75th	3.36	3.16	4.96	2.14	3.93	2.68	3.60
P99th	7.88	6.62	8.02	5.22	7.82	12.79	8.01
Max	7.97	6.66	8.04	5.29	7.84	12.80	12.80

Figure 14 illustrates the CDF of these results, presenting the error distribution for each trajectory. Notably, trajectories T4 and T6 exhibit comparable performance up to the 75th percentile, achieving positioning errors of 2.14 m and 2.86 m, respectively. However, it is worth mentioning that despite having the second-best 75th percentile error, T6 demonstrates the highest positioning error, measuring 12.80 m. Although the maximum error of T6 is higher than the one observed in the other trajectories, it is a rare occurrence. This is evident from the fact that its 95th percentile positioning error is 7.89 m, which is similar to the maximum error seen in the other trajectories. Consequently, 95% of the evaluated position estimates lie below 7.89 m, thus being a more reliable indication of typical performance.

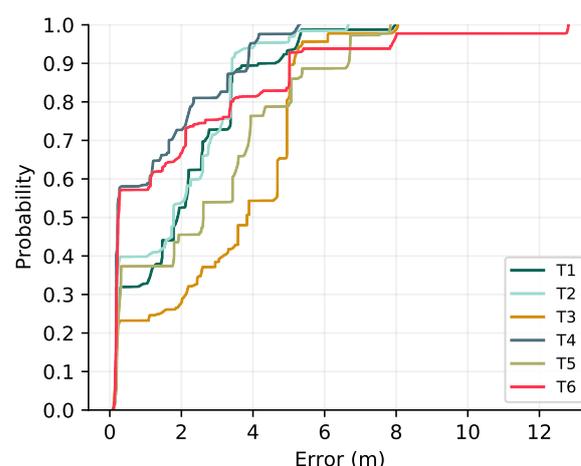


Figure 14. CDF of Wi-Fi fingerprinting using averaged Wi-Fi samples.

6. Conclusions

This paper presented a new dataset for the indoor tracking of vehicles in industrial environments. The dataset was collected in a factory-like environment with a mobile unit, which was purposely designed to emulate an industrial vehicle equipped with Wi-Fi and motion sensors. In addition to the documentation of the dataset, we detailed the process to

collect the data, describing the software used to collect data from each sensor type and how the GT data were obtained.

Despite the existence of other datasets for indoor positioning, the dataset proposed in this paper contributes to the pool of available datasets with distinct characteristics: data were collected in an industrial environment with heavy machinery that can influence indoor radio propagation and magnetometer readings; Wi-Fi data (radio map and test samples) from multiple Wi-Fi interfaces; orientation data obtained from two low-cost IMUs; displacement data obtained from an absolute encoder attached to the wheel, allowing accurate measurement of the travelled distance; and GT collected by a video camera to annotate timestamps when the mobile unit passed by reference tags.

Due to having data from two low-cost IMUs, this dataset can be explored to devise new sensor fusion approaches that combine data from both sensors to improve orientation estimates while keeping the overall cost of the system more economical. These IMUs allow greatly reducing the cost of the positioning system especially when compared to industrial-grade sensors, which are costly.

Industry 4.0 and the IoT are the primary research areas that can benefit from this dataset. Potential applications include the development of vehicle tracking and monitoring systems, autonomous robot positioning, as well as indoor transportation systems for materials. Additionally, the indoor positioning and indoor navigation community can explore this dataset to develop, test, and evaluate their systems as well as contribute new methods for indoor tracking in industrial environments. Finally, adding a new public dataset contributes to the fair comparison and benchmarking between different solutions.

Author Contributions: I.S., C.P. and A.M. designed the data collection solution and conducted the data collection campaign; I.S. performed the radio map site survey, developed the software for collecting the data, conceived the data formats, and wrote the scripts. C.P. designed and developed the solution to obtain the ground truth. I.S., C.P. and J.T.-S. wrote the original draft of the paper; I.S., C.P., J.T.-S. and A.M. participated in the review and editing of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope UIDB/00319/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset is available on <https://doi.org/10.5281/zenodo.7826540>, accessed on 19 October 2023.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Access Point
BLE	Bluetooth Low Energy
CSV	Comma-Separated Values
GT	Ground Truth
GPS	Global Positioning System
IoT	Internet of Things
IPIN	International Conference on Indoor Positioning and Indoor Navigation
IPS	Indoor Positioning System
IMU	Inertial Measurement Unit
k-NN	k-Nearest Neighbour
RPi	Raspberry Pi
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
UWB	Ultrawide-Band

References

1. Aheleroff, S.; Xu, X.; Lu, Y.; Aristizabal, M.; Pablo Velásquez, J.; Joa, B.; Valencia, Y. IoT-enabled smart appliances under industry 4.0: A case study. *Adv. Eng. Inform.* **2020**, *43*, 101043. [CrossRef]
2. Ssekidde, P.; Steven Eyobu, O.; Han, D.S.; Oyana, T.J. Augmented CWT Features for Deep Learning-Based Indoor Localization Using WiFi RSSI Data. *Appl. Sci.* **2021**, *11*, 1806. [CrossRef]
3. Sittón-Candanedo, I.; Alonso, R.S.; Rodríguez-González, S.; García Coria, J.A.; De La Prieta, F. Edge Computing Architectures in Industry 4.0: A General Survey and Comparison. In Proceedings of the 14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019), Seville, Spain, 13–15 May 2019.
4. Sigov, A.; Ratkin, L.; Ivanov, L.A.; Xu, L.D. Emerging Enabling Technologies for Industry 4.0 and Beyond. *Inf. Syst. Front.* **2022**. [CrossRef]
5. Yu, J.G.; Selby, B.; Vlahos, N.; Yadav, V.; Lemp, J. A feature-oriented vehicle trajectory data processing scheme for data mining: A case study for Statewide truck parking behaviors. *Transp. Res. Interdiscip. Perspect.* **2021**, *11*, 100401. [CrossRef]
6. Silva, I.; Pendão, C.; Torres-Sospedra, J.; Moreira, A. TrackInFactory: A Tight Coupling Particle Filter for Industrial Vehicle Tracking in Indoor Environments. *IEEE Trans. Syst. Man, Cybern. Syst.* **2022**, *52*, 4151–4162. [CrossRef]
7. Bahl, P.; Padmanabhan, V. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Tel Aviv, Israel, 26–30 March 2000; Volume 2, pp. 775–784. [CrossRef]
8. He, S.; Chan, S.H.G. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Commun. Surv. Tutorials* **2016**, *18*, 466–490. [CrossRef]
9. Potorti, F.; Park, S.; Crivello, A.; Palumbo, F.; Girolami, M.; Barsocchi, P.; Lee, S.; Torres-Sospedra, J.; Ruiz, A.R.J.; Perez-Navarro, A.; et al. The IPIN 2019 Indoor Localisation Competition—Description and Results. *IEEE Access* **2020**, *8*, 206674–206718. [CrossRef]
10. Poulouse, A.; Kim, J.; Han, D.S. A Sensor Fusion Framework for Indoor Localization Using Smartphone Sensors and Wi-Fi RSSI Measurements. *Appl. Sci.* **2019**, *9*, 4379. [CrossRef]
11. Panyov, A.A.; Golovan, A.A.; Smirnov, A.S. Indoor positioning using Wi-Fi fingerprinting pedestrian dead reckoning and aided INS. In Proceedings of the 2014 International Symposium on Inertial Sensors and Systems (ISISS), Laguna Beach, CA, USA, 25–26 February 2014; pp. 1–2. [CrossRef]
12. Chen, L.-H.; Wu, E.H.K.; Jin, M.-H.; Chen, G.-H.. Intelligent Fusion of Wi-Fi and Inertial Sensor-Based Positioning Systems for Indoor Pedestrian Navigation. *IEEE Sens. J.* **2014**, *14*, 4034–4042. [CrossRef]
13. Bi, J.; Wang, Y.; Yu, B.; Cao, H.; Shi, T.; Huang, L. Supplementary open dataset for WiFi indoor localization based on received signal strength. *Satell. Navig.* **2022**, *3*, 25. [CrossRef]
14. Lohan, E.S.; Torres-Sospedra, J.; Gonzalez, A. WiFi RSS measurements in Tampere University multi- building campus, 2017. *Zenodo* **2021**. [CrossRef]
15. Aranda, F.J.; Parralejo, F.; Álvarez, F.J.; Torres-Sospedra, J. Multi-Slot BLE Raw Database for Accurate Positioning in Mixed Indoor/Outdoor Environments. *Data* **2020**, *5*, 67. [CrossRef]
16. Moreira, A.; Nicolau, M.J.; Silva, I.; Torres-Sospedra, J.; Pendão, C.; Meneses, F. Wi-Fi Fingerprinting Dataset with Multiple Simultaneous Interfaces. *Zenodo* **2019**. [CrossRef]
17. Laoudias, C.; Piché, R.; Panayiotou, C. KIOS WiFi RSS Dataset 2013. Available online: https://www.researchgate.net/profile/Christos-Laoudias/publication/256482916_KIOS_WiFi_RSS_dataset (accessed on 11 May 2023).
18. Torres-Sospedra, J.; Montoliu, R.; Martínez-Usó, A.; Avariento, J.P.; Arnau, T.J.; Benedito-Bordonau, M.; Huerta, J. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Republic of Korea, 27–30 October 2014; pp. 261–270. [CrossRef]
19. Lopez Pastor, J.A.; Ruiz Ruiz, A.; García Sánchez, A.J.; Gómez Tornero, J.L. Wi-Fi RSSI fingerprint dataset from two malls with validation routes in a shop-level for indoor positioning. *Zenodo* **2020**. [CrossRef]
20. Nor Hisham, A.N.; Ng, Y.H.; Tan, C.K.; Chieng, D. Hybrid Wi-Fi and BLE Fingerprinting Dataset for Multi-Floor Indoor Environments with Different Layouts. *Data* **2022**, *7*, 156. [CrossRef]
21. Spachos, P. RSSI Dataset for Indoor Localization Fingerprinting. *IEEE Dataport* **2020**. [CrossRef]
22. Zsolt Tóth, J.T. Miskolc IIS Hybrid IPS: Dataset for Hybrid Indoor Positioning. In Proceedings of the 26st International Conference on Radioelektronika, Kosice, Slovakia, 19–20 April 2016, pp. 408–412.
23. Salimibeni, M.; Hajiakhondi-Meybodi, Z.; Malekzadeh, P.; Atashi, M.; Plataniotis, K.N.; Mohammadi, A. IoT-TD: IoT Dataset for Multiple Model BLE-based Indoor Localization/Tracking. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, The Netherlands, 18–21 January 2021; pp. 1697–1701. [CrossRef]
24. Alhomayani, F.; Mahoor, M.H. OutFin, a multi-device and multi-modal dataset for outdoor localization based on the fingerprinting approach. *Sci. Data* **2021**, *8*, 66. [CrossRef] [PubMed]
25. Jimenez, A.R.; Mendoza-Silva, G.M.; Montoliu, R.; Seco, F.; Torres-Sospedra, J. Datasets and Supporting Materials for the IPIN 2016 Competition Track 3 (Smartphone-based, off-site). *Zenodo* **2016**. [CrossRef]
26. Jimenez, A.R.; Mendoza-Silva, G.M.; Seco, F.; Torres-Sospedra, J. Datasets and Supporting Materials for the IPIN 2017 Competition Track 3 (Smartphone-based, off-site). *Zenodo* **2017**. [CrossRef]

27. Jiménez, A.R.; Mendoza-Silva, G.M.; Ortiz, M.; Perez-Navarro, A.; Perul, J.; Seco, F.; Torres-Sospedra, J. Datasets and Supporting Materials for the IPIN 2018 Competition Track 3 (Smartphone-based, off-site). *Zenodo* **2018**. [[CrossRef](#)]
28. Ortiz, M.; Perul, J.; Torres-Sospedra, J.; Renaudin, V. Datasets and Supporting Materials for the IPIN 2018 Competition Track 4 (Foot-Mounted IMU based Positioning, off-site), 2019. *Zenodo* **2019**. [[CrossRef](#)]
29. Jiménez, A.R.; Perez-Navarro, A.; Crivello, A.; Mendoza-Silva, G.M.; Seco, F.; Ortiz, M.; Perul, J.; Torres-Sospedra, J. Datasets and Supporting Materials for the IPIN 2019 Competition Track 3 (Smartphone-based, off-site). *Zenodo* **2019**. [[CrossRef](#)]
30. Ortiz, M.; Perul, J.; Renaudin, V.; Torres-Sospedra, J. Datasets and Supporting Materials for the IPIN 2019 Competition Track 4 (Foot-Mounted IMU based Positioning, off-site). *Zenodo* **2020**. [[CrossRef](#)]
31. Torres-Sospedra, J.; Quezada-Gaibor, D.; Jiménez, A.R.; Seco, F.; Perez-Navarro, A. Datasets and Supporting Materials for the IPIN 2020 Competition Track 3 (Smartphone-based, off-site). *Zenodo* **2020**. [[CrossRef](#)]
32. Ortiz, M.; Zhu, N.; Renaudin, V.; Ramesh, A. Datasets and Supporting Materials for the IPIN 2020 Competition Track 4 (Foot-Mounted IMU based Positioning, off-site). *Zenodo* **2021**. [[CrossRef](#)]
33. Torres-Sospedra, J.; Polo, F.A.; Parralejo, F.; Parent, V.B.; Alvarez, F.; Pérez-Navarro, A.; Jimenez, A.R.; Seco, F. Datasets and Supporting Materials for the IPIN 2021 Competition Track 3 (Smartphone-based, off-site). *Zenodo* **2021**. [[CrossRef](#)]
34. Torres-Sospedra, J.; Silva, I.; Pendo, C.; Moreira, A.; Meneses, F.; Costa, A.; Nicolau Maria, J.; Gonzalez-Perez, A.; Jiménez Antonio, R.; Pérez-Navarro, A. Datasets and Supporting Materials for the IPIN 2022 Competition Track 3 (Smartphone-based, off-site). *Zenodo* **2022**. [[CrossRef](#)]
35. Shu, Y.; Xu, Q.; Liu, J.; Choudhury, R.R.; Trigoni, N.; Bahl, V. Indoor Location Competition 2.0 Dataset, 2021. Available online: <https://www.microsoft.com/en-us/research/publication/indoor-location-competition-2-0-dataset/> (accessed on 11 May 2023).
36. Luo, J.; Pronobis, A.; Caputo, B.; Jensfelt, P. The KTH-IDOL2 database. Technical Report CVAP304. In *Technical Report CVAP304*; KTH Royal Institute of Technology, CVAP/CAS: Stockholm, Sweden, 2006.
37. Ullah, M.M.; Pronobis, A.; Caputo, B.; Luo, J.; Jensfelt, P. The COLD Database. In *Technical Report TRITA-CSC-CV 2007:1*; KTH Royal Institute of Technology, CVAP/CAS: Stockholm, Sweden, 2007.
38. Moreira, A.; Silva, I.; Meneses, F.; Nicolau, M.J.; Pendo, C.; Torres-Sospedra, J. Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning. In *Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, Japan, 18–21 September 2017; pp. 1–8. [[CrossRef](#)]
39. Chen, C.; Zhao, P.; Lu, C.X.; Wang, W.; Markham, A.; Trigoni, N. Deep-Learning-Based Pedestrian Inertial Navigation: Methods, Data Set, and On-Device Inference. *IEEE Internet Things J.* **2020**, *7*, 4431–4441. [[CrossRef](#)]
40. Asraf, O.; Shama, F.; Klein, I. PDRNet: A Deep-Learning Pedestrian Dead Reckoning Framework. *IEEE Sens. J.* **2022**, *22*, 4932–4939. [[CrossRef](#)]
41. Yang, X.; Zhuang, Y.; Gu, F.; Shi, M.; Cao, X.; Li, Y.; Zhou, B.; Chen, L. DeepWiPos: A Deep Learning-Based Wireless Positioning Framework to Address Fingerprint Instability. *IEEE Trans. Veh. Technol.* **2023**, *72*, 8018–8034. [[CrossRef](#)]
42. Alitaleshi, A.; Jazayeriy, H.; Kazemitabar, J. EA-CNN: A smart indoor 3D positioning scheme based on Wi-Fi fingerprinting and deep learning. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105509. [[CrossRef](#)]
43. Yu, D.; Li, C.; Xiao, J. Neural Networks-Based Wi-Fi/PDR Indoor Navigation Fusion Methods. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2503514. [[CrossRef](#)]
44. Guo, G.; Chen, R.; Niu, X.; Yan, K.; Xu, S.; Chen, L. Factor Graph Framework for Smartphone Indoor Localization: Integrating Data-Driven PDR and Wi-Fi RTT/RSS Ranging. *IEEE Sens. J.* **2023**, *23*, 12346–12354. [[CrossRef](#)]
45. Silva, I.; Pendo, C.; Moreira, A. Real-World Deployment of Low-Cost Indoor Positioning Systems for Industrial Applications. *IEEE Sens. J.* **2022**, *22*, 5386–5397. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.