



**POLITECNICO
DI TORINO**

Master of Science in Mechatronic Engineering

Master Thesis

Three-dimensional magnetic fingerprinting for indoor localization

Supervisor:

Prof. Alessandro Rizzo

Candidate:

Luca Palmisano

Company supervisors

SONY Europe STC

Dr. Eng. Ben Eitel

M.Sc. Akin Ayanoglu

December 2020

Summary

Nowadays, it is widely known that localization systems changed our daily routine. A lot of people are using them but, currently, with localization systems, we are mainly referring to outdoor localization and navigation systems based on global positioning system (GPS). Despite outdoor localization being widely used and highly developed, there is still no standard technology for indoor localization purposes. This issue arises because the satellite based signals are largely attenuated when passing through walls and ceilings, leading to a loss of precision.

An interesting alternative, for indoor applications, that does not require installing and maintaining costly equipment in the building is the geomagnetic fingerprinting. In modern buildings, the earth's magnetic field is distorted due to the presence of ferromagnetic construction materials. The distortion can be used to define every location by its magnetic fingerprint. All the current smartphones are equipped with the magnetometer, thus, it is possible to use this sensor associated with the magnetic map of the building, to design a smartphone-based indoor localization system. In the thesis, a reliable 3-dimensional magnetic map fingerprint method has been investigated. Subsequently, an already existing 2D indoor localization system has been extended to work in 3D. Finally, the performance of the indoor localization engine using the 2D and 3D magnetic maps has been compared.

Contents

List of Figures	V
List of Tables	VIII
1 Introduction	1
1.1 State of the art	1
1.2 Context of the work	4
1.3 Objective of the work	5
1.4 Organization	5
2 The Magnetic Indoor Localization Engine	7
2.1 Sensors	7
2.2 Sensor processing	8
2.2.1 Calibration	8
2.2.2 Magnetic Field Representation and Orientation Compensation	9
2.3 Pedestrian Dead Reckoning - PDR	10
2.4 Database magnetic map	11
2.5 Particle filter localization	11
3 2-dimensional magnetic map generation	16
3.1 Map generation framework	16
3.2 2D magnetic map issues	18
3.3 Additional interpolation and extrapolation approaches	21
3.4 2D methods evaluation	24
4 Magnetic measurement setup	28
4.1 New 3D measurement setup	28
4.2 V-SLAM analysis	31
4.2.1 Google ARCore	32
4.2.2 PSG V-SLAM	36
4.3 Trajectory correction	38

5	3-dimensional magnetic map generation and MILE extension	41
5.1	3D map generation	41
5.2	3D map evaluation	44
5.2.1	Smoothness	47
5.2.2	Statistical behaviour of the 3D magnetic field	48
5.2.3	Calibration methods evaluation	50
5.3	MILE 3D extension	53
6	Performance evaluation	55
6.1	Test bed and dataset collection	55
6.2	Localization accuracy	58
6.3	Height tracking capability	67
6.4	Barometer sensor analysis	70
7	Conclusions and future work	73
A	Test bed for 3D maps evaluation	75
B	Test bed for performance evaluation	80
C	Model training results	92
	References	96

List of Figures

2.1	Magnetic Indoor Localization Engine (MILE) architecture	8
2.2	Relation between xyz and mia coordinates	10
2.3	Example of magnetic grid map.	12
2.4	Particle filter workflow	13
2.5	Floorplan mode behaviour	14
2.6	Particle filter localization example	15
3.1	Convex hull	17
3.2	Example of non natural reconstruction of the magnetic field using MATLAB built-in functions.	19
3.3	Magnetic magnitude map using griddata natural interpolation and nearest extrapolation.	20
3.4	Inpaint PDE magnetic magnitude map	22
3.5	Magnetic magnitude maps obtained using inpaint NaNs and inpaint DCT	23
3.6	Peak of the magnetic magnitude maps obtained using inpaint NaNs and inpaint DCT	24
4.1	Cart setup for 3D magnetic captures	29
4.2	3D-printed folder to locate the smartphone	30
4.3	Interference test result, x axis	31
4.4	Example of ARCore result affected by jump discontinuity	33
4.5	Example of ARCore result affected by length drift	33
4.6	Trajectory obtained using ARCore applied to a Sony Xperia 5, fixed to the cart at 1m height	35
4.7	Trajectory obtained using ARCore applied to a Sony Xperia XZ3, fixed to the cart at 0.5m height	35
4.8	Trajectory obtained using PSG V-SLAM applied to a video recorded by a GoPro, fixed to the cart at 1.5m height	36
4.9	Trajectory obtained using PSG V-SLAM applied to a video recorded by a GoPro, fixed to the cart at 1.25m height	37
4.10	Transformed ARCore V-SLAM trajectory	39

4.11	Transformed PSG V-SLAM trajectory	39
5.1	GoPro fixed to the cart	42
5.2	Cross correlation example and alignment	43
5.3	Magnetic map magnitude vertical section generated by inpaint DCT 3D	44
5.4	Magnetic map magnitude vertical section generated by inpaint PDE 3D	45
5.5	Magnetic map magnitude vertical section generated by inpaint NaNs 3D	45
5.6	Magnetic map magnitude vertical section generated by interp3 spline	45
5.7	Magnetic map magnitude vertical section generated by interp3 makima	46
5.8	Magnetic map magnitude vertical section generated by interp3 cubic	46
5.9	Comparison between inpaint DCT 3D and inpaint PDE 3D	50
5.10	Test scheme used for calibration methods evaluation	51
6.1	Test bed for performance evaluation	56
6.2	Ground truth walk (in green) and MILE estimated localization (in red)	58
6.3	CDF of the estimation error for a sample walk	59
6.4	PDR results for a 1.75m and a 1.25m heights recordings	65
6.5	Xperia 5 results at different heights	66
6.6	Comparison between the best 3D against the best 2D performances, using Dataset 2	67
6.7	Example of bad height tracking result. The target height is 1.00m.	68
6.8	Example of good height tracking result. The target height is 1.00m.	69
6.9	PDR height estimation outcome	70
6.10	Barometric sensor test outcome	71
A.1	Original and transformed trajectories used for 3D maps evaluation	75
A.2	Smartphone heading along the trajectory used for 3D maps evaluation	76
A.3	Magnetic magnitude at height 0.25m obtained using inpaint DCT 3D	77
A.4	Magnetic magnitude at height 0.50m obtained using inpaint DCT 3D	77
A.5	Magnetic magnitude at height 0.75m obtained using inpaint DCT 3D	78
A.6	Magnetic magnitude at height 1.00m obtained using inpaint DCT 3D	78
A.7	Magnetic magnitude at height 1.25m obtained using inpaint DCT 3D	79
A.8	Magnetic magnitude at height 1.50m obtained using inpaint DCT 3D	79
B.1	Original and transformed trajectories used for 3D performance evaluation	81
B.2	Smartphone heading along the trajectory used for 3D performance evaluation	82
B.3	Magnetic magnitude at height 0.25m obtained using inpaint DCT 3D	83

B.4	Magnetic magnitude at height 0.50m obtained using inpaint DCT 3D	84
B.5	Magnetic magnitude at height 0.75m obtained using inpaint DCT 3D	85
B.6	Magnetic magnitude at height 1.00m obtained using inpaint DCT 3D	86
B.7	Magnetic magnitude at height 1.25m obtained using inpaint DCT 3D	87
B.8	Magnetic magnitude at height 1.50m obtained using inpaint DCT 3D	88
B.9	Magnetic magnitude at height 1.75m obtained using inpaint DCT 3D	89
B.10	Magnetic magnitude at height 2.00m obtained using inpaint DCT 3D	90
B.11	Magnetic magnitude obtained using inpaint DCT	91

List of Tables

3.1	Average absolute difference between mean values of measured grid points and estimated grid points.	25
3.2	Average absolute difference between standard deviation of measured grid points and estimated grid points.	26
3.3	Smoothness coefficients, lower value represents smoother map. . . .	27
5.1	Smoothness coefficients for the 6 heights 3D magnetic map, lower value represents smoother map.	47
5.2	Average absolute difference between mean values of measured grid points and estimated grid points of the 3D magnetic map.	48
5.3	Average absolute difference between standard deviations of measured grid points and estimated grid points of the 3D magnetic map. . . .	49
5.4	Evaluation test results for ellipsoid calibration	51
5.5	Evaluation test results for android calibration	51
5.6	Evaluation test results for the faked ellipsoid calibration	52
6.1	Magnitude and inclination covariances for 2D and 3D maps	60
6.2	3D performance of the test dataset.	61
6.3	2D performance of the test dataset.	61
6.4	3D performance of the test tracks captured at 1.00m height.	62
6.5	2D performance of the tracks captured at 1.00m height.	62
6.6	3D performance of the test tracks captured at 1.25m height.	63
6.7	2D performance of the test tracks captured at 1.25m height.	63
6.8	3D performance of the test tracks captured at 1.50m height.	63
6.9	2D performance of the test tracks captured at 1.50m height.	63
6.10	3D performance of the test tracks captured at 1.75m height.	64
6.11	2D performance of the test tracks captured at 1.75m height.	64
6.12	Estimated heights mean values for test Dataset 1.	67
6.13	Estimated heights standard deviations for test Dataset 1.	68
6.14	Estimated heights mean values for test Dataset 2.	69
6.15	Estimated heights standard deviations for test Dataset 2.	69

6.16	Barometer BMP380 differences of the barometric pressure median values for different height jumps.	72
6.17	Barometer LPS22HB differences of the barometric pressure median values for different height jumps.	72
C.1	3D performance of the test dataset for $C^m = 100\mu T^2$ and $C^i = 400\text{deg}^2$.	92
C.2	2D performance of the test dataset for $C^m = 100\mu T^2$ and $C^i = 400\text{deg}^2$.	92
C.3	3D performance of the test dataset for $C^m = 60\mu T^2$ and $C^i = 150\text{deg}^2$.	93
C.4	2D performance of the test dataset for $C^m = 60\mu T^2$ and $C^i = 150\text{deg}^2$.	93
C.5	3D performance of the test dataset for $C^m = 40\mu T^2$ and $C^i = 60\text{deg}^2$.	93
C.6	2D performance of the test dataset for $C^m = 40\mu T^2$ and $C^i = 60\text{deg}^2$.	94
C.7	3D performance of the test dataset for $C^m = 20\mu T^2$ and $C^i = 30\text{deg}^2$.	94
C.8	2D performance of the test dataset for $C^m = 20\mu T^2$ and $C^i = 30\text{deg}^2$.	94
C.9	3D performance of the test dataset for $C^m = 10\mu T^2$ and $C^i = 15\text{deg}^2$.	95
C.10	2D performance of the test dataset for $C^m = 10\mu T^2$ and $C^i = 15\text{deg}^2$.	95

Chapter 1

Introduction

During the last few decades, positioning systems have become one of the main research fields among an increasing number of universities, research centers and companies, which can be evenly seen in the growing number of location based services (LBS). The impact of the global positioning system (GPS) has been significant as it has changed completely the way we move in the cities. Despite GPS being very well known and highly developed, it is suitable for outdoor localization purposes only. GPS is based on the Global Navigation Satellite System (GNSS). Satellite signals suffer of significant degradation when passing through materials and structures of buildings [5]. This fact results in a limited capability of using GPS as an indoor positioning system. Regarding indoor localization applications, the interest is justified by a wide range of LBS. One of the most common applications could be localization and navigation when moving inside unknown buildings, such as airports, hospitals or shopping malls. Furthermore, indoor localization systems can be very interesting for companies in the proximity marketing business or for museums willing to offer personalized tours to the visitors. Other new applications exist such as shopping routing (e.g. inside grocery shops), management of emergencies or new spatial games. Considering the high number of applications of LBS, there isn't still a common standard technology available to pursue an efficient result, as a trade-off between precision and cost deploying of the infrastructure.

1.1 State of the art

A wide range of approaches have been tried to find a solution to the indoor localization problem: light based systems (i.e. LiDAR technology), radio frequency (RF) based systems (i.e. RADAR technology), ultra wide band (UWB) based systems, bluetooth based systems, WiFi based systems, computer vision based systems, dead reckoning based systems and magnetic field based systems [5]. All the listed methods can be classified essentially in 2 classes depending on how they are used:

infrastructure-free and infrastructure-based approaches.

As indicated by the names, the infrastructure-free approaches are the ones not needing any hardware infrastructure to be deployed, while, in contrast, infrastructure-based approaches are the ones requiring certain hardware to be installed on the site where the indoor localization system is deployed [22]. Although they can be based on different core technologies, the infrastructure-based systems all rely on "ranging", measuring the distance between two (or more) devices. Usually one of the devices is the target device, the one to be localized. Between the infrastructure-based systems, the UWB technology is reportedly known to be very precise, achieving an accuracy below $0.5m$. Its drawback is the infrastructure overhead in terms of costs and deploying time, that make it difficult to be widely used [5][22]. A common approach for both the infrastructure-based and the infrastructure-free systems is the so called "fingerprinting". This technique consists of two stages; in the first stage, also known as "offline stage" or "training stage", some signals (usually WiFi, bluetooth or the magnetic field) are measured at defined positions and stored as fingerprints in database. During the second stage, known as online stage, "the position corresponding to new measured signal quantities is estimated using the positions associated with the stored fingerprints that are the most similar when compared to the new measurements" [5]. Between the infrastructure-based systems using the fingerprinting approach, there are the WiFi based systems, where the measured signals are the wireless signals present in the area of interest. An example could be a shopping mall, where there are usually several WiFi routers giving the internet connection to the users. For the WiFi based systems, the stored measure for the fingerprint is usually the received signal strength (RSS) at specific locations [5]. On the opposite side, there are the infrastructure-free systems where location estimation relies only on the sensors typically embedded in smartphones and tablets (such as gyroscope, accelerometer, magnetometer). For fingerprinting approaches, the relevant importance of magnetic field fingerprinting has been found out. The magnetic field doesn't need any hardware deployed at all on the site and several surveys on the topic are showing it has better accuracy in localization than WiFi fingerprinting [5][18][22].

The magnetic field fingerprinting is based on a physical phenomenon called magnetic distortion, strictly related to the magnetic induction. The earth's magnetic field, also known as geomagnetic field, can be considered constant and pointing toward the magnetic north pole of the planet. However, inside buildings, the presence of ferromagnetic construction materials and objects lead to the magnetic distortion phenomenon [18][6]. The ferromagnetic materials, previously magnetized due to the magnetic induction phenomenon, keep a residual induction that is responsible of a new magnetic field. As a consequence, inside buildings, the local magnetic field is the sum of two components: the earth's magnetic field and the magnetic field created by ferromagnetic materials. The total magnetic field measured inside buildings

can be seen as if the earth's magnetic field is distorted by the ferrous objects (magnetic distortion). Due to this physical behaviour, each position in indoor spaces is defined by its own magnetic fingerprint and it's been proved that the magnetic fingerprint can be considered as stable in time [2].

As described before, the fingerprinting approach includes an offline stage and an online stage. The offline stage is usually realized in two steps: at first the magnetic measures are collected and then, during the map construction step, the magnetic map is created [18]. In this context, the majority of the relevant studies focus on smartphone-based indoor localization systems. Taking into account that most of the people own a smartphone, these systems are the most attractive ones.

In Magicol [27] the proposed magnetic measurement step consists of a surveyor walking along predefined paths, keeping the smartphone in a fixed position. Using the inertial measurements, coming from the accelerometer and the gyroscope, it's finally possible to detect the user changes of direction. In this way, it's possible to match the magnetic measures with a position on the floorplan. The measured magnetic data are finally interpolated to create the magnetic map. The drawback of this procedure is the fact that the data are user dependent and prone to errors: a data cleaning step is further required. Magicol suggests also to extend the magnetic capture procedure via crowdsourcing. This technique consists to query the final users of the indoor localization system to capture the magnetic field, following the previously exposed procedure.

Another approach is proposed by mPILOT [7]. The study field of mPILOT is a corridor. They propose to take 100 magnetic measurements on the same walking straight line, with a distance of $1m$ between each measure. Consequently, an interpolation is performed to find the missing magnetic values all over the corridor. In this survey, there is no investigation about the 2D behaviour of the magnetic field. Measuring the magnetic field along a straight line only is a limiting unidimensional constraint.

From the magnetic map reliability point of view, a step further is made by MaLoc [6]. MaLoc's aim is to generate a magnetic map reproducing the 2D behaviour of the magnetic field in a defined area of interest of the building. They propose to measure the magnetic field along "short" parallel lines, with spacing of $60cm$ between each parallel line, till the whole area is completely measured. As in the previously described studies, it is performed an interpolation to find the missing values between the measured lines. The MaLoc drawback is the time consuming procedure. Despite being very accurate, capturing the magnetic field along short lines spaced by $60cm$ takes long time even for small maps.

All the three methods previously described were based on the assumption that the magnetic field doesn't change along the height. Considering that the user's smartphone moves in the 3D space, ignoring the height information could result in a loss of precision of the system. The importance of the 3-dimensional behaviour of the

magnetic field in fingerprinting approaches has been investigated in several studies and it has been proved that the height of the magnetic sensor influences the measurement's value. In other words, the magnetic field is height-dependent inside buildings [1][4][3].

Currently, one of the biggest shortcomings of 3-dimensional magnetic fingerprinting is the measurements collection. An high number of measurements is required to have a good model of the magnetic map. Furthermore, for each measure an accurate positioning is needed to assign the measurements to the ground truth positions [1].

1.2 Context of the work

This thesis is based on the Magnetic Indoor Localization Engine (MILE or engine) that Sony is developing in the Stuttgart Technology Center (STC). The MILE combines measurements from several sensors, embedded in the user's smartphone, to obtain a final system that, based on the geomagnetic fingerprint, is able to overcome magnetic field ambiguities. Even though a magnetic map gives absolute information about the location, it could happen that multiple locations have a very similar magnetic fingerprint. This ambiguity is overcome through a Pedestrian Dead Reckoning (PDR) solution, estimating the user motion.

As exposed in 1.1, geomagnetic fingerprint based indoor localization solutions generally require an offline phase during which the magnetic fingerprint of a selected area is created. In this regard, the MILE needs a surveyor that walks in the localization area with a magnetic sensor (e.g. the smartphone), in order to capture the magnetic field following multiple predefined reference points. The reference points are usually selected as corner points or other characteristic points of the floorplan. Once the surveyor accomplishes his task, all the collected data are interpolated to estimate the magnetic field values at non surveyed points of the floorplan. In this way, the 2-dimensional magnetic map of the area is generated. The magnetic map constitutes the database of the MILE and in the online phase, it is compared to the magnetic field values measured by the user's smartphone. There are some drawbacks in the described procedure, firstly related to the surveying procedure. The current surveying method could result to be very tedious, as the surveyor has to accurately follow the reference points, trying to walk with a constant speed between them. The accuracy of the surveyor directly influences the accuracy of the position where each magnetic measure is assigned. Overall, the matching between the magnetic field and its position has direct impact on the final magnetic fingerprint map. A non-reliable magnetic map, not describing the real magnetic field, will lead to lack of precision of the position estimation. Another drawback is that, even assuming to have an accurate surveyor, the described procedure is unsuitable for large scale indoor environments because of its time consuming requirements. Moreover, like other indoor localization techniques exposed in 1.1, the fingerprint is based on the assumption of having a magnetic field that does not vary along the

height. This means that, if we describe the localization area with a 3D reference frame K^{xyz} , with xy coincident with the floor and z indicating the height, ignoring the height, the magnetic field B is:

$$B = f(x, y) \tag{1.1}$$

In other words, the final 2-dimensional magnetic map is assumed to describe the magnetic field from the floor till the ceiling, not depending on the height of the measurements. This simplification could influence the final accuracy of the MILE.

1.3 Objective of the work

Considering the context of the work and the current drawbacks it presents, there are different areas the thesis wants to investigate. The main objective is to study the 3D behaviour of the magnetic field in indoor spaces and to extend the MILE to be able to work in 3D. To extend the MILE, two steps are performed. At first the map generation process is extended to generate a 3D magnetic map. The new obtained 3-dimensional magnetic map constitutes the magnetic fingerprint of the chosen localization area and it's fed to the MILE. Thereafter, the engine framework is extended to be able to work in the 3D space.

In the end, the final goal is to carry out an analysis of the possible improvements of the engine positioning accuracy, using a 3D magnetic map against the state of the art 2D magnetic map. In this regard, during the final analysis it is of interest to find out if using a 3D magnetic map, it is possible to achieve the height tracking of the user's smartphone.

To accomplish the main goals, at first, the current 2D map generation framework is modified to improve the interpolation and extrapolation technique targeting the creation of a more realistic map. Consequently, it is investigated and chosen, based on a thorough evaluation, the best 3-dimensional interpolation and extrapolation method for the 3D map generation. Additionally, two V-SLAM (visual - simultaneous localization and mapping) systems are evaluated with the aim of providing an alternative process to the current tedious magnetic measurements collection.

1.4 Organization

The thesis development is organized as follows:

- In Chapter 2, the magnetic indoor localization engine (MILE) is described.
- In Chapter 3, the current 2D magnetic interpolation-extrapolation methods are described and analysed. Additional 2D methods are introduced. Finally, a detailed evaluation of all the 2D methods is performed.

- In Chapter 4, the magnetic measurement setup is designed to capture new magnetic measurements.
- In Chapter 5, the 3D magnetic map generation procedure is implemented. The 3D interpolation-extrapolation methods are introduced and evaluated. Finally, the MILE is extended to be able to support the 3D mode.
- In Chapter 6, a thorough evaluation of the 3D against 2D performances is conducted. The height tracking capability of the 3D MILE is analysed and a possible integration of the smartphones' barometers is investigated.
- In Chapter 7, the final conclusions and future work suggestions are reported.

Chapter 2

The Magnetic Indoor Localization Engine

In this chapter, the Magnetic Indoor Localization Engine, MILE, introduced in Section 1.2, is analyzed in depth. As described before, the MILE uses a sensor fusion approach to perform the localization of the user equipped with his smartphone. The key idea of this localization system is to locate the user employing the magnetic anomalies present inside buildings. The only magnetic measurements are not enough to properly reach the localization goal. There exist magnetic ambiguities in the localization area. These are present because many locations inside buildings have similar magnetic fingerprints. As a consequence, even if the magnetic fingerprint gives absolute information about the location, the MILE could return an incorrect localization estimation. To overcome this drawback, the engine uses a Pedestrian Dead Reckoning (PDR) technique, able to estimate the user motion. Using the PDR along with the magnetic fingerprint of the localization area, it is possible to compensate for magnetic ambiguities. Moreover, the PDR does not provide the exact motion estimation of the user. Its output is affected by a cumulative error that leads to drifts of the user motion result. In this case, the magnetic field is capable of compensating for cumulative errors of the PDR. The integration of the magnetic fingerprint and the PDR is exploiting the sensor fusion approach. As already seen in Section 2.1, different sensors are used to reach a common goal. In Figure 2.1, it is shown the MILE architecture, explained in detail in the next sections.

2.1 Sensors

The hardware equipment needed to run the engine consists of standard smartphone sensors, as the ones included in the IMU (inertial measurement unit) of nowadays

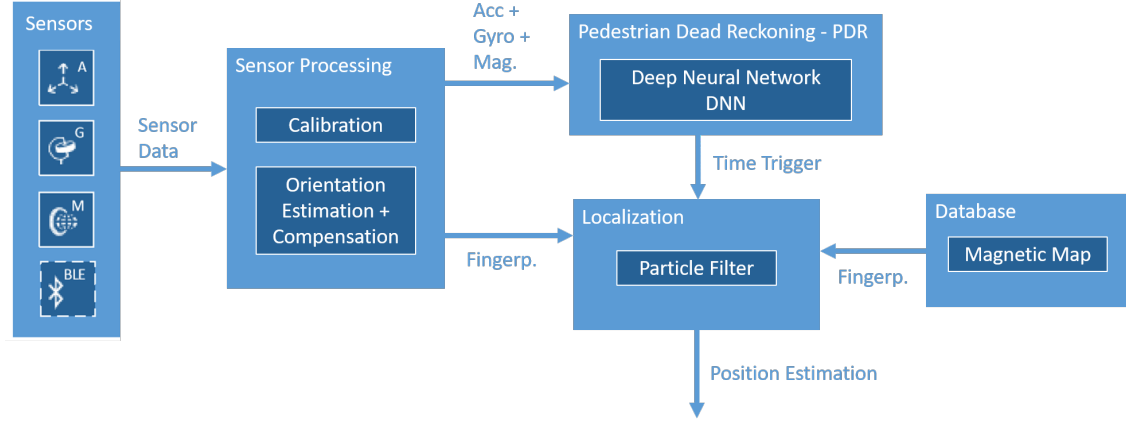


Figure 2.1: Magnetic Indoor Localization Engine (MILE) architecture

smartphones. The sensors are the accelerometer, measuring the accelerations of the smartphone, the gyroscope, measuring the angular velocities of the smartphone, and the magnetometer used to measure the local magnetic field sensed by the smartphone. The outputs of these sensors are 3-dimensional vectors in the cartesian coordinate system of the device. Moreover, to solve the initialization problem of the particle filter (see Section 2.5), some Bluetooth beacons are deployed in the localization area (by the provider of the MILE system).

2.2 Sensor processing

2.2.1 Calibration

In order to make the MILE independent from the smartphone model, it is necessary to introduce a calibration process for the magnetometers. This necessity arises because, without a proper calibration of the magnetic sensor, different smartphones measure different magnetic fields at the same point in space. As known from literature [6][18], the magnetic field vector B_p sensed by a smartphone in the sensor coordinate system can be modeled as:

$$B_p = WRB_e + V \quad (2.1)$$

where W is the vector representing the contribution of the soft iron effect, V is the vector representing the contribution of the hard iron offset, B_e is the local magnetic field vector in the earth coordinate system and R is the rotation matrix to transpose the measurements from the earth coordinate system to the smartphone coordinate system. The hard iron offset V is generated by ferromagnetic materials close to the magnetic sensor, which create their own constant magnetic field (e.g.

the smartphone speaker magnet). The soft iron effect W is instead generated by unmagnetized ferromagnetic materials. When close to the magnetic sensor, they react to the external magnetic field introducing their magnetic interference (e.g. iron and nikel on the chip-set PCB, they do not necessarily generate a magnetic field by themselves). The vectors W and V are smartphone dependent and they can change over time. They represent the contributions that an appropriate calibration method should cancel. After calibration, all the smartphones should measure the magnetic field without hard iron offset and soft iron effect.

In this regard, there are two possible approaches implemented in the MILE. The first calibration approach, known as ellipsoid, is a calibration method based on the least squares ellipsoid fitting algorithm [8]. This method is directly applied, inside the engine, to the raw uncalibrated magnetic measurements sensed by the smartphone. It requires a user iteration to compute the ellipsoid fitting parameters. To do so, the user has to follow a routine before each magnetic capture, rotating the smartphone around its geometrical center. The second calibration approach, known as android, is a calibration method directly implemented inside the smartphone firmware. The android calibration is based on an ellipsoid fitting algorithm similar to the one used in the ellipsoid calibration method. Unfortunately, it is not possible to check the algorithm differences between the two approaches because, in the android calibration, the smartphone outputs directly the calibrated magnetic field values. Currently, there is no recommended calibration method between the ellipsoid and the android. The choice of a preferential method will be further discussed during the thesis development, in Section 5.2.3.

2.2.2 Magnetic Field Representation and Orientation Compensation

A necessary condition to properly run the MILE is that the magnetic fingerprint map and the magnetic field sensed by the target smartphone have to refer to the same reference system. For this purpose, the earth cartesian system K^{xyz} is chosen as the reference system, with xy defining the earth floor and z describing the height variation. In general, the smartphone magnetic sensor outputs the magnetic field vector $B = [b_i, b_j, b_k]$ in the internal sensor cartesian reference system K^{ijk} . To obtain the correct orientation estimation of the magnetic field in the earth K^{xyz} reference system, Android smartphones provide directly the orientation (in K^{xyz} reference frame) through an internal algorithm providing the output in form of quaternions. The smartphone orientation is used inside the MILE for two objectives. At first, to transpose the magnetic sensor measurements in the K^{xyz} reference system, $B = [b_x, b_y, b_z]$. Secondly, it is a necessary input for the PDR, as will be explained in Section 2.3.

Once the magnetic measures are in the earth reference system, there is a change

of coordinates used to describe the magnetic field B . Instead of using the cartesian coordinates $[b_x, b_y, b_z]$, it's used a different representation, describing the magnetic field B by its magnitude, inclination and azimuth. In the new set of coordinates, named as *mia*, the magnitude is the intensity of the magnetic field vector B , measured in microtesla μT . The inclination is the angle between the magnetic field vector B and the earth floor (the plane xy in the earth coordinate system). The azimuth is the angle between the magnetic field vector B and the true magnetic north. In figure 2.2 the relation between the cartesian coordinates $[b_x, b_y, b_z]$ and the new set of coordinates $[m, i, a]$ it's shown. The choice of using the *mia* coordi-

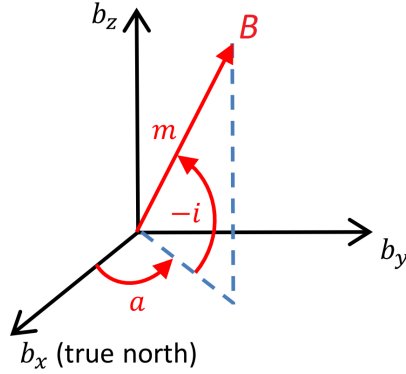


Figure 2.2: Relation between xyz and mia coordinates

nates is justified by practical advantages they can bring in a development phase. For example, the possibility to easily switch, if needed, to different fingerprinting modes. In fact, as reported by [18], some indoor localization systems are using a 1D fingerprint of the magnetic field, storing only the magnetic magnitude.

2.3 Pedestrian Dead Reckoning - PDR

Inside the MILE, the particle filter is the core part and it is updated through a time trigger every 1s interval. At each trigger the PDR, whose objective is to compute the user movement, passes to the particle filter the new estimated relative displacement and heading change. The PDR is implemented as a deep neural network (DNN) using as input accelerometer, gyroscope and magnetic readings. Its output is a 3D vector describing the movement in the smartphone reference system K^{ijk} . As the particle filter works in the earth reference system K^{xyz} , a 3D rotation is performed knowing the smartphone orientation (section 2.2.2). In this way, the output vector belongs to the required reference frame. The PDR accuracy suffers of an error accumulating over time. This error will be considered during the covariance matrix tuning of the likelihood modes, as will be described in Section 2.5.

2.4 Database magnetic map

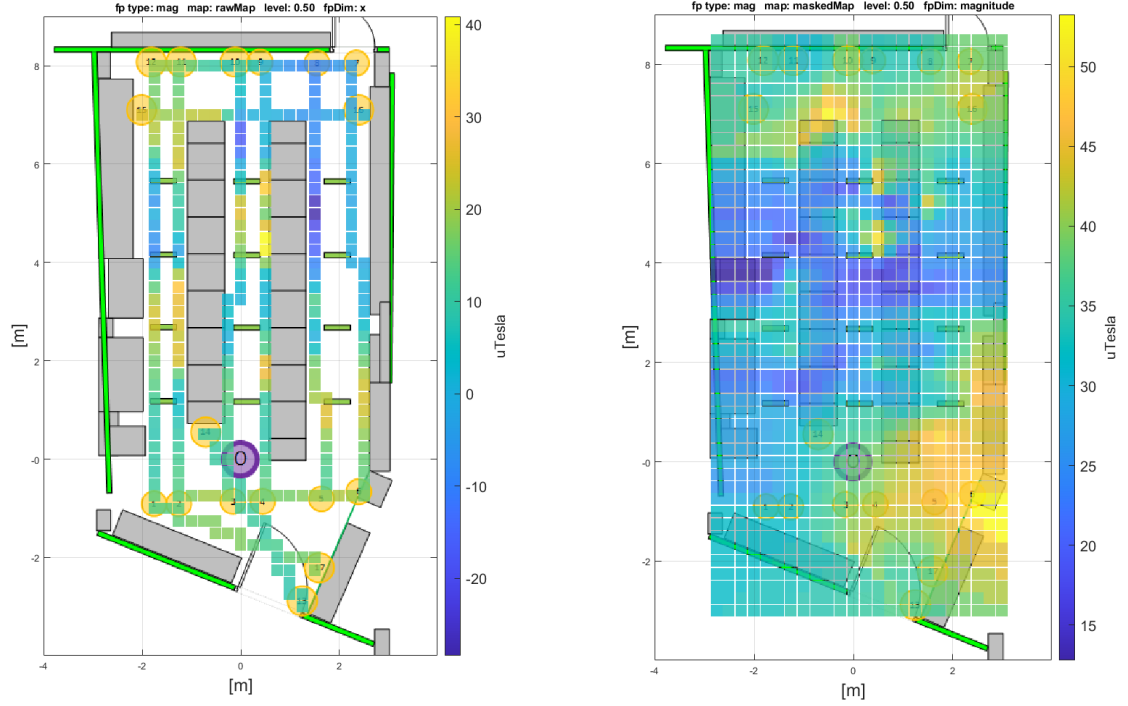
The database magnetic map is the ground truth magnetic map of the area where the magnetic localization system is deployed. The magnetic map is the result of the map generation process, investigated in Chapter 3 and Chapter 5. In its starting version, it is saved inside the MILE as a 2-dimensional grid map, in form of a matrix. Each grid point is defined by its 2D coordinates (x, y) in the absolute reference system xyz . For each grid point, the three magnetic field components, magnitude, inclination and azimuth, defined in Section 2.2.2, are saved. The default resolution of the grid map is set to $0.25m$ and there is the possibility to change it. In this way, the default grid map is made up of squared grid points of dimension $0.25m \times 0.25m$, like in Figure 2.3b. Referring to Figure 2.3b, it is necessary to note that not all the grid points are obtained through a direct measure of the magnetic field. As described in Section 1.2, the surveyor, measuring the magnetic field of the localization area, walks on a predetermined trajectory. The grid points derived from the trajectory are labelled as "measured grid points" (Figure 2.3a) and their value is the actual measured value. All the other grid points, labelled as "non-measured grid points", are computed through an interpolation or extrapolation procedure.

2.5 Particle filter localization

The core of the engine is a particle filter, a popular filtering method used for localization purposes. A particle filter is a non-parametric recursive Bayes filter [17]. It aims to find the posterior distribution of the states of a Markov process using a finite number of samples, called particles. Each of the particles represents a hypothetical state of the process. According to the likelihood of the hypothesis compared to real measurements, a weight is assigned to all the particles. The weighted particle set represents the posterior distribution mentioned before. One of its main features is the fact that it can describe even multimodal distributions. In the MILE, the Markov process is the user movement. The engine uses a 4-dimensional particle filter, defining the user state as $\mathbf{s} = [x, y, \theta, l]$. Where x and y are the 2D coordinates of the user in the localization area, θ is the motion heading and l is the step length. The number of particles is set to 5000 and each particle k is defined as $\chi_k^t = (s_k^t, w_k^t)$ with s_k^t state and w_k^t weight of the k^{th} particle at time t . Areas of the state space where the particles are dense will represent areas where it is most likely to find the user true location.

As shown in Figure 2.4, the first step of the particle filter is the initialization. In this step, the 5000 particles are created. Their states are initialized around a known starting point according to a gaussian distribution. For this purpose, a bluetooth beacons approach is under development.

Once the particles are initialized, there are three core steps recursively repeated inside the filter: updating, weighting and resampling. The three steps are triggered



(a) Measured grid points, reflecting the surveyed trajectory

(b) Magnetic magnitude map, note the presence of both measured and non-measured grid points

Figure 2.3: Example of magnetic grid map.

every 1s interval, when, as seen in Section 2.3, the PDR inputs the particle filter with the estimated walked distance and heading change. In the updating step, the particle states at time t are updated following the motion model (2.2), starting from the previous step states, at time $t - 1$.

$$\theta_k^t = \theta_k^{t-1} + \Delta\theta^t + n_\theta \quad (2.2a)$$

$$l_k^t = l^t + n_l \quad (2.2b)$$

$$x_k^t = x_k^{t-1} + l_k^t \cdot \cos(\theta_k^t) \quad (2.2c)$$

$$y_k^t = y_k^{t-1} + l_k^t \cdot \sin(\theta_k^t) \quad (2.2d)$$

In the motion model (2.2), $\Delta\theta^t$ and l^t are, respectively, the heading change and the movement length in the earth plane xy at time t , derived from the PDR output. n_θ and n_l are two additive gaussian noises used to handle, respectively, the heading change and the movement length uncertainties.

The second of the core particle filter steps is the weighting step. In this step, also known as likelihood update step, the particles are weighted according to their similarity with some measured observations z^t . The likelihood is performed through

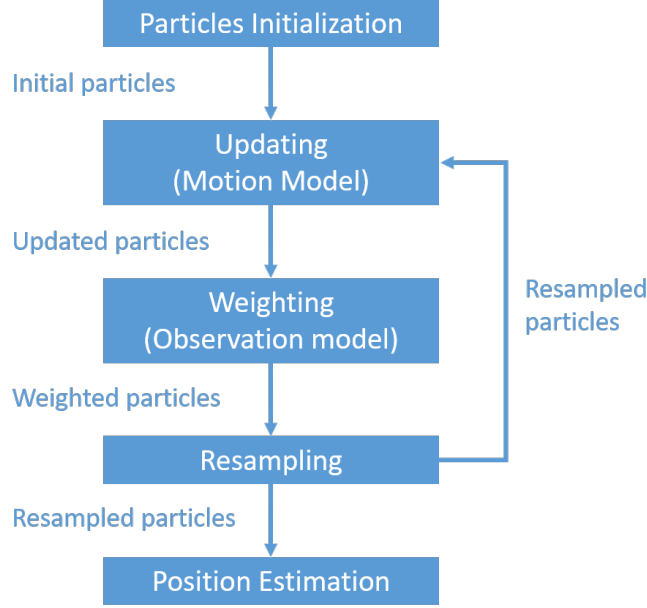


Figure 2.4: Particle filter workflow

the following gaussian pseudo-distribution observation model $P(z^t|s^t)$:

$$P(z^t|s^t) = \frac{1}{\sqrt{(2\pi)^n \det(C)}} \exp \left[-\frac{1}{2} (z^t - \text{obs}(s^t))^T C^{-1} (z^t - \text{obs}(s^t)) \right] \quad (2.3)$$

where n is the dimension of the observation z^t , C is the covariance matrix and $\text{obs}(s^t)$ is the function returning the observation values for the considered state s^t . The MILE likelihood step uses three sub-likelihood observation modes. Subsequently, to obtain the final overall likelihood, the product of the sub-likelihoods is performed. The three likelihood modes are denoted as P^{mag} , magnetic likelihood, P^{pdr} , PDR likelihood and P^{flr} , floorplan likelihood.

For the magnetic mode P^{mag} , the measurement vector is $z^t = [m^t, i^t, a^t]$. It is a 3D vector containing the magnitude m^t , the inclination i^t and the azimuth a^t of the magnetic field measured at time t . In this mode the $\text{obs}(s^t)$ function performs a linear interpolation based on the magnetic database grid maps (Section 2.4). The aim of the interpolation is to get the magnitude, the inclination and the azimuth estimates for all the k particles at time t , χ_k^t , according to their states s_k^t . The magnitude, the inclination and the azimuth are assumed to be uncorrelated [10], therefore the covariance matrix C is a 3×3 diagonal matrix.

For the PDR mode, P^{pdr} , the measurement vector is $z^t = [l^t, \theta^t]$, containing the step length and the motion heading coming from the PDR (section 2.3). In this mode the $\text{obs}(s^t)$ function returns the step length and the motion heading for each

particle. As for the previous mode, P^{mag} , the two measurements are assumed to be uncorrelated and the covariance matrix C is a 2×2 diagonal matrix. Note that usually the PDR is affected by a cumulative error (Section 2.3) and the motion model already includes the position estimation. Consequently, a low weight is given to this sub-likelihood setting high values to the covariance matrix.

The last sub-likelihood, floorplan mode P^{flr} , differs from the others because it does not use the gaussian pseudo-distribution observation model. In this mode, the floorplan of the localization area brings the information about the "no-go areas" and the walls. Particles that are falling into a no-go area or passing through a wall are assigned with a floorplan likelihood equal to $P^{\text{flr}} = 0$. On the opposite, all the particles that are following the floorplan restrictions are assigned with a floorplan likelihood equal to $P^{\text{flr}} = 1$. In Figure 2.5 an example of the floorplan mode is shown, with green particles following the floorplan constraints and red ones trying to perform non-allowed movements.

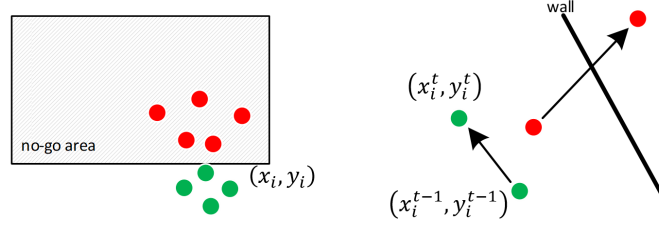


Figure 2.5: Floorplan mode behaviour

Finally, the overall likelihood for each particle k is computed as:

$$P_k = P_k^{\text{mag}} \cdot P_k^{\text{pdr}} \cdot P_k^{\text{flr}} \quad (2.4)$$

The resulting likelihoods are then normalized to obtain the particle weights. In this way, the sum of the weights of all the N particles is equal to 1: $\sum_{k=1}^N w_k = 1$.

The third particle filter inner step is the resampling step. In this phase, the particles are resampled to keep and duplicate only the particles with higher weight while discarding the lower weighted ones. This procedure is based on the assumption that particles with higher weight describe the user's true state more likely than particles with a lower weight. Thus, all the particles are resampled with a probability proportional to their weight. To do this, inside the MILE, a set of N samples between 0 and 1 is generated, with N number of particles. Subsequently, using the CDF (cumulative distribution function) given by the normalized particle weights w_k , a new particle set is drawn with replacement from the current particle set. For the sample generation, a systematic approach is used. It generates N uniformly distributed samples (between 0 and 1) where each of the samples in the spaces of width $\frac{1}{N}$ is located relatively in the same position.

The resampled particles are used as input for the successive particle filter call.

As can be visualized in Figure 2.6, their weighted average gives the user position estimation.

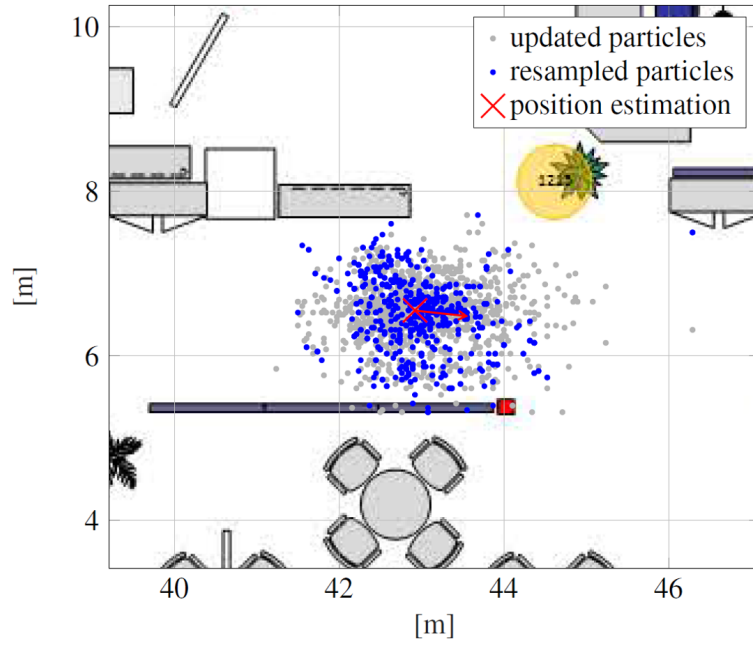


Figure 2.6: Particle filter localization example

Chapter 3

2-dimensional magnetic map generation

In this chapter, the 2D magnetic map generation and the improvements aimed to generate a reliable map are analyzed. All the results shown in this chapter are obtained from magnetic measurements that were captured prior to the thesis. The test bed consists of two different office environments that were surveyed using the magnetic sensor of a smartphone.

3.1 Map generation framework

The 2D map generation process aims to generate the 2D magnetic map based on magnetic measurements taken along a known trajectory (Section 1.2). As explained in Section 2.4, the MILE works on a 2-dimensional grid map where only part of the grid points are directly measured (measured grid points). The other part of the grid points is obtained through an interpolation or extrapolation process. The interpolation-extrapolation process influences the accuracy of the final map since not all the interpolation-extrapolation techniques give reliable results. In this context, it's necessary to differentiate between interpolation and extrapolation. The interpolation is the mathematical process used for estimating values lying within a known dataset. On the opposite, the extrapolation is the mathematical process aimed to extrapolate values lying beyond a known dataset [26]. The boundary between the two areas, the interpolated and the extrapolated one, is known as convex hull and it is visualized in Figure 3.1 by the orange line.

Before analyzing the available interpolation and extrapolation techniques, it's necessary to understand how differences between sensors frequencies are handled. Usually, different sensors refer to different clocks. This means that data coming from the IMU, the magnetometer or from the Android internal orientation estimation, have different rates. To manage these differences, before the map generation starts,

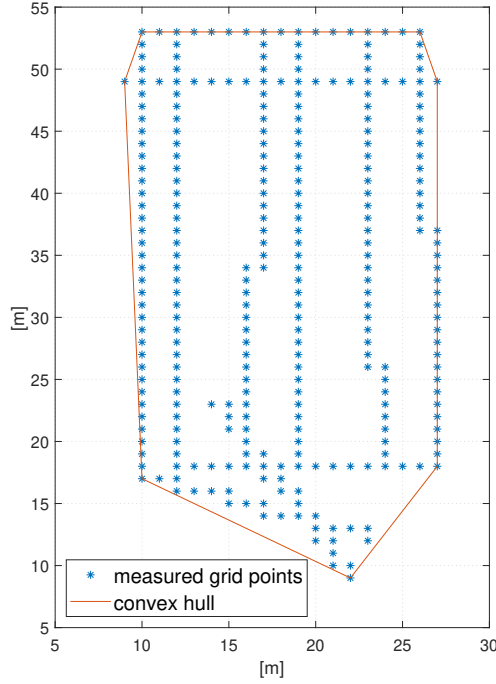


Figure 3.1: Convex hull

a resampling of all the sensors measures to a common sampling frequency, typically set to 50Hz, is performed. The chosen frequency is relatively high compared to the surveyor walking speed during the magnetic field capture. This fact leads to grid points defined by more than a single measurement. The following approach is applied: at first all the measures are assigned to the closest grid points. Subsequently, for all the grid points defined by more than a single measurement a median filter is applied. In this way, each grid point is defined by a single magnetic measure that is the median value of all the measures previously assigned to that grid point.

At the current phase of the MILE development, the whole framework is developed on MATLAB [23], including the map generation step. Considered this, the applied interpolation and extrapolation techniques were all based on the software's built-in functions. The development environment gives the possibility to choose between two families of 2D interpolation (and extrapolation) functions. The difference between the two classes is the type of the data they handle, that can be gridded or scattered. In a 2D environment as the one we are considering, gridded data are those data that are equally spaced in both x and y dimensions. Referring to Figure 3.1, our data belong to the scattered category in the considered scenario, since there is no constant pattern for their spatial distribution. For interpolation and extrapolation

of scattered data, there are two MATLAB built-in functions: *scatteredInterpolant* and *griddata*. Both functions use the Delaunay triangulations of the scattered sample points to perform the interpolation-extrapolation process. Even though the two functions are based on Delaunay triangulations, they differ in the triangulation method they use. Furthermore, they have the possibility to choose between several interpolation and extrapolation methods.

The *scatteredInterpolant* function performs both interpolation and extrapolation of some input data. In particular, the available interpolation methods are the linear interpolation, the nearest neighbor interpolation and the natural neighbor interpolation. For the extrapolation, *scatteredInterpolant* gives the possibility to choose between linear extrapolation and nearest neighbor extrapolation.

In contrast to *scatteredInterpolant*, the *griddata* function performs only interpolations. To overcome this shortcoming of *griddata*, an integration with *scatteredInterpolant* is performed to obtain a complete map. In other words, when using *griddata* in the map generation framework is performed a *griddata* interpolation coupled with a *scatteredInterpolant* extrapolation. The available interpolation methods for the *griddata* function are the linear interpolation, the nearest neighbor interpolation, the natural neighbor interpolation, the cubic interpolation and the biharmonic spline interpolation (called *v4* in MATLAB).

All the available interpolation and extrapolation methods are based on different approaches that lead to different output results. The quality of the interpolation can be evaluated by the continuity of the result. It can vary from discontinuous, in the case of nearest neighbor interpolation, to continuity C^2 for cubic and biharmonic interpolations. At the current development step of the map generation framework, there is no preferential function or preferential interpolation-extrapolation method for the magnetic map creation. This is because all the interpolation-extrapolation methods are producing very similar results between them. The only method clearly producing a discontinuous result is the nearest neighbor interpolation, not feasible with the expected real magnetic field behaviour. Before evaluating which of the available methods is most suitable for reproducing a realistic magnetic field behaviour, it is necessary to deal with some recurrent interpolation-extrapolation problems in the next Section 3.2.

3.2 2D magnetic map issues

A first step to choose the best interpolation-extrapolation method is to analyze the results and try to find a key metrics to evaluate them. In order to do this, it's necessary to understand what is the physical meaning of the interpolation extrapolation result. With this in mind, it is finally possible to evaluate the reliability of the outcome. In the map generation process, the physical quantity that we want to estimate, based on some known measurements, is the distorted earth's magnetic

field. As already described in Chapter 1, while outdoor the magnetic field is moreover constantly pointing towards the earth magnetic north pole, indoor there is not a constant model for it. Thus, in this case, it's not possible to know the a priori ground truth of the magnetic field behaviour. Nevertheless, we can expect that the magnetic field, as a natural field, should vary in a continuous manner and without strong discontinuities all across indoor spaces. The previous one is the basic assumption to start the evaluation of all the map generation methods. Consequently, the methods producing a smooth result are considered the more suitable ones. During the evaluation of the magnetic maps that are obtained through the MATLAB built-in functions, it is observed that their outcomes occasionally suffer of some errors. The errors are usually located around the convex hull and they tend to propagate in the extrapolation area of the map. The reason for the errors to be in the extrapolation area is that the extrapolation is usually more difficult and unconstrained than the interpolation. It can be considered the main weakness point of the map generation framework. In Figure 3.2a and 3.2b it is possible to notice, in the red squared areas, the typical errors found when using, respectively, linear extrapolation and nearest neighbor extrapolation.

Considering Figure 3.2a, it is clear that when using a linear extrapolation there is

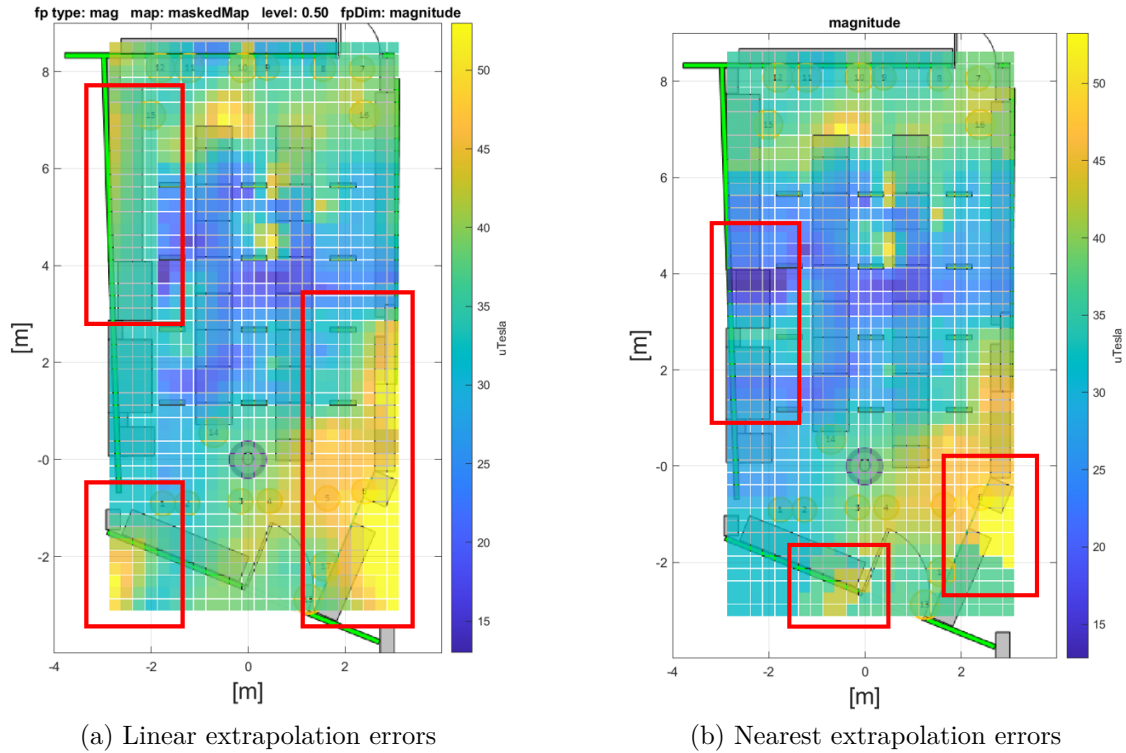


Figure 3.2: Example of non natural reconstruction of the magnetic field using MATLAB built-in functions.

the generation of high peaks in extrapolated areas. Furthermore, for some of the evaluated methods using linear extrapolation there could be also a step discontinuity along the edge of the convex hull (top left red box in Figure 3.2a). Whereas, when using the nearest neighbor extrapolation there is the creation of the errors shown in figure 3.2b. In this case, there is a step discontinuity starting from the last grid point on the convex hull edge and propagating inside the extrapolated area. A necessary observation is that, even if the extrapolation area of the map is usually smaller than the interpolation area, it is not possible to underestimate the errors happening in that area. This because, as will be described in later stages, the engine performance is very sensitive to magnetic fluctuations.

After a first visual analysis, it is possible to conclude that all the considered MATLAB built-in functions were suffering the described defects. Some of the methods were suffering them more and some others less. In Figure 3.3, the result for the magnetic magnitude obtained using `griddata` function with natural interpolation and nearest extrapolation is shown. The overall map looks smooth but it still presents the typical error related to nearest extrapolation.

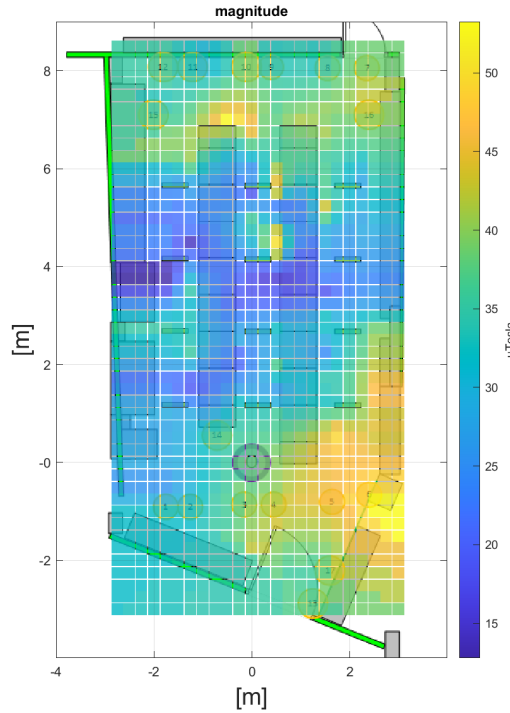


Figure 3.3: Magnetic magnitude map using `griddata` natural interpolation and nearest extrapolation.

As a consequence, before numerically evaluate the best performing function, additional map generation methodologies not relying on the MATLAB built-in

functions are studied in Section 3.3.

3.3 Additional interpolation and extrapolation approaches

To avoid the previously exposed problems deriving from the MATLAB built-in interpolation and extrapolation functions, new methods are investigated. In this regard, a similarity between the magnetic map generation problem and the so called image inpainting problem has been found. Image inpainting is a term that comes from the art restoration field, describing a craftsman trying to restore damaged paintings following the style of the original artist [11]. Referring to the already seen Figure 2.3a, the map containing the only measured grid points can be seen as the corrupted painting. All the missing values are the points of the painting we want to restore, following the original behaviour, that in this case is the real magnetic field behaviour.

The digital image inpainting is a well known problem and nowadays there are many methods used to solve it [20]. One of the most common methods relies on using partial differential equations (PDE) for each of the missing values of the image or the magnetic map. To use this inpainting method in the map generation framework, a function [13], that is available on the MATLAB "File Exchange" community, is integrated into the framework. The function enables to inpaint a matrix over missing values with five different boundary conditions. Between the five boundary conditions, only two of them are found to be suitable for map generation purposes, returning realistic magnetic maps as a result. The first of the two methods is based on PDE, thus, from now on, it will be mentioned as inpaint PDE. It solves the Laplacian equation for each missing value by considering the four adjacent neighbors (two on the x axis and two on the y axis) as constraints. In Figure 3.4, it is noticeable that with this method there is still the same problem as the linear extrapolation, with the magnetic field diverging outside the convex hull. The second of the two feasible methods has been named inpaint NaNs (NaN stands for Not-a-Number). It uses the "spring metaphor", which means that each of the missing values is modelled as if it is connected with a spring to the 8 neighbors. In this method, a linear system of 8 equations is solved for each of the missing values. A "spring" weight equal to 1 is assigned to each of the neighbors along the x and y axis while "spring" weight equal to $\frac{1}{\sqrt{2}}$ is assigned to the diagonal neighbors.

Another common methodology that is employed for image inpainting purposes uses the discrete cosine transform (DCT). In this regard, another inpainting technique [15][16], taken from the MATLAB "File Exchange" community, is added to the map generation framework. The function, in the following referred as inpaint DCT, performs an iterative process based on discrete cosine transforms. At first, a DCT of the input data is performed. After that, a nearest interpolation is made

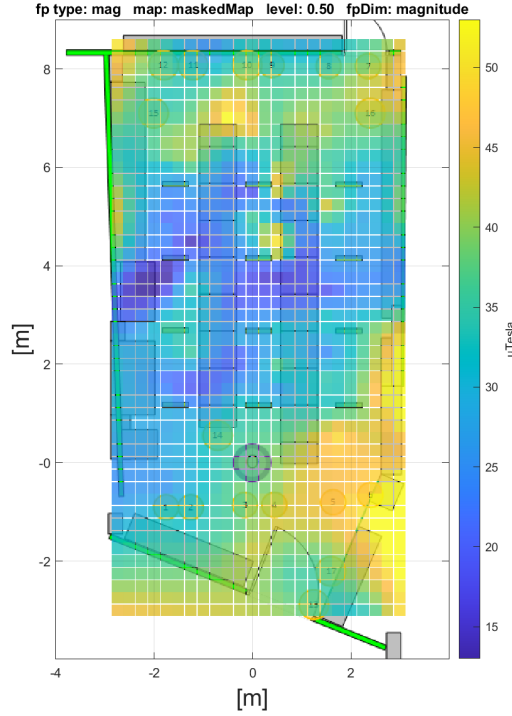


Figure 3.4: Inpaint PDE magnetic magnitude map

in the transform domain. Finally, an inverse discrete cosine transforms (IDCT) is performed. The iterative process converges to the estimated magnetic map.

In Figure 3.5a and 3.5b, it is possible to see the result of the magnitude map obtained using, respectively, inpaint NaNs and inpaint DCT. A relevant result is that using these two methods, the generated maps are not suffering of the previously described problems. Even though both methods provide smooth results, there is a trend of inpaint NaNs to create sharp peaks coincident with some of the measured grid points. Furthermore, sometimes, it tends to flatten interpolated and extrapolated areas. This effect, known as ghost behaviour [20], usually arises when there are large "holes" between the measured grid points, or, in the image processing field, between the known pixels. In Figure 3.6a and 3.6b, the described behaviour can be noticed. In Figure 3.6a, referring to inpaint NaNs, the brightest grid point is a measured grid point. All around it, it is possible to notice that the other grid points are flattened, thus, a sharp peak of the magnetic field is generated. In Figure 3.6b, referring to inpaint DCT, the same area of Figure 3.6a is shown. In this case, it is possible to notice that inpaint DCT generates a smoother peak. Generally, since extrapolation is more prone to errors, it is necessary to consider that the surveyor should maximize the area he is surveying during the magnetic capture for map

generation. In this way, there are more measured grid points for the interpolation-extrapolation procedure and the ghost behaviour could be avoided. In fact, if the surveyed area is well covered, it should be unlikely to have big missing values holes. Note that, as the final objective of the thesis is to evaluate the 3D magnetic map,

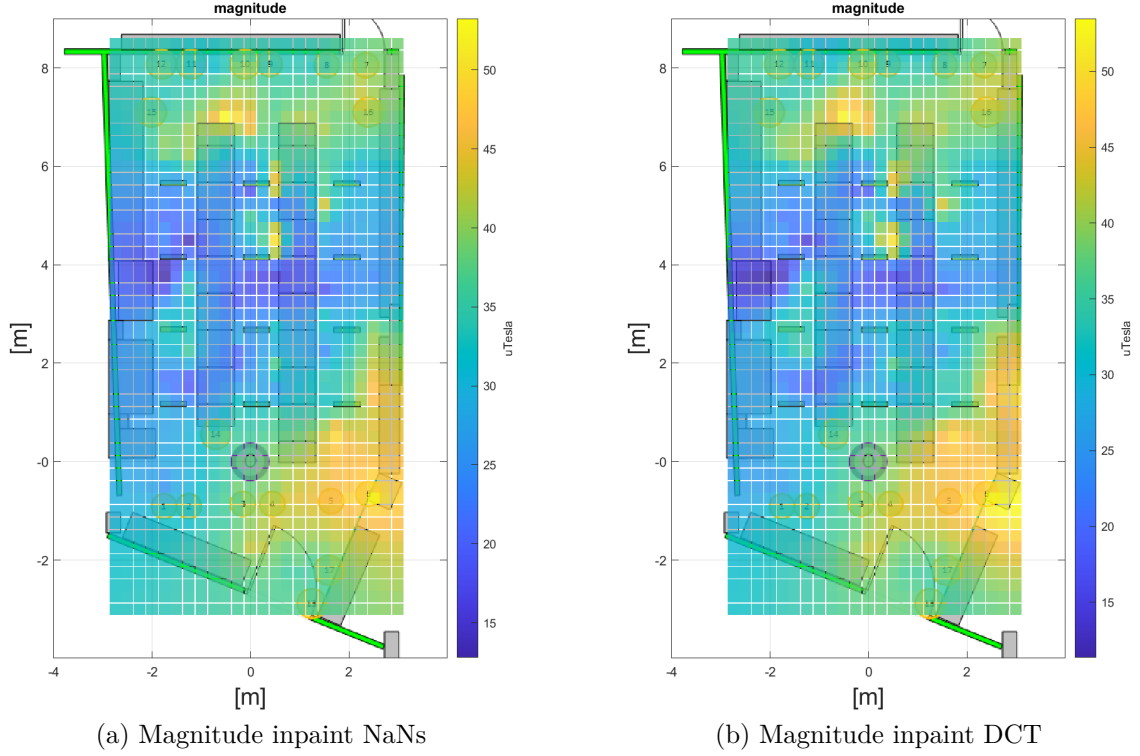


Figure 3.5: Magnetic magnitude maps obtained using inpaint NaNs and inpaint DCT

all the three added map generation methods (inpaint PDE, inpaint NaNs and inpaint DCT) have corresponding functions able to perform a 3D inpainting. The newly added methods differ from the MATLAB built-in functions because they do not provide a parametric function as output of their computations. Their output is a 2D (for the three dimensional case a 3D) matrix, same size as the input matrix, with all the missing values replaced with estimated values. In this way, the framework generates magnetic map whose resolution change is constrained, currently, only to multiples of the smallest grid point dimension (i.e. from $0.25m \times 0.25m$ to $0.5m \times 0.5m$, $0.75m \times 0.75m...$).

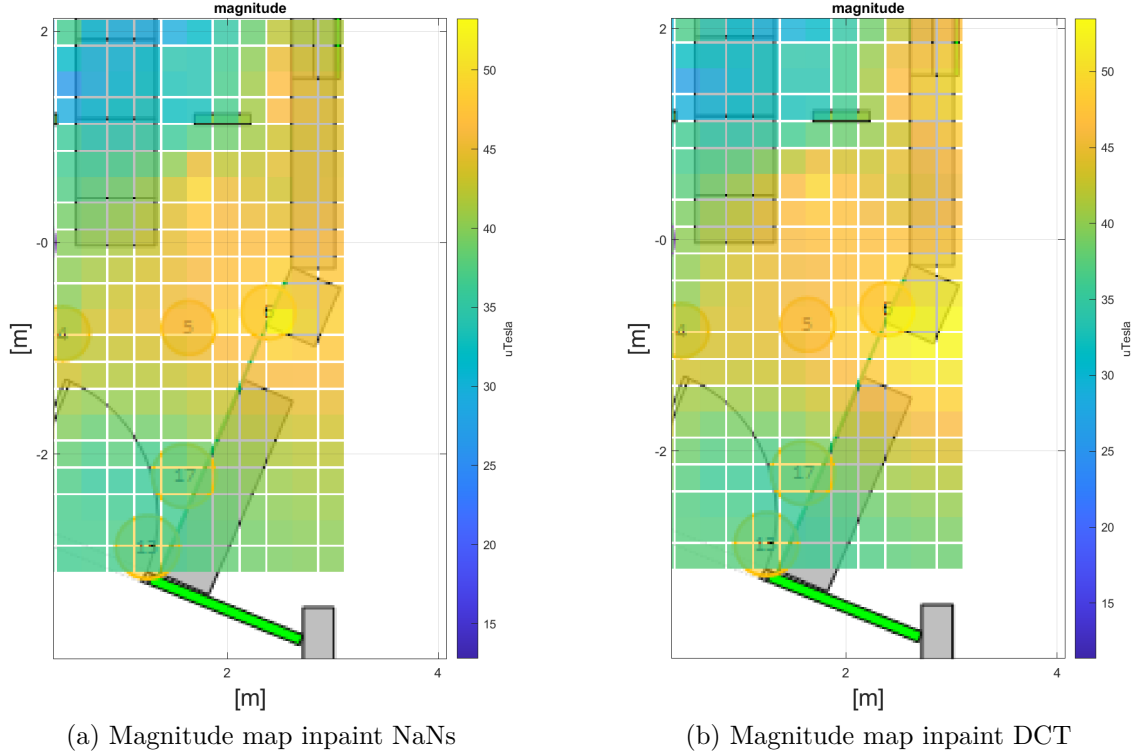


Figure 3.6: Peak of the magnetic magnitude maps obtained using inpaint NaNs and inpaint DCT

3.4 2D methods evaluation

After the integration of the three additional interpolation-extrapolation methods in the map generation framework is concluded, it is possible to perform a detailed evaluation of all the available approaches. The objective is to choose which one is the most suitable map generation method for the MILE. At this step of the thesis development, the only available data are two magnetic captures taken in two different office environments prior the thesis. They will be used to generate two maps: map 1 and map 2.

In a first approach, the idea is to assume that the key metrics to evaluate the methods could be the magnetic field behaviour of the interpolated-extrapolated grid points compared to the magnetic field behaviour of the measured grid points. To analyse these behaviours, some statistics about the measured and the estimated (through interpolation-extrapolation) values are calculated: the mean value and the standard deviation. In Table 3.1 the average absolute difference between the measured values and the interpolated-extrapolated values is shown. Considering this table, it is not possible to derive meaningful conclusions. The only clear result

is that using griddata cubic interpolation and linear extrapolation the estimated map is far from the real magnetic field behaviour. The statistics confirm the visual analysis of the output map. Furthermore, there is a tendency of the methods using linear extrapolation to have mean values difference higher than the correspondent methods using nearest extrapolation. This is mostly due to the divergence of the magnetic field, seen in Section 3.2, when using linear extrapolation. Beyond the discussed considerations, there is no further indication about which method reproduces the natural magnetic field the best.

Mean values Δ [μT]				
function	interp.	extrap.	map 1	map 2
scatteredInterpolant	linear	linear	1.47	1.02
scatteredInterpolant	linear	nearest	0.61	0.56
scatteredInterpolant	natural	linear	1.45	1.11
scatteredInterpolant	natural	nearest	0.60	0.65
griddata	linear	linear	1.01	1.21
griddata	linear	nearest	0.44	0.81
griddata	natural	linear	1.06	1.22
griddata	natural	nearest	0.41	0.92
griddata	cubic	linear	8.44	4.76
griddata	cubic	nearest	1.48	0.57
griddata	v4	linear	1.18	1.35
griddata	v4	nearest	0.54	0.96
inpaint PDE	-	-	0.76	1.84
inpaint NaNs	-	-	0.51	0.74
inpaint DCT	-	-	0.64	1.10

Table 3.1: Average absolute difference between mean values of measured grid points and estimated grid points.

In Table 3.2, the average absolute difference between standard deviations of measured grid points and estimated grid points is shown. The standard deviation analysis should give information about the magnetic field oscillation around its mean value. Comparing the standard deviations of measured grid points and estimated grid points it would be possible to find which interpolation-extrapolation methods generate a map close to the natural magnetic field behaviour.

Considering Table 3.2, it is possible to notice that there are four methods with less than $1\mu T$ difference, for both the maps, between the measured and the estimated standard deviations. Consequently, these four methods generate a magnetic field whose oscillation around its mean value is comparable with the real measured one. This analysis is still not capable to detect the interpolation and extrapolation errors shown in Section 3.2. Furthermore, the inpaint NaNs method, that has been previously described as a good method, is not performing well in this evaluation. The presence of the flattened values outside the border of the convex hull is influencing the computation of the standard deviation. As mentioned before, these

Standard deviation Δ [μT]				
function	interp.	extrap.	map 1	map 2
scatteredInterpolant	linear	linear	1.05	1.26
scatteredInterpolant	linear	nearest	0.98	0.76
scatteredInterpolant	natural	linear	1.02	1.24
scatteredInterpolant	natural	nearest	1.16	0.83
griddata	linear	linear	1.14	1.17
griddata	linear	nearest	1.15	1.16
griddata	natural	linear	0.99	1.18
griddata	natural	nearest	1.34	1.26
griddata	cubic	linear	21.46	32.45
griddata	cubic	nearest	1.22	2.57
griddata	v4	linear	0.60	0.64
griddata	v4	nearest	0.64	0.09
inpaint PDE	-	-	1.45	3.02
inpaint NaNs	-	-	2.15	1.25
inpaint DCT	-	-	0.94	0.41

Table 3.2: Average absolute difference between standard deviation of measured grid points and estimated grid points.

flattened areas could be avoided with a careful survey of the floorplan when capturing the magnetic field. Following these results, there is the need to design a further evaluation metric that is able to consider the errors happening in the interpolation-extrapolation step.

A new evaluation approach is applied based on computing the smoothness of the maps generated with the available methods. As earlier described, the real magnetic field is assumed to change smoothly and without discontinuities. To evaluate the smoothness, a convolution approach is used. The new evaluation approach computes the standard deviation of the difference between each grid point with its eight neighbors. The difference between diagonal neighbors is evaluated with a weight of $\frac{1}{\sqrt{2}}$, while the neighbors along the x and y directions are weighted 1. With this evaluation metric, the interpolation-extrapolation technique showing the lowest standard deviation is supposed to be the smoothest. In Table 3.3, the results for the smoothness computed in the described way are shown.

Based on the evaluation of the smoothness, it is evident that inpaint NaNs and inpaint DCT (as expected) produce a magnetic map smoother than the ones produced by the MATLAB built-in functions. Furthermore, the method using griddata natural interpolation and nearest extrapolation is quite close to the smoothness performance obtained by inpaint DCT. As seen in Figure 3.3, although the overall map looks smooth it still suffers of the nearest neighbor extrapolation error.

Finally, considering the results coming from the previous evaluations and the visual analysis of the generated maps, it is noticed that inpaint NaNs and inpaint DCT provide promising results. They do not suffer of the extrapolation errors described

Smoothness				
function	interp.	extrap.	map 1	map 2
scatteredInterpolant	linear	linear	3.20	2.11
scatteredInterpolant	linear	nearest	2.91	2.18
scatteredInterpolant	natural	linear	3.04	2.03
scatteredInterpolant	natural	nearest	2.73	2.11
griddata	linear	linear	3.23	2.24
griddata	linear	nearest	2.77	1.88
griddata	natural	linear	3.07	2.13
griddata	natural	nearest	2.57	1.78
griddata	cubic	linear	7.39	14.73
griddata	cubic	nearest	3.79	6.42
griddata	v4	linear	3.10	2.18
griddata	v4	nearest	2.74	1.84
inpaint PDE	-	-	2.97	2.16
inpaint NaNs	-	-	2.32	1.53
inpaint DCT	-	-	2.55	1.71

Table 3.3: Smoothness coefficients, lower value represents smoother map.

before and they are performing well in the smoothness evaluation. Inpaint DCT is performing good also in the magnetic field statistical behaviour analysis. Nevertheless, the results obtained from the previous key metrics analysis are not enough to conclude that the new added methods provide better performances in the MILE. As a matter of fact, some of the MATLAB built-in methods provided results close to the ones of the new added methods. All the methods will be further evaluated during the performance evaluation of Chapter 6.

Chapter 4

Magnetic measurement setup

In this chapter, a magnetic measurement setup able to measure the 3-dimensional magnetic field is proposed. In contrast to the previous Chapter 3, where the data were already available and captured prior the thesis, new data is required to pursue the main goal of the thesis: the 3-dimensional magnetic fingerprinting. The new data are captured in the office location where the thesis is developed. For this purpose, it is possible to use the surveying method (Section 1.2) described in the previous chapters or, alternatively, to design a new setup able to capture magnetic data in a more efficient way. Considering that one additional dimension has to be captured for the 3D map, it has been decided to design a new approach. In fact, for 3D magnetic measurements, the state of the art method is time consuming. It would require to carefully follow the surveying procedure for different heights.

4.1 New 3D measurement setup

To be able to design a new magnetic recording setup, it is necessary to know the requirements for a 3-dimensional magnetic map. In the 2D map generation process, the map was seen as a grid, in form of a matrix, with some measured grid points and some interpolated or extrapolated grid points. For the 3D map generation the map has to be a multi level grid, with each of the levels representing a different height of the magnetic map. This is translated to an array of matrices in the MATLAB map generation framework. As in the 2D map generation, also in 3D, there will be measured grid points and estimated (interpolated or extrapolated) grid points. The difference is that in 3D, the measured grid points belong to different levels, i.e. different heights. With this in mind, to record the data for the 3-dimensional map generation process, a recording setup able to capture the magnetic field at different heights at the same time is designed. With the state of the art surveying process of

the magnetic field, the measurements are taken one height level at the time. With the new proposed method, it is possible to measure several height levels at the same time.

To accomplish the simultaneous measurement of the magnetic field at different heights, the cart setup shown in Figure 4.1 is built.



Figure 4.1: Cart setup for 3D magnetic captures

A basic requirement for building a functional cart is to use non-magnetic materials. The usage of magnetic materials would interfere with the magnetic measurements, thus, it has to be avoided. For this purpose, the main components of the cart are made of polypropylene, a non-magnetic polymer. The only magnetic materials are the bearings and the mechanical parts used to fix the wheels to the moving base. Their presence can be neglected as investigations showed that they are producing a magnetic interference in a small area around them. The magnetic interference can be measured in a range of 5 to 10 centimeters away from them.

The objective of the cart setup is to use different sensors simultaneously at different heights. To hold the smartphones, the folder shown in Figure 4.2 is 3D-printed using PLA, a non-magnetic polymer.



Figure 4.2: 3D-printed folder to locate the smartphone

As can be seen from Figure 4.1, the 3D-printed folders can be set at different height levels. A total of 7 levels is provided, from 25cm to 175cm. The minimum distance between the available levels is set to 25cm, following the 2D spacing of the grid points seen in 2.4.

Considering the materials used to build smartphones, they can be seen as active magnetic devices. This observation leads us to think that smartphones located in different levels could interfere reciprocally during magnetic measurements. To check if it is feasible to have 2 smartphones measuring the magnetic field, with 25cm spacing, a test is conducted. One smartphone is located inside one of the folders and a magnetic capture is started. The cart is kept in the same position. In this way, the smartphone measures a constant magnetic field. After a defined time interval, another smartphone is continuously moved around the fixed smartphone, at a distance of 25cm. Finally the magnetic signals measured by the fixed smartphone are analyzed. The aim is to check if there are differences, in the magnetic capture, between the two phases of the experiment. Figure 4.3 shows the result of the test for the magnetometer x axis. The blue signal is the magnetic measurement taken when the second smartphone was not moving around. The corresponding phase of the experiment is named "interference-off". The orange signal describes the magnetic

measurement carried out with the second smartphone moving around, at 25cm distance. This phase is named "interference-on".

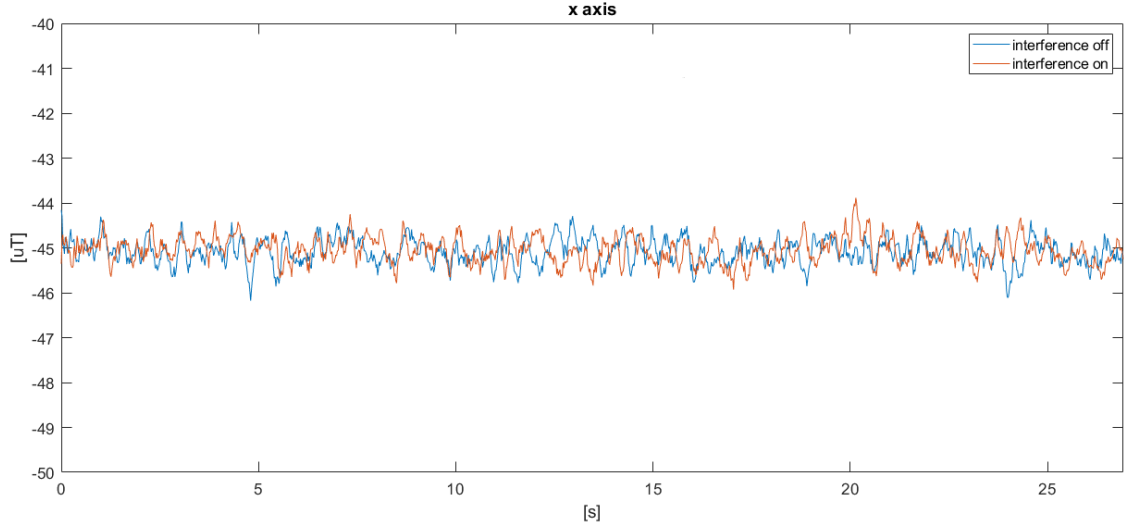


Figure 4.3: Interference test result, x axis

From a visual analysis, it is possible to observe that the interference-off and the interference-on signals are two very similar noisy signals. The interference-off mean value is $-45.054\mu T$ and its standard deviation is equal to $0.292\mu T$. The interference-on mean value is $-45.093\mu T$ and its standard deviation is equal to $0.289\mu T$. Considering the similarity of these parameters, confirmed also from the analysis of the y and z axis, it is concluded that there is not any reciprocal interference between smartphones spaced by 25cm .

4.2 V-SLAM analysis

As seen so far, a crucial step of the map generation process is the matching of the magnetic measurements with their ground truth position. In the state of the art method, this step usually takes long time and requires precision from the surveyor. The surveyor has to carefully follow some reference points with constant speed and he, or she, has to hold the smartphone in a fixed position (Section 1.2). With the cart setup, the aim is to propose an approach to speed up the procedure. For this purpose, the idea is to introduce a V-SLAM setup in the cart. The V-SLAM is a class of computational algorithms widely used for robotics, autonomous driving, augmented reality (AR) and virtual reality (VR) applications. The objective of a V-SLAM algorithm is to simultaneously perform localization and mapping in an

unknown environment [28]. Introducing a V-SLAM in the cart setup, it is possible to keep track of the position of the cart while capturing magnetic measurements. The history of all the positions gives the trajectory of the cart. The V-SLAM returns also the timestamp related to each position. This means that the surveyor is not constrained in driving the cart at a constant velocity, the V-SLAM will anyway keep track of the position at each time instant. The trajectory, associated with the time information, enables a straightforward coupling of the magnetic measurements with their ground truth position. Two V-SLAM algorithms are analyzed in the following subsections.

4.2.1 Google ARCore

ARCore is the platform developed by Google to create augmented reality applications. It provides three main capabilities: motion tracking, environmental understanding and light estimation of the environment [9]. To do this, ARCore uses the smartphone's camera to extract the main features of the environment. The features are used as references to understand how the smartphone is moving in the new scenario. The estimated movement is combined with data coming from the IMU to get the smartphone position and orientation within the environment.

In the context of this thesis, the focus is on tracking the smartphone position and orientation while it takes magnetic measurements in parallel. For this purpose, ARCore is employed on the android smartphones that are fixed in the cart setup.

The capture routine for the usage of ARCore consists in walking twice around the localization area. During the first walk, named "preparatory" walk, the smartphone runs ARCore without saving the estimated trajectory. This phase is performed to allow ARCore to capture the main features of the environment. During the second walk, the actual estimated trajectory is saved. The double step procedure is introduced because preinvestigations showed that ARCore performs better when it already knows the main features of the environment.

To evaluate ARCore performances, a first test is carried out walking in the office area while holding the smartphone in the hands, not using the cart. The aim is to find if there are some influencing factors of the final ARCore result.

Different ARCore captures are performed with different approaches. Walking tests at low velocity and high velocity are performed. Single and double directional "preparatory" walk are tested, to check possible improvements of the preliminary feature extraction from the unknown environment. Also, different trajectories (randomly intersecting or non-intersecting paths) are tested.

Analyzing all the estimated trajectories, it is not possible to choose a best performing ARCore approach. All the proposed approaches give similar results and are affected by the same errors, shown in Figure 4.4 and 4.5. In Figure 4.4, a trajectory affected by a noticeable jump discontinuity is shown. For this error, the trajectory suffers from sudden jumps that produce a discontinuous trajectory instead of a

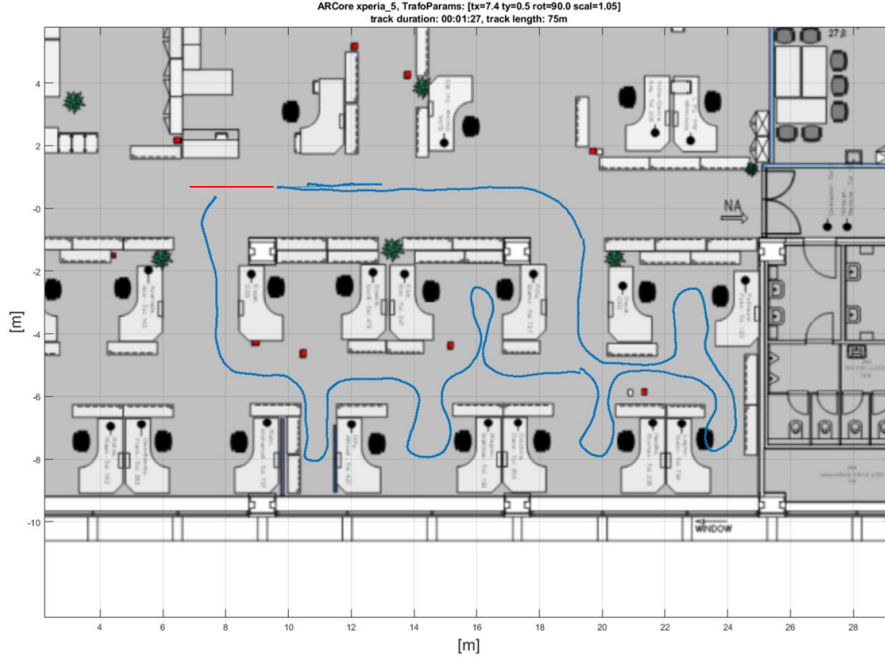


Figure 4.4: Example of ARCore result affected by jump discontinuity

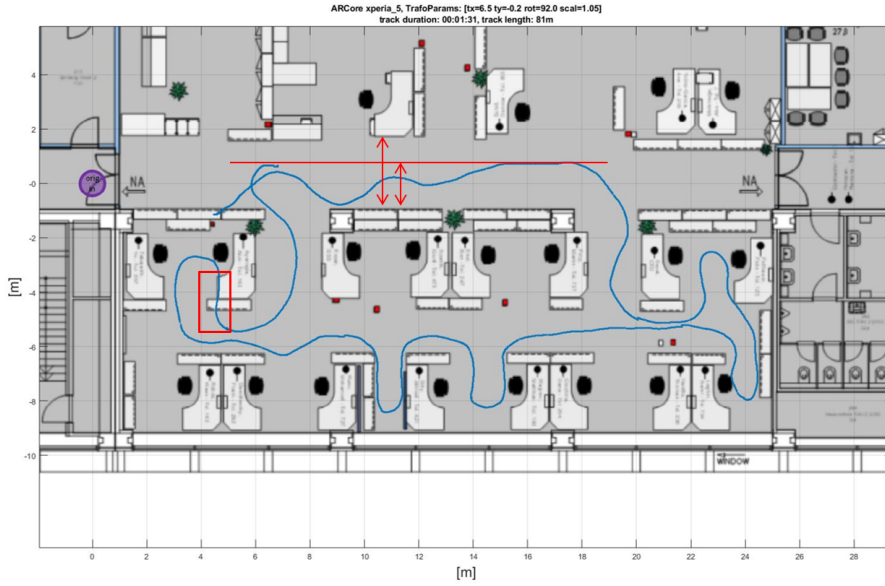


Figure 4.5: Example of ARCore result affected by length drift

continuous one. In Figure 4.5, a trajectory affected by a length drift problem is shown. The length drift problem in V-SLAM systems is a well known problem and

it represents an open issue in the research field [30]. It leads to a non-reliable estimation of the trajectory length, with extension and/or constriction of some parts of the trajectory. In Figure 4.5, this is shown as a shrinking of the walked area of the upper corridor and as an overlapping of the trajectory with furnishings of the floorplan.

New issues are detected using ARCore on smartphones fixed in the cart's folders. The cart setup does not provide any suspension to the smartphones. All the high frequency mechanical vibrations are transmitted directly from the cart's wheels to the smartphones. The vibrations are translated to a blurry image captured by the smartphone's camera. To handle this problem, the cart is moved at slow velocity, trying to reduce the impact of the vibrations on the captured images. Eventually, the slow velocity is not enough to reduce the images blurriness and the VSLAM produces worse results. Even though ARCore uses IMU data, it is not able to compensate for all the vibrations. The results show more jump discontinuities and length drifts than the ones obtained when the smartphone is held in the hand (as Figure 4.4 and 4.5).

Another issue is the fact that the results are device and height dependent. Smartphones fixed at different heights produce different trajectories. The same smartphone produces different results depending on the height it is fixed. Generally, it is noticed that if the V-SLAM is run on smartphones at higher height levels, it produces better results. This is explained by a larger view of the environment and, consequently, a more effective feature extraction for ARCore. Regarding the device dependence, usually, more powerful smartphones provide more accurate results.

In Figure 4.6 and 4.7, two trajectories obtained from two different smartphones, a Sony Xperia 5 at $1m$ height and a Sony Xperia XZ3 at $0.5m$ respectively, are shown. As expected, the higher smartphone gives a better result. Nevertheless, it still returns a lot of jump discontinuities and length drifts (visible where the trajectory overlaps the floorplan furnishings).

The choice of the trajectory for the map generation and a post-processing step, that is necessary to fix the described errors, will be introduced in Section 4.3.

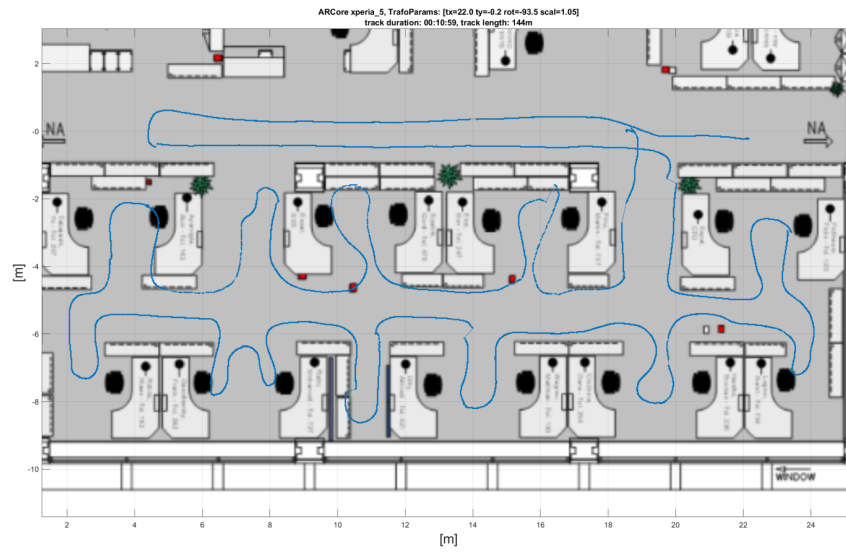


Figure 4.6: Trajectory obtained using ARCore applied to a Sony Xperia 5, fixed to the cart at 1m height

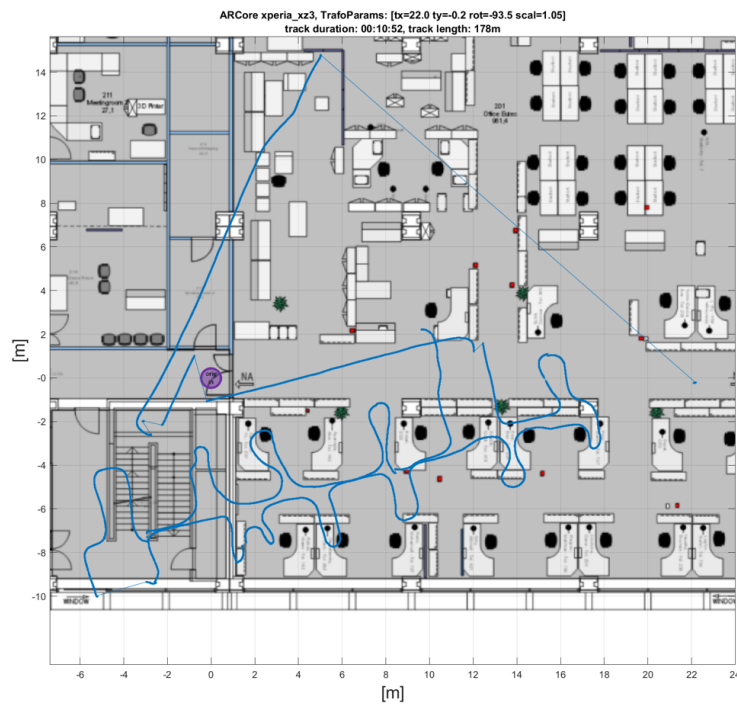


Figure 4.7: Trajectory obtained using ARCore applied to a Sony Xperia XZ3, fixed to the cart at 0.5m height

4.2.2 PSG V-SLAM

The PSG V-SLAM is a V-SLAM system developed by the "Perception Systems Group" in the Sony Technology Center. As ARCore, the PSG V-SLAM is based on a sensor fusion approach. It uses camera images and inertial measurements coming from the device's IMU. The main features extracted from the images come from the floor and from the ceiling of the environment. For this reason, it is recommended to record the images at a middle height between the floor and the ceiling. The IMU data and the images are used, as in ARCore, to get the device position, orientation and to better understand its movements within the environment.

In contrast to ARCore, the PSG V-SLAM runs offline and it is not executed on the device. The V-SLAM is performed on a server with high computational power after the data are recorded. Considering that the V-SLAM is introduced for magnetic map generation purposes and that the aim is not to design a real-time map generation process, the PSG V-SLAM offline execution is not a drawback.

To use the PSG V-SLAM, a GoPro Hero 8 is added to the measuring cart setup shown in Figure 4.1. The GoPro captures video and IMU data that is subsequently processed by the PSG V-SLAM. In Figure 4.8 and 4.9, two trajectory results are shown. Both results are obtained using PSG V-SLAM based on GoPro videos that are recorded using the cart setup.

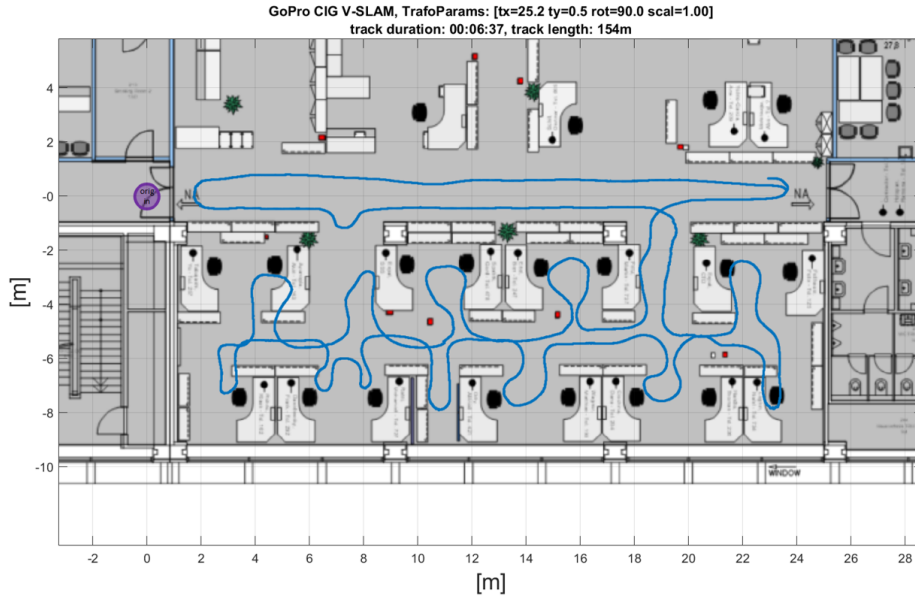


Figure 4.8: Trajectory obtained using PSG V-SLAM applied to a video recorded by a GoPro, fixed to the cart at 1.5m height

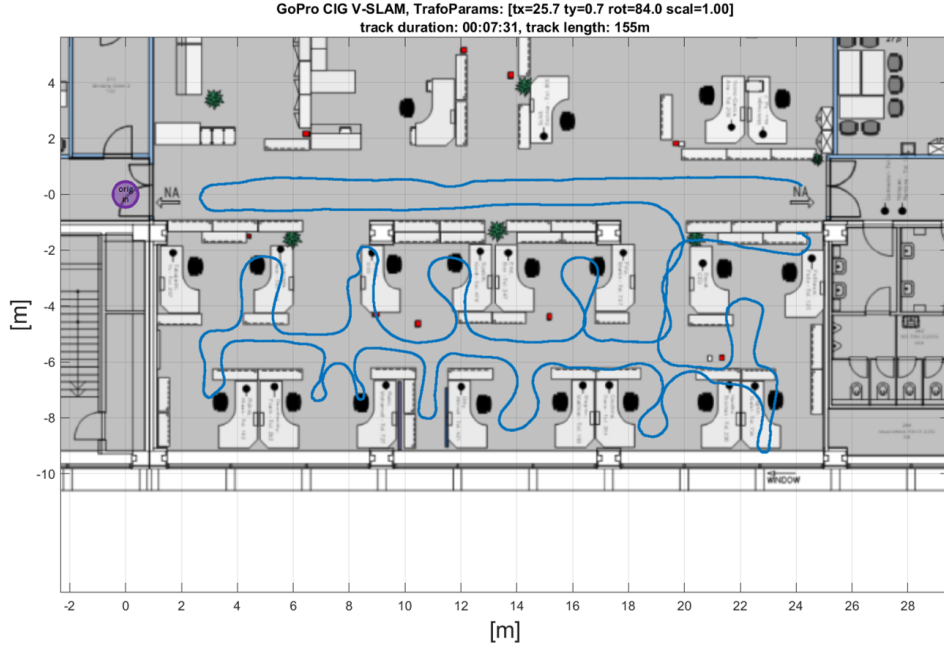


Figure 4.9: Trajectory obtained using PSG V-SLAM applied to a video recorded by a GoPro, fixed to the cart at 1.25m height

In Figure 4.8 and 4.9, the PSG V-SLAM produces continuous output trajectories. The jump discontinuity problem, described for Google ARCore in Section 4.2.1 and visible in Figure 4.4, is not present when using the PSG V-SLAM. Despite solving the jump discontinuity problem, the PSG V-SLAM output trajectories are still affected by the length drift problem. As described before, this is a non-trivial problem and its solution is not object of study in the context of this thesis. In the following Section 4.3, a post-processing routine is introduced to mitigate trajectories errors.

In contrast to ARCore V-SLAM, the PSG V-SLAM does not require a double walk within the analyzed environment. Furthermore, considering multiple tests, the PSG V-SLAM's capability to provide good results handling the steady cart vibrations is proven. This robustness of the PSG V-SLAM enables the possibility to increase the cart velocity during the data capture phase. In this regard, the same area that was covered using ARCore and considering only the second walk in 10 minutes and 59 seconds (Figure 4.6), is covered with PSG V-SLAM in 6 minutes and 37 seconds (Figure 4.8).

Summarizing the observations of the previous Section 4.2, it was found that ARCore results suffer of the jump discontinuity and of the length drift problem. Furthermore, for ARCore it is necessary to walk twice in the localization area. The

advantage of using ARCore is that the trajectories results are available right after the capture is finished due to the fact that the V-SLAM runs directly online on the smartphone. On the opposite, considering PSG V-SLAM, it was found that its results suffer of the length drift problem only. Moreover, with PSG V-SLAM, there is not the requirement to walk twice in the localization area and it is possible to move the cart with higher velocity due to its robustness. The disadvantage of using PSG V-SLAM is that the trajectory results are not available right after the capture is taken. It is necessary to perform the V-SLAM on the server. Usually, this step takes up to 2 hours for a 8 minutes track.

4.3 Trajectory correction

In the previous Section 4.2, two V-SLAM systems were introduced with the aim to keep track of the cart position within the environment during the magnetic measurement phase. In this section, a post-processing routine is presented to improve the V-SLAM trajectories results. Afterwards, the final choice on the most suitable V-SLAM approach for magnetic map generation is made.

For both the V-SLAM systems, a trajectory correction step is performed. The correction step is necessary to ensure a high quality of the magnetic map. As a matter of fact, using the trajectory as it is given by the V-SLAM systems leads to a wrong magnetic map. Due to the incorrect matching of the magnetic measures and their ground truth positions, the generated magnetic map does not describe the real magnetic field. To improve the V-SLAM trajectories, a correction step based on reference points and Helmert transformations is introduced. The Helmert transformation is a 3D transformation consisting of a roto-translation and a scaling of a set of points [29]. It is a similarity transformation. At first, some reference points are introduced in the floorplan. The reference points are some known points manually set on the floorplan. Usually, they are set by checking the recorded video from the cart during the data capture. Whenever a misalignment between the position shown in the recorded video and the raw trajectory result is found, a reference point is set in correspondence of the position estimated by the video analysis. Once N reference points are set, the raw trajectory is divided in $N - 1$ segments. For each segment, a Helmert transformation is performed forcing the trajectory to pass through the reference points. Finally, the transformed trajectory is similar to the true trajectory.

In Figure 4.10 and 4.11, two corrected trajectories are shown, returned respectively from ARCore and PSG V-SLAM.

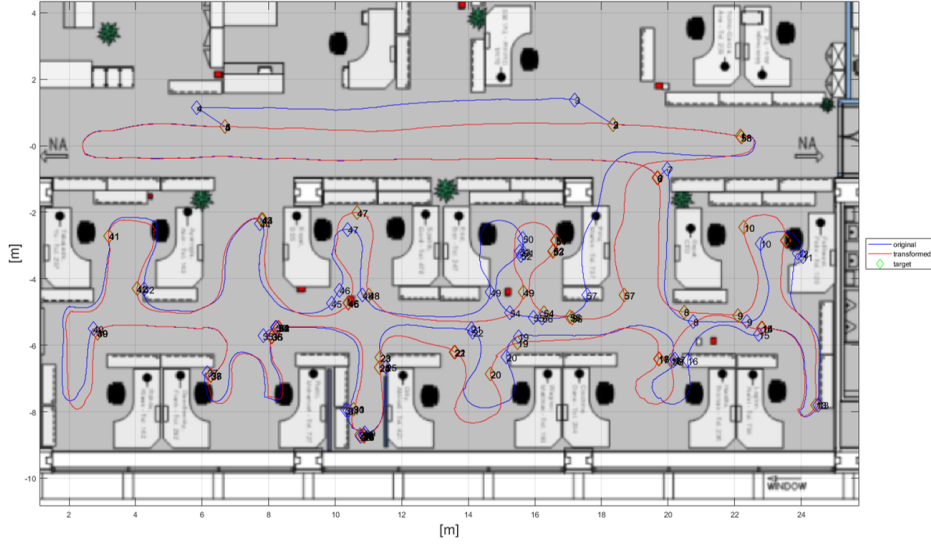


Figure 4.10: Transformed ARCore V-SLAM trajectory

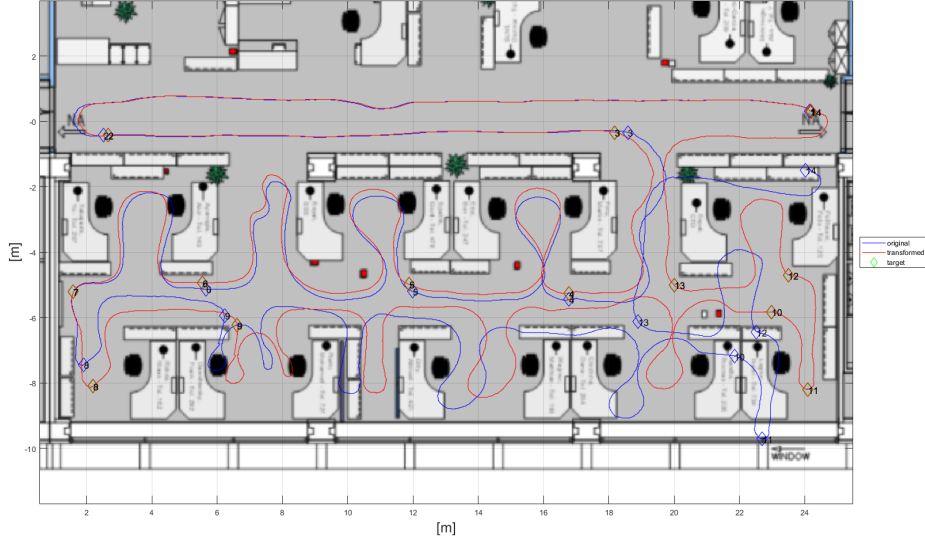


Figure 4.11: Transformed PSG V-SLAM trajectory

In this context, the reference points choice is important, as it is influencing the final trajectory. Currently, the choice of the reference points is performed manually and it depends on the accuracy of the person performing this task. To obtain a

precise reconstruction of the trajectory, this procedure is time consuming and takes up to 3 hours, as in the case of the 58 reference points set in Figure 4.10.

From Figure 4.10 and 4.11, it is evident that the PSG V-SLAM trajectory, affected by the only length drift problem, requires almost a quarter of the number of reference points used for ARCore trajectory. In Figure 4.10, referring to ARCore V-SLAM, 58 reference points are used. In Figure 4.11, referring to PSG V-SLAM, 14 reference points are used. The difference between the two approaches is mainly due to the absence of the jump discontinuities in PSG V-SLAM trajectories. In fact, the jump discontinuities require a lot of reference points more. Consequently, more time is needed to set all the reference points. Overall, considering the previous observations, the PSG V-SLAM is chosen as the system used during the magnetic measurement process, responsible to track the cart position.

Chapter 5

3-dimensional magnetic map generation and MILE extension

In this chapter, the central topic is the 3-dimensional magnetic map generation process. Once a reliable magnetic measurement setup is designed, the 3D magnetic map of the localization area can be generated. At first, the 3D map generation procedure is described. Consequently, different 3D interpolation-extrapolation approaches are introduced and evaluated to have a realistic magnetic map. Finally, the MILE is extended to be able to work in 3-dimensional space.

5.1 3D map generation

Once the V-SLAM system has been chosen, it is possible to execute the 3D map generation process. As described in the previous Section 4.3, a GoPro is added to the cart setup of to use the PSG V-SLAM, as shown in Figure 5.1. The task of the GoPro is to record the video, that will be processed with the PSG V-SLAM, while, simultaneously, some smartphones are used to capture the magnetic field of the surveyed environment.

To capture the magnetic recording on the smartphones and the video on the GoPro, the procedure is manually executed following the next consecutive steps:

1. All the needed smartphones are calibrated. The choice on which calibration method to be used will be further discussed in the next Section 5.2.
2. The magnetic capture is started for all the smartphones.
3. The GoPro video recording is started.



Figure 5.1: GoPro fixed to the cart

4. The cart is moved around the surveyed environment, accurately covering the whole area.
5. The GoPro video recording is ended.
6. The magnetic capture is ended for all the smartphones.

As described before, the GoPro is used to keep track of the position of the cart during the magnetic measurement phase. Considering the described procedure, there is a delay between the start of the magnetic capture of each of the used smartphones and the start of the GoPro video. It is noticeable also the delay between the end of the GoPro video recording and the end of the magnetic measurements. In this context, a synchronization step between the video and the magnetic measurements is needed. To perform this task, a cross correlation is applied to the accelerometer data recorded by the smartphones and from the GoPro. The cross correlation is a signal processing technique useful to align two discrete-time signals, if one is delayed with respect to the other [24]. The accelerometer data are discrete-time signals whose sampling frequency is defined by the sensor. To perform a successful cross correlation, it is necessary that the two signals, in our case the accelerometer data, have the same sampling frequency. In the considered case, all the devices and the GoPro have different sampling frequencies. Thus, before performing the cross correlation, a preliminary resampling to a common frequency is needed. All the accelerometer signals are resampled to 50Hz, the sampling frequency of the GoPro accelerometer. Subsequently, the cross correlation is performed between the accelerometer data of each of the smartphones and the GoPro. In Figure 5.2, an example of the cross correlation between one of the smartphones and the GoPro is

visualized.

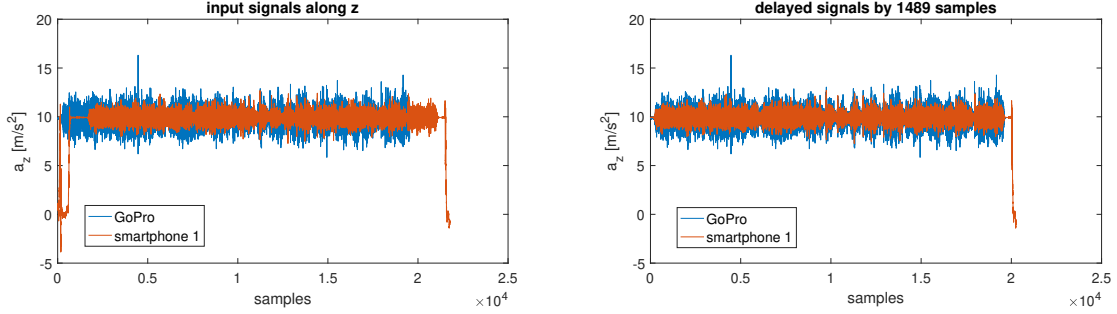


Figure 5.2: Cross correlation example and alignment

In Figure 5.2, in the left graph, two asynchronous signals are shown. In the right graph, one of the two signals, the "smartphone 1" signal is delayed by 1489 samples in order to synchronize it to the GoPro signal. The timestamp of the 1490th "smartphone 1" sample is used as first magnetic measure timestamp. All the magnetic measures captured before are not considered. Actually, the discarded magnetic measures belong to the time interval during which the operator is setting up the cart before starting the GoPro video recording. After synchronization, the last magnetic measure corresponds to the last synchronous sample between the GoPro and the smartphone. After that sample, all the magnetic measures belong to the time interval starting after the operator ends the GoPro video recording.

Once the magnetic measurements are correctly matched with the positions on the floorplan, it is necessary to get the smartphones orientation along the walked trajectory. The orientation is required to correctly transform the magnetic measurements from the sensor reference frame K^{ijk} to the earth reference frame K^{xyz} . During the magnetic measurement phase, the smartphones and the GoPro move rigidly fixed to the cart. If their pose on the cart is carefully fixed, it is possible to assume that all the devices have the same orientation. In this way, the GoPro orientation obtained from V-SLAM is associated to all the smartphones. To do so, it is necessary to notice that the V-SLAM reference frame K^{abc} is different from the sensor reference frame K^{ijk} . Consequently, a rotation is performed from K^{abc} to K^{ijk} in order to have the sensor orientation in the correct reference frame.

Finally, having the magnetic field sensed by different smartphones at different heights and knowing the smartphones orientation, the 3-dimensional magnetic map is generated. In this context, four 3-dimensional interpolation-extrapolation methods are evaluated: inpaint DCT 3D, inpaint PDE 3D, inpaint NaNs 3D and interp3. The first three methods, inpaint DCT 3D, inpaint PDE 3D and inpaint NaNs 3D are the natural extensions to 3D of the methods described in Section 3.3, inpaint DCT, inpaint PDE and inpaint NaNs. The fourth method, interp3, is a MATLAB built-in function to perform 3-dimensional interpolation and extrapolation. It provides

different interpolation-extrapolation algorithms. The `interp3` function requires, as input, an array of 2D matrices where each of the array elements is a 2D map related to a specific height. A basic requirement is that each of these 2D maps has to be fully defined, without missing values. This means that, as we measure only some trajectory grid points for each height, a preliminary 2D interpolation-extrapolation is performed for each measured height before using `interp3`.

5.2 3D map evaluation

In this section, the 3D map generation methods inpaint DCT 3D, inpaint PDE 3D, inpaint NaNs 3D and `interp3` are evaluated. Subsequently, in Section 5.2.3, an evaluation of the available calibration methods is performed.

Besides evaluating the 2D behaviour of the magnetic field at a fixed height, for a 3D magnetic map the analysis of the vertical section of the map is additionally possible. Through the vertical section analysis, some early observations about the magnetic field behaviour along the height can be formulated. In Figure 5.3, 5.4 and 5.5, the vertical section is generated using, respectively, inpaint DCT 3D, inpaint PDE 3D and inpaint NaNs 3D. In Figure 5.6 and 5.7, the same vertical sections are obtained using, respectively, `interp3` spline and `interp3` makima. Both spline and makima algorithms are based on two different modified cubic interpolations provided by MATLAB [19]. Other interpolation algorithms for `interp3` exist but, in this context, they provide unreliable results. In figure 5.8, an example of wrong result is shown, obtained using `interp3` cubic.

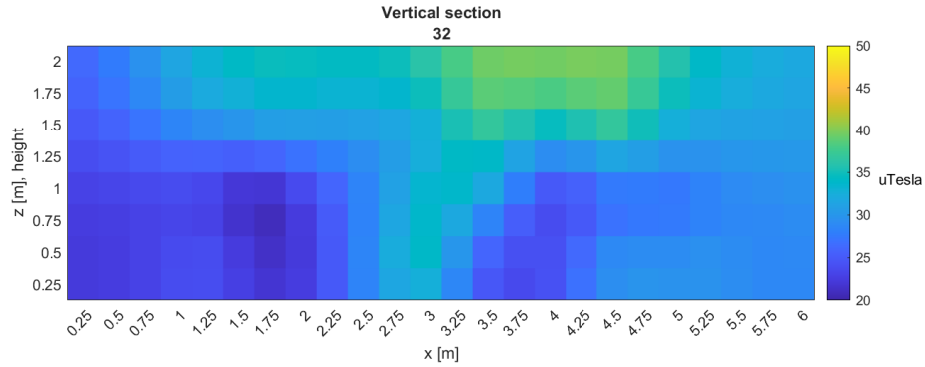


Figure 5.3: Magnetic map magnitude vertical section generated by inpaint DCT 3D

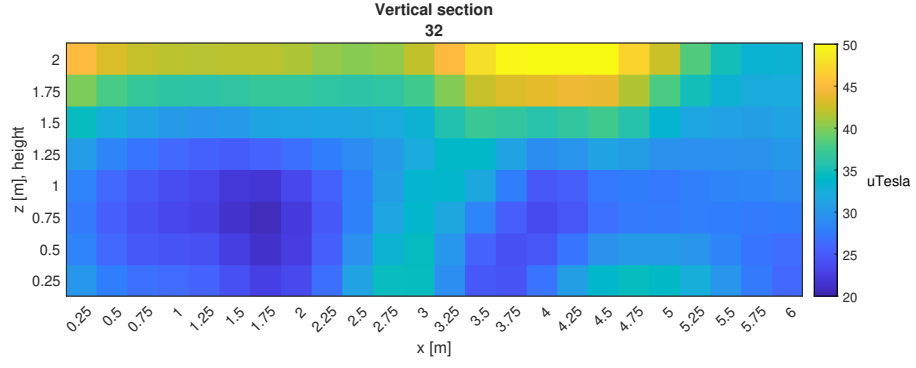


Figure 5.4: Magnetic map magnitude vertical section generated by inpaint PDE 3D

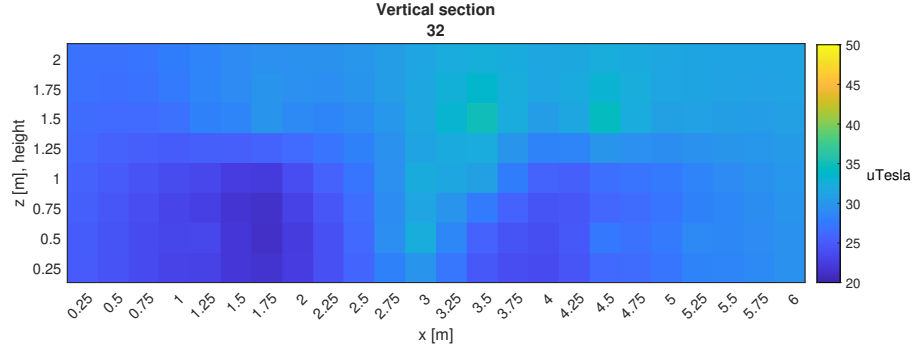


Figure 5.5: Magnetic map magnitude vertical section generated by inpaint NaNs 3D

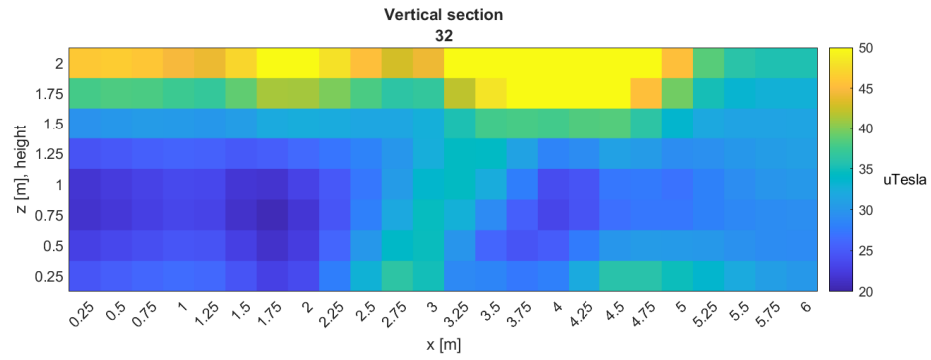


Figure 5.6: Magnetic map magnitude vertical section generated by interp3 spline

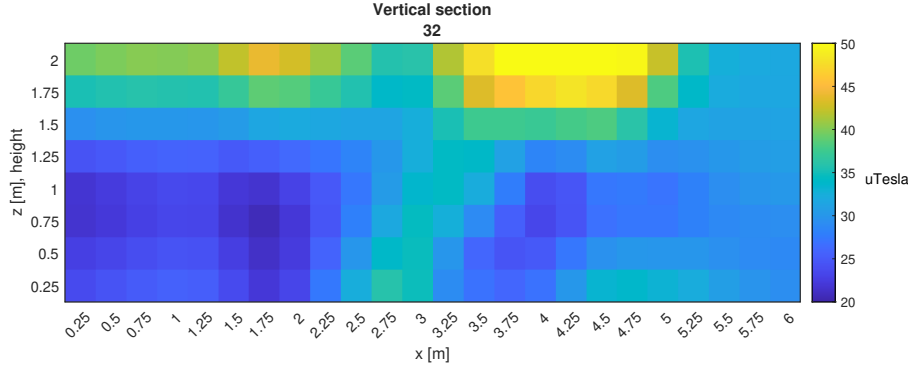


Figure 5.7: Magnetic map magnitude vertical section generated by interp3 makima

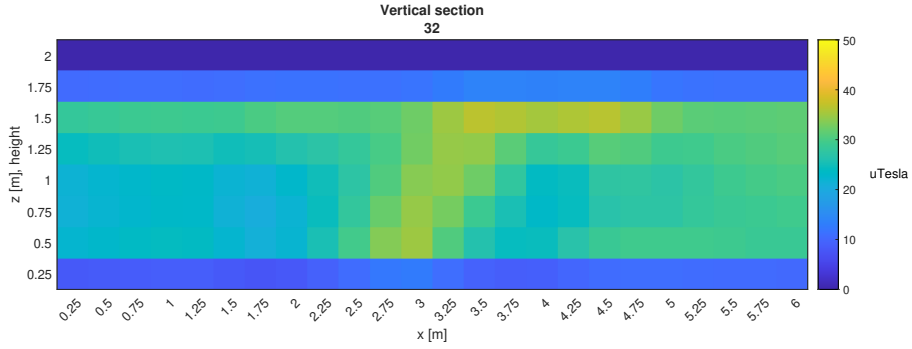


Figure 5.8: Magnetic map magnitude vertical section generated by interp3 cubic

In the previous pictures, the measured levels are the ones at height 0.5m, 1.0m and 1.5m. Thus, in the vertical sections from Figure 5.3 to 5.8, there are two interpolated levels (heights 0.75m and 1.25m) and three fully extrapolated levels (heights 0.25m, 1.75m and 2.00m). From the vertical sections, it is clear that the extrapolation is critical. For inpaint PDE 3D (Figure 5.4), interp3 spline (Figure 5.6) and interp3 makima (Figure 5.7), it is evident the creation of unexpected peaks in extrapolated levels. These peaks at heights 1.75m and 2.00m are not realistic because, considering the magnetic field behaviour at low heights, a smoother evolution at high heights is expected. The vertical sections obtained using inpaint DCT 3D and inpaint NaNs 3D provide more realistic results. With respect to Figure 5.3 and 5.5, the magnetic field behaviour seems to be smoother and more natural compared to the results obtained using inpaint PDE 3D and interp3.

To evaluate 3D magnetic maps, the most accurate way would be to directly compare them with the ground truth magnetic map. In this way, the map generation method that shows less differences with the ground truth map is the one producing the best

result. Unfortunately, the ground truth map is by itself a generated map, affected by uncertainties as the smartphones sensitivity, the accuracy of the trajectory used for map generation and from the chosen interpolation-extrapolation algorithm. In the following subsections, some evaluation metrics are introduced that are useful to evaluate the generated magnetic maps without the need of a ground truth map. For a final evaluation of the magnetic maps, their performances within the MILE will be investigated in Chapter 6.

5.2.1 Smoothness

A first approach to evaluate the 3D magnetic maps, is the evaluation of their smoothness. For this metric, the same smoothness computation approach explained in Section 3.4 is adopted. In the case of a 3D map, the computation is iterated for each of the height levels. For this purpose, a new magnetic map is generated based on measurements taken at three heights, 0.25m, 0.75m and 1.25m. The measurements are taken using the cart setup and following the 3D map generation procedure described in Section 5.1. The test bed and the generated magnetic maps can be found in Appendix A. The final magnetic map consists of six height levels, with three of them containing some measured grid points, two of them fully interpolated and one fully extrapolated. A total of five magnetic maps are generated using inpaint DCT 3D, inpaint PDE 3D, inpaint NaNs 3D, interp3 spline and interp3 makima. For both interp3 methods, the preliminary 2D interpolation-extrapolation is performed using inpaint DCT (the 2D version). The results of the smoothness analysis are shown in the next Table 5.1.

3D smoothness					
height [m]	inpaint DCT 3D	inpaint PDE 3D	inpaint NaNs 3D	interp3 spline	interp3 makima
0.25	0.76	1.62	0.74	0.81	0.81
0.50	0.57	1.17	0.39	0.66	0.64
0.75	0.62	0.99	0.61	0.64	0.64
1.00	0.48	0.86	0.32	0.55	0.54
1.25	0.65	1.08	0.65	0.69	0.69
1.50	0.63	1.46	0.35	1.40	1.12

Table 5.1: Smoothness coefficients for the 6 heights 3D magnetic map, lower value represents smoother map.

Referring to the previous table, it is noticeable that, like it was for the 2D evaluation, the inpaint NaNs method produces the smoothest map also in 3D. Furthermore, comparing the smoothness of inpaint NaNs 3D between height levels that contain some measured grid points (0.25m, 0.75m and 1.25m) and the 3 interpolated-extrapolated levels (0.50m, 1.00m and 1.50m), there is a big difference between the smoothness coefficients values. This result is confirmed from the visual

analysis of the inpaint NaNs 3D map, where a flattened behaviour of the magnetic field is noticed for the interpolated-extrapolated levels. In other words, also in 3D, inpaint NaNs suffers of the previously described ghost behaviour (Section 3.3).

For what concerns inpaint DCT 3D and inpaint PDE 3D, they show opposite smoothness behaviours. The inpaint DCT 3D shows a constantly low value in Table 5.1, confirming a smooth map for all the levels. On the other side, inpaint PDE 3D presents high values of the smoothness coefficients, describing a map with different spikes in all the levels.

Regarding the 3D magnetic maps obtained using `interp3`, it is observed that the extrapolated levels show a clearly worse performance in terms of smoothness. The last observation is justified by the previous considerations about the vertical sections shown in Figure 5.6 and 5.7.

However, the only smoothness metric is not enough to accurately evaluate the 3D map generation methods; further evaluation metrics are required.

5.2.2 Statistical behaviour of the 3D magnetic field

As in the 2D methods evaluation (Section 3.4), another approach for the evaluation of the 3D magnetic map generation methods is based on the magnetic field statistical behaviour. The method consists in comparing the mean values and the standard deviations of measured grid points and estimated (through interpolation-extrapolation) grid points. From the comparison, an evaluation on which of the map generation methods generate a realistic magnetic field is obtained. For the purpose of this evaluation, the same 3D magnetic maps of the smoothness analysis (Section 5.2.1) are used. In the 3-dimensional case, the comparison between measured grid points and interpolated-extrapolated grid points is made for all the three measured levels (0.25m, 0.75m and 1.25m). In the following Tables 5.2 and 5.3, the average results of the three levels comparisons are shown.

Method	Mean values Δ [μT]
inpaint DCT 3D	0.11
inpaint PDE 3D	2.62
inpaint NaNs 3D	0.05
<code>interp3</code> spline	0.19
<code>interp3</code> makima	0.19

Table 5.2: Average absolute difference between mean values of measured grid points and estimated grid points of the 3D magnetic map.

Method	Standard deviation Δ [μT]
inpaint DCT 3D	0.37
inpaint PDE 3D	5.73
inpaint NaNs 3D	1.31
interp3 spline	0.43
interp3 makima	0.42

Table 5.3: Average absolute difference between standard deviations of measured grid points and estimated grid points of the 3D magnetic map.

Referring to Table 5.2, it is not possible to reach conclusive results. Apart from inpaint PDE 3D, the average difference between the mean values of measured grid points and estimated grid points is very similar for the other four proposed 3D map generation methods.

A slightly different result is found in Table 5.3. In this case, inpaint PDE 3D and inpaint NaNs 3D show a clear high standard deviation difference. Inpaint DCT 3D and the two interp3 methods still provide similar results. Table 5.3 shows the average absolute difference between standard deviations of measured grid points and estimated grid points. As described in Section 3.4, the standard deviation of the magnetic field can be seen as a number that describes its oscillation around its mean value. The difference of the standard deviations shows how different the oscillation of the measured grid points' magnetic field is compared to the one of estimated grid points. Low values of this metric indicate map generation methods that generate a realistic magnetic field behaviours. Consequently, according to the described approach, inpaint PDE 3D is the worse of the considered methods. Moreover, it follows that inpaint DCT 3D and the two interp3 methods generate the most realistic magnetic field behaviour between the analyzed methods.

Nevertheless, comparing the magnetic maps generated using inpaint PDE 3D and inpaint DCT 3D, it is found that the biggest difference between the two maps belongs to grid points outside the convex hull or grid points coincident with no-go areas. In Figure 5.9, the green areas of the map are those areas where the difference between inpaint DCT 3D and inpaint PDE 3D is less than the mean difference between the two maps. On the opposite, the red areas of the map are those areas where the difference between inpaint DCT 3D and inpaint PDE 3D is higher than the mean difference.

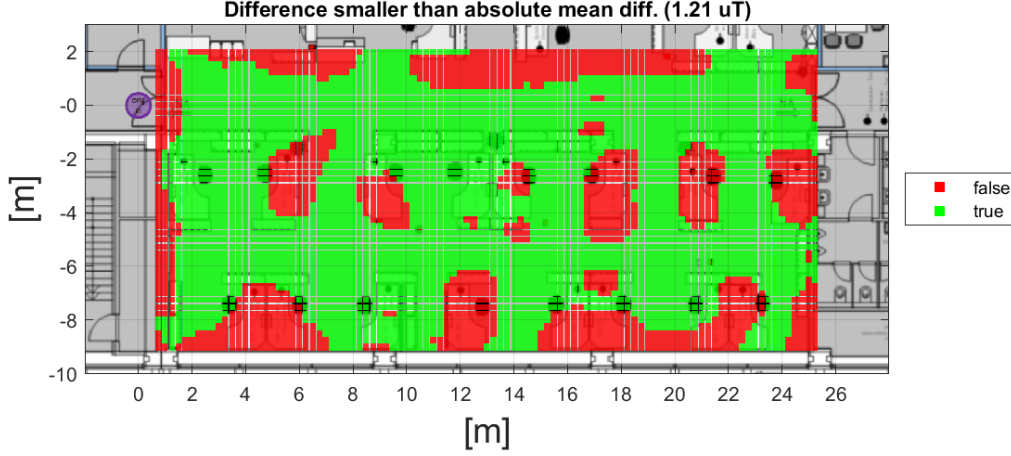


Figure 5.9: Comparison between inpaint DCT 3D and inpaint PDE 3D

Referring to the previous comparison and considering that most of the differences belong to not commonly walked areas, it is not possible to conclude which is the preferable map generation method.

The evaluations will be further discussed in Chapter 6, in the context of the performance evaluation inside the MILE.

5.2.3 Calibration methods evaluation

In Section 2.2.1, the two calibration approaches available in the MILE were described. They are the ellipsoid method and the android method. To evaluate which of the two calibration methods is the most reliable one, a test is performed as schematized in Figure 5.10. A magnetic capture is performed, in the same test bed area used in the previous Section 5.1 and 5.3. For the purpose of the test, five levels are measured from 0.25m to 1.25m using the cart setup with five different smartphones. Subsequently, the five levels are split in two groups. The first group, that consists of level 1 (0.25m), level 3 (0.75m) and level 5 (1.25m), is used to generate Map 1. Thus, Map 1 consists of three measured levels and 2 interpolated-extrapolated levels. The second group, that consists of level 2 (0.50m) and level 4 (1.00m), is used to generate Map 2. Thus, Map 2 consists of two measured levels and three interpolated-extrapolated levels. The obtained 3D magnetic maps are compared level by level. In each comparison, the measured grid points of one map are compared with the correspondent grid points of the other map.

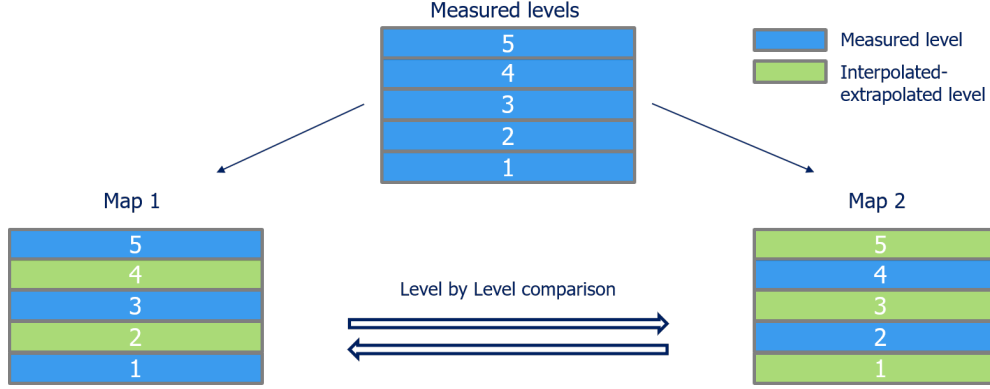


Figure 5.10: Test scheme used for calibration methods evaluation

The expected outcome of the comparisons is to have higher differences for those levels where the calibration did not perform well. As now the focus is on the calibration methods evaluation, for the described test both the maps, Map 1 and Map 2, can be generated using independently one of the previously described 3D map generation methods. The following Tables 5.4 and 5.5 show results referring to maps generated using inpaint DCT 3D.

Ellipsoid calibration			
Level	Magnitude Δ [μT]	Inclination Δ [deg]	Azimuth Δ [deg]
1	1.85	3.11	9.12
2	1.44	1.64	5.30
3	1.84	0.94	2.63
4	1.60	0.84	2.35
5	1.23	2.53	6.21

Table 5.4: Evaluation test results for ellipsoid calibration

Android calibration			
Level	Magnitude Δ [μT]	Inclination Δ [deg]	Azimuth Δ [deg]
1	2.84	5.66	18.83
2	3.58	5.96	16.44
3	5.20	4.85	11.11
4	2.15	1.90	4.03
5	0.91	1.57	2.88

Table 5.5: Evaluation test results for android calibration

From the tables, it is possible to conclude that the ellipsoid calibration shows more stable results compared to the ones of android calibration. Referring to Table 5.5, the first three levels have a high difference comparing the measured and the estimated grid points. This outcome can be explained assuming that at least one of the three levels has magnetic measurements affected by a poor calibration. In fact, if level 2 is the level that suffers from a poor calibration, also the levels adjacent to it are influenced by level 2 values. To prove this assumption, a new map is generated following the same procedure that is used for obtaining the results of Table 5.4 and 5.5. This time an ellipsoid calibration is used, where the calibration parameters for level 2 are manually changed to emulate a wrong calibration for level 2. The result of this test is shown in Table 5.6.

Faked ellipsoid calibration			
Level	Magnitude Δ [μT]	Inclination Δ [deg]	Azimuth Δ [deg]
1	7.30	24.09	57.47
2	6.74	21.30	60.02
3	3.40	10.22	29.59
4	1.60	0.84	2.35
5	3.13	2.89	8.70

Table 5.6: Evaluation test results for the faked ellipsoid calibration

In Table 5.6, it is evident that if one level is affected by a poor calibration, it influences also the adjacent levels. Consequently, it is proven that in Table 5.5 at least one of the first three levels is affected by a bad calibration. Eventually, it is possible to conclude that the android calibration method results are smartphone dependent, thus, it is preferable to use ellipsoid calibration.

In the context of the test of Figure 5.10, it has been tried to use the level by level comparison to benchmark the available 3D map generation methods. Theoretically, comparing the measured grid points of one of the two maps with the correspondent estimated grid points of the other map, it could be possible to evaluate which of the map generation methods provide a result most similar to the real measured magnetic field. In practice, in the considered test, the estimated grid points are always found in between the two measured grid points of the two adjacent measured levels. Thus, as a result of the comparison, it is not possible to find differences between the different 3D interpolation-extrapolation methods. In fact, the main differences between the 3D maps are located on grid points far from the measured grid points, where the ground truth data are not available.

5.3 MILE 3D extension

In this section, the MILE framework is extended to work in 3-dimensional space. As seen so far, a new 3D magnetic map generation process has been introduced in the previous sections. To be able to use a 3D magnetic map inside the MILE, an extension of the engine capabilities is needed. As seen in Section 2.5, the state of the art MILE is based on a 4-dimensional particle filter. It defines the user state as a 4-dimensional vector $\mathbf{s} = [x, y, \theta, l]$. Where x and y are the 2D coordinates of the user in the localization area, θ is the motion heading and l is the step length. In order to use a 3D magnetic map instead of the state of the art 2D map, the height z is added to the user state vector. The extended user state is a 5-dimensional vector $\mathbf{s} = [x, y, z, \theta, l]$.

Considering the previous vector state extension, the core of the MILE becomes a 5-dimensional particle filter. The particle filter workflow remains the same described in Section 2.5 and in Figure 2.4. Nevertheless, some modifications are necessary to handle the additional state variable z .

In the particles initialization step, z is initialized according to a uniform distribution between 0m and 2m. During the performance evaluation phase (Chapter 6), the height tracking capability of the engine will be evaluated. As a second option, the initialization of z according to a gaussian distribution that is centered at 1.25m is implemented.

Concerning the updating step, a fifth equation is added to the 2D motion model (Equations 2.2). The new motion model is given by:

$$\theta_k^t = \theta_k^{t-1} + \Delta\theta^t + n_\theta \quad (5.1a)$$

$$l_k^t = l^t + n_l \quad (5.1b)$$

$$x_k^t = x_k^{t-1} + l_k^t \cdot \cos(\theta_k^t) \quad (5.1c)$$

$$y_k^t = y_k^{t-1} + l_k^t \cdot \sin(\theta_k^t) \quad (5.1d)$$

$$z_k^t = z_k^{t-1} + \Delta z^t + n_z \quad (5.1e)$$

where $\Delta\theta^t$, l^t and Δz^t are, respectively, the heading change, the movement length (in the earth plane xy) and the estimated height variation derived from the PDR output. n_θ , n_l and n_z are three additive gaussian noises used to handle, respectively, the heading change, the movement length and the estimated height variation uncertainties.

Regarding the weighting step, the observation model $P(z^t | s^t)$ is the same as in the 2D case, described by equation 2.3. In this step, the only modification is related with the observation function $obs(s^t)$ used in the magnetic likelihood mode P^{mag} . The observation function returns the magnetic map values at the particles positions for the considered state s^t , at time t . In the 3D MILE, $obs(s^t)$ performs a 3-dimensional linear interpolation based on the 3D magnetic map grid points. For the resampling step and the position estimation step is not required any change.

They execute the same instructions both for the 2D and the 3D MILE, with the difference that the 3D MILE estimated position is returned in 3-dimensional space. Due to the described modifications, the MILE is provided of the 3D capability. If a 2D database magnetic map is chosen, the MILE executes the 2D version described in Section 2.5. If a 3D database map is chosen, the MILE executes the 3D version described in the current section.

Chapter 6

Performance evaluation

In this chapter, the performances of the MILE using 3D and 2D maps are investigated and compared. The objective is to analyse the positioning accuracy, using 3D magnetic maps instead of 2D magnetic maps, and to find out the height tracking capabilities of the 3D engine.

6.1 Test bed and dataset collection

For the performance evaluation, a new office area is chosen as test bed. The area used for the 3D map evaluation (Section 5.2), shown in Appendix A, is not suitable for the MILE performances analysis. For a reliable performance evaluation, a bigger area is needed, where multiple paths can be walked. The office area shown in Figure 6.1, defined by the red rectangle, is chosen. The selected area is a 452m² office area and, as visible from the picture, it gives the possibility to walk along different loops.

To generate the magnetic map using the procedure described in Section 5.1, four smartphones are fixed at different heights. A Nokia 9 at 0.25m, a Sony Xperia XZ3 at 0.75m, a Sony Xperia 5 at 1.25m and a Google Pixel 4 at 1.75m. Based on the magnetic measurements at the four levels, the 3D magnetic map is generated with eight levels, from 0.25m to 2.00m heights, with 25cm spacing between the levels. To generate the magnetic map, the ellipsoid calibration of the magnetic sensor is used, according to the results of Section 5.2.3. As one of the objectives is to compare the 3D vs 2D performances, the 2D magnetic map is generated using only the measurements from the smartphone at the height 1.25m. The choice of the height for the 2D magnetic map is based on the average height a smartphone is typically held by an adult user in portrait mode. The raw trajectory, the corrected trajectory and the magnetic maps are shown in Appendix B. For what concerns the 3D magnetic map (for this example generated using inpaint DCT 3D), it is noticeable that the average vertical difference of the magnitude, between 0.25m and 2.00m, is $3.46\mu T$.

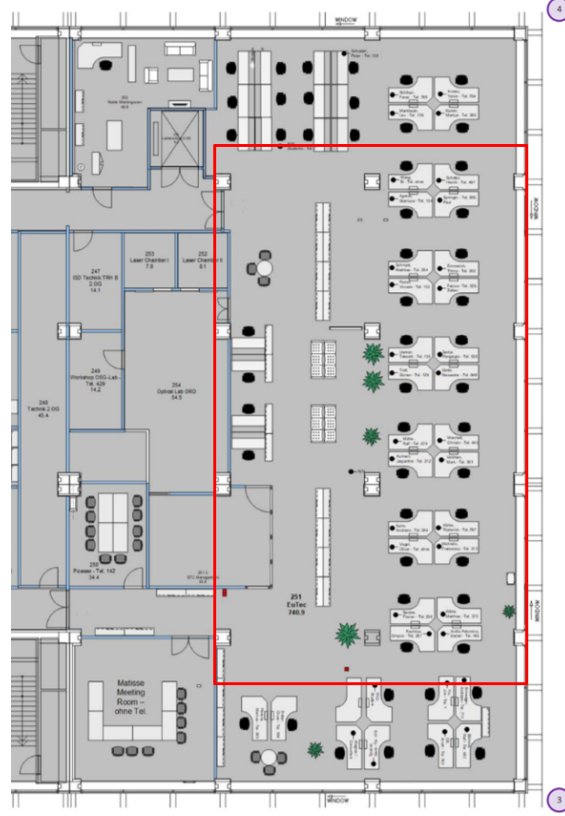


Figure 6.1: Test bed for performance evaluation

The highest vertical difference is $29.85\mu T$ and the lowest is $0\mu T$. If we limit the investigation to the typical utilization range of a smartphone, i.e. analyzing the vertical differences between 1.00m and 2.00m, the following results are obtained: the average vertical difference becomes $2.51\mu T$, the highest vertical difference is $18.55\mu T$ and the lowest is $0\mu T$. From the previous observation, possible advantages are expected when using the 3D magnetic map instead of the 2D map. In general, the average vertical difference seems to be relatively low ($2.51\mu T$) compared to the full range of vertical differences ($0\mu T$ - $18.55\mu T$). Thus, in different areas of the localization area, the 3D magnetic map will not increase the amount of magnetic information given by the 2D map.

Once the magnetic maps are generated, a first dataset of human walks is captured for the performance evaluation. For this purpose, the following five Android smartphones are used:

- Sony Xperia 5
- Sony Xperia XZ3

- Google Pixel 4
- Nokia 9
- Samsung Galaxy s10e

The dataset, named Dataset 1, consists of measurements captured at four different heights: 1.00m, 1.25m, 1.50m and 1.75m. Each of the smartphones is used to capture six walks of 2 minutes, for each of the measured heights. In this way, for each of the measured heights, a total of 1 hour of walking is recorded. Overall, the complete dataset consists of 4 hours of human walks.

A second dataset, named Dataset 2, is recorded using the five smartphones at variable heights during the walk. Each of the smartphones is used to capture two walks of 3 minutes. During the first minute of walk, the smartphones are held at 1.00m height. During the second minute they are held at 1.25m height and, finally, during the third minute they are held at 1.50m height. Overall, the second dataset consists of 30 minutes of human walks at variable heights that will be useful to evaluate the height tracking capability of the MILE. For both the datasets, the ellipsoid calibration of the magnetic sensor is used according to the results described in Section 5.2.3.

To keep the smartphone height at a constant level during the measurements, a reference signal in form of a tape on the surveyor's body is fixed at a reference height. Thus, the surveyor walks around the selected area trying to keep the smartphone in correspondence of the reference height. In this regard, the smartphone oscillations during the walk are used (and necessary) to perform the PDR estimation (Section 2.3). The human walk track capture procedure consists of two steps. In the first step the smartphone's sensor is calibrated according to the previously described procedure (Section 2.2.1). In the second step, the surveyor walks around the evaluation area keeping track of his ground truth positions within the floorplan. To keep track of the ground truth 2D positions, an app is used that was developed for the purpose of the indoor localization project. Using the app, the surveyor sets some reference points by tapping on the floorplan during his walk, in real time. Usually, some characteristic points are set as reference points, like the end of a corridor, a shelf or a specific table. The ground truth position between two successive reference points is computed as a linear interpolation over the time between the two reference points. In Figure 6.2, the ground truth trajectory in green and the MILE estimated positions in red are shown. Using the described procedure, the ground truth trajectory depends on the surveyor's accuracy during the reference points setting. For this reason, to perform a reliable performance evaluation, the whole dataset is captured by the same surveyor. In this way, the obtained results are affected by the same surveyor's accuracy.

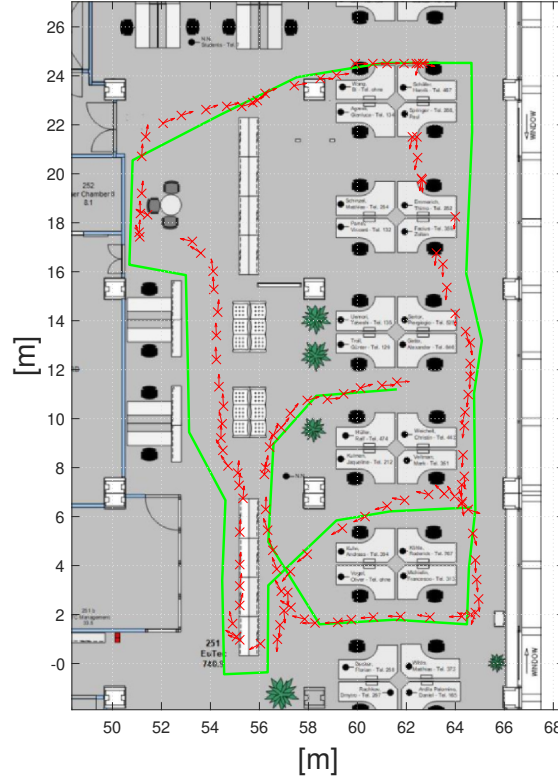


Figure 6.2: Ground truth walk (in green) and MILE estimated localization (in red)

6.2 Localization accuracy

The localization accuracy performances are used to benchmark the magnetic maps performances within the MILE. All the previously described 2D and 3D magnetic maps are tested, one at the time, as database magnetic map of the MILE. In this context, the walk recordings introduced in the previous Section 6.1 are used to simulate a user of the indoor localization system. After each of the walk recordings is processed by the MILE, a cumulative distributive function (CDF) of the estimation error is used to evaluate the MILE performance. The estimation error is computed as the distance between the estimated position and the ground truth position, obtained as described in the previous Section 6.1. In Figure 6.3, the CDF for a walk simulation example is shown. Referring to Figure 6.3, the 50% of the CDF can be used as metric to evaluate the accuracy of the MILE. The 80% and 90% of the CDF can be used as metrics to evaluate the robustness of the MILE.

As described in Section 2.5, one of the core steps of the particle filter is the weighting step. In this step a gaussian pseudo-distribution model is employed as

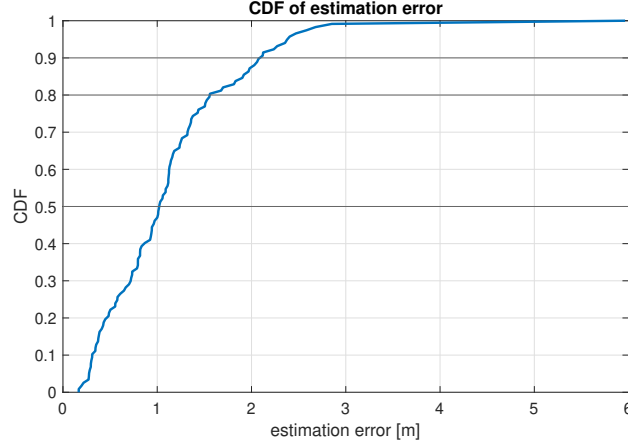


Figure 6.3: CDF of the estimation error for a sample walk

observation model, to weight all the particles according to their similarity with some measured observations. As described in Section 2.5, the weighting step (or likelihood update step) is divided into three sub-likelihood observation modes: P^{mag} , P^{pdr} and P^{flr} . It was described that only the magnetic mode P^{mag} and the PDR mode P^{pdr} use the gaussian pseudo-distribution model. The models for the two modes, P^{mag} and P^{pdr} , require to set a covariance matrix, respectively C^{mag} and C^{pdr} . As in this thesis no modifications are made to the PDR, for the covariance matrix the focus will be on C^{mag} , covariance matrix of the magnetic mode.

As seen in 2.5, the covariance matrix C^{mag} is a 3×3 diagonal matrix equal to $C^{\text{mag}} = \text{diag}\{C^{\text{m}}, C^{\text{i}}, C^{\text{a}}\}$. C^{m} , C^{i} and C^{a} are respectively the magnitude, the inclination and the azimuth covariances. High values for these parameters mean that in the weighting step a low weight is assigned to the correspondent magnetic field dimension. On the opposite, low values for the covariances mean that in the weighting step a high weight is assigned to the correspondent magnetic field dimension. In general, the covariances are set to have a magnetic measurements weight that grows in parallel with the database magnetic map accuracy. In fact, the difference between the measured magnetic field and the correspondent magnetic field obtained from database map at the same position will be lower for accurate magnetic maps. Thus, the covariance can be set low. Early investigations on the MILE found that the azimuth covariance C^{a} has to be set to a high value also for accurate magnetic maps. The azimuth obtained from the Android internal algorithm, described in Section 2.2.2, is not accurate since there is not any absolute reference for its computation. On the opposite, the inclination computation is more accurate because it can be referenced to the earth's gravity acceleration. Following the previous observations, the azimuth covariance is set to $C^{\text{a}} = 270\text{deg}^2$, the same value used in the state of the art MILE. In this way, a low weight is assigned to the magnetic azimuth

measurement during the weighting step.

To tune the magnitude covariance C^m and the inclination covariance C^i , a parameter optimization is performed. Dataset 1, described in the previous Section 6.1, is split into two chunks of 2 hours each: 50% of walk captures of each height is assigned to the train Dataset 1, the other 50% of walk captures is assigned to the test Dataset 1. The train set is used to perform an iterative process with the aim to choose the covariances C^m and C^i that optimize the engine performances for a given magnetic map. Subsequently, once the covariances C^m and C^i are correctly set for each of the evaluated magnetic maps, the test set is used to perform the final localization accuracy evaluation.

In the training procedure, the MILE processes the train dataset walks and the outcomes are evaluated. The MILE observation model covariances C^m and C^i are iteratively changed to search for the covariances that provide the best performance. At first, high covariances values are set, equal to $C^m = 100\mu T^2$ and $C^i = 400\text{deg}^2$. Successively, these values are decreased, with variable step size, till when there is not any performance improvement. In this regard, the performance is evaluated as a balanced trade-off between localization accuracy (50% of the CDF) and localization robustness (80% of the CDF). The improvement is evaluated looking at the average value between accuracy and robustness for each step. In Appendix C, the results of the training process are shown for all the maps. In general, the best performances for the 2D magnetic maps are obtained with higher covariance values, $C^m = 20\mu T^2$ and $C^i = 30\text{deg}^2$, than for the 3D maps, $C^m = 10\mu T^2$ and $C^i = 15\text{deg}^2$. This means that, in the 2D MILE a lower weight is assigned to the magnetic measurements. In fact, as the 2D magnetic map is measured at 1.25m, it cannot describe accurately the magnetic field at different heights. As a result, if the user holds the smartphone at other heights, the difference between the measured magnetic field and the magnetic map value at the same position is larger than the 3D case. The covariances obtained as results of the training are shown in the next Table 6.1.

Map	C^m [μT^2]	C^i [deg^2]
2D	20	30
3D	10	15

Table 6.1: Magnitude and inclination covariances for 2D and 3D maps

Once the model’s covariances are correctly set, the test dataset is used to evaluate the magnetic maps performances. The test dataset consists of 2 hours of captures: 30 minutes at 1.00m, 30 minutes at 1.25m, 30 minutes at 1.50m and 30 minutes at 1.75m. Different simulations are performed. At first, the whole test dataset is fed to the MILE to evaluate the overall performance of the engine considering different

heights simultaneously. In Tables 6.2 and 6.3, the results of the first simulations obtained using, respectively, 3D magnetic maps and 2D magnetic maps are shown. Note that for interp3 maps, the preliminary 2D interpolation-extrapolation is performed using inpaint DCT 2D. Comparing the results, all the 3D maps outperform the 2D maps. Considering the best 3D performance, obtained using interp3 spline, against the best 2D performance, obtained using griddata v4 nearest, it is possible to conclude that with the 3D map there is an improvement of 0.5m of localization accuracy and 0.4 – 0.6m of robustness.

3D performance - test dataset				
function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	1.9	4.6	6.9
inpaint PDE 3D	-	1.9	4.7	6.8
inpaint NaNs 3D	-	2.2	4.7	6.8
interp3	makima	1.9	4.6	6.9
interp3	spline	1.8	4.6	7.0

Table 6.2: 3D performance of the test dataset.

2D performance - test dataset					
function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	linear	linear	2.6	5.6	8.0
scatteredInterpolant	linear	nearest	2.3	5.1	7.7
scatteredInterpolant	natural	linear	2.7	5.6	7.9
scatteredInterpolant	natural	nearest	2.4	5.2	7.7
griddata	linear	linear	2.7	5.8	8.0
griddata	linear	nearest	2.4	5.4	7.9
griddata	natural	linear	2.6	5.6	7.9
griddata	natural	nearest	2.4	5.3	7.7
griddata	cubic	linear	2.6	5.6	8.1
griddata	cubic	nearest	2.3	5.3	7.9
griddata	v4	linear	2.5	5.4	7.6
griddata	v4	nearest	2.3	5.0	7.6
inpaint PDE	-	-	2.4	5.3	7.7
inpaint NaNs	-	-	2.4	5.2	7.6
inpaint DCT	-	-	2.3	5.1	7.7

Table 6.3: 2D performance of the test dataset.

Another result of the previous Tables 6.2 and 6.3 is that, both for 3D and 2D, there is not any clear preferable map generation method. For the 2D case of Table 6.3, it is possible to observe that, considering the same function and the same interpolation method, the nearest extrapolated maps usually perform better than the correspondent linearly extrapolated maps. This observation roughly confirms the result found during the 2D smoothness analysis shown in Table 3.3. Apart from the linearly extrapolated maps, all the other 2D map generation methods performances are very similar between them. Different maps show a localization accuracy

performance of 2.3 – 2.4m and robustness around 7.6 – 7.7m. It is not possible to choose one single method as reference for the 2D magnetic maps. Also in Table 6.2, that refers to the 3D case, it is not possible to choose a single 3D map generation method. All the 3D performances show similar results. The average localization accuracy is 1.8 – 1.9m and the robustness 6.8 – 6.9m. In Section 5.2.2 it was found that the magnetic field statistical behaviour of inpaint PDE 3D was quite different compared to the other methods. Moreover, it was also found that the main differences belong to not commonly walked areas or to no-go areas. Considering this, the results of Table 6.2 can be explained by the fact that along commonly walked paths the magnetic differences between different maps are lower.

To better evaluate the magnetic maps performances, the second set of simulations is performed one height at the time. The test Dataset 1 is divided and it is processed level by level inside the MILE. In the following tables the main results for each of the heights is shown. Considering the results similarity, only the best performing maps are shown.

3D performance - height 1.00m				
function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	1.3	2.7	3.7
inpaint PDE 3D	-	1.4	2.7	3.7
inpaint NaNs 3D	-	1.4	3.0	4.1
interp3	makima	1.3	2.6	3.8
interp3	spline	1.3	2.6	3.6

Table 6.4: 3D performance of the test tracks captured at 1.00m height.

2D performance - height 1.00m					
function	interp.	extrap.	50% [m]	80% [m]	90% [m]
griddata	v4	linear	1.6	3.0	3.7
griddata	v4	nearest	1.6	3.0	3.8
inpaint PDE	-	-	1.6	3.0	3.8
inpaint NaNs	-	-	1.7	3.1	3.9
inpaint DCT	-	-	1.5	2.9	3.8

Table 6.5: 2D performance of the tracks captured at 1.00m height.

3D performance - height 1.25m				
function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	1.9	4.0	5.6
inpaint PDE 3D	-	2.0	4.2	5.7
inpaint NaNs 3D	-	2.2	4.7	6.8
interp3	makima	2.0	4.3	6.0
interp3	spline	1.8	4.0	5.6

Table 6.6: 3D performance of the test tracks captured at 1.25m height.

2D performance - height 1.25m					
function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	natural	nearest	2.4	4.6	6.0
griddata	cubic	nearest	2.3	4.7	6.3
griddata	v4	nearest	2.3	4.8	6.2
inpaint PDE	-	-	2.3	4.6	6.1
inpaint NaNs	-	-	2.3	4.8	6.2

Table 6.7: 2D performance of the test tracks captured at 1.25m height.

3D performance - height 1.50m				
function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	1.2	2.4	3.3
inpaint PDE 3D	-	1.2	2.4	3.2
inpaint NaNs 3D	-	1.6	3.3	4.1
interp3	makima	1.2	2.3	3.3
interp3	spline	1.2	2.2	3.2

Table 6.8: 3D performance of the test tracks captured at 1.50m height.

2D performance - height 1.50m					
function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	linear	nearest	1.7	3.2	3.9
scatteredInterpolant	natural	nearest	1.8	3.2	4.0
griddata	cubic	nearest	1.7	3.1	4.0
griddata	v4	nearest	1.8	3.3	4.0
inpaint DCT	-	-	1.8	3.4	4.2

Table 6.9: 2D performance of the test tracks captured at 1.50m height.

3D performance - height 1.75m				
function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	5.0	8.2	9.8
inpaint PDE 3D	-	5.3	8.4	10.2
inpaint NaNs 3D	-	4.6	7.8	9.6
interp3	makima	4.9	8.4	10.1
interp3	spline	5.2	8.5	10.4

Table 6.10: 3D performance of the test tracks captured at 1.75m height.

2D performance - height 1.75m					
function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	linear	nearest	6.0	9.1	10.5
griddata	natural	nearest	6.0	9.2	10.6
griddata	v4	nearest	5.9	9.1	10.4
inpaint NaNs	-	-	6.0	8.8	10.4
inpaint DCT	-	-	6.0	9.2	10.8

Table 6.11: 2D performance of the test tracks captured at 1.75m height.

Referring to the previous tables, the MILE performances with both 3D and 2D maps at height 1.75m are worse compared to lower heights. Considering, as an example for 3D maps, inpaint DCT 3D, the accuracy and robustness values at 1.75m height are, respectively, 5.0m and 9.8m. At lower heights the worse performance is found at level 1.25m, where the accuracy and robustness values are, respectively, 1.9m and 5.6m. It is possible to notice a consistent difference between the two heights. The same result is found for 2D maps. As an example, considering griddata v4 nearest, the accuracy and robustness values at 1.75m height are, respectively, 5.9m and 10.4m. At lower heights the worse performance is found at level 1.25m, where the accuracy and robustness values are, respectively, 2.3m and 6.2m. For the 2D maps, the result can be explained by the fact that the 2D maps are generated at 1.25m while the considered walks are recorded at 1.75m. For the 3D maps, it is more difficult to find an explanation of the result. A first hypothesis to explain the result is that the magnetic field at height 1.75m can have a lower magnetic diversity along the whole level. If this is the case, it would be more difficult for the MILE to correctly obtain localization information from the magnetic field. To check if the hypothesis is confirmed, the ranges of the magnetic magnitude maps are compared for different levels. At 1.75m the range is equal to $18.14\mu T$, while at lower levels they are $15.88\mu T$ at 1.50m, $18.16\mu T$ at 1.25m and $19.57\mu T$ at 1.00m. The range at 1.75m is found to be in line with lower heights. A further comparison is made analysing the average standard deviation of the magnetic field for the considered heights. At 1.75m the standard deviation is equal to $2.39\mu T$, while at lower levels they are $2.27\mu T$ at 1.50m, $2.30\mu T$ at 1.25m and $2.31\mu T$ at

1.00m. It is found that the standard deviations at the considered heights are similar. The hypothesis of lower magnetic diversity at height 1.75m is not confirmed. As the MILE localization is performed by a sensor fusion approach, also the results coming from the PDR are evaluated. In the next Figures 6.4a and 6.4b, two PDR results of respectively 1.75m and 1.25m heights are shown.

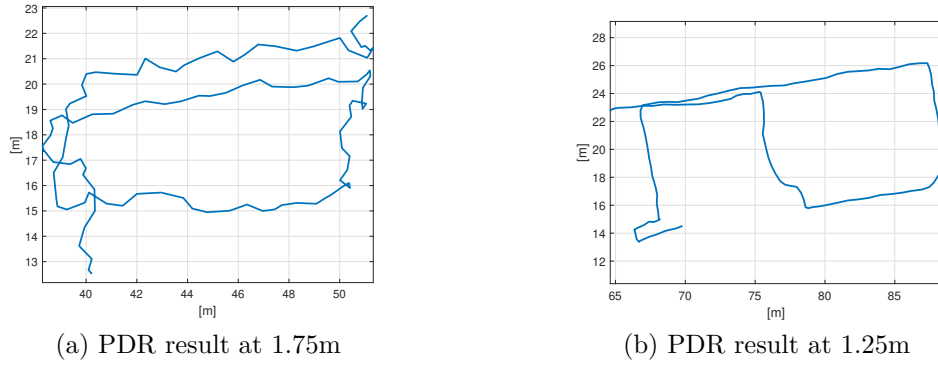


Figure 6.4: PDR results for a 1.75m and a 1.25m heights recordings

In the previous Figure 6.4a, it is possible to observe that the PDR result at 1.75m is a discontinuous trajectory. Comparing it to the PDR result at 1.25m, shown in Figure 6.4b, the PDR output trajectory at 1.75m is definitely worse than the result obtained at 1.25m. The PDR results are depending on the walking style of the user. Thus, the PDR outcome difference for 1.75m and 1.25m heights is explained by a different user walking style for the two cases. Finally, the PDR bad results at 1.75m height explain also the MILE poor localization performance at that height. Referring to the results shown from Table 6.4 to Table 6.9, the MILE performances at non measured levels, 1.00m and 1.50m, are better than the performance at the measured level, 1.25m. This result is found both for 3D and 2D magnetic maps. This observation might be explained by the fact that measured levels depend on the smartphone used to measure it. On the opposite, interpolated levels as 1.00m and 1.50m are located between two measured levels. Consequently, their magnetic field, obtained as interpolation of the 2 adjacent measured levels, might host the influence of two smartphones. To check the validity of the hypothesis, the walk tracks measured by the Sony Xperia 5 are tested level by level. If the hypothesis is correct, the Sony Xperia 5, used to generate the map at 1.25m height, will not experience a different outcome at that height. The result is shown in Figure 6.5.

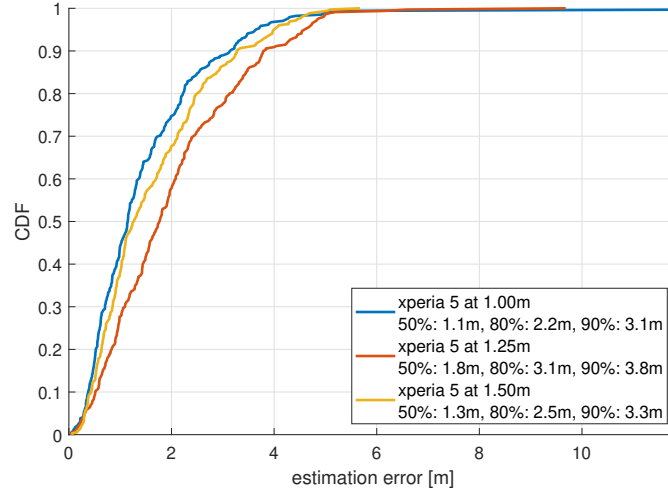


Figure 6.5: Xperia 5 results at different heights

In the CDFs shown in the previous figure it is noticeable that also the Sony Xperia 5 performances are worse at 1.25m than 1.00m and 1.50m. Thus, it is not possible to conclude that the measured levels depend on the smartphones used to measure it. From the statistical behaviour of the magnetic field at 1.25m, described before, it is not possible to notice strange results for that specific height. Also considering the visual analysis of the magnetic maps, shown in Appendix B, the 1.25m height does not seem to be different than the other heights. A further analysis of the described engine performances at measured heights is left as future work.

In general, from Table 6.4 to 6.11, the improved performance of the 3D MILE is confirmed. Both for localization accuracy and robustness, the 3D mode shows better results than 2D mode. Also for the considered set of simulations, apart from the linearly extrapolated maps, all the 2D maps show similar results. Regarding the 3D maps, the inpaint NaNs 3D maps usually perform slightly worse than the other 3D maps. Apart from inpaint NaNs 3D, the 3D maps performances are all similar.

A final performance evaluation is performed using Dataset 2, with walk recordings captured at variable heights from 1.00m to 1.50m. In this dataset, 1.75m height is not considered due to the previously described PDR issue. In Figure 6.6, the best 3D performance, obtained using inpaint DCT 3D, is compared to the best 2D performance, obtained using griddata cubic nearest. Inpaint DCT 3D shows a localization accuracy of 1.1m and a robustness of 3.0m. Griddata cubic nearest returns a localization accuracy of 1.5m and a robustness of 3.6m. As expected, the 3D map improves the localization accuracy by 0.4m and the localization robustness by 0.6m.

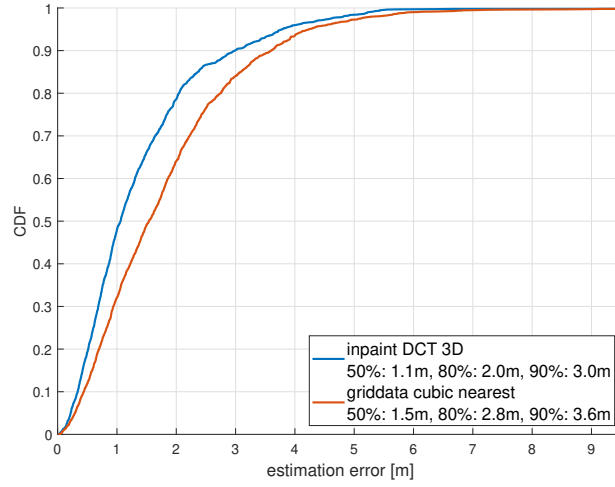


Figure 6.6: Comparison between the best 3D against the best 2D performances, using Dataset 2

6.3 Height tracking capability

To test the height tracking capability of the 3D MILE, two evaluations are performed. For both the evaluations, the estimated heights mean values and standard deviations are used as evaluation metrics.

At first, the test Dataset 1 is processed by the 3D MILE one height level at the time. All the walk recordings of the considered dataset consist of 2 minutes recordings at fixed heights: 1.00m, 1.25m, 1.50m and 1.75m. In Table 6.12, the mean values of the estimated heights are shown for each of the levels of the test Dataset 1. In Table 6.13, the standard deviations of the estimated heights are shown for each of the levels of the test Dataset 1. Both the mean values and standard deviations are reported for all the available 3D magnetic maps.

Height mean values [m] for test Dataset 1					
target height [m]	inpaint DCT 3D	inpaint PDE 3D	inpaint NaNs 3D	interp3 makima	interp3 spline
1.00	1.13	1.10	1.09	1.11	1.11
1.25	1.44	1.36	1.41	1.41	1.43
1.50	1.50	1.45	1.47	1.47	1.47
1.75	1.77	1.76	1.75	1.77	1.75

Table 6.12: Estimated heights mean values for test Dataset 1.

Height standard deviations [m] for test Dataset 1					
height [m]	inpaint DCT 3D	inpaint PDE 3D	inpaint NaNs 3D	interp3 makima	interp3 spline
1.00	0.38	0.36	0.39	0.37	0.37
1.25	0.25	0.24	0.29	0.25	0.26
1.50	0.21	0.20	0.22	0.22	0.22
1.75	0.21	0.21	0.22	0.21	0.22

Table 6.13: Estimated heights standard deviations for test Dataset 1.

In the previous tables, a first result is that all the five 3D maps show similar height tracking performances. Overall, it is found that the mean value of the estimated heights is really close to the target height for levels 1.50m and 1.75m. On the opposite, for lower levels, especially 1.25m, the mean estimated height differs from the target height. In general, referring to Table 6.13 the estimated heights standard deviations are relatively high. This means that the estimated height is not converging to the target height for a lot of walk recordings of Dataset 1. In Figure 6.7 and 6.8, two examples of height tracking performances are shown, with target height equal to 1.00m. In Figure 6.7, the estimated height is constantly over the target height of 1.00m, while, in Figure 6.8, the estimated height converges to 1.00m after an initial phase of wrong height estimation. Consequently, the first result represents an example of bad height tracking of the 3D MILE and the second result represents an example of good height tracking of the 3D MILE.

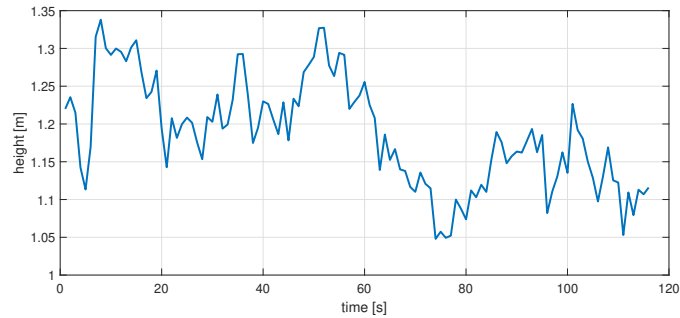


Figure 6.7: Example of bad height tracking result. The target height is 1.00m.

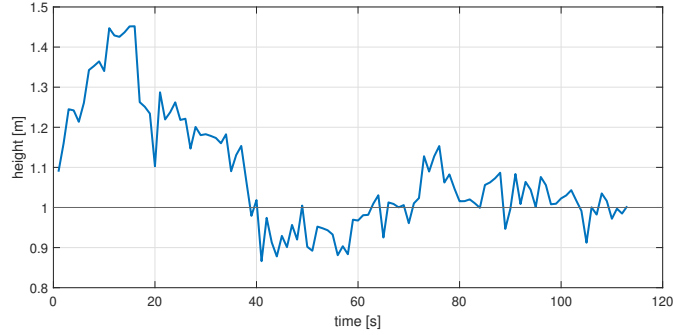


Figure 6.8: Example of good height tracking result. The target height is 1.00m.

A further evaluation of the 3D MILE height tracking capability is performed employing Dataset 2. The Dataset 2 captures consist of 3 minutes of recordings at variable heights. During the first minute, the smartphone is held at 1.00m height, during the second minute at 1.25m and during the third minute at 1.50m. As in the previous case, in Tables 6.14 and 6.15 are shown, respectively, the mean values and the standard deviations of the estimated heights relative to different target heights.

Height mean values [m] for Dataset 2					
target height [m]	inpaint DCT 3D	inpaint PDE 3D	inpaint NaNs 3D	interp3 makima	interp3 spline
1.00	1.45	1.40	1.42	1.40	1.39
1.25	1.49	1.39	1.47	1.42	1.43
1.50	1.60	1.51	1.59	1.55	1.54

Table 6.14: Estimated heights mean values for test Dataset 2.

Height standard deviations [m] for Dataset 2					
height [m]	inpaint DCT 3D	inpaint PDE 3D	inpaint NaNs 3D	interp3 makima	interp3 spline
1.00	0.25	0.24	0.25	0.25	0.26
1.25	0.26	0.28	0.26	0.29	0.32
1.50	0.22	0.26	0.23	0.26	0.28

Table 6.15: Estimated heights standard deviations for test Dataset 2.

In Table 6.14, it is possible to observe that the average height tracking accuracy for Dataset 2 is lower than the one found in Table 6.12 for Dataset 1. This result might be explained by the fact that the smartphone is held for only 1 minute at a same height in Dataset 2, against the 2 minutes in Dataset 1. Thus, for Dataset 2, the MILE height tracking is not able to converge to the target height and the results are less precise. Referring to Table 6.15, the standard deviations are confirmed to

be high, describing high variability of the height estimations.

As described so far, the MILE performance depends on many factors like the magnetic map precision and the PDR accuracy. The PDR has a central role in the motion model (5.1). Beside computing the estimated movement length and heading change, the PDR provides also the estimated height variation in equation (5.1e). With this in mind, the PDR results for the z dimension are evaluated for Dataset 2. In this case, the expected outcome is to have a 25cm height increase after the first minute and after the second minute of walk. The evaluation shows that the outcome does not satisfy the expectations: the PDR height variation is uncorrelated with respect to the real smartphone height variations. In Figure 6.9, an example of PDR height estimation is shown for a Dataset 2 capture. The initial height is referenced to 0m.

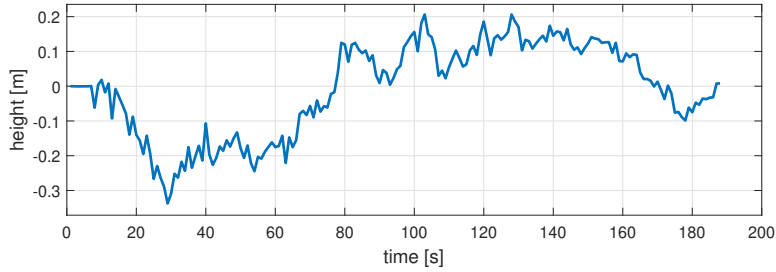


Figure 6.9: PDR height estimation outcome

Finally, it is possible to conclude that the 3D MILE height tracking capability is not accurate. The state of the art PDR is not precise for computing height variations and the only magnetic data are not enough to reach a precise height estimation. From the result of Figure 6.6, the 2D localization accuracy of the 3D MILE is 1.1m. Considering this, the height estimation is a challenging task due to the required accuracy, that is higher than the current engine capabilities.

6.4 Barometer sensor analysis

To improve the height tracking performance of the 3D MILE, a possible integration of the smartphones barometers with the engine is investigated. Barometers are sensors used to measure air pressure and, nowadays, they are integrated in many smartphones. The measured barometric pressure depends on environmental factors like the altitude, the weather and the temperature [25]. Due to the previous dependencies, the barometric pressure cannot be used to obtain absolute estimation of the height position of the sensor. Nevertheless, it can be used to acquire useful information about relative movements along the height dimension. Theoretically,

the barometer pressure increases moving the barometer from a starting height to a lower height and viceversa, thus, it is possible to measure a height variation. In the context of the thesis, the two barometer sensors that are embedded in the previously presented five smartphones are investigated. They are the Bosch pressure sensor BMP380 [12] and the STMicroelectronics sensor LPS22HB [21]. The BMP380 is embedded in the Sony Xperia 5, Sony Xperia XZ3, Pixel 4 and Nokia 9 while the LPS22HB is embedded in the Samsung s10e. In the Bosch sensor datasheet, it is indicated that the BMP380 sensor has a relative accuracy of $\pm 0.06\text{hPa}$ that corresponds, according to the Bosch documentation, to height variation of $\pm 50\text{cm}$. In the STMicroelectronics sensor datasheet, only the relative accuracy of the LPS22HB sensor as $\pm 0.1\text{hPa}$ is indicated. Based to the specification sheets of the two sensors, the usage of both the barometers seems to be challenging for the purposes of the thesis. Nevertheless, a simple offline investigation is performed. The objective is to check if the barometer sensors are capable to detect height variations in the order of $\pm 0.25\text{cm}$, which is the distance between different levels of the 3D maps used in Section 6.3. The test consists of measuring the barometric pressure at three heights spaced by 0.25cm . The smartphones are moved from one level to the next one every 60 seconds intervals. Subsequently, as the signal is very noisy, the median value of each phase of the experiment is computed. Using the median value, the aim is to consider only the direct component of the signal. The objective is to check if there is a constant difference between the median values of measured pressures at different heights. The raw pressure signal is shown in Figure 6.10 by the blue line. In the same Figure 6.10, the median value during the three phases of the experiment is shown in red.

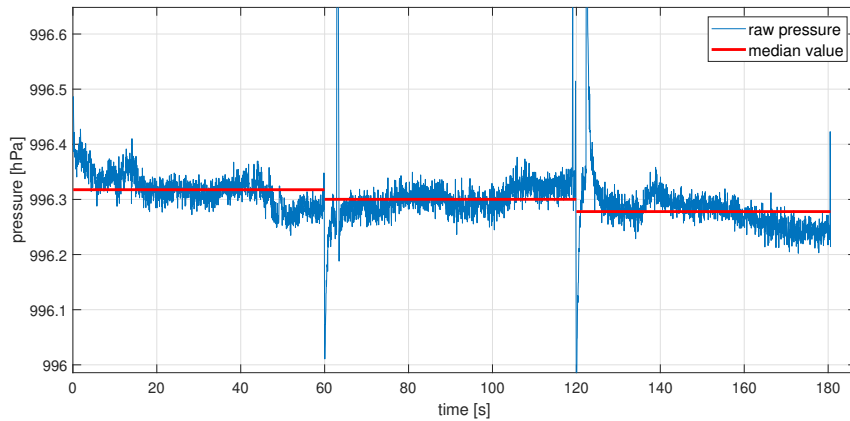


Figure 6.10: Barometric sensor test outcome

In the previous Figure 6.10, the spikes at $t = 60\text{s}$ and $t = 120\text{s}$ are generated by the smartphone movement from one of the heights to the successive height. In

general, the median values of the three phases of the experiment decrease every time the smartphone height is increased, following the expectations. Moreover, the noise component of the signal is relatively high compared to the median values differences. To better evaluate the pressure signal behaviour, the test is repeated for eight measurements for each barometer model. The results are shown in the following Tables 6.16 and 6.17. For Table 6.16, the Bosch BMP380 embedded in the Pixel 4 is used. For Table 6.17, the Samsung s10e is used as it is the only available smartphone equipped with the LPS22HB barometer. In the two tables the difference between the median values of the three phases of the experiment are reported.

BMP380 - difference of the median values [hPa]								
height jump	1	2	3	4	5	6	7	8
from 1.00m to 1.25m	0.019	0.034	0.017	0.018	0.041	0.011	0.059	-0.008
from 1.25m to 1.50m	0.042	0.031	0.022	-0.001	-0.004	0.005	-0.045	0.048
from 1.00m to 1.50m	0.061	0.065	0.040	0.017	0.037	0.016	0.015	0.040

Table 6.16: Barometer BMP380 differences of the barometric pressure median values for different height jumps.

LPS22HB - difference of the median values [hPa]								
height jump	1	2	3	4	5	6	7	8
from 1.00m to 1.25m	0.028	0.022	0.015	0.004	0.010	0.011	0.029	0.025
from 1.25m to 1.50m	0.020	0.023	-0.022	0.035	0.003	0.048	0.010	0.030
from 1.00m to 1.50m	0.048	0.046	-0.007	0.039	0.013	0.059	0.039	0.054

Table 6.17: Barometer LPS22HB differences of the barometric pressure median values for different height jumps.

Referring to the two tables, the differences between median values do not converge to a unique value. It is observed that for some height increases the barometric pressure difference does not confirm the expected outcome and it is negative. For some other height increases, the pressure difference is almost zero. Overall, the random behaviour shown in Tables 6.16 and 6.17 is explained by the fact that the accuracy of the sensors is less than the resolution needed for height tracking purposes. Nevertheless, in a future work the barometer can be integrated in the MILE for floor level detection. In fact, there are some surveys showing the reliability of the pressure sensor results for the purpose of floor level detection [14].

Chapter 7

Conclusions and future work

In the thesis, a 3-dimensional magnetic fingerprinting approach for indoor localization engines is discussed.

At first, a survey about the state of the art magnetic fingerprinting approaches is conducted. It is found that, although some surveys show promising results, there still does not exist any reliable indoor localization system based on 3D magnetic maps.

The magnetic indoor localization engine, MILE, is introduced. The interpolation step of the available 2D map generation method is evaluated. Furthermore, additional interpolation approaches are added to improve the quality of 2D magnetic maps.

A new measurement setup is designed to measure the 3-dimensional magnetic field. The new proposed setup is designed to simplify the time consuming state of the art magnetic measurement procedure. To reduce the effort during the measurement step, two V-SLAM systems are integrated into the measurement setup and evaluated. The map generation framework is extended to generate 3D magnetic maps using the 3D magnetic measurements captured with the new setup. In this context, different 3-dimensional interpolation-extrapolation methods are introduced in the framework.

An approach is proposed for 2D and 3D magnetic maps in order to evaluate the quality of the magnetic maps. For this evaluation, the magnetic field statistical behaviour and the smoothness of the maps are considered as key metrics.

Moreover, the MILE is extended to be able to work in 3D. The vertical dimension z is added to the particle filter states, that corresponds to the height information. Both the map generation framework and the MILE are tested in office areas. A final evaluation of the localization performances of the 3D framework against 2D framework is performed. The most important result is that, using 3D maps, there is a performance improvement both in terms of localization accuracy, 40 – 50cm, and localization robustness, 40 – 60cm. Also, it is found that the choice of a single 3D magnetic interpolation method and a single 2D magnetic interpolation method is a

challenging task. Although there is a clear performance difference between 3D and 2D maps, most of the considered methods have similar performances within the two classes. Nevertheless, the recommended 3D maps are inpaint DCT and interp3 spline. Regarding the several 2D maps with similar outcomes, the recommended map is griddata v4 nearest, as it is found to be between the best performing 2D maps in every simulation. Moreover, it is found that the results of the evaluation approach used to evaluate the quality of magnetic maps do not completely match the performances evaluation. The only matching result between map quality evaluation and performance analysis is that, in general, the 2D magnetic maps using linear extrapolation perform worse than the other 2D maps.

Finally, the height tracking capability of the 3D MILE is evaluated. The results show that it is not possible to reach an accurate height tracking of the smartphones in the current stage of the development. Furthermore, to improve the height tracking, two smartphone's embedded barometers are analysed. As a result of the analysis, the accuracy of the barometers is not enough to reach the required resolution.

The thesis conclusions open the way to new developing fields of the MILE. Different tasks are suggested as future works.

The smartphones dependencies of the measured magnetic maps have to be further analysed. In fact, from the evaluations performed in the thesis, it is still not clear why the 3D MILE performances are better for interpolated levels than for measured levels.

Considering the different performances of different 3D maps height levels, a possible development of an adaptive parameter optimization for the observation model covariances is suggested.

Regarding the 2D MILE, there is the possibility to use the vertical variation of the magnetic field, taken from 3D maps, to assign a reliability value to the 2D magnetic maps. In the proposed way, 2D grid points with high vertical variation of the 3D magnetic field will be given with a lower reliability, thus, lower weight.

Regarding the map generation process, it is found that the trajectory correction phase requires a time consuming manual reference point setting. Further V-SLAM systems can be investigated to obtain better trajectories, reducing the number of reference points, thus, the time impact of the current procedure.

Appendix A

Test bed for 3D maps evaluation

In Figure A.1, the original and the corrected (after Helmert transformations) trajectories used for 3D maps evaluation are shown. In Figure A.2, the heading for the corrected trajectory is shown.

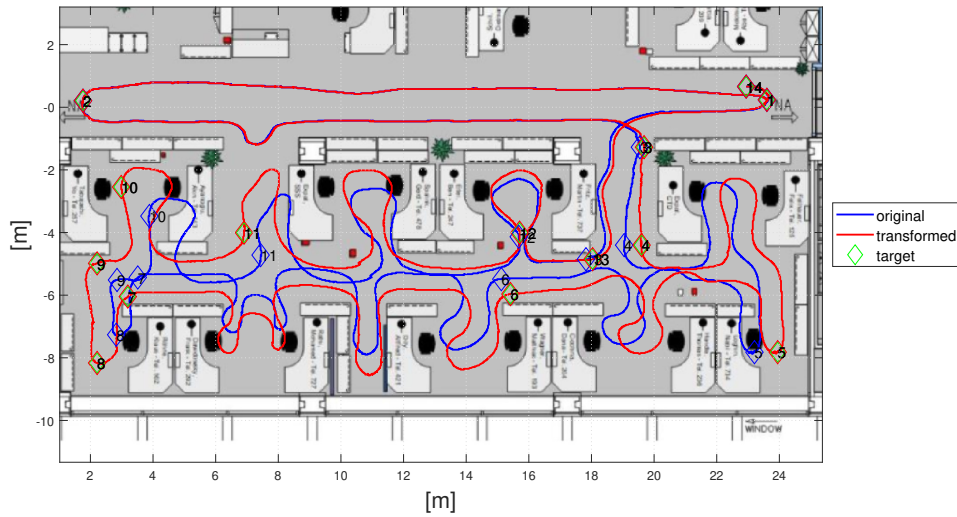


Figure A.1: Original and transformed trajectories used for 3D maps evaluation

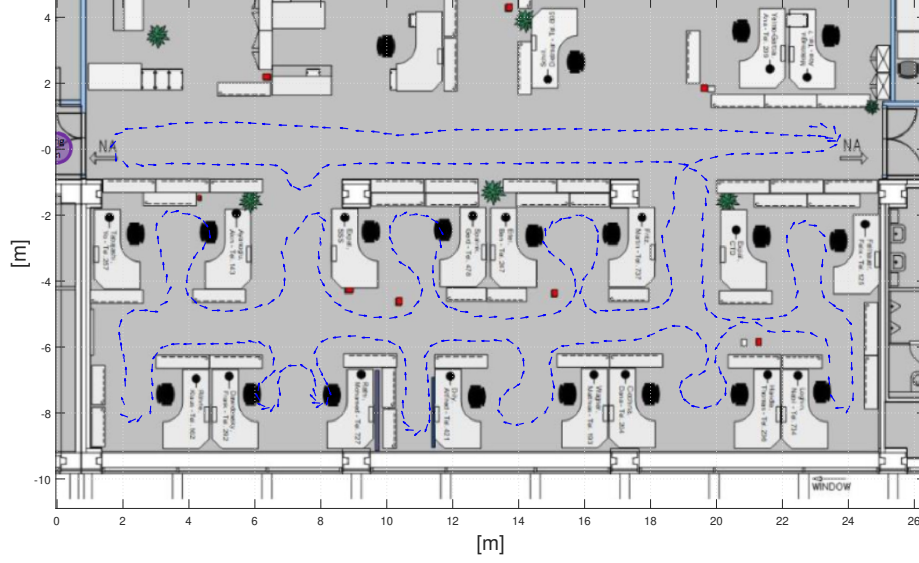


Figure A.2: Smartphone heading along the trajectory used for 3D maps evaluation

In Figures A.3-A.8, an example of magnetic map is shown. For conciseness, only the magnitude map obtained using inpaint DCT 3D is visualized. In the images, it is possible to notice that low heights, 0.25m and 0.50m, show more peaks (visible in bright yellow and dark blue) due to the proximity to the floor. This is explained by the presence of ferromagnetic construction materials inside the floor.

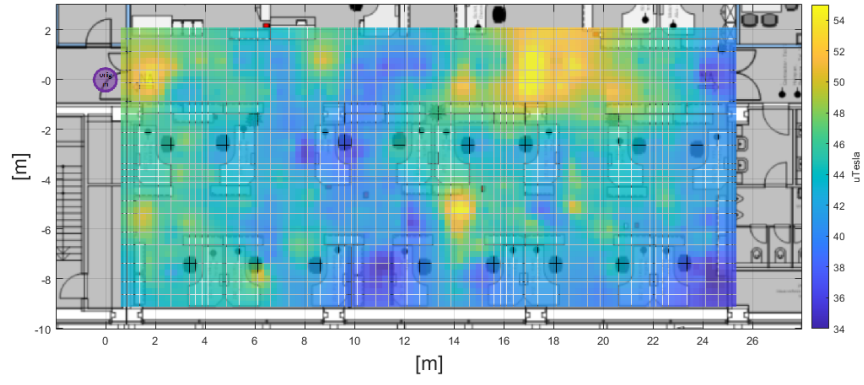


Figure A.3: Magnetic magnitude at height 0.25m obtained using inpaint DCT 3D

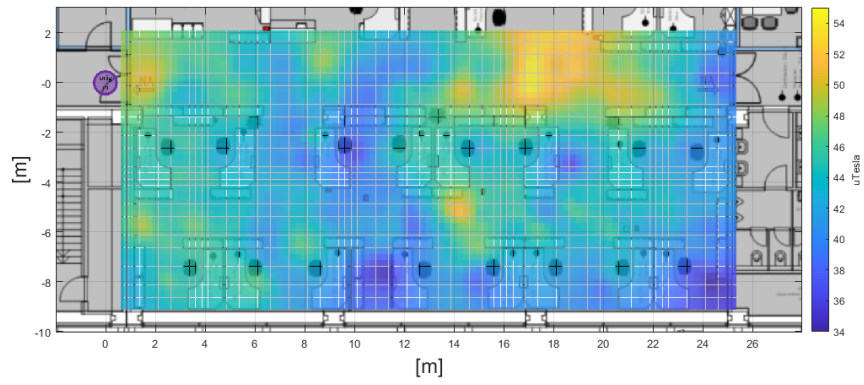


Figure A.4: Magnetic magnitude at height 0.50m obtained using inpaint DCT 3D

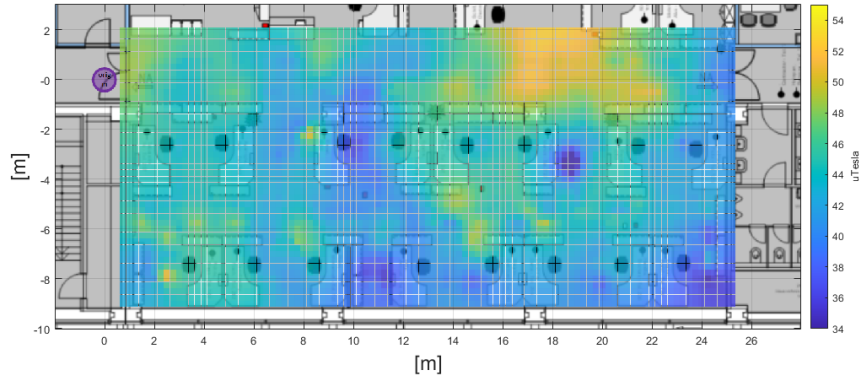


Figure A.5: Magnetic magnitude at height 0.75m obtained using inpaint DCT 3D

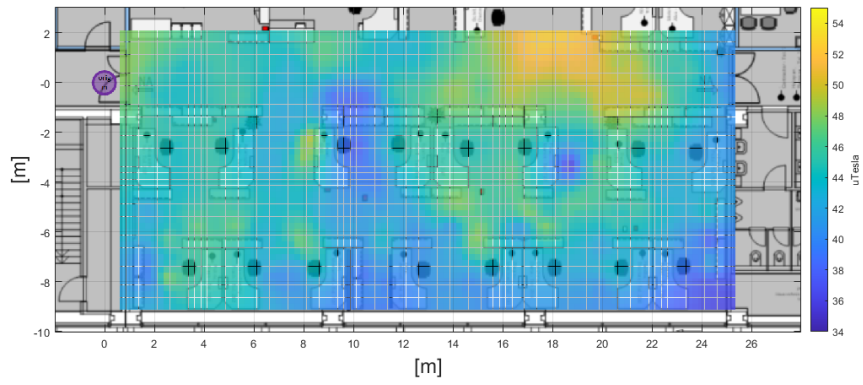


Figure A.6: Magnetic magnitude at height 1.00m obtained using inpaint DCT 3D

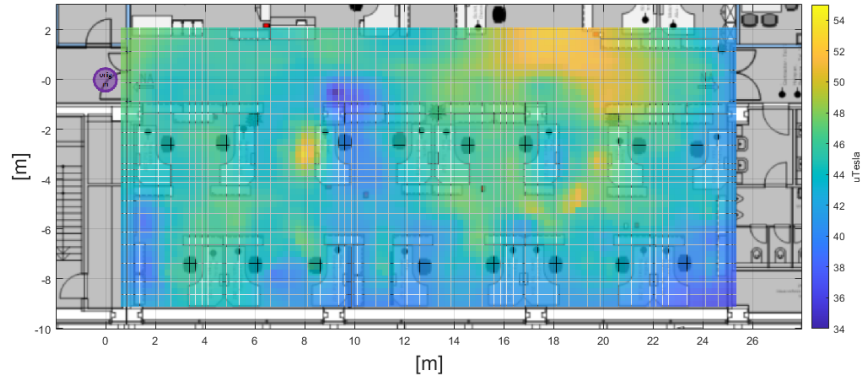


Figure A.7: Magnetic magnitude at height 1.25m obtained using inpaint DCT 3D

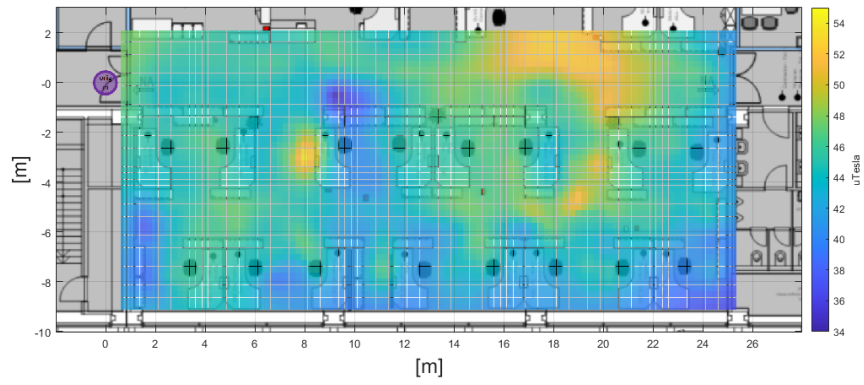


Figure A.8: Magnetic magnitude at height 1.50m obtained using inpaint DCT 3D

Appendix B

Test bed for performance evaluation

In Figure B.1, the original and the corrected (after Helmert transformations) trajectories used for 3D performance evaluation are shown. In Figure B.2, the heading for the corrected trajectory is shown.

In Figures B.3-B.10, an example of 3D magnetic map used for performance evaluation is attached. For conciseness, only the magnitude map obtained using inpaint DCT 3D is visualized. In the images, it is possible to notice that low heights, 0.25m and 0.50m, show more peaks (visible in bright yellow and dark blue) due to the proximity to the floor. This is explained by the presence of ferromagnetic construction materials inside the floor. For this reason, the magnitude color scale at heights 0.25m and 0.50m has been adapted to a different range than the other heights.

In Figure B.11, an example of 2D magnetic magnitude map used for performance evaluation is shown. For conciseness, only the result obtained using inpaint DCT is visualized.

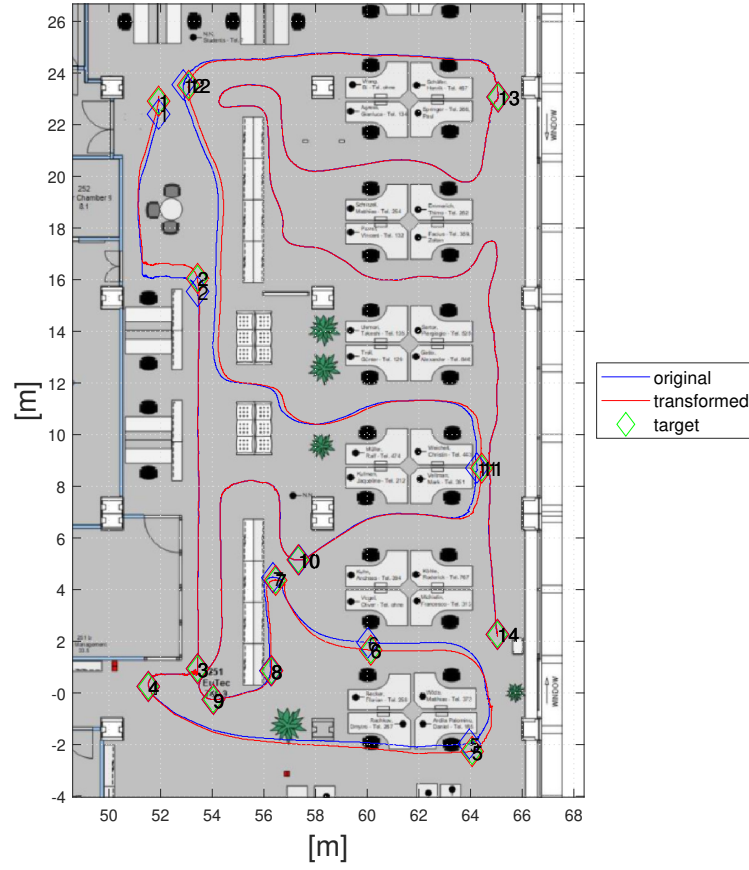


Figure B.1: Original and transformed trajectories used for 3D performance evaluation

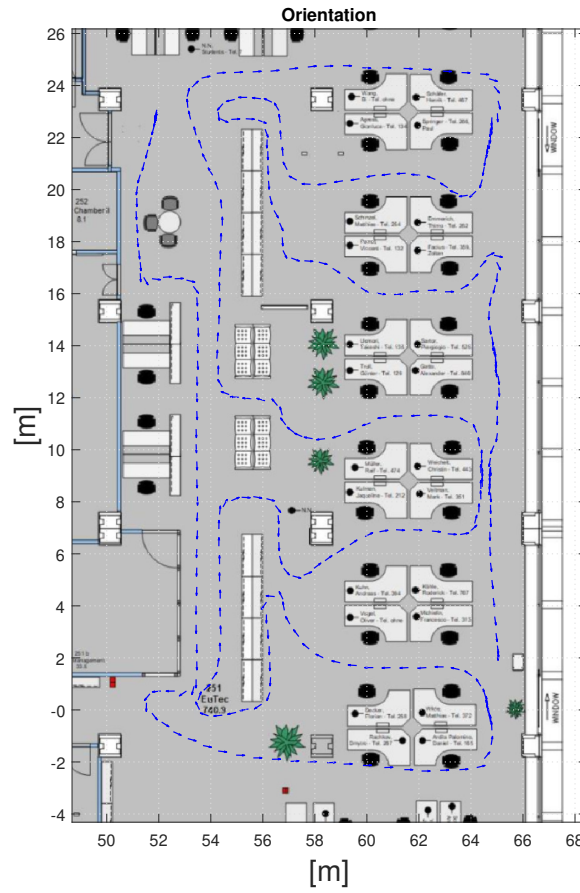


Figure B.2: Smartphone heading along the trajectory used for 3D performance evaluation

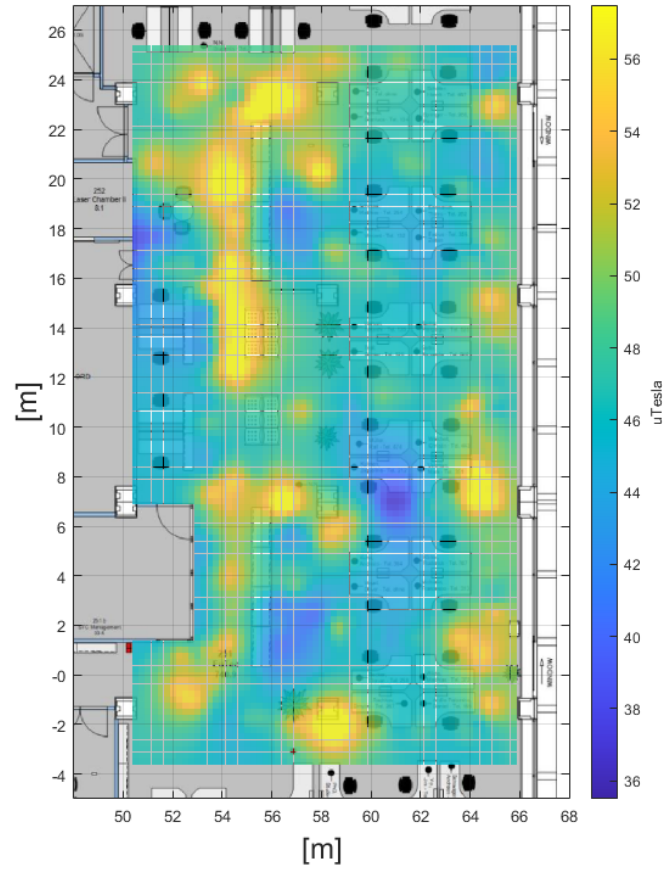


Figure B.3: Magnetic magnitude at height 0.25m obtained using inpaint DCT 3D

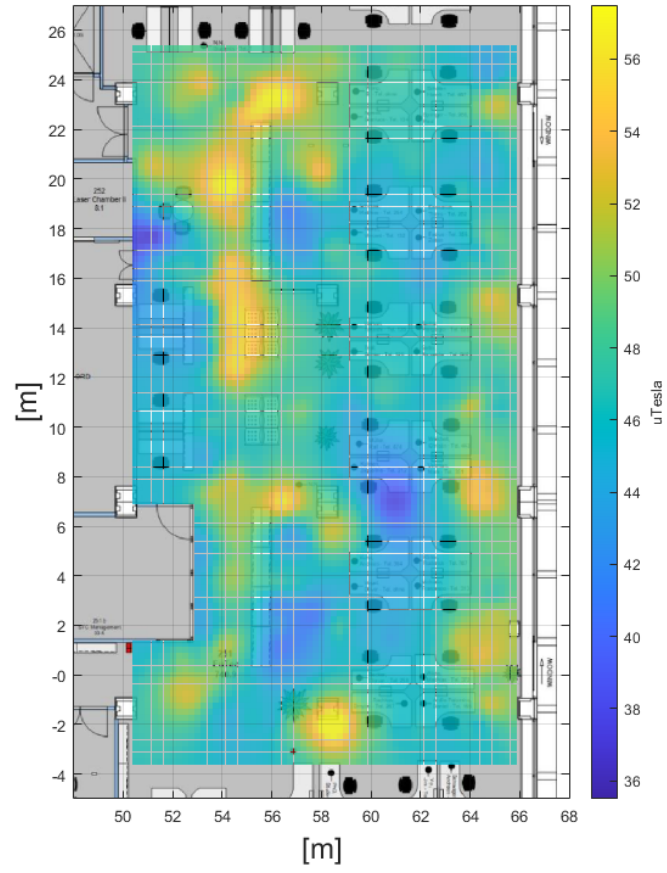


Figure B.4: Magnetic magnitude at height 0.50m obtained using inpaint DCT 3D

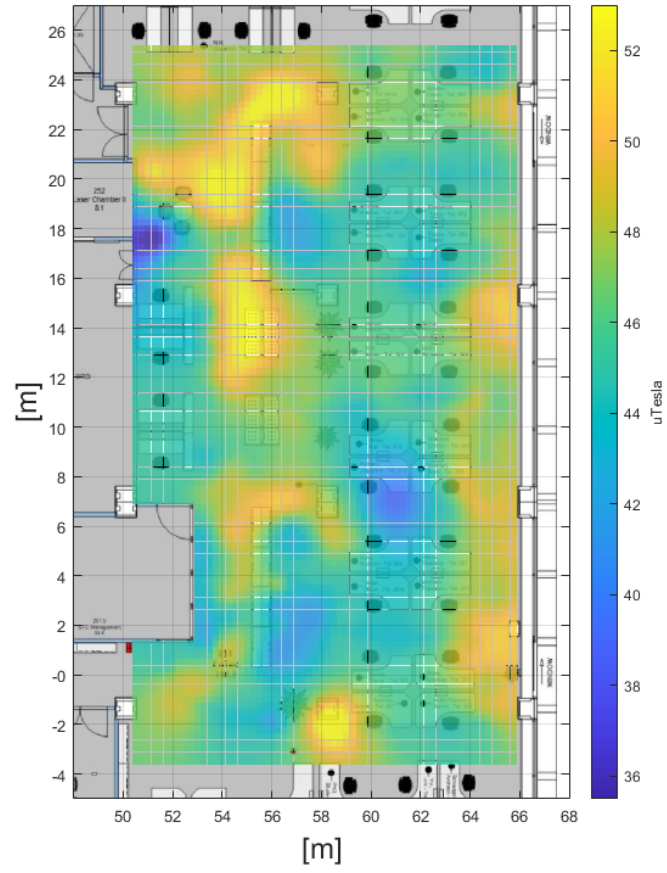


Figure B.5: Magnetic magnitude at height 0.75m obtained using inpaint DCT 3D

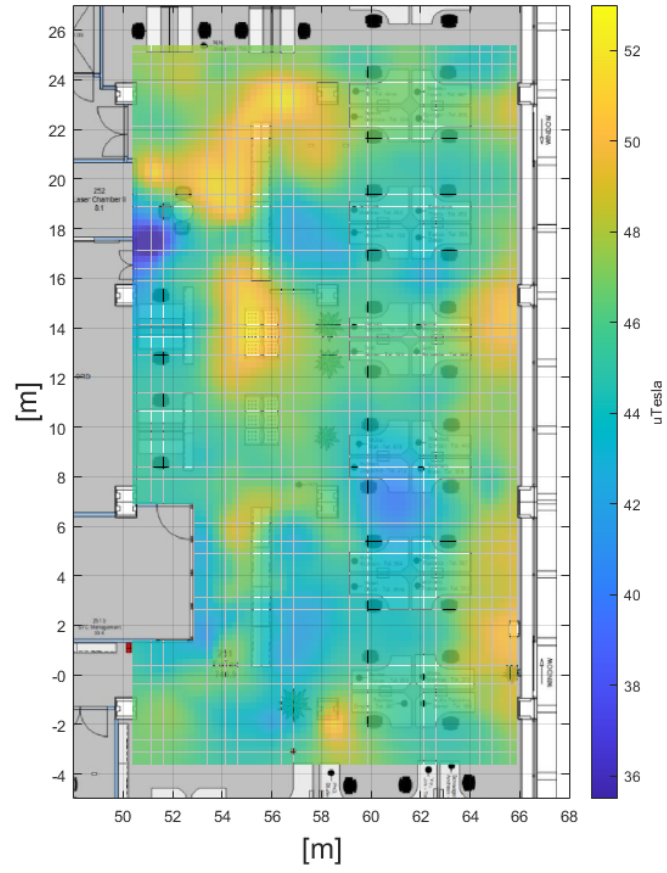


Figure B.6: Magnetic magnitude at height 1.00m obtained using inpaint DCT 3D

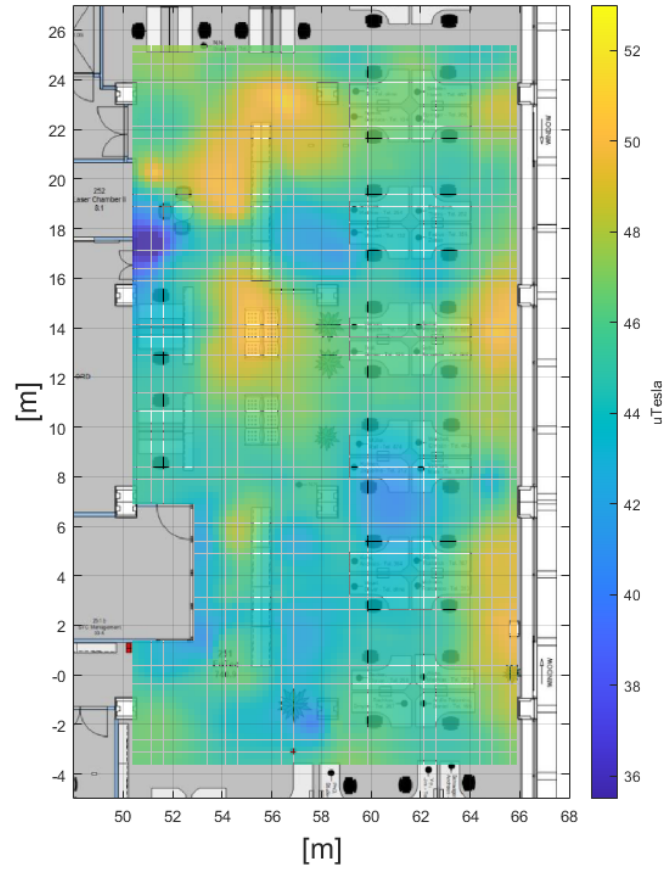


Figure B.7: Magnetic magnitude at height 1.25m obtained using inpaint DCT 3D

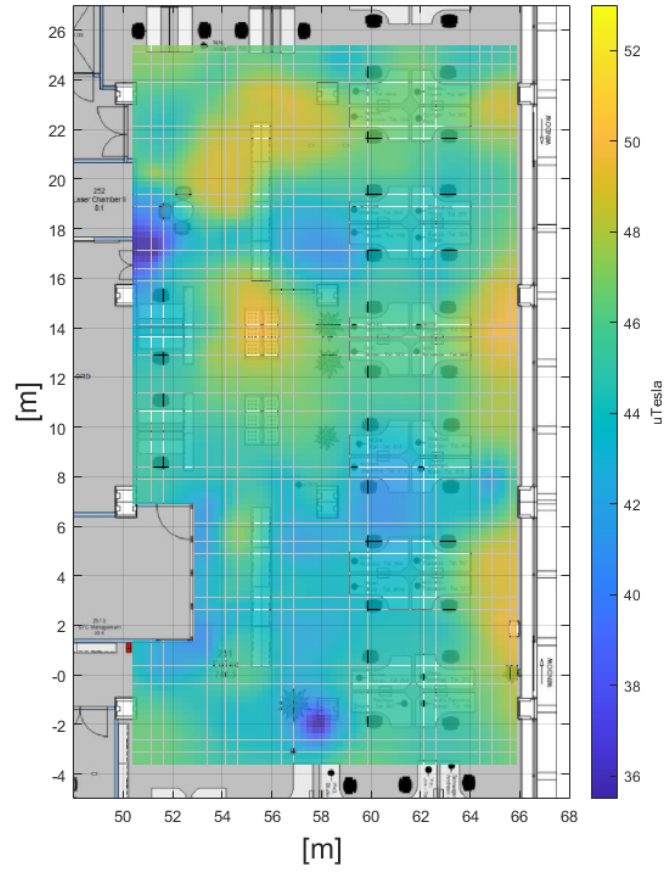


Figure B.8: Magnetic magnitude at height 1.50m obtained using inpaint DCT 3D

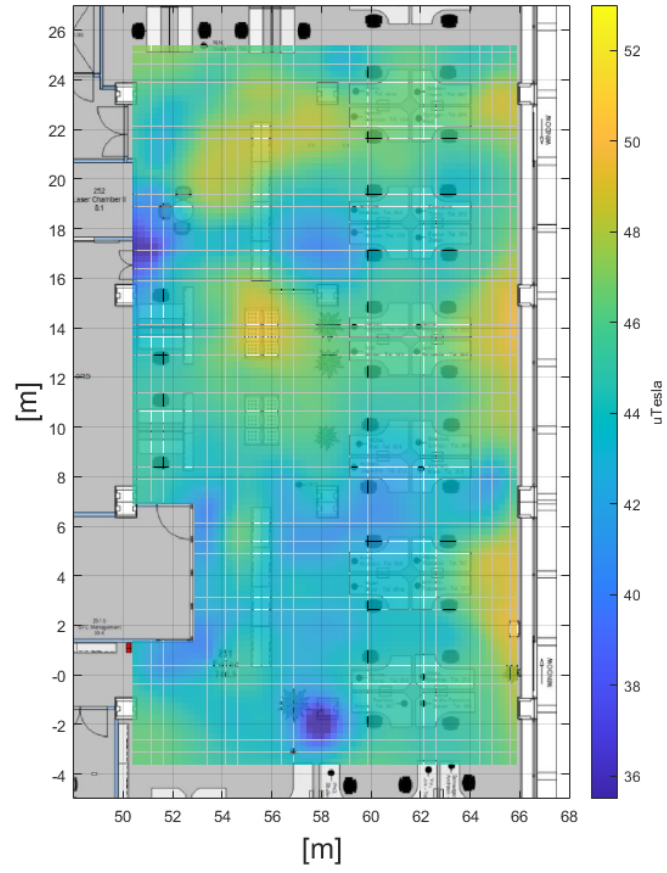


Figure B.9: Magnetic magnitude at height 1.75m obtained using inpaint DCT 3D

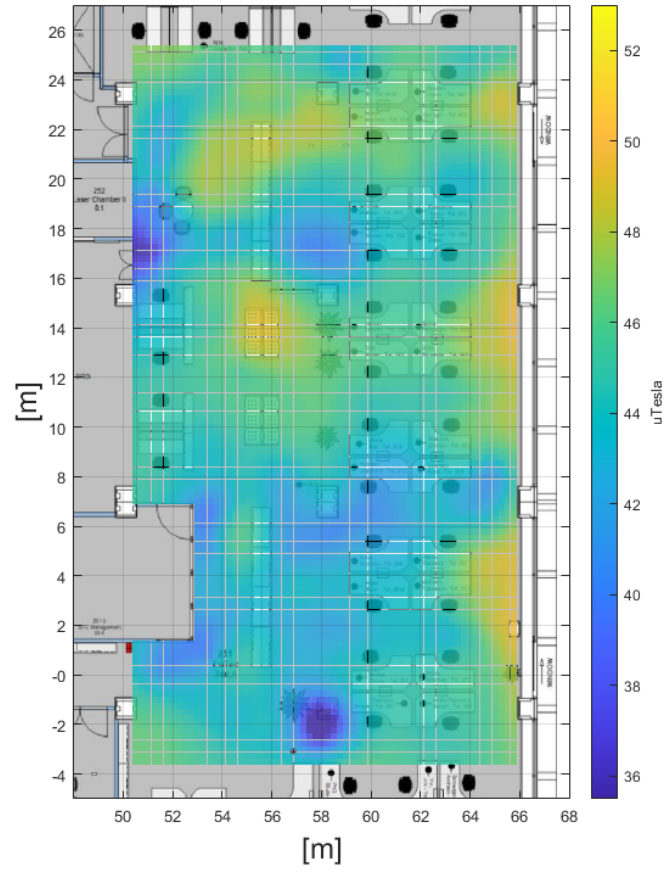


Figure B.10: Magnetic magnitude at height 2.00m obtained using inpaint DCT 3D

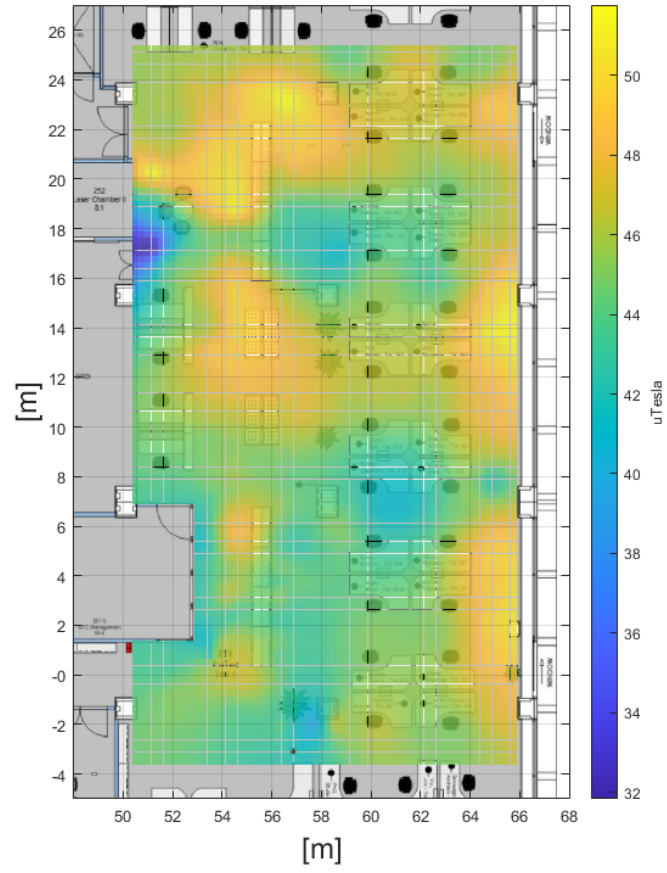


Figure B.11: Magnetic magnitude obtained using inpaint DCT

Appendix C

Model training results

In Tables C.1-C.10, the results of the training procedure for different covariances sets are reported. The 2D maps best performances are obtained using $C^m = 20\mu T^2$ and $C^i = 30\text{deg}^2$. The 3D maps best performances are obtained using $C^m = 10\mu T^2$ and $C^i = 15\text{deg}^2$. In this regard, it is chosen to not set covariances lower than $C^m = 10\mu T^2$ and $C^i = 15\text{deg}^2$. In fact, from preliminary analysis, it was necessary to leave space for device variabilities. Thus, too tight covariances are avoided.

function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	2.5	5.0	7.8
inpaint PDE 3D	-	2.6	5.1	7.8
inpaint NaNs 3D	-	2.6	5.1	7.6
interp3	makima	2.5	5.1	7.8
interp3	spline	2.6	5.2	7.6

Table C.1: 3D performance of the test dataset for $C^m = 100\mu T^2$ and $C^i = 400\text{deg}^2$.

function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	linear	linear	2.9	5.4	7.4
scatteredInterpolant	linear	nearest	2.6	5.1	7.3
scatteredInterpolant	natural	linear	2.8	5.3	7.6
scatteredInterpolant	natural	nearest	2.6	5.2	7.3
griddata	linear	linear	2.8	5.3	7.5
griddata	linear	nearest	2.5	5.1	7.3
griddata	natural	linear	2.8	5.3	7.5
griddata	natural	nearest	2.7	5.0	7.3
griddata	cubic	linear	2.7	5.2	7.6
griddata	cubic	nearest	2.5	5.1	7.4
griddata	v4	linear	2.5	5.1	7.3
griddata	v4	nearest	2.5	5.0	7.4
inpaint PDE	-	-	2.6	5.1	7.3
inpaint NaNs	-	-	2.5	5.0	7.3
inpaint DCT	-	-	2.6	5.0	7.4

Table C.2: 2D performance of the test dataset for $C^m = 100\mu T^2$ and $C^i = 400\text{deg}^2$.

function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	2.3	4.7	7.1
inpaint PDE 3D	-	2.4	4.9	7.3
inpaint NaNs 3D	-	2.5	5.0	7.3
interp3	makima	2.4	4.8	7.3
interp3	spline	2.4	4.8	7.4

Table C.3: 3D performance of the test dataset for $C^m = 60\mu T^2$ and $C^i = 150\text{deg}^2$.

function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	linear	linear	2.8	5.3	7.4
scatteredInterpolant	linear	nearest	2.5	5	7.2
scatteredInterpolant	natural	linear	2.7	5.2	7.5
scatteredInterpolant	natural	nearest	2.5	5.0	7.3
griddata	linear	linear	2.7	5.2	7.4
griddata	linear	nearest	2.4	5.0	7.2
griddata	natural	linear	2.7	5.3	7.4
griddata	natural	nearest	2.5	5.1	7.4
griddata	cubic	linear	2.5	5.3	7.6
griddata	cubic	nearest	2.4	5.0	7.3
griddata	v4	linear	2.4	5.1	7.2
griddata	v4	nearest	2.4	4.9	7.1
inpaint PDE	-	-	2.5	5.0	7.2
inpaint NaNs	-	-	2.4	4.9	7.2
inpaint DCT	-	-	2.5	5.0	7.4

Table C.4: 2D performance of the test dataset for $C^m = 60\mu T^2$ and $C^i = 150\text{deg}^2$.

function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	2.3	4.7	7.0
inpaint PDE 3D	-	2.3	4.7	7.1
inpaint NaNs 3D	-	2.3	4.9	7.3
interp3	makima	2.2	4.8	6.9
interp3	spline	2.3	4.6	6.8

Table C.5: 3D performance of the test dataset for $C^m = 40\mu T^2$ and $C^i = 60\text{deg}^2$.

function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	linear	linear	2.7	5.1	7.1
scatteredInterpolant	linear	nearest	2.4	4.9	7.0
scatteredInterpolant	natural	linear	2.6	5.1	7.5
scatteredInterpolant	natural	nearest	2.4	4.9	7.0
griddata	linear	linear	2.6	5.2	7.2
griddata	linear	nearest	2.4	5.0	7.2
griddata	natural	linear	2.6	5.2	7.4
griddata	natural	nearest	2.5	5.0	7.1
griddata	cubic	linear	2.5	5.2	7.3
griddata	cubic	nearest	2.4	5.0	7.2
griddata	v4	linear	2.4	5.0	7.1
griddata	v4	nearest	2.3	4.9	7.1
inpaint PDE	-	-	2.4	4.9	7.0
inpaint NaNs	-	-	2.4	4.9	7.2
inpaint DCT	-	-	2.4	4.9	7.1

Table C.6: 2D performance of the test dataset for $C^m = 40\mu T^2$ and $C^i = 60\text{deg}^2$.

function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	2.1	4.9	7.3
inpaint PDE 3D	-	2.0	4.3	6.3
inpaint NaNs 3D	-	2.2	4.7	6.5
interp3	makima	2.0	4.6	7.0
interp3	spline	2.0	4.6	6.9

Table C.7: 3D performance of the test dataset for $C^m = 20\mu T^2$ and $C^i = 30\text{deg}^2$.

function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	linear	linear	2.5	5.3	7.2
scatteredInterpolant	linear	nearest	2.3	4.9	6.8
scatteredInterpolant	natural	linear	2.5	5.3	7.2
scatteredInterpolant	natural	nearest	2.3	4.9	7.0
griddata	linear	linear	2.5	5.4	7.3
griddata	linear	nearest	2.4	5.0	7.0
griddata	natural	linear	2.5	5.4	7.4
griddata	natural	nearest	2.4	5.1	7.1
griddata	cubic	linear	2.5	5.2	7.3
griddata	cubic	nearest	2.3	4.9	7.0
griddata	v4	linear	2.4	5.0	6.9
griddata	v4	nearest	2.3	5.0	6.9
inpaint PDE	-	-	2.3	4.8	6.8
inpaint NaNs	-	-	2.3	4.9	6.8
inpaint DCT	-	-	2.3	5.0	6.9

Table C.8: 2D performance of the test dataset for $C^m = 20\mu T^2$ and $C^i = 30\text{deg}^2$.

function	mode	50% [m]	80% [m]	90% [m]
inpaint DCT 3D	-	1.8	4.2	5.9
inpaint PDE 3D	-	1.9	4.3	6.1
inpaint NaNs 3D	-	2.0	4.6	6.4
interp3	makima	1.9	4.4	6.7
interp3	spline	1.8	4.3	6.1

Table C.9: 3D performance of the test dataset for $C^m = 10\mu T^2$ and $C^i = 15\text{deg}^2$.

function	interp.	extrap.	50% [m]	80% [m]	90% [m]
scatteredInterpolant	linear	linear	2.7	5.5	7.5
scatteredInterpolant	linear	nearest	2.3	4.9	6.8
scatteredInterpolant	natural	linear	2.5	5.5	7.4
scatteredInterpolant	natural	nearest	2.3	5.0	6.9
griddata	linear	linear	2.7	5.7	7.6
griddata	linear	nearest	2.4	5.1	7.1
griddata	natural	linear	2.8	5.8	7.8
griddata	natural	nearest	2.3	5.2	7.1
griddata	cubic	linear	2.6	5.6	7.5
griddata	cubic	nearest	2.4	5.2	7.3
griddata	v4	linear	2.5	5.4	7.2
griddata	v4	nearest	2.3	5.0	7.0
inpaint PDE	-	-	2.4	5.1	7.0
inpaint NaNs	-	-	2.4	5.0	6.7
inpaint DCT	-	-	2.4	5.2	7.0

Table C.10: 2D performance of the test dataset for $C^m = 10\mu T^2$ and $C^i = 15\text{deg}^2$.

Bibliography

- [1] Naoki Akai and Koichi Ozaki. “3D Magnetic Field Mapping in Large-Scale Indoor Environment Using Measurement Robot and Gaussian Processes”. In: *IEEE* september 18-21 (2017). DOI: [10.1109/IPIN.2017.8115960](https://doi.org/10.1109/IPIN.2017.8115960).
- [2] Binghao Li et al. “Using Geomagnetic Field for Indoor Positioning”. In: *Journal of Applied Geodesy* 27 november (2013). DOI: [10.1515/jag-2013-0016](https://doi.org/10.1515/jag-2013-0016).
- [3] Daniel Carrillo et al. “MagicFinger: 3D Magnetic Fingerprints for Indoor-Location”. In: *MDPI sensors* 15 july (2015). DOI: [10.3390/s150717168](https://doi.org/10.3390/s150717168).
- [4] David Hanley et al. “Experimental Evaluation of the Planar Assumption in Magnetic Positioning”. In: *IEEE* september 24-27 (2018). DOI: [10.1109/IPIN.2018.8533828](https://doi.org/10.1109/IPIN.2018.8533828).
- [5] German Mendoza-Silva et al. “A Meta-Review of Indoor Positioning Systems”. In: *MDPI sensors* 17 october (2019). DOI: [10.3390/s19204507](https://doi.org/10.3390/s19204507).
- [6] Hongwei Xie et al. “MaLoc: A Practical Magnetic Fingerprinting Approach to Indoor Localization using Smartphones”. In: *UBICOMP’14* september 13-17 (2014). DOI: [10.1145/2632048.2632057](https://doi.org/10.1145/2632048.2632057).
- [7] Imran Ashraf et al. “mPILOT - Magnetic Field Strength Based Pedestrian Indoor Localization”. In: *MDPI sensors* 14 july (2018). DOI: [10.3390/s18072283](https://doi.org/10.3390/s18072283).
- [8] Liu Yan X. et al. “Novel calibration algorithm for a three-axis strapdown magnetometer”. In: *Sensors* may (2014). DOI: [10.3390/s140508485](https://doi.org/10.3390/s140508485).
- [9] *ARCore overview; Google Developers*. URL: <https://developers.google.com/ar/discover>.
- [10] A. Ayanoglu, D. M. Schneider, and B. Eitel. “Crowdsourcing-Based Magnetic Map Generation for Indoor Localization”. In: *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Sept. 2018, pp. 1–8. DOI: [10.1109/IPIN.2018.8533832](https://doi.org/10.1109/IPIN.2018.8533832).
- [11] Marcelo Bertalmio, Guillermo Sapiro, and C. Ballester. “Image Inpainting”. In: *Proceedings of SIGGRAPH* (Jan. 2002). DOI: [10.1145/344779.344972](https://doi.org/10.1145/344779.344972).
- [12] *BMP380*. URL: <https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp380/>.

- [13] John D'errico. “inpaint nans”. In: *MATLAB Central File Exchange* Retrieved October 18 (2020). URL: https://www.mathworks.com/matlabcentral/fileexchange/4551-inpaint_nans.
- [14] EDN. *Air pressure sensors in smartphones: Transforming navigation and fitness tracking*. Apr. 2020. URL: <https://www.edn.com/air-pressure-sensors-in-smartphones-transforming-navigation-and-fitness-tracking/>.
- [15] Damien Garcia. “Inpaint over missing data in 1-D, 2-D, 3-D,... ND arrays”. In: *MATLAB Central File Exchange* Retrieved September 4 (2020). URL: <https://www.mathworks.com/matlabcentral/fileexchange/27994-inpaint-over-missing-data-in-1-d-2-d-3-d-nd-arrays>.
- [16] Damien Garcia. “Robust smoothing of gridded data in one and higher dimensions with missing values”. In: *Computational Statistics & Data Analysis* 54.4 (Apr. 2010), pp. 1167–1178. DOI: [10.1016/j.csda.2009.09.020](https://doi.org/10.1016/j.csda.2009.09.020). URL: <https://doi.org/10.1016%2Fj.csda.2009.09.020>.
- [17] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”. In: *IEEE Proceedings F - Radar and Signal Processing* 140.2 (1993), pp. 107–113. DOI: [10.1049/ip-f-2.1993.0015](https://doi.org/10.1049/ip-f-2.1993.0015).
- [18] Suining He and Kang G. Sun. “Geomagnetism for Smartphone-Based Indoor Localization: Challenges, Advances, and Comparisons”. In: *ACM Computing Surveys* Vol. 50, No. 6, Article 97 (2017). DOI: [10.1145/3139222](https://doi.org/10.1145/3139222).
- [19] The MathWorks Inc. *Interpolation for 3-d gridded data in meshgrid format - matlab interp3*. URL: <https://de.mathworks.com/help/matlab/ref/interp3.html>.
- [20] Tsz Ho Kwok and Charlie Wang. “Interactive Image Inpainting Using DCT Based Exemplar Matching”. In: vol. 5876. Nov. 2009, pp. 709–718. DOI: [10.1007/978-3-642-10520-3_67](https://doi.org/10.1007/978-3-642-10520-3_67).
- [21] *LPS22HB*. URL: <https://www.st.com/en/mems-and-sensors/lps22hb.html>.
- [22] Dimitrios Lymberopoulos and Jie Liu. “The Microsoft Indoor Localization Competition”. In: *IEEE Signal Processing Magazine* 6 september (2017). DOI: [10.1109/MSP.2017.2713817](https://doi.org/10.1109/MSP.2017.2713817).
- [23] MATLAB. *version 9.8.0 (R2020a)*. Natick, Massachusetts: The MathWorks Inc., 2020.
- [24] William Menke and Joshua E. Menke. *Environmental data analysis with Matlab*. Academic Press, 2016.
- [25] Carrie Perles. *The Causes of Barometric Pressure*. Mar. 2019. URL: <https://sciencing.com/causes-barometric-pressure-5101048.html>.

- [26] Margaret Rouse. *What is extrapolation and interpolation?* Sept. 2015. URL: <https://whatistechtarget.com/definition/extrapolation-and-interpolation>.
- [27] Y. Shu et al. “Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing”. In: *IEEE Journal on Selected Areas in Communications* 33.7 (2015), pp. 1443–1457. DOI: [10.1109/JSAC.2015.2430274](https://doi.org/10.1109/JSAC.2015.2430274).
- [28] T. Taketomi, H. Uchiyama, and S. Ikeda. “Visual SLAM algorithms: a survey from 2010 to 2016”. In: *IPSJ Transactions on Computer Vision and Applications* (June 2017). DOI: [10.1186/s41074-017-0027-2](https://doi.org/10.1186/s41074-017-0027-2).
- [29] G. A. Watson. “Computing Helmert transformations”. In: *Journal of Computational and Applied Mathematics* 197 (Dec. 2005), pp. 387–394. DOI: [10.1016/j.cam.2005.06.047](https://doi.org/10.1016/j.cam.2005.06.047).
- [30] Khalid Yousif, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. “An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics”. In: *Intelligent Industrial Systems* 1 (Nov. 2015). DOI: [10.1007/s40903-015-0032-7](https://doi.org/10.1007/s40903-015-0032-7).