On Indoor Localization and Indoor Mapping: A Deeper Look into Dynamic Mapping

Georgios Pipelidis

Fakultät für Informatik

Informatik 4 - Software und Systems Engineering

Technische Universität München

Dissertation

On Indoor Localization and Indoor Mapping: A Deeper Look into Dynamic Mapping

Georgios Pipelidis

Fakultät für Informatik

Technische Universität München

On Indoor Localization and Indoor Mapping: A Deeper Look into Dynamic Mapping

Georgios Pipelidis

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines *Doktors der Naturwissenschaften* (Dr. rer. nat.) genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Jörg Ott
Prüfer der Dissertation:	1. Priv.-Doz. Dr. Christian Prehofer
	2. Prof. Raja Sengupta

Die Dissertation wurde am 31.05.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 28.02.2022 angenommen.

Acknowledgement

"Is the circle of live
and it rules us all,
through despair and hope"

— Lion King
Elton John &
Tim Rice

First, I would like to thank my supervisor PD. Dr. Christian Prehofer, who believed on me and gave me this unique opportunity to pursue this Ph.D. His guidance was always of great importance to me and the quality of this work. Second, I would like to thank my wife Thu Kim Co, whose support during difficult days gave me the strength to continue pursuing this Ph.D. Third, I would like to thank my colleague Nikolaos Tsiamitros, without his dedication and ambition, the quality of this thesis would have been limited. I would also like to thank Prof. Dr. Manfred Broy who was always there for the last four years, whenever I needed him.

I would also like to thank my mother Eleni Pirpiri who taught me what empathy is. Additionally, I would like to thank my father who taught me the meaning of hard work and the rewards that comes through it. I should thank my sister, whose support during my life taught me to respect myself.

I would like to devote this Ph.D. to a person who is no longer with us, my cousin Eva, who taught me that there is much more to life than what our eyes can see.

I have to thank my cousin Rita who taught me to appreciate art and helped me to evolve as a human. I would like to thank my uncle Theologos who brought my first computer and inspired me to pursue this long journey. I would like to thank Dr. Ilias Gerostathopoulos who taught me how to write scientific documents and Tanmaya Mahapatra who pushed me every time I needed to advance.

Last but not least, I would like to thank everyone who contributed to this dissertation with feedback or even shared work, without them, this thesis would not have the value it has. Their names, with alphabetical order, are: Amy Yun, Anton Moritz Rohr, Christian Becker, Efdal Ustaoglu, Frederik Fraaz, Hasim Koc, Ioanis Passalidis, Jason Bouroutis, Kankamon Rujiranun, Kaushik Gopalan, Moawiah Assali, Mustafa Parlak, Nam Le Duc, Nithish Raghunandanan, Omid Reza Moslehi Rad, Pavlos Tzianos, Romeo Kinzler, Sahana Prasanth, Salma Farag, Shobhit Agrawal, Syed Adeebul Hafiz and Wasiq Kasam Rumaney.

Abstract

Location-Based Based Services (LBS) are widely used among smart device users. It has been estimated that LBS users are increasing exponentially with the accuracy of the services [KYL14a]. The first Geographic Information Systems (GIS) for example, were almost exclusively used by urban planners. When GPS satellites became available to the public, airlines had begun adopting them [PFL+95], while after digital map products emerged, navigating devices were widely adopted by millions of drivers.

Today the vast majority of smartphones are equipped with navigation or other location-based services and are daily used by hundreds of millions of users. Furthermore, the Internet of Things (IoT) is a rapidly developing field, and it is expected that soon billions of devices will be in daily use. Most of these devices is expected to be mobile, with high demand on localization services.

Unfortunately, indoor maps and indoor localization methods cannot be realized following scalable approaches such as satellite photography and Global Navigation Satellite System (GNSS) for two major reasons. The first reason is that indoor areas cannot be scalably photographed using satellites or airborne photography, and hence cannot be scalable mapped. The second reason is that GNSS satellites cannot be used for localization indoors since their signal cannot penetrate the solid objects of the structure [Hen12].

This thesis aims to explore, experiment, and propose enhancements, for technologies and methods that can boost the realization of precise and ubiquitous LBS. Additionally, this thesis explores alternative solutions for indoor navigation, localization, and mapping, some emerged purely via crowdsourced information. We implement and evaluate such systems, obtaining clear improvements. We build an enhanced LBS and evaluate it in challenging environments. We introduce a novel method for precise localization through efficient sensor fusion via particle filter. Finally, we used our developed method to generate radio maps used to provide precise localization and navigation.

Contents

List of Figures

1 Introduction

"In God we trust, the rest bring data"

— W. Edwards Deming

In the recent years devices, such as smartphones, equipped with sensors able of estimating the location of an entity (e.g. a human, an object, etc.), called Location Services (LS), are becoming pervasive (e.g. smartphones, wearables, etc.). LSs are commonly used for adding value to the functionality of existing tools, such as navigation or recommendation engines. These services are called Location-Based Services (LBS) [Kue05]. A LBS implies the existence of localization technology (i.e. LS) and a map. A map can be defined as a model that describes the geometry, the topology and semantic information of a place [A+05].

Even though people spend approximately 80% of their time indoors [bui04], [KNO+01], LBSs are mostly developed for the outdoor world, where localization and mapping problems have largely been addressed. Unfortunately, the same does not apply to indoor environments. Global Navigation Satellite System (GNSS) cannot work indoors, since their signals cannot penetrate solid objects, such as building walls. Additionally, most of the indoor places lack digital indoor maps. Understanding the indoor environments is therefore of great importance.

Furthermore, it has been estimated [LL10] that when the accuracy of LBS systems increase, the number of users increases exponentially. Hence, methods for generating, integrating and enhancing indoor maps need to be researched (i.e. [EY15]), in order to cover the increasing demand. Additionally, modeling semantic information for indoor places (i.e. information that can be used for localization or navigation) is equally important to modeling the geometric and topological properties of a place [ARC10]. Finally, Indoor LBS need to be constantly enhanced for allowing future technologies, such as the Internet of Things (IoT) or Augmented Reality (AR), to be fully exploited.

1.1 Overview and Challenges of Indoor Location Based Services

Open challenges and drawbacks of indoor LBSs can be organized into three categories, referring to **indoor localization challenges**, **indoor mapping challenges** and **indoor spatial information modeling challenges**.

Indoor Localization Techniques, Overview and Challenges.

The most critical drawback of indoor localization is the lack of a positioning technology, such as GPS at the outdoor world. From the various existing technologies, provided for localization in the indoor world, every technology has its benefits and drawbacks. Consider the most prevailing technologies:

1. ***BLE Beacons on Indoor Localization:*** BLE stands for Bluetooth Low Energy and is part of Bluetooth 4.0 [29], [NO20]. Low energy protocol allows Beacons to work on a single battery for a long time. The spectrum of the signal ranges between (2.4GHz - 2.4835GHz) and it can be detected in up to 100m. The difference to Bluetooth is that it has lower transfer rates. Such dedicate hardware is a resource-demanding technology, since BLE beacons have to be densely installed. As a result, large organizations avoid to install them in large building (i.e. airports [Ber]), while BLE beacons require complicated installations that demand smartphone applications. BLE beacons are mostly operating with batteries and as a result they are an energy-constrained technology. Their clocks are impossible to be synchronized [WCS10], in at least on nanosecond precision, which makes precise localization impossible.

2. ***UWB on Indoor Localization:*** Various methods of localization exist based on Ultra-Wideband technologies [GTG+05]. Similar to WiFi localization techniques, localization can be achieved: (1) by the angle of arrival (AOA), which measures the angles between a given node and a number of reference nodes to estimate the location. (2) The signal strength (SS), where the distance between nodes is estimated by computing the energy intensity of the signal. (3) Time delay information estimates the distance between nodes by estimating the travel time of the received signal. UWB share some of the advantages and disadvantages of BLE beacons, for example there is no clear economic driving force for it. Their cost is even higher than BLE and their adoption rate is even lower since they cannot be supported by smartphones to this day. UWB cannot co-existence with other radio-based technologies, due to interference [HHT+02].

3. ***Magnetic field-based localization:*** This technique, the location is estimated based on disturbances of the earth's magnetic field caused by structural steel elements in a building [CDS+11]. Its unique characteristics are that spatially it varies but it is a permanent characteristic of space. For accurate mapping and localization, a 3D axis electronic compass equipped with an internal tilt-compensated algorithm to measure the heading of the sensor, can be used. This technology requires permanent structures in a building (i.e. walls) reach in structural steel elements, that will vary on the steel content and structure. Usually, this is not the case. Additionally, the disturbances tent to occur near walls which disables the technique from operating in large indoor areas like big halls etc. Additionally, existing techniques for mapping magnetic field landmarks are not robust against user orientation and velocity.

4. ***WiFi based localization:*** There are several approaches for indoor localization based on WiFi, among the most popular are: (1) Based on proximity sensing [KKB14], this demands a database of station IDs and their geolocation, then the position is determined by measuring the RSS. (2) Trilateration, the distance is cal-

culated from the station to a device. With more than one stations the device position can be approximately estimated. Several methods for trilateration exist, some of them are (a) based on Time of Arrival (ToA) [LCBA08], it estimates the distance based on the Round Trip Time (RTT) of a message. (b) Time Difference of Arrival (TDoA) [NB08], it uses the difference between the arrival times of the signals to determine the position. (c) RSS [FAVT12], uses propagation-loss of the WiFi signals to compute the distance. (3) Another method for localization is based on triangulation or Angle of Arrival (AoA) [RM09], where the distance is trigonometrically estimated but special antennas are mandatory for this approach. (4) Another popular approach for indoor localization based on WiFi is by wave propagation estimation based on Friis formula, where the received and transmit power need to be known, as well as the signal wavelength, and then the distance can be derived by the Friis formula similar to [CLCS10b]. (5) Finally, localization can also be done by pattern recognition and fingerprinting methods. The advantage of such methods is that they are infrastructure agnostic and hence, they can easily be crowdsourced. Such technologies seem to be ubiquitous, but they work only under specific circumstances.

Algorithms that use trilateration for positioning – where the distance to the target is being estimated based on RF propagation time – presume the synchronization of the clocks of the access points (AP). Keeping the AP clocks synchronized is a challenge due to high clock crystal oscillations or low transmission bandwidth for device-to-device synchronization. Algorithms that use angle of arrival require optimized antennas for localization and cannot be used with existing smartphones due to limitations on clock synchronization. Algorithms that used received signal strength for localization, can be influenced by the presence of people, since the microwave frequency used in WLAN can be absorbed by the human body. Additionally, signals are constantly reflected caused the multipath [ZD02] problem where the correct distance estimation is impossible to be estimated. Additionally, mapping areas with their unique characteristics commonly called "fingerprints" is a resource-intensive procedure, and often suffers from heterogeneity due to the variations of different WiFi antennas on smartphones.

5. ***Computer Vision-Based Localization:*** Localization via computer vision techniques [GZY+14, ZXQ+20, LCS11, Zha04, BBM87, Can86, C62] works as follows: After images are captured, from a camera for which the intrinsic and extrinsic parameters [Zha04] are known and unique features have been extracted (e.g. BRISK features [LCS11]), using a Structure from Motion (SfM) algorithm [BBM87], a 3D point cloud of the building can be extracted. Finally, by applying edge detection algorithm (i.e. Cuny algorithm [Can86]), shape recognition algorithm (i.e. Hough Transform [C62]) and segmenting the results, by grouping parallel lines into groups and rejecting lines of no geometric importance, the landmark contours can be identified. Localization can be also performed using Depth Sensors. It works as follows: An infrared projector projects a unique pattern (i.e. a speckle pattern [Dai13]). An infrared sensor, whose relative distance to the projector and rotation is known, recognizes these markers. A depth map is constructed by analyzing the unique pattern of infrared light markers by triangulating the distance between the sensor the projector and the object. The technique of analyzing a known pattern is called structured light (project a known pattern onto the scene and infer depth from the

deformation of that pattern). Finally, combining structured light with computer vision techniques, for example, depth from focus (uses the principle that objects that is blurrier is further away) and depth from stereo (objects gets shifted more when are close and the scene is in angle than objects that is far away), the depth of different areas can be estimated. Computer vision-based approaches have the highest power consumption, higher processing demand, higher response time for localization queries, high user involvement, by requesting from users to capture multiple photos of an area per time – hence low user experience –, while it requires high upfront investment since photos of the entire area have to be captured with specialized equipment and be updated every time changes occur.

6. **IMU-based localization**: IMU based localization (e.g. Dead Reckoning) is a promising but challenging solution since its error is cumulative and very soon it emerges to the hundreds of meters. However, often IMU data is fused with one or more of the above solutions. An additional problem is that the commercial off-the-shelf sensors, which equip smartphones, are very noisy and as a result produce highly uncertain readings. Additionally, these approaches are computationally expensive and as a result they are often executed on the cloud increasing costs and localization intervals. Finally, they often require well detailed indoor maps which are time consuming and expensive to obtain.

7. **Vertical localization :** is an open challenge [BOBZ15]. All the above approaches have been mainly focusing on planar localization, and very few have ever tested in a multi-storey environment. The problem with vertical localization is that existing methods cannot easily distinguish between floors, while phones equipped with barometric sensors cannot utilize them for vertical localization due to the fact that atmospheric pressure highly depends on temperature, humidity, and other environmental constraints. As a result, for estimating altitude, reference barometric, temperature and humidity sensors are required, while calibration between the sensors is essential.

Indoor Mapping, Overview and Challenges

Indoor localization and navigation, in most cases, require indoor maps. Indoor maps indicates the existence of models that describe geometry of places and objects, topological relationships between places (i.e. adjacency and connectivity) and semantic annotation of spaces (i.e. the way that the place is used (e.g. stairs, elevator, etc.) and unique identifiers of the place (e.g. the received signal strength in a room from multiple APs) that – when mapped – can be used for localization, these maps commonly called 'radio maps'). Challenges here can be summarized as follows:

1. **Scalability:** Generating indoor maps is a demanding procedure since there is no airborne or satellite imagery available to support this procedure, as it has happened with the outdoor maps. This increases manpower demands and as a result costs.

2. **Temporality:** Indoor environment characteristics that can be used as landmarks to support the navigation, and even the localization procedure, are never static (i.e.

objects displaced etc.), which soon results to outdated indoor maps, while their maintenance effort increases their overall cost.

3. **Legality:** Legal challenges are often the case since it is common for indoor areas to be privately owned [ZSN+13].

4. **3D-Mapping:** Mapping the vertical dimension of buildings, while including storey altitude, or storey height from the reference area, is an open challenge.

5. **Topological Mapping:** The last mapping challenge emerges from the hypothesis that not all areas are equally visited and transition between areas is governed via constraints, as it has been also assumed by [RAK14].

Overview and Challenges of Modeling Indoor Areas

1. **Storage overhead:** Describing indoor maps requires a tremendous amount of data, considering the fact that recently, only the building footprints in OSM surpassed the amount of data of streets [GZ12]. Additionally, there is not a well agreed-upon model for describing indoor areas [GZ11]. Filtering outliers, extraction of topological information from spatial information and enhancement of existing models with semantic information, are technologies under research.

2. **Heterogeneity:** Additionally, since the process of mapping is often crowdsourced, there is a need for mechanisms that manage the heterogeneity of various sources and can integrate different inputs for the same floorplans.

3. **Semantics Mapping:** Furthermore, indoor localization cannot use the maps without them being semantically enhanced with uniquely identified locations.

4. **Other Challenges:** Finally, modeling the accuracy of indoor localization method (provide the correct position), availability (provide results within a constrained time limit), stability (provide consistent results) and ambiguity (provide uncertainty of the results) remain open challenges. Last but not least, there is not an explicitly defined taxonomy of indoor environments.

1.2 Research Goals

As, a response to selected challenges presented in Section 1.1 *"Overview and Challenges of Indoor Location Based Services"*, this thesis focuses on mapping indoor areas by empowering the crowd. To design solutions and processes to address these challenges, we introduce a method based on customized, crowdsourced and scalable solutions that may be followed by communities to dynamic map indoor areas in different stages, combining multiple indoor mapping generation techniques whenever necessary.

Since the area is broad, the thesis primarily focuses on clarifying existing attempts and

state-of-the-art and proposes methods that can provide essential information that will bootstrap solutions for mapping, localizing and describing indoor areas. The thesis thus targets the following research goals:

1. The first goal is to investigate the **"Vertical localization"** and **"3D mapping"** challenges, (1.1.7 and 1.1.4 respectively), while diving into the **"Indoor Mapping"** challenge 1.1. In particular, we examine a simplified mapping approach, where data collected by user devices can be used to precisely map the vertical dimension of buildings, and where this information can be described by existing standard models for indoor mapping or they will need to be extended.

2. Our second goal is to investigate if semantic characteristics of indoor areas could be identified and be mapped, which will enable us to answer the **"Temporarity"** challenge 1.1.2.

3. Our third goal is to investigate potential solutions of the **"IMU based-localization"** challenge (1.1.6) and hence develop a simultaneously localization and mapping, and investigate whether the use of additional information might improve the localization effort or accuracy.

4. Our fourth goal is to dive deeper into **"WiFi based localization"** and **"Heterogeneity"** challenges (1.1.4 and 1.1.2 respectively), and examine whether *"radio-maps"* could be obtained via crowdsourced approaches and if the heterogeneity problem could be addressed.

1.3 Contribution and Publications

The main contribution presented in this thesis consists of a commented collection of publications. Most of the results presented in these publications stem from research work and collaboration within the TUM Living Lab Connected Mobility project, in which the author participated. And has been funded by the Bavarian State Ministry for Economic Affairs and Media, Energy, and Technology.

The following peer-reviewed papers from the core contribution presented in this thesis. An overview of the contribution is presented in Chterapter 2, while the summaries and full texts of these publications are included in Sections 3.1-3.5.

1. Georgios Pipelidis, Christian Prehofer and Gerostathopoulos, Ilias. *"Bootstrapping the Dynamic Generation of Indoor Maps with Crowdsourced Smartphone Sensor Data."* Geographical Information Systems Theory, Applications and Management", 2019, Springer International Publishing, pages:70–84. DOI: 10.1007/978-3-030-06010-7_5

2. Georgios Pipelidis, Omid Reza Moslehi Rad, Dorota Iwaszczuk, Christian Prehofer, and Urs Hugentobler. *"Dynamic Vertical Mapping with Crowdsourced Smartphone Sensor Data."* MDPI, Sensors, Vol.: 18, Year: 2018, Article-Number:480 2018. DOI: 10.3390/s18020480

3. Georios Pipelidis, Frederic Fraaz, Christian Prehofer. *"Extracting Semantics of Indoor Places based on Context Recognition."* 2018 IEEE International Conference on Pervasive Computing and Communications 2018. DOI: 10.1109/PERCOMW.2018.8480187

4. Georgios Pipelidis, Nikolaos Tsiamitros, Petteri Nurmi, Efdal Ustaoglu, Romeo Kienzler, Huber Flores, Christian Prehofer. *"Cross-Device Radio Map Generation via Crowdsourcing"*. IEEE, International conference on Indoor Positioning and Indoor Navigation (IPIN). 2019. DOI: 10.1109/IPIN.2019.8911766

5. G. Pipelidis, N. Tsiamitros, C. Gentner, D. B. Ahmed and C. Prehofer. *"A Novel Lightweight Particle Filter for Indoor Localization."* 2019, IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN), Pisa, Italy, 2019, pp. 1-8. DOI: 10.1109/IPIN.2019.8911744

Main contributions of this thesis were also included in the following peer-reviewed publications, either as an extension of the work or as alternative use cases of the main components developed during this thesis:

1. A comprehensive state-of-the-art review is presented in: Georgios Pipelidis, Christian Prehofer. *"Models and Tools for Indoor Maps."* MediaTUM, Digital Mobility Platforms and Ecosystems, 154 2016. DOI: 10.14459/2016md1324021

2. A high level idea of this thesis is presented in: Georgios Pipelidis, Xiang Su, Christian Prehofer. *"Generation of indoor navigable maps with crowdsourcing."* ACM Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia, Pages: 385-387 2016. DOI: 10.1145/3012709.3018007

3. Our position on the crowdsourcing of indoor maps is presented in: Georgios Pipelidis, Christian Prehofer, Ilias Gerostathopoulos. *"Adaptive Bootstrapping for Crowdsourced Indoor Maps."*, SitePress International Conference on Geographical Information Systems Theory, Applications and Management 2017. https://doi.org/10.5220/0006369302840289

4. An earlier approach of the vertical mapping was presented in: Georgios Pipelidis, Omid Reza Moslehi Rad, Dorota Iwaszczuk, Christian Prehofer and Urs Hugentobler. *"A novel approach for dynamic vertical indoor mapping through crowd-sourced smartphone sensor data."* IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN) 2017. DOI: 10.1109/IPIN.2017.8115902

5. An extensive evaluation of our developed localization component is presented in: V. Renaudin and M. Ortiz and J. Perul and J. Torres-Sospedra and A. R. JimÃ©nez and A. Perez-Navarro and G. MartÃn Mendoza-Silva and F. Seco and Y. Landau and R. Marbel and B. Ben-Moshe and X. Zheng and F. Ye and J. Kuang and Y. Li and X. Niu and V. Landa and S. Hacohen and N. Shvalb and C. Lu and H. Uchiyama and D. Thomas and A. Shimada and R. Taniguchi and Z. Ding and F. Xu and N. Kronenwett and B. Vladimirov and S. Lee and E. Cho and S. Jun and C. Lee and S. Park and Y. Lee and J. Rew and C. Park and H. Jeong and J. Han and K. Lee and W. Zhang and X. Li and D. Wei and Y. Zhang and S. Y. Park and C. G. Park and S. Knauth and G.

Pipelidis and N. Tsiamitros and T. Lungenstrass and J. P. Morales and J. Trogh and D. Plets and M. Opiela and S. Fang and Y. Tsao and Y. Chien and S. Yang and S. Ye and M. U. Ali and S. Hur and Y. Park. *"Evaluating Indoor Positioning Systems in a Shopping Mall: The Lessons Learned From the IPIN 2018 Competition."* IEEE Access, Pages: 148594-148628, Vol.: 7, Year: 2019, DOI: 10.1109/ACCESS.2019.2944389, ISSN: 2169-3536s

6. A method for the quantification of the quality of indoor maps in presented in: Assali, M., Pipelidis, G., Podolskiy, V., Iwaszczuk, D., Heinen, L., Gerndt, M. (2019). Quantifying The Quelity of Iindoor Maps. ISPRS, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. DOI: 10.5194/isprs-archives-XLII-2-W13-739-2019

7. A use case of our developed tracking method is presented in: G. Pipelidis, N. Tsiamitros, M. Kessner and C. Prehofer, *"HuMAn: Human Movement Analytics via WiFi Probes"*, 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kyoto, Japan, 2019, pp. 370-372. DOI: 10.1109/PERCOMW.2019.8730703

2 Overview of the Contributions and Evaluation Methodology

"In theory, theory and practice are the same, in practice they are not."

— Albert Einstein

This chapter describes the entire thesis as one cohesive story. It highlights the key results of the thesis and provides a detailed overview of the contribution. Additionally, it presents the methodology followed to evaluate each presented component.

2.1 High Level Overview

From an autonomous vacuum cleaner to an autonomous car, Simultaneously Localization And Mapping (SLAM) [CCC+16] is the algorithm used for localization. SLAM is the most common approach followed to address a localization problem. The advantage of SLAM is that it addresses the problems of localization and mapping as one [BD06, DB06, CCC+16, DWB06], and in particular the one process supports the other. Its main contribution is that it uses the correlations between mapped landmarks for reducing the localization error (Figure 2.1). As a result, an "entity" can construct a map of the environment (from landmarks) and use this map to deduce its location [CKKC15].

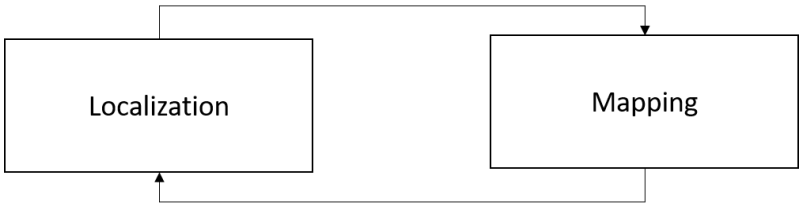

Figure 2.1: The two main components of a SLAM algorithm. Inspired by [CCC+16]

SLAM algorithms consist of two major components, the localization and the mapping components, as depicted in Figure 2.1 The localization component is often a kinematic model, the role of the mapping component is to detect loop closures able to support the reset of the often cumulative localization error. Loop closures occur when the same location is being revisited by the tracking object – or objects – and they can enhance the localization accuracy by assisting the precise mapping of areas.

Assuming that we are tasked with tracking an object's location relative to the landmarks of its environment. In a specific times, the location and the orientation of the object can be retrieved by computing the posterior probability of observed features such as the current location and orientation of the surrounding landmarks, the set of its prior actions, and its previous location and orientation. Hence, the current position of an object can be computed using probabilistic approaches (i.e.Bayes Theorem), if a state transition model and an observation model are defined. The state transition model is usually assumed to be a Markov process since the next state depends only on the previous state (Markov property) and the directions are independent of the observations and the map (Time invariance). The observation model describes the probability to observe a landmark when the location, orientation, and other landmark locations are known.

Since this problem can be formulated in a probabilistic way, there is a need for representing of the observation and the motion models in such a way that it will enable efficient and consistent computation of its prior and the probabilities of the posterior distribution. There are two popular computational solutions for this problem, he extended Kalman filter (EKF-SLAM) [EW99] and the use of Rao-Blackwellised particle filters (FastSLAM) [DM96].

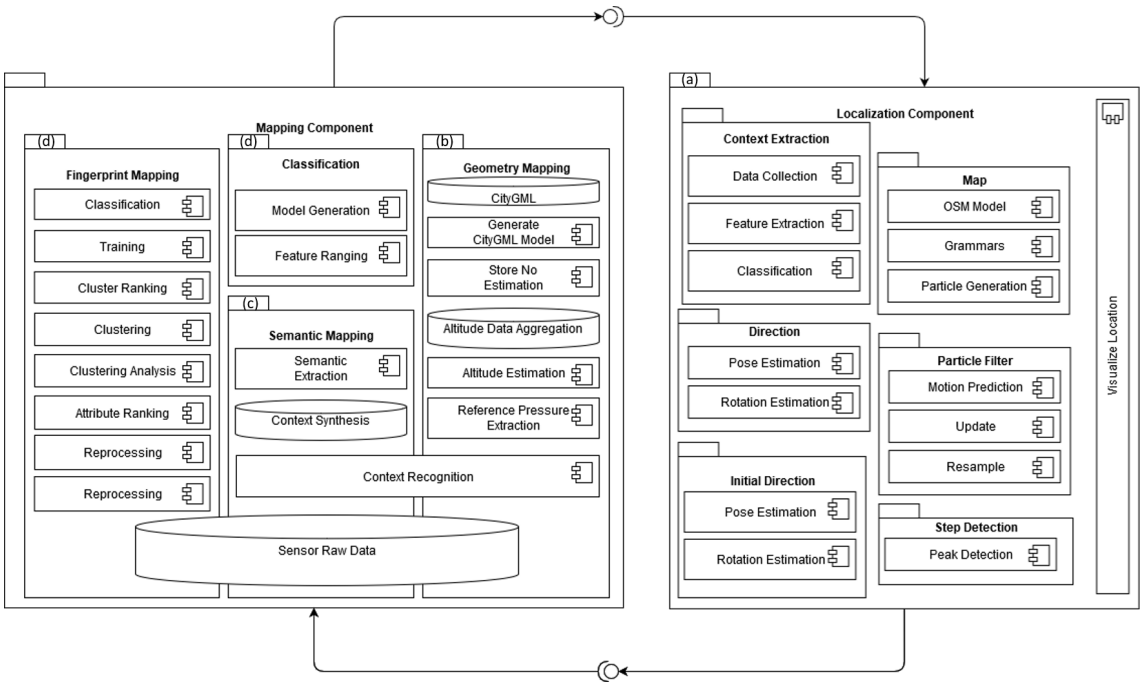

Figure 2.2: The high level architecture of the system conceived, developed and evaluated during this thesis. This is an extension of Figure 2.1. Where (a) is presented in Section 3.4, (b) is presented in Section 3.2, (c) is presented in section 3.3 and the two components that assemble (d) are presented in 3.5

The approach that has been proposed, implemented and extensively evaluated during this thesis, operates under the particle filter idea, and it extends between two major components, as presented in Figure 2.2. The first component we will refer to as the "Localization Component", since it mostly focuses on the localization problem and it operates in an off-the-shelf smartphone. The second component, which we will refer to as the "Mapping Component", since it mostly focuses on the mapping problem and operates on the cloud. Functionalities for localization and mapping are shared between both compo-

nents. In particular each component consists of multiple sub-components, responsible for tasks such as identifying the movement direction or the users context that can contribute in both, localizing and mapping. In this section we will delve deeper to the high level architecture and the sub-components that embodies it.

2.1.1 The Localization Component

The term "location" [Kue05] is associated with a certain place in the real world. Location denotes a place of an object in the real world, and hence this kind of location belongs to the class of physical locations. The cyberspace Internet has brought another concept of the location where virtual meetings take place (e.g. a distributed computer game). This is called a virtual location. LBSs predominantly refer to physical locations, with an exception of augmented reality.

Physical locations can be further broken down into three subcategories: (1) Descriptive locations: natural geographic objects (2) Spatial locations: a single point in the Euclidean space (position) expressed by coordinates. (3) Network locations: the topology of a communications network. The target persons of an LBS can be pinpointed by all these location description models. Spatial location or position information represents an appropriate means for exactly pinpointing an object on Earth.

A LBS needs to map between different location categories. For example, distance calculations can only be done by descriptive locations, while routing can be expressed better by descriptive locations. For expressing spatial locations, it is necessary to use: (1) a coordinate system, (2) a datum, or else a frame that can enable the coordinate system to describe positions in the three-dimensional space, and (3) a projection (i.e. on a map). Coordinate systems used for describing locations are the Cartesian and ellipsoidal. The Cartesian describes a location by specifying its distances to predefined axes. The ellipsoidal describes a location by its angles to an equatorial and polar plane. A datum defines the size and shape of the Earth as well as the origin and orientation of the coordinate system that is used to reference a certain position.

The localization component developed during this thesis is designed to operate in a smartphone, and considers the limited processing capacity of a smartphone, as well as the limitations that high battery consumption may cause. It consists of various subcomponents all optimized to maximize accuracy, while minimizing energy consumption and processing capacity. Diving deeper in the localization component, its operations are responsible to perform localization using the IMU sensors of the smartphone, while fusing this information with the geometry, topology, semantics of a building, as well as user's context.

Context Extraction

Contact can be any information that can characterize the situation of an entity [SAW94]. An abstraction of the context recognition module can be presented through a Venn Diagram, as depicted in Figure 2.3. Four outcomes can be imagined, when attempting to

monitor movement from two sensors. The sensors to be monitored are the acceleration and atmospheric pressure, or barometer, sensors and their outcomes can be described following a binary approach, as "recording disturbances" or "not recording disturbances". An accelerometer monitors disturbances in movement while the barometer monitors disturbances on atmospheric pressure. Since earths atmospheric pressure varies depending on altitude, barometric sensor can be used for monitoring vertical movements, such as floor transitions. Hence, pressure disturbances can be interpreted as vertical disturbances.

From the above, four outcomes are possible to perceive, with four possible interpretations. Consequently, when high disturbances on acceleration are perceived, and high disturbances on pressure, it can be interpreted as the "climbing stairs" activity, while when there are no pressure disturbances, it can be interpreted as the walking activity. Additionally, when there are no disturbances observed in the acceleration, but there are disturbances on pressure, then it can be interpreted as the "using elevator" activity, while when no disturbances on pressure are observed, it can be interpreted as a stationary activity such as "sitting" or "standing".

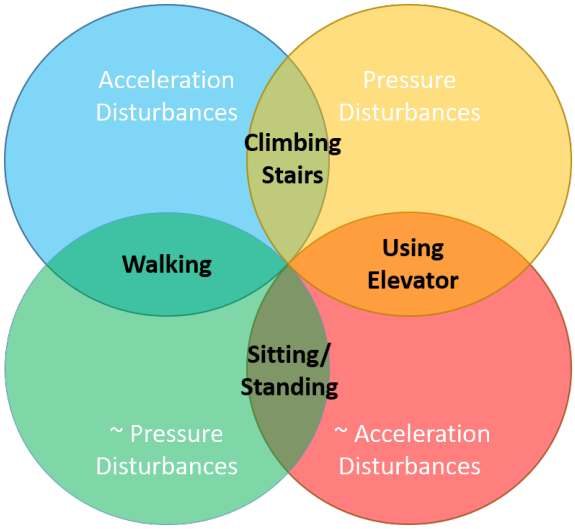

Figure 2.3: A Venn diagrams that illustrates the reasoning behind the context recognition subcomponent.

Certainly the possibilities do not end with these four states, instead we can monitor the pressure derivative, which if positive it can interpret as downward movement, and upward when positive. Additionally, depending on the phones pose, (e.g. phone in trouser pocket), a distinction between "sitting" and "standing" activities can be perceived, by monitoring the direction of the earth's acceleration. Finally, we should clarify that in practice the above assumptions are more complicated to implemented since both sensors produce floating point measurements that carry considerable noise. Therefore, this task can better achieved with the use of machine learning models able to handle this uncertainty.

Direction Estimation

For obtaining reliable direction estimation the phone pose is essential to be acquired. The phone pose can be calculated by orchestrating three key sensors. The first and most important is the gyroscope sensor, the second is the acceleration sensor and last is the magnetic sensor. Often, the phone pose can be estimated as a classification problem, hypothesizing that during a gait cycle the maximum acceleration occurs towards the walking direction, while at the same time there is a minimum in the lateral acceleration [HGTH14], [CR15], [PPC+12]. Attitude can be expressed with three mathematical representations. Euler angles, rotation matrix or quaternions. Although the phone can be freely moved, the gravity direction is always static towards the earths core, as can be seen in the illustrated example of Figure 2.4.

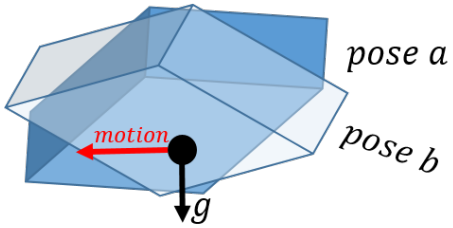

Figure 2.4: An example of two different phone poses.

After the gravity direction is being identified, with the help of the acceleration sensor, the search of the direction of motion is limited to a planar surface, instead of the 3D environment, and can be identified by monitoring the perpendicular motions, using the acceleration sensor, which are acting on the phone, and are caused due to the human acceleration. The next and final step is to translate the user direction from relative orientation of a user's coordinate system to a the global, or reference, coordinate system, which is the earth's coordinate system. The earth's coordinate axes are aligned towards the north and south pole of the earth and it is perpendicular to the direction of gravity.

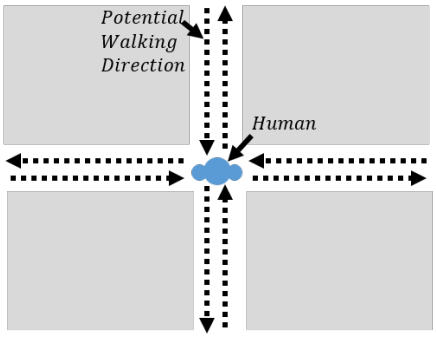

Figure 2.5: An example of topological influence on possible direction.

For translating the user's orientation in the global coordinate system a local coordinate system that identifies the direction of the phone is used, as a first step. Ideally, this could be achieved by translating the local direction, sensed by the accelerometer and gyroscope to magnetometer, which is responsible to detect the flowing direction of earth's magnetic field. Unfortunately, the magnetometer measurements are highly disturbed [RAL] by

steel elements inside the walls of the infrastructure. As a result, at this step we deduce the walking direction with the help of the geometry of a building.

During our study, we followed a different approach. Although humans may potentially walk towards any direction, often, our walking direction is parallel to directions dictated by the topological characteristics of a building. For example, we are mostly walking parallel to the topological direction in a corridor, while even when we walk perpendicular, let's say because we wish to enter in a room, we will only walk for a very short time. Hence, we can deduct the walking direction for the majority of the cases, to just a few. An example is visualized in Figure 2.5, where the topology limits potential walking directions of the user to set of only four potential directions.

Step Detection

The step detection module is responsible for identifying steps while being performed by a person [SRL13]. In order to perform step detection, being aware of the phone's pose is essential, and it is performed as described in the direction estimation module. The difference in this module is the fact that we select the axis of motion parallel to earth's gravity axis.

Considering the fact that bipedal movement is a periodic task, patterns are expected to emerge, when monitoring signal sensed by the accelerometer. In particular, as can be seen in Figure 2.6, the moment that the main heel strikes the ground a peak on acceleration axis parallel to earth's axis is observed.

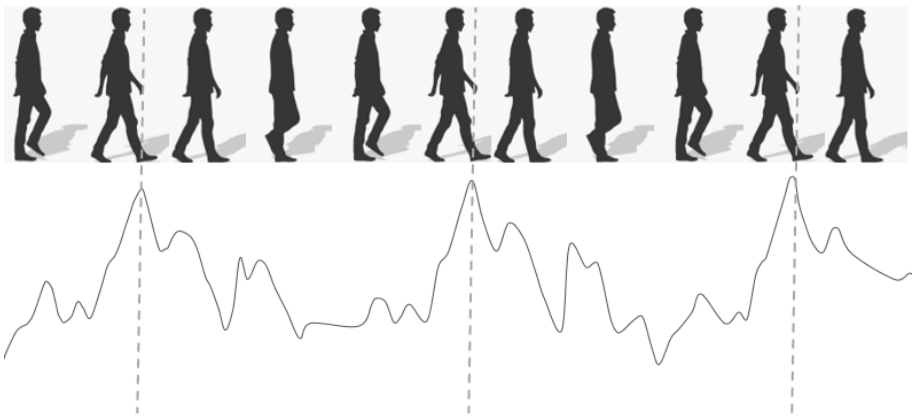

Figure 2.6: An example of bibedal movement and its pattern, as observed by the acceleration sensor.

This peak can be detected via pattern recognition since its entire period often follows similar pattern. As can be seen in Figure 2.6 a gait cycle can be divide in up to eight phases, and it can broken down to patterns such as (a) rapid increase, (b) slight decrease, (c) stabilization, etc. These fluctuations can be easily captured by monitoring the trend of the the graph (e.g. by monitoring its derivative), while a state machine can be used to assign the likelihood of being a signal produced during a gait cycle.

Furthermore, the length of the signal can give information about the duration of a step,

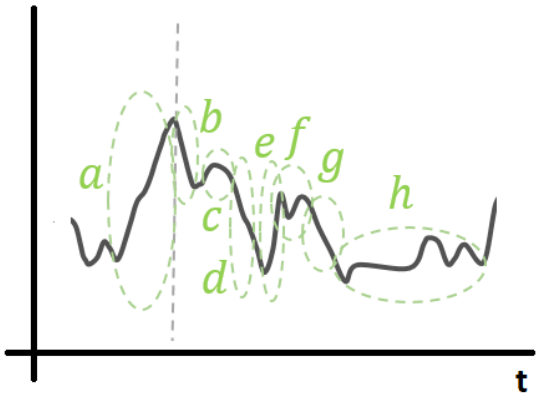

Figure 2.7: An example of a period of gait

which its magnitude can provide information about the intension or velocity of the step. Hence, if accurate estimation of the displacement of person and its velocity can be obtain, even with a single off-the-shelf accelerometer [RSL12].

Map

A map is a data model that aims to represent elements of the physical world, such as buildings, roads, corridors, etc. This data can be dynamically retrieved via remote or local databases. A map can be used to identify the elements that exist within an area of the physical world as well as their geometry. These elements may be used for location identification or simply for routing. Examples of such elements are: stairs, corridors, pathways, walls, rooms, elevators, etc.

When using lightweight maps [EKS13] the aforementioned elements are not complete, as a result, we came up with an additional module that can enhance their representation. We call this module "grammars" and an example of its function is to enhance the points in a topology object, i.e. topology usual is described through a graph, in which the node distance depends on the plurality of intersections, meaning, the more the intersections, the denser the nodes, hence the smaller their distances.

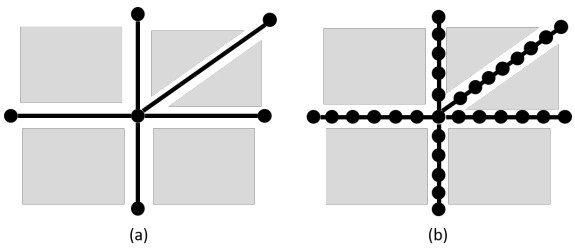

Figure 2.8: A topology example before (a) and after (b) the execution of our grammars module.

On the other hand the localization algorithm and in particular the particle filter updating component can be highly benefited from obtaining a much denser topological network. This is one of the responsibility of the grammars module. In particular, this module will

parse and interpret data queried by Open Street Maps, and its output will be used to augment the localization procedure.

Particle Filter

Particle filter [DM97] is an efficient filtering method, especially when multi-sensor information is being fused. It works based on a set of Monte Carlo simulations that operate sequentially, and helps to improve the quality of the signal. In more details, a particle filter operates by generating a series of particles often following a normal distribution. When the initial location is unknown, particle filters may be generated even uniformly along walkable areas.

Particle filters often operate in two phases. The novelty of the particle filter that was implemented during this thesis is that it operates in three phases, which enabled us to include additional mechanisms for better quantification of the uncertainty the particles. In particular, the context recognition contributed on vertical transition discovery, while the direction estimation and its constant comparison with the topology of the building enabled us to correct the error, when direction was misidentified. The first phase is the motion prediction phase. In this phase Pedestrian Dead Reckoning (PDR) is used for obtaining an estimation of the user's location at any moment. PDR enables localization only based on IMU sensors and estimates the user motion in the context of the offset when compared to her previous location. As can be seen in Figure 2.9, where three consecutive steps are visualized.

The idea of dead reckoning, also known as a deduced reckoning, is that the current location can be estimated based on the previous location, the distance traveled and the direction of motion. Today dead reckoning can be applied to pedestrian data collected by smartphones [KK14], although there are still several challenges.

Unfortunately, PDR tends to accumulate error in both coefficients, direction (θ_n) and length of displacement (δ). As a result, unfortunately, PDR alone is not an ideal mechanism for localization.

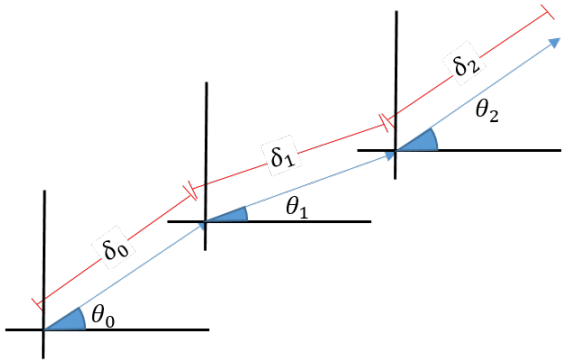

Figure 2.9: An example of a dead reckoning.

The second phase of our particle filter is the update phase, as presented in Figure 2.10. In this phase, weights are assigned to all particles according to a series of criteria. These criteria are: (a) their distance to the walls of the structure, (b) the similarity between

their direction and the user's walking direction, (c) the user's activity (e.g. stairs, etc.) and (d) if a complementary localization method is available (e.g. WiFi or BLE based) the similarity of features extracted between the two is also taken into consideration.

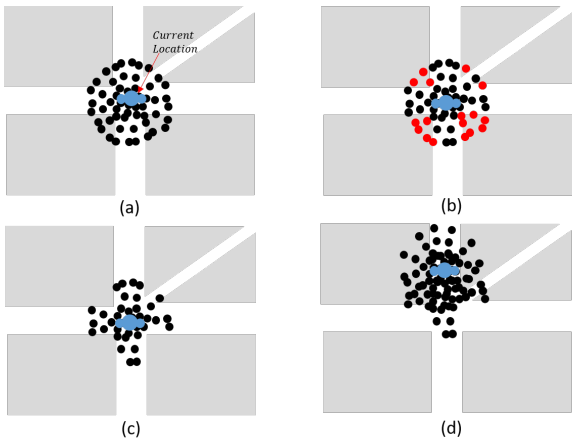

Figure 2.10: An illustrated example of the phases of a particle filter.

The third phase of our particle filter is the resample phase. In this phase particles that have been generated following a normal distribution around the estimated user's position, and inside the estimated uncertainty radius. The initial estimation of the users location is being computed on the server side, and it will be discussed in a later section. Each particle's weight is assigned in the update phase.

2.1.2 The Mapping Component

The main role of the mapping component is to support the mapping of our SLAM approach. The map is constantly updated with context extracted by the localizations component, which later support the localization component through loop closure identification, which has as a result the constant improvement of the localization accuracy. The mapping component consist of four smaller components, each with a unique task, and being responsible for a unique aspect of our mapping problem.

More precisely, the **Geometry Mapping** component, focuses on extracting geometric characteristics of a building, such as the height of every floor or the height and location of entrances in the building. The **Semantic Mapping** attempts to identify the function of locations inside buildings. For example stairs, elevators, corridors, etc. Furthermore, the **Fingerprint Mapping** component, aims to identify unique "landmarks" in the area that can be used later for localization. Such landmarks may be extracted based on WiFi properties (i.e. WiFi AP visibility and RSSI for each AP) in particular areas. The **Classification** Component is developed with all other components in mind and its function is to provide localization based on particular features as monitored in the environment.

Geometry Mapping

The Geometry Mapping component attempts to identify geometric characteristics in a building, and map them following existing standard models, inspired by [BAOBZ15]. A relatively easy geometric characteristic to be mapped is the floor height of every building, which is essential for the navigation indoors. Provided that all modern smartphones are equipped with barometric sensors, estimating height should not be a challenge. However, estimating altitude with a barometer requires a reference pressure and temperature, that can be extracted either by a reference barometer or by a reference location.

Reference Pressure Extraction In this work, we decided to use the transition between outdoors and indoors as a reference point. Many approaches have been suggested for outdoor to indoor transition identification, and vice versa. A simple technique has been suggested by [WSE+12] and [RNBM11], where the drop of confidence or inability of GPS is obtained as an indication of this transition. Digital cameras in smartphones have been also suggested [LV10] together with image processing techniques. A promising technique has been suggested by [ZZL+12] and [RKSM14], where light sensors, cell tower signal and magnetic field sensors, together with assistive technologies, such as the acceleration and proximity sensor and time, are fused for identifying the outdoor to indoor transition.

As a result a major challenge was to identify the moment of the indoor transition of a user from outdoors. Our hypothesis here was to monitor simultaneous fluctuations in readings sensed by three sensors, as can also be seen in Figure 2.11. These sensors are:

1. Magnetic sensor fluctuations, since magnetic disturbances tend to be more common inside buildings, due to the steel element inside the walls of the structure due to steel elements inside the walls of a building.

2. The GPS uncertainty fluctuations, as already mentioned in the motivation of this thesis, GPS cannot work indoors. More precisely, when a device is located indoors the uncertainty of the signal increases, often to the size of the entire building.

3. Light intensity fluctuations, since light intensity is expected to be reduced during the transition to indoors during the day since no artificial light can reach the light intensity that the sun can emit. Additionally, indoor fluorescent light exhibits a periodical pattern, due to alternating power (AC) of lighting infrastructure. However, an additional sensor here needs to be consulted, able to identify whether or not the phone sensor is blocked by objects (e.g. because is located in the pocket).

Altitude Estimations After a reference pressure is being extracted, the altitude difference can be quantified using the barometric sensor. By taking into consideration the recorded pressure during the transition as reference pressure, the temperature can be easily extracted by open APIs, while the trust of the reference pressure can be quantified either counting the time since the extraction, or by monitoring the context extractions component and assigning trust when vertical transitions are recorded.

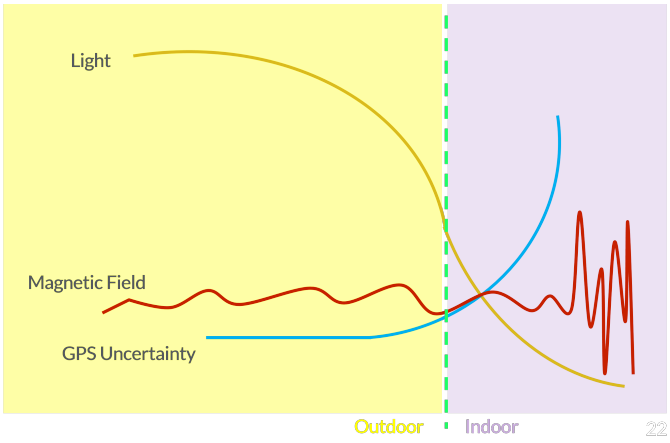

Figure 2.11: An illustrated example of the expected behavior of the three sensors during the user transition from outdoors to indoors.

Storey Number Estimation As already mentioned, the estimation of the number of stores is made following an unsupervised cluster ranking technique called Elbow Method. The Elbow Method is working hand in hand with unsupervised clustering techniques, and it aims to identify the most likely number of cluster in a dataset by identifying an Elbow formed when the mean uniformity inside clusters is monitored in relation to the number of clusters used. The main idea behind this is that uniformity inside clusters will be abruptly flatten after the optimum number of clusters is being reached.

Generate CityGML Model This module is responsible for the generation of the map. CityGML model was selected for presenting the information due to its robustness, continues support by its open source community, and its focus to detail. After the number of floors in a building is being identified, the height of every floor can be computed by calculating the median height value of each cluster. Finally, for mapping the floors, with CityGML, we introduced the *"storeyAltitude"* attribute, to the Level Of Detail 2 (LoD2). Additionally, we used *"StoreyHeightAboveGround"* attribute which was introduced by [KYL14b].

Semantic Mapping

The semantic mapping component is mapping the semantic aspect of objects in space. Examples of such places are corridors, stairs, escalators, elevators, travelators, entrances, doors, tables, chairs, and many others. During this thesis, we experimented with mapping areas such as elevators, stairs and escalators. For mapping this objects, we used our localization and context extraction components.

The reason for mapping these semantics is the fact that they can highly support the localization process. Context recognition is independent of pose, direction, and floor recognition, and while all the above are prone to accumulative error context is not. Hence, if semantics are properly mapped, context can support corrections to misidentification for many of our components.

However, our semantic recognition component has to incorporate the accumulated localization uncertainty, while categorizing the semantics properties of areas. Hence, we performed a fusion between activities extracted from multiple users at the same location and we managed to extract higher-level information that can indicate the object's semantics. For example we were able to differentiate between a stair and an escalator, because people tend to perform two different activities at the same location. In particular, while many people tend to be standing at an escalator, and hence the activity to be recognized is "using elevator", at the same time other people are walking and hence, the activity to be recognized is the "climbing stairs" activity.

For computing higher level context, we had to introduce belief functions for the quantification of the uncertainty of user's context and the place semantics, we had to implemented a prototype system and evaluated its components, and finally we had to provide a framework for the dynamic semantic annotation of indoor places from inertial motion unit data from smart-phones.

Fingerprint Mapping

The fingerprint mapping component is responsible for the collection and the labeling, of signals emitted in the WiFi with locations. It ensures the quality by performing filtering to incoming data based on their location labels. The filtered data is further processed in the attribute ranking subcomponent, which aims to quantify the quality of information that each attribute provides to the localization method. Furthermore, a cluster analysis is performed to identify the optimum number of clusters in the available dataset. Finally, a clustering analysis is performed to ensure that only clusters that contribute to the localization method survives.

Preprocessing The first component in our system, parses the streaming data into data structures. Each device identifier is stored with its corresponding latitude and longitude positions and the normalized received signal strength from its entire trajectory. Data is transformed into segments, forming attributes, described by their trajectory identifier, timestamp, latitude, longitude and floor. The preprocessing subcomponent ensures that all data have been extracted by stationary access points by aggregating data collected from multiple time periods and multiple different phones.

Heterogeneity Corrections Heterogeneous devices require correction before sharing fingerprint databases, due to calibration offsets between different manufacturers of WiFi antennas. In our experiment, we use common landmarks to calibrate incoming data based on their known changes. The calibration was made by fitting a hyperbolic curve between signals captured at the same locations from different users.

Attribute Ranking The attribute ranking component is responsible for ranking and removing bad attributes. In particular, this component assigns a score to each attribute,

which score is later used for the decision of the inclusion or exclusion of the particular attribute. The score is being calculated based on two factors. The first factor is the frequency of the appearance of an attribute at the same area. The second factor is the density of appearance in the particular area.

Cluster Analysis The cluster analysis component identifies the optimum number of clusters in every training set. For achieving this, the elbow method was chosen. The elbow method examines the percentage of variance explained as a function of the number of clusters. The optimum number of clusters can be obtained by performing constant clustering in our dataset, after constantly adding clusters in the algorithm, and monitoring the moment that by adding another cluster the variance of different clusters stops to be significantly increased.

Clustering and Cluster Ranking We used k-Means clustering algorithm for performing unsupervised clustering. In cluster analysis we remove lousy clusters. Lousy clusters are identified due to their spatial limitations. In particular, clusters are expected to be formed following uniform pattern around an imaginary center. However, this is not always the case since there is a number of cluster that exceed boundaries of other cluster and are mixed with multiple others. Unfortunately, these clusters will not contribute to the localization process, or in the best case, their contribution is limited. The cluster ranking component aims to eliminate such bad clusters, and hence enhance the localization accuracy.

Training and Classification The final step in our architecture is the training and classification phase. Due to the nature of our limited and heterogeneous data, we selected a classifier that can score lower error, when trained with limited data and its learning speed when further datasets are introduced is the highest. The selected classifier is Linear Discriminant Analysis (LDA) [Fis36]. Operates by first performing a dimensionality reduction to the provided dataset that can accelerate the learning curve, and its goal is to maximize the separability between different clusters. Finally, the classification of incoming streams is performed by performing a sliding window approach, where incoming streams are segmented into equal duration batches. As already mentioned, the data segment is already tagged with a location estimated through a particle filter algorithm executed on the smartphone.

2.2 Contribution

In summary, the key results of this thesis include:

1. To begin, we provide a model for crowdsourcing indoor maps by combining various techniques and enabling useful intermediate services presented in [PPG17] and [PPG19]. To our knowledge, such model has never been provided in the past, and it focuses on combining existing mapping and localization techniques for the scalable

mapping and maintenance of indoor maps. We focused on providing a research direction focused on flexible, customized mapping of buildings, which allows the integration of existing data and manual techniques. We provided a solution to the problem of obtaining the critical mass of user data for self-starting crowdsourcing mapping techniques.

2. As a response to the "Vertical Localization", "WiFi based localization" and "Indoor Localization" challenges (1.1.7, 1.1.4 and 1.1 respectively), we introduced a novel infrastructure-independent method for the dynamic vertical mapping of structures, presented in [PRI+17] and [PMRI+18]. Our method for calculating altitude based on barometric sensor readings uses as a reference pressure and reference altitude at the moment of the Outdoor-Indoor Transition (OITransition) of the user. This is recognized through the fusion of four different sensors. Following this novel approach, the need for calibration between different barometric sensors becomes obsolete.

   We evaluated our approach in data collected during the pass of six months in three different buildings. We achieved a true positive score of 99.3% for the identification for OITransition, 100% on the number of floors identification and 0.51m average error on the floor height estimation, which is comparable to rival methods that use entire networks of reference pressure sensors.

   The benefit of our method is the fact that it does not require reference barometric sensors. Hence, it can operate in various environments, that have never been visited in the past. Additionally, our developed framework can enable the dynamic mapping of the vertical characteristics in a building, while it takes into consideration the uncertainty of data collected via crowdsourcing.

   We introduced an approach for the extraction of the reference pressure at the Outdoor-to-Indoor-Transition (OITransition) of the user, which is identified by the GPS uncertainty, the magnetic disturbances and the ambient light, which we only took into consideration when the proximity sensor was indicating that the phone surface is not blocked by objects.

   We treated the floor number identification as an unsupervised classification problem, where the number of clusters, or floors, and the altitude of each floor, or cluster label, were unknown. Hence, clustering analysis was applied to our dataset. We deployed the elbow method for the floor number identification and the popular K-means clustering algorithm for the precise altitude identification. Finally, We proposed an enhancement of the CityGML Level of Detail Two Plus (LoD2+) that provides the indoor geometry of buildings at lower levels of detail.

3. As a response to "Semantics Mapping" challenge (1.1.3), we examined the possibility of the dynamic extraction of the semantics of indoor places based on human context, presented in [PFP18b]. We identify user's context from inertial motion unit sensor data, and we use this context to dynamic extract the semantic properties of indoor places. We introduced belief functions for the quantification of the uncertainty of the user context and the place semantics. We implemented a prototype system and evaluated its components, obtaining promising results. We

demonstrated that the semantic recognition of indoor places through user context is feasible. We open-sourced the project and it is available here [Pip17].

We instantiated our framework, in a client-server model. In the client side data is collected from multiple sensors of a smartphone, and is segmented into groups, from which various features are then extracted. Those features are later used to train a support vector machine classifier, which was used to identify seven different activities. These activities are then streamed to the server, where they are aggregated using fuzzy logic with other user activities. This enabled us to execute a rule-based approach, through which we semantically annotate indoor places. The approach was evaluated and the results are presented.

4. As a response to the "IMU-based localization" (challenge 1.1.6), we invented a method, presented in [PTG+19],[GP] and [ROP+19], which can be used to localize people in high accuracy in buildings for which no detailed map and infrastructure is available.

   The benefit of our method is the fact that it performs localization with up to two orders of magnitude fewer particles than state-of-the-art approaches. Indicatively, we obtained a median error of 2.3m, in real-time, in off-the-shelf smartphones, while using only 40 particles instead of 400 or up to 4000 particles that other methods require for achieving the same accuracy.

   The novel contribution lies on the fact that particles are updated after taking into consideration contextual information extracted from the geometry, topology, semantics, as well as the user's direction and activity.

   Our approach executes 3D indoor localization method, in diverse indoor environments. It is running on smartphone sensors and for its execution, it only requires a crude floor plan, which commonly is openly available.

   We provided a comparison of the system against all systems participated in EVAAL competition since 2016 and until 2018.

   Our algorithm is resilience to a certain degree of sensor errors due to a series of unique attributes. A major attribute was the context recognition module, which has the ability to detect floor transitions, which renders it a complete 3D localization and mapping solution.

   A second unique characteristic of our algorithm is the novel use of the topology of the place, that enabled us to use the allowed path directions of the initial position, and hence to identify the best match, which is chosen as the initial or current direction.

   Finally, our algorithm has the ability to autonomously increase its uncertainty when the human movement does not entirely agree with the building geometry, which makes it the most robust in its category, according to the 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN),

5. As a response to "WiFi based localization" and "Heterogeneity" challenges (1.1.4 and 1.1.2 respectively), we delve deeper into the problem of heterogeneity when cross-training a localization model along multiple devices, presented in [PTU+19]. We introduced an architecture that enables the "growth" of organic maps, following a crowdsourcing approach. We implemented and evaluated the proposed architecture, obtaining up to 1.8m localization accuracy, and 18.7% improvements in robustness, which is up to three times better than state-of-the-art solutions. We demonstrate the accuracy of our method in a subway station and we open-sourced the dataset.

   More specifically, we conceptualized, implemented and evaluated a fingerprint technique for indoor localization, which continuously stays up-to-date via crowdsourced information that has as an effect for the localization accuracy to be continuously improving.

   Additionally, we introduced a novel approach for clustering RSS data in real-time. Once data is collected, our approach autonomously clusters, ranks and classifies it. We trained a ML algorithm for enabling indoor localization and provided a s detailed study of how classification accuracy increases based on the number of datasets collected. We evaluated different classifiers that can be used for localization for limited data. We achieve a median error of 1.8 meters purely through fingerprint technology due to the novel approach we followed for managing the heterogeneity between different smartphones.

2.3 Evaluation Methodology

This section presents the steps followed to evaluate the quality of all reported frameworks and components that have been developed or proposed in this dissertation.

Evaluation of localization methods is an open challenge, since many environmental characteristics can influence their accuracy. Different infrastructures can favor some methods while disfavouring others. Some evaluation criteria as described by [MF09] are:

1) Scalability: It concerns whether the algorithm is accurate enough for hundreds, or even thousands of nodes as it is for less than ten. It also examines in case the localization system is centralized, if there are some potential bottlenecks or in case the localization system is distributed, whether an algorithm can be easily developed and deployed for a distributed system easily.

2) Accuracy: It concerns whether the estimated positions match the ground truth positions. Since this is an application-dependent task, accuracy is expressed based on the inter-node spacing. (i.e. if the average node spacing is 100m, up to 1m error may be acceptable, it cannot be the same when the average node spacing is 0.5 m). Metrics applied on this technique are:

- When the actual node position (ground truth) and physical network topology are given, the error can be expressed as: (a) Mean absolute error: by the residual error

between the estimated and actual node positions for every node in the network, after summarizing them and averaging the result. (b) FROB: (Frobenius): by computing the residual error between all nodes in the network. Assuming that the estimated and actual inter-node distances are determined, it determines the root mean square of the total residual error, which represents the global quality of the localization algorithm. (c) GER (Global Energy Ratio): by the normalized distance error between all nodes. (d) GDE (Global Distance Error): by taking the RMS error over the network of n nodes and normalizes it by the average radio range. (e) ARD (Average Relative Deviation): by normalizing the average of the estimated distances between all nodes in the network and the estimated location. (f) BAR: the sum-of-squares normalized error taken from matching the estimated location with the actual location.

- Without ground truth, the error can be estimated based on: (a) Average Distance Error: by subtracting from the observed range between two nodes their estimated distance. (b) SPFROB (shortest-path FROB): based on the shortest path between two nodes, rather than Euclidean distance.

3) Resilience to Error and Noise: It concerns whether the localization algorithm can deal with errors and noise in the input data, as well as, whether noise, bias or uncorrelated error in the input data affect the algorithm's performance.

4) Coverage: It concerns the area covered by the network and the algorithm can perform localization, given a specific network topology/deployment. Usually, it depends on the deployed network density. It also concerns the effort needed to add another node to the network after the initial localization algorithm has completed.

Due to the variation between different methods followed for assessing the quality of performance, each methodology for evaluation is presented separately. In total two main approaches for evaluation were followed, either systematic reviews of the available literature or empirical studies, where our methods were compared against existing methods either by generating datasets or using open-sourced datasets. The list of the different strategies followed for the evaluations follows:

1. The Section 3.1 (*"Bootstrapping the Dynamic Generation of Indoor Maps with Crowdsourced Smartphone Sensor Data"*) aims to set the base of different ideas that we aim to examine during this dissertation. Hence, a systematic approach was followed to collect and cluster information regarding different methods and tools for indoor map generation. Scientific papers published in conferences, which are focusing on indoor localization were systematically researched. These conferences were mainly the IEEE International Conference on Indoor Positioning and Indoor Navigation, but also the the ACM International Conference on Mobile Computing and Networking, the ACM International Workshop on Indoor Spatial Awareness, the ACM International Conference on Mobile Systems, Applications, and Services and the ACM Conference on Embedded Network Sensor Systems.

2. The Section 3.2 (*"Dynamic Vertical Mapping with Crowdsourced Smartphone Sensor Data"*) has been empirically evaluated. In particular all components of our system have been individually evaluated, while a holistic approach was followed for the

entire system as a whole. Because for in this system, the barometric and light sensor were used, among others; and due to their high influence on the atmospheric conditions, data used for the evaluation of our system was evaluated in a period of six months, different times of the day, including three seasons (i.e. winter, autumn and summer). Additionally, data for contacting this evaluation was collected from multiple buildings. The groundtruth was obtained using laser scanners that obtain very accurate information.

3. In the Section 3.3 (*"Extracting Semantics of Indoor Places based on Context Recognition"*) we provided a feasibility study, and we show proof of concept our approach followed a component based evaluation.

   This section is limited on examining the potential future directions of such frameworks instead of providing a solid solution.

   As a result, this framework was evaluated in the a subway station, where semantic properties of areas tend to vary a lot and the size of this

   The Groundtruth was obtainsed from areas mapped and are openly available in Open Street Maps.

4. The Section 3.4 (*"A Novel Lightweight Particle Filter For Indoor Localization"*) has been empirically evaluated on 13 different people, in a 2 level environment and 7 different corridors with total length of 500m, in a university environment. Groundtruth was obtained using laser scanning devices that enabled us to map the precise location of 50 landmarks. Additionally, this framework has been evaluated against prestigious organization as the ETRI, CLE, Google, IBM, Sony, and others in the IEEE International Conference on Indoor Localization and Indoor Navigation 2018 and obtained very promising results (3^{rd} place in 75^{th} percentile and 1^{st} place in 90^{th} percentile) among on-site smartphones that do not use camera for localization. A detailed documentation on the competition is available [ROP+19].

5. The Section 3.5 ("Cross-Device Radio Map Generation via Crowdsourcing") has been empirically evaluated using open-sourced data, as well as data collected from us in three different environments, which include university and subway station environment, and later being open-sourced. Each component of the system is evaluated separately of the system, which was evaluated in a holistic approach.

2.4 Structure

This thesis begins with introduction and motivation in Chapters 1. In Chapter **??** the current state-of-the-art and state-of-practice are presented, while necessary background is being provided. In Chapter 2 the overview of the contribution of this thesis is being listed. In Chapter 4 the summaries and full texts of the publication that ensemble this thesis are included. Finally, the Chapter 4 concludes the thesis.

3 Commented Collection of Papers

The main contributions of this thesis were published separately in various international conference proceedings. This Chapter includes both summaries and full versions of the selected papers in the order presented in Section 1.4, as well as comments on the workshops and conferences where the papers were presented.

3.1 Bootstrapping the Dynamic Generation of Indoor Maps with Crowdsourced Smartphone Sensor Data

"Most important than knowledge is imagination"

— Albert Einstein

Georgios Pipelidis
Ilias Gerostathopoulos
Christian Prehofer

In the Communications in Computer and Information Science book series. Published by Springer International Publishing 2019
Editor: Lemonia Ragia, Robert Laurini, and Jorge Gustavo Rocha
Book Title: Geographical Information Systems Theory, Applications and Management
Pages: 70–84
Year: 2019
ISBN: 978-3-030-06010-7
DOI: 10.1007/978-3-030-06010-7_5
The attached version is the author's manuscript, the original version is available electronically from the publisher's site at: https://link.springer.com/chapter/10.1007/978-3-030-06010-7_5

This paper was published as [PPG19], and it serves as an introduction to our vision and sets the context for the rest of the work presented in this thesis. In particular, we outline a model that can support crowdsourcing activities for indoor mapping, through the use of information mined by a pool of users that voluntarily contribute, while receiving intermediate services. To our knowledge, such model has never been provided in the past, and it focuses on combining existing mapping and localization techniques for the scalable mapping and maintenance of indoor maps.

In this paper, we introduced how different sensors, that most smartphones are already equipped, and how through the combination of these sensors interesting services can emerge that will motivate users to contribute with information that will enhance the quality of the collected data. Our examples beginning from the sensor level to the navigable maps.

Comments on Authorship

My personal contribution to this paper lies in analyzing sensor functionality for each sensor that smartphones are equipped with. I conceived and design the architecture diagram that presents the process which starts from sensor data and ends with indoor navigable maps. Under the guidance of the other authors, I extended the architecture to a paper that includes examples of intermediate services and how these services can contribute to accelerating the indoor map generation. Finally, under the helpful guidance and supervision of the other authors, I authored a majority of the text.

Bootstrapping the Dynamic Generation of Indoor Maps with Crowdsourced Smartphone Sensor Data

Georgios Pipelidis, Christian Prehofer and Ilias Gerostathopoulos

Fakultät für Informatik, Technische Universtität München, Munich, Germany
{georgios.pipelidis, christian.prehofer,
ilias.gerostathopoulos}@in.tum.de

Abstract. Although there is a considerable progress in mapping the indoor places, most of the existing techniques are either expensive or difficult to apply. In this paper, we articulate our view on the future of indoor mapping, which is based on customized, crowdsourced and scalable approaches. On the basis of this approach, we discuss the research challenges that we envision to face in this world of customized bootstrapping and diverse techniques and services. We focus our interest in the combination of multiple of indoor mapping generation techniques and discuss challenges and various indoor mapping techniques. We introduce our adaptive method for bootstrapping the procedure of indoor mapping in multiple ways through intermediate services. Those emerged services enable the obtaining of useful data for this procedure, while they increase the quality of those data. We discuss the necessary components for such approach and we give an example of a bootstrapping procedure.

Keywords: Indoor Mapping; Crowdsourcing; Bootstrapping Process

1 Introduction

Indoor mapping is an enabler for many applications such as indoor navigation systems, augmented reality or even robotics. This is a useful service even if indoor localization is not available, since it enables people to have a view of the indoor place. Together with indoor localization techniques, which have been an active area of research [1], indoor mapping can help materialize the vision for ubiquitous indoor positioning system on a worldwide scale[2].

There is considerable progress in the mapping of indoor places, and many diverse techniques have been proposed, ranging from vision-based [3] and robot-based [4], up to crowdsourced mapping [2]. However, most of the existing techniques are either expensive or difficult to apply, since sensors and methods are usually prone to error due to a variety of the building structures. It remains a challenge to provide cost-effective, easy-to-apply mapping techniques which can cover the large volume and variety of indoor places with their often unique characteristics and semantics. Furthermore, there is a large volume of indoor places

to be considered. For instance, the building footprints in Open Street Maps (OSM) recently surpassed the amount of the street data—not even considering the indoor maps [5].

In this paper, we articulate our view on the future of indoor mapping, which is based on the fact that (i) mapping techniques differ in terms of complexity, required resources and output and (ii) compared to outdoor maps, indoor mapping is more challenging for several reasons:

- *Indoor places are very diverse* in nature and many of them also change frequently; consider e.g. remodeling of floors or new shops in a shopping mall. This also refers to the semantic description of the objects in the buildings.
- *Indoor mapping techniques are very diverse* and range from manual with ad hoc tuning to crowdsourcing techniques. While manual techniques are often more reliable, the abundance of new personal devices with advanced sensors (e.g., motion sensors, cameras, gyroscopes, pedometers) also enable sophisticated crowdsourcing of indoor maps [6].
- *Services related to indoor mapping are also very diverse* in terms of end-user needs and technical assumptions. For instance, architects have different needs than pedestrians or fire fighters. Also, some services require localization, some only mapping, and some only user traces or landmark identification.

To emphasize the diversity of end-user needs and assumptions in the services related to indoor mapping, consider a hospital: the main service is finding doctors, patients, or equipment, assuming a well administered building with well defined tags for tracing and localization. Here, manually created maps can be used—a costly, yet worthy, investment for the hospital administration. On the other hand, in a shopping mall with diverse shop owners, diverse infrastructure and no central management of tags, users also aim to discover places, find other people and explore the map. Here, users may have time to contribute to crowdsourced map creation in exchange for some useful apps. Finally, in an automated factory, highly accurate indoor maps can be important in guiding robots, augmented reality and help avoiding accidents.

Following the above, in this paper we argue that there will be no single way for mapping indoor places, but rather *a diverse set of techniques and services will be used to build up maps and services for indoor locations in a customized way.* Some services may actually not even require proper maps, as in the case of a "take me to the exit" service for which only user traces can be sufficient. We also posit that we will move towards custom solutions for combining indoor mapping techniques in order to improve accuracy and enable a number of diverse services. On the basis of this approach, we discuss the research challenges that we envision to face in this world of customized bootstrapping and diverse techniques and services.

This paper focuses on the combination of indoor mapping techniques and the services they enable. It presents a research direction that focuses on flexible, customized mapping of buildings. This can integrate existing data, manual techniques as well as crowdsourcing from user data. It specifically targets the

problem of obtaining the critical mass of user data for self-starting crowdsourcing mapping techniques. A main point here is that some services can be offered earlier in order to collect data for crowdsourcing. This, we also call intermediate service, as these do not require fully detailed and accurate maps. To illustrate and exemplify the approach, we show a way to describe such flexible bootstrapping of indoor maps that combines techniques as well as services.

In particular, we contribute by highlighting the need for a bootstrapping process that can be customized to the available techniques and building characteristics and by providing an example of such a process.

The rest of the paper is structured as follows. Section 1.2 overviews the most promising indoor mapping techniques. Section 2 provides an overview of our approach, while Section 2.3 exemplifies it on a specific bootstrapping process. Section 3 provides a short assessment of the current state of the art, while Section 4 puts forward a research roadmap and concludes by summarizing the key points. This paper is extension of the work already presented by [7]. More specific, in this paper the approach, the methods and the related work have been extended.

1.1 Indoor Mapping and Challenges

An indoor map implies the existence of a model that describes the geometry, the topology and the semantics of an indoor space [8]. The geometry of an indoor space indicates the morphology of important places or objects in the space. For example the shape and the location of a room or a desk. Topological relationships signify the explicit description of adjacent and connected places in that space. The semantics indicate the way that places in the space are used. For example the existence of stairs, elevator, toilet etc. Semantics may also indicate unique characteristics of locations in that space. For example the Received Signal Strength Indicator in a place with multiple WiFi Access Points.

Indoor maps are typically created via a manual process that starts off with obtaining the architecture blueprints of a building, enhancing them with Places of Interest (POIs), and submitting the result to a floorplan database. The problems of this traditional approach are that (i) it is labor-intensive and slow; (ii) it is not always economically viable, as many times the cost of creating the maps can surpass the revenue they create; (iii) it relies on having the building blueprints in the first place, which is not always true, as e.g., in the case of developing countries; and (iv) there is a huge effort in keeping the maps up-to-date, since the manual process has to be repeated to capture changes in the environment.

Additionally, there is not a well agreed upon model for these procedure. Beyond the technical challenge of generating the maps, mapping indoor places is a resource demanding procedure with an expansive cost. Additionally, environment characteristics are never static (i.e. objects displaced etc.). Hence, indoor maps can often become outdated, while their maintenance effort increase the overall cost. Legal challenges are often present, since in most cases indoor places

are privately owned. Furthermore, indoor localization cannot use the maps without semantically enhanced and uniquely identified nodes which can be used by an entity for successfully localized.

As a result, there is clearly potential in automating the map creation and update process. In particular, we see a great potential in automated techniques that rely on user data, i.e. crowdsourcing, for creating maps that are cost-effective, semantically-rich and dynamically updated. In this vision, crowdsourced maps are created based on fusion of data sensed by modern ubiquitous devices such as smart phones.

1.2 Mapping Techniques

In this chapter we describe the available technologies for a potential use in a bootstrapping process for indoor maps. We posit that those technologies can be used to provide services, which by their turn can be used as the means for incentivizing users to participate in the envisioned crowdsource-based system. These initial users can provide the critical data mass allowing the creation of more sophisticated services leading to full-blown indoors maps. In this chapter, on top of articulating our generic bootstrapping model, we exemplify how the presented different techniques and technologies for indoor mapping fit within the model.

Light Detection And Ranging (LiDAR). LIDAR uses lasers to measure the distance between objects inside a building (i.e., walls, floors, ceilings etc.) like [4]. A LiDAR unit, often mounted on a robot or vehicle, scans the environment. The position of the unit is estimated by vSLAM [9]. A point cloud is generated and by identifying contours (i.e. points of similar distance), a map can be extracted. Semantic annotations are usually manually made by expert surveyors.

Usage of existing architectural blueprints. If blueprints are encoded in formats such as Industry Foundation Classes (IFC) [10] or Building Information Modeling (BIM) [11], they contain the geometric information that can be readily used in indoor maps. However, such formats do not include topological nor semantic information. The last is usually added manually by expert surveyors, resulting into mapping data encoded into formats such as IndoorGML [12]. Approaches for automatic derivation of topological relations (e.g., adjacency and connectivity of rooms) from IFC models have also been suggested [13].

Structure from motion. In this technique, a 3D structure of a building can be extracted from a camera [3] by capturing many images of an indoor place and translating them into a single 3D view. To do this, the camera's internal and external parameters, e.g. lens-generated distortion, translation and rotation matrix have to be known or be retrievable from common features of the captured images.

Depth sensors. In this technique, a typical setting is to have an infrared projector that projects a unique pattern. An infrared sensor, whose relative distance to the projector and rotation are known, recognizes this pattern. A depth map

is constructed by analyzing the unique pattern of infrared light markers by triangulating the distance between the sensor, projector and the object. Finally, a 3D point cloud is extracted from stereoscopic view algorithms, from which a map can be generated [14].

Smart phone 3D modeling tools. In this technique, specialized smart phone apps enable users construct components of a building [15]. After initial versions of the maps have been created, other users can enhance the maps or vote on their accuracy and completeness.

Activity-based map generation. An indoor map can be transparently and autonomously generated based on activity recognition of users as it has been suggested by [2]. This technique works as follows: After extracting steps of users by their `x` and `y` coordinates or by a series of trajectories, a point cloud can be extracted. A map of the indoor place can be created by fusing data from different users and identifying places with common patterns. For example, places where users performing the same activity (i.e., stairs) can be identified.

From the above, the use of **Structure from motion** or **Depth sensors**, the use of **Smart phone 3D modeling tools** as well as **Activity-based map generation** lend themselves to crowdsourcing, whereas **Lidar** and **Usage of existing architectural blueprints** do not.

2 Adaptive Bootstrapping

In this section, we outline our envisioned approach towards indoor mapping, based on the following observations on the present and future research and development in indoor mapping:

- *Techniques need to be combined.* There are many indoor mapping techniques which differ in terms of complexity, required resources, and output. For instance, if one wants to use LiDAR, a localization technique has to be in place, and also sophisticated laser equipment has to be available. Activity-based map generation, on the other side, does not make any major assumptions in terms of equipment; however, it assumes a plethora of data. We argue that a combination of different techniques will be used to create or maintain indoor maps that are both cost-effective and accurate.
- *Bootstrapping is needed for crowdsourcing.* As discussed, we posit there will be no "single-shot" solution towards indoor mapping; combined solutions, as shown below, will also involve crowdsourcing. Therefore an incremental, stepwise bootstrapping will be needed to obtain user data. This is substantiated by crowdsourcing techniques which not only need user data, but also other inputs like building floorplans or points of interest.
- *No single bootstrapping process.* We believe that the diversity of buildings, mapping techniques, as well as services will lead to individual and custom processes for such bootstrapping. The processes will be adapted to end-user needs, available infrastructure, available budget, and other factors.

2.1 Services Related to Indoor Mapping

A number of services with different characteristics, users, and assumptions on crowdsourcing effort can be supported by our approach, e.g.:

Wellness monitoring. This is a family of emerging services that provide feedback to users based on their activities during the day. For example, services that can track the number of steps that a user did during a day can be used for identifying the distance traveled by the user.

Card swiping. This service may substitute the Magnetic stripe cards with smart phone build-in NFC chips. In combination with other sensor data, it can be used to generate a general model for identifying outdoor-indoor transitions and vice versa.

"Take me to the exit". This service can work as a digital Ariadne's thread, where users will be able to find their way back to the entrance of indoor places by following their own captured route in reverse. User traces collected from this service can be used for generating a point cloud.

Instruction-based navigation. This service can provide basic instructions on how to visit an office or a classroom in the form of instructions such as "Enter from the north entrance, walk straight for 10 secs, then turn right, walk up the stairs and enter the door on the right".

Location-aware ticketing. This service can free users of public transportation from the need to purchase tickets in advance, as users can be billed based on the actual distance traveled. In addition, companies that run the transportation services will be able to acquire an accurate view of the usage patterns and optimize their services.

Elderly monitoring. This service can be used to identify accidents involving elderly or people with special needs in real time by detecting problems in mobility or patterns that correspond to sudden falls. Data from such service can be used for semantically enhancing indoor maps, via adding the use of a room.

Call forwarding. This service can use the information of a person's position inside a building (e.g. a specific office) and the position of land lines within the building in order to automatically forward calls to the nearest land line.

Dynamic meeting scheduler. This service can use the (indoor) user position (or an approximation, e.g. a room) and possibly user calendar data, in order to propose meeting locations that fit the participants' locations. Data from this service can be used for labeling indoor spaces.

It is clear that the services related to indoor mapping are rather diverse, and make different assumptions regarding the maturity and completeness of the supporting indoor mapping systems. For instance, wellness monitoring does not assume any complete mapping or localization system (even though the data captured from such services can actually allow for activity-based mapping tech-

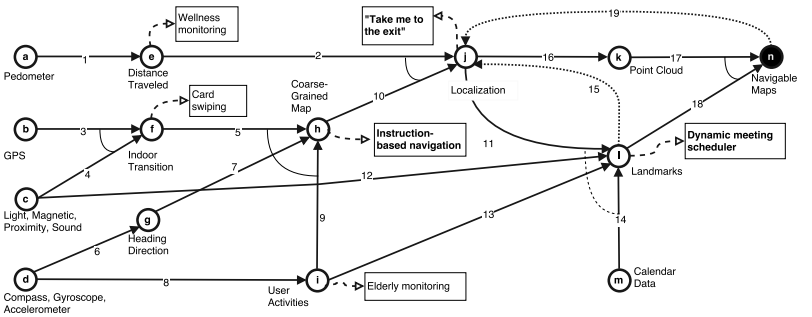

Fig. 1. Customized bootstrapping process for a university campus building. Circular nodes are artifacts, arrows are tasks with inputs and outputs, rectangles are intermediate services (services in bold are described in the text) [7].

niques). Also, "take me to the exit" does not assume the existence of a complete navigable map, but only of a single well-defined route from a *single* user.

Even though one might argue that a single indoor mapping techniques will prevail and allow for the creation of different services, including the ones outlined above, we do not believe this will be the case. Instead, we believe that the current fragmented picture of the techniques and services will continue to be the norm. The question then becomes, how can we combine different techniques in a specific indoor setting towards creating innovative applications? The naïve answer is to just use all the available techniques in parallel and pick the best results out of their execution. In reality, though, different techniques have different starting and ending points. Hence, a more realistic view of the composition is as a chain of tasks with inputs and outputs, dependencies between them, even loops, representing that a task receives inputs from other tasks and provides output to them. (As an aside, the simultaneous localization and mapping (SLAM) technique features this exact loop between the localization and mapping module.)

An important observation is that services with rudimentary assumptions in terms of indoor mapping can act as catalysts for gaining the critical mass of user data that can enable services with more advanced mapping needs. For instance, in a hospital building, the target service might be full-blown indoor navigation, whereas intermediate services might be call forwarding for medical personnel, room-based localization of equipment, elderly monitoring, and others. Potential users are the medical personnel, patients, and visitors. In contrast, consider a university campus building: the target service can be the same as in the hospital case, but now intermediate services could be room finders, "take me to the exit", wellness monitoring, etc., whereas potential users are now students and academic employees. Finally, in the case of a subway station, a promising intermediate service is, e.g., location-aware ticketing.

In the following, we are providing a way to model such bootstrapping processes. Our modeling technique is based on the fact that each indoor mapping technique can be broken down to a number of tasks with inputs and outputs. The input of the initial task indicates the technique's assumptions. As a result, a bootstrapping process can be represented as a graph of tasks. We present an example of this in the next section.

2.2 Bootstrapping Components

In this section we discus the number of components needed for initiating the bootstrapping procedure.

Distance. An important component of the bootstrapping process, is the one responsible for estimating the traveled distance from the user. The user's travel distance can be estimated from pedometer data. Pedometer applications have become ubiquitous in the today smart-phones, while their accuracy has been dramatically increased [16]. This information is essential for indoor localization, while services can be emerged indicating to the user's distance in a particular time interval.

Indoor Transition. A mechanism for detecting the indoor transition is also important, since this will provide an accurate initial location for localization. By fusing GPS and sensor data (i.e. light sensor, wifi RSS etc.), the location of the transition from outdoors to indoors, and vice versa, can be detected [17]. The accurate detection from outdoor to indoor environment will provide with an accurate initial position, which can enhance the localization.

Heading Direction. A mechanism for estimating the user direction is equally important for a localization algorithm, since together with the estimated distance traveled can contribute on a pedestrian dead reckoning algorithm. From Inertial Motion Unit IMU data (i.e. accelerometer, gyroscope, compass etc.), the walking or even standing direction of the user can be estimated in various ways [18]. This information can be used for estimating the position of the user.

User Activity. A component for recognizing the user's performing activity is needed for more robust localization, for identifying unique landmarks as well as for enhancing the procedure of the semantically annotation of places, by indicating the use of this place. Identifying the activity a user is performing from a given set of activities, using IMU data can be performed accurate enough [19]. Having this information the localization procedure will be improved and the final map can be enhanced with semantic information.

Localization. Knowing the orientation, distance traveled and activity a basic pedestrian dead reckoning mechanism can be put in action, since the orientation together with the distance traveled and an initial location (i.e. entrance) can be used for estimating the current user location, while the activity can be used for improving this procedure. For example, the standing activity can be used for re-calibrating the sensors (i.e. gravity direction identification or gather more measurements for restarting the location etc.) or walking can be used for

resetting the pedometer error. On this step, applications such as “guide me back to the entrance”, or ”share my indoor path” can be emerged. Delivery services will have the precise location of the delivery address and will not only be limited on the building location. It can enhance outdoor navigation by suggesting the entrance which is nearest to the destination, or services such as subway transportation suggestions, since the distance from outdoor to the indoor station can be more accurate estimated or even be personalized.

Landmarks. After segmenting sensor data based on discrete characteristics, uniquely identified locations will be emerged. For example the activity performed by the user on a specific area, the RSS of the WiFi or the magnetic field intensity, can be used to characterize the area. After mapping these places on a basic map, the localization procedure can be enhanced, thus better localization implies better landmark locations. Similar to a Simultaneous Localization And Mapping (SLAM) algorithm. Services such as ”Find an available meeting room” will be emerged. This service can work as follows: It will identify users who are in a meeting room, based on calendar data and similarities in sensor data (i.e. WiFi RSS). Then it will broadcast the name of this room to users who have been delayed and are going to join the meeting, according to their calendar and the room name is either unknown or has been changed.

2.3 Bootstrapping Example

This section introduces an example of a bootstrapping process for a university campus building. To illustrate the bootstrapping process, we use a data-flow-like diagram depicted in Figure 1.

In this diagram, circular nodes correspond to artifacts. Each artifact enables the creation of one or more services. For example, `Distance Traveled` (`e`) can enable a service such as wellness monitoring, since the walked distance is directly related with exercising. Inputs and outputs of artifacts are visually presented as solid enumerated arrows which indicate data flow. For example, the input of `Indoor Transition` (`f`) is GPS signal (`3`) and IMU (`4`) data (i.e. ambient light, magnetic field, proximity and sound). By reasoning on these input data, similar to [20], the output is the locations of entrances (`5`). In case of more than one input, a solid line connecting them implies conjunction (e.g. lines `5`, `9` and `7`); a dashed line implies disjunction (e.g. `11`, `12`, `13`, `14`). Finally, dotted connections imply additional inputs which can improve the data quality (e.g. `15`).

An artifact can be connected to a number of intermediate services. A service is represented by a rectangle and implies a set of software functionalities which can be a user-facing application. Finally, the target artifact is represented as a filled circular node (e.g. `n`).

Figure 1 presents a set of possible bootstrapping options. One would start at one or more of the nodes on the left, e.g. assuming devices with GPS (`b`) or compass/gyroscope and accelerometer (`d`). Informally speaking, we can then proceed to some of the connected nodes (e.g. `f` or `g`), based on user data generated from operating services possible at this point. Based on the new data, we can proceed with further steps in this graph.

As depicted in Figure 1, the entire bootstrapping process could emerge via existing services, such as wellness monitoring or card swiping. Of course, alternative paths are also available. For example the `Coarse-Grained Map` step could be skipped; similarly, `User activities` might not be needed if semantically-rich calendar data are available.

In our example, the target service is to enable indoor navigation based on dynamically created maps that capture the geometry, topology and semantics of the building. The above information needs to be integrated in a data model, e.g. by using and extending the IndoorGML standard [12]. IndoorGML provides the constructs to denote subdivisions of indoor places (i.e. rooms), spaces that connect two indoor places (e.g., inner doors), spaces that connect indoor places to outdoor ones (e.g., entrance doors), spaces acting as passages between indoor places (e.g., corridors, stairs), and other important properties.

There are a number of intermediate services among the ones described in the beginning of this Section. We describe here the indoor mapping techniques and associated artifacts they rely upon:

Instruction-based navigation. To provide this service, a `Coarse-Grained Map` is needed. This is a model that includes the elements essential for routing, such as corridors, stairs, doors, and entrances. This is the outcome of merging three other artifacts: `Indoor Transition`, `Heading Direction` and `User Activities` (tasks `5, 7, 9`). The first one is derived by using GPS data (task `3`) and fusing them with other mobile sensor data such as light, magnetic, and proximity data (task `4`). The intuition is that the sensors' behavior changes during the outdoor-indoor transition, where the GPS uncertainty and the WiFi received signal strength are both increasing.

`Heading Direction` can be derived via machine learning algorithms (embodied in task `6`) that work on compass, gyroscope and accelerometer data. The intuition is, if a phone's pose is identified, it can be used to extract the user's local direction (i.e. in the phone's coordinate system) via monitoring the acceleration changes due to the gait movement, then relate this direction to a global system using the compass.

Finally, `User Activities` can be derived from the same data using machine learning techniques with high accuracy (task `8`), since moving and stationary activities can be detected from disturbances in the acceleration sensor, while movements on the vertical space can be detected from disturbances in the barometric sensor.

Dynamic Meeting Scheduler. This service is based on the `Landmarks` artifact. Landmarks are distinctive locations in a building. They are either locations where users consistently perform the same activity (e.g., stairs)—contributed by the `User Activities` (task `13`)—or locations with distinct characteristics of a measured quantity (e.g., WiFi RSS, geomagnetism, sound, light)—contributed by the `Light, Magnetic, Proximity, Sound` (task `12`). In both cases, landmarks need to be localized in a building—hence the dependence on `Localization` (task `11`). Landmarks can also be derived from `Calendar Data` (task `14`) via semantics (e.g., meeting room name).

"Take me to the exit". In our example, we assume that there is no localization infrastructure in place. As a result, we would need to resort to pedestrian dead reckoning techniques [21]. Pedestrian dead reckoning is based on approximating the position of a user by measuring the distance traveled when walking towards a direction from a known point. This explains why `Localization` depends on the `Distance Traveled` (task `2`) and the `Coarse-Grained Map` (task `10`). The former is derived directly from pedometer data (task `1`). The latter contains information regarding the heading direction (task `7`) and the indoor transition points (task `5`). These points are the initial *known points* in the dead reckoning algorithm. `Localization` can also depend on `Landmarks` for re-calibrating the algorithm (restarting the error) in distinct locations (task `15`).

Finally, `Localization` provides input for the creation of `Point Cloud` (task `16`) using existing techniques, and subsequently of `Navigable Maps` (task `17`). `Navigable Maps` are also enhanced by the identified `Landmarks` (task `18`). In particular, activity-related landmarks can be a rich source of semantic annotation for maps (e.g., places where people sit together for long time can be labeled as meeting rooms). At the same time, `Navigable Maps` can enhance `Localization` by error recalibration on the basis of non-navigable places (task `19`). This can be achieved either by relating user traces to sets of possible routes or via uniquely identified locations (e.g. stairs), in which case the context of users (e.g. "climbing stairs") can be used for re-positioning them.

It is important to note that the example bootstrapping process illustrates a cost-effective solution without dedicated equipment and expensive manual work. As an alternative, consider hiring an indoor localization company, for performing tasks `1` and `2` in our example—this would have led to a different customization of the same bootstrapping process.

Being aware of the orientation, distance traveled and activity of the user, a basic pedestrian dead reckoning mechanism — for *Localization* — can be put in action, since the orientation together with the distance traveled and an initial location (i.e. entrance) can be used for estimating the current user location, while the activity can be used for improving this procedure. For example, the standing activity can be used for recalibrating the sensors (i.e. gravity direction identification etc.) or walking can be used for resetting the pedometer error. On this step, applications such as "guide me back to the entrance", or "share my indoor path" can be emerged.

Side-services such as "Find an available meeting room" can provide with labels of the locations, while in combination with a localization technique can provide *Landmark* locations. The side-service can work as follows: It will identify users who are in a meeting room, based on calendar data and similarities in sensor data (i.e. WiFi RSS). Then it will broadcast the name of this room to users who have been delayed and are going to join the meeting, according to their calendar and the room name is either unknown or has been changed. After segmenting sensor data based on discrete characteristics,uniquely identified locations will be emerged. For example the activity performed by the user on a specific area (i.e. door handling events **??**), the RSS of the WiFi or the magnetic field intensity,

can be used to characterize the area. After mapping these places on a basic map, the localization procedure can be enhanced, thus better localization implies better landmark locations. Similar to a Simultaneous Localization And Mapping (SLAM) algorithm.

3 Related Work

To our understanding, there is no prior work on systematic bootstrapping of indoor maps. There are several works which integrate different intermediate techniques, which we list below. More mapping techniques can be found here **??**.

Heading direction. [22] detect the discreate signal vibration when the heel strikes the ground during a gait circle. Then they use this data point as a reference and scan the signal to identify the dominant body's movement partition from the entire signal segment. Finally, they translate the walking direction to the global magnetic system. However, their framework is highly dependent on the terrain as well as on user behavior.

Indoor-Outdoor transition. [20] do not only use the drop of GPS accuracy as an indication of the I/O transition, but also use light censors, cell tower signals, and magnetic field sensors. The acceleration and proximity sensor time series are fused for identifying the I/O transition.

Activity Recognition. [23] use a Support Vector Machine classifier to distinguish among moving activities such as walking, running, and ascending and descending stairs and improve existing position systems. Their observation is that the step length varies when a user is walking, running or climbing stairs. Their approach is argued to work in various phone poses. However, their approach uses a large amount of features, which can result in high computational demands.

Localization. [24] have developed a ZUPT algorithm for localization. However, they point out the need to identify vertical transitions due to the limitation the vertical displacements cause. To solve this problem they introduce a moving platform detection module. It works by combining accurate sensors, and not those available on a smart-phone, such as accelerometer, barometer and magnetometer. They estimate altitude using the barometric sensor, while they are also using it to identify instance phases.

4 Discussion and Outlook

Following the diversity of indoor places, techniques and services, we have outlined our position for an adaptive bootstrapping process. This includes mapping techniques but also intermediate services which enable data collection for improving maps and offering enhanced services. We have illustrated examples of customizations of the process in a visual way and argue that the bootstrapping

Our view integrates many existing mapping techniques as well as services and also assumes considerable progress in each of these disciplines. As we focus more on how the different processes for mapping can be integrated, our vision is orthogonal to research roadmaps of specific techniques.

Our new bootstrapping approach also gives rise to the several challenges:

Bootstrapping processes. We need research to understand and model bootstrapping processes, similar to our example, in order to obtain a more complete picture of the techniques and services that are available. Also, most of the services described in Section 2.3 are open challenges mainly due to the inherent complexity of indoor localization: existing sensors (both in phones and specialized devices) fail to effectively propagate a discrete signal patterns in indoor space, making simple triangulation-based techniques infeasible. Additionally, robust heading direction identification independent of the phone's pose remains an open challenge [20].

Intermediate targets/artifacts. We need to understand what can be useful intermediate targets/artifacts, which are both feasible w.r.t mapping techniques and also enable useful services. Moreover, protocols need to be emerged to enable information exchange through APIs between the different services. Importantly, we need to manage the uncertainty inherent to both sensor reading and human users, filter out outliers, and in general work with noisy data. Trust models to manage ambiguous information extracted from multiple users need to be emerged. Existing indoor data models have to be enhanced in order to cope with such incomplete, ambiguous or inaccurate models.

Process customization. We need research to understand when and how to apply different bootstrapping processes to specific buildings. This can also lead to easier or automatic customization of bootstrapping to specific classes of buildings.

5 Conclusion

In this paper, we discuss our view on the future of techniques for indoor mapping. We propose customized, crowdsourced and scalable approaches and we discuss the research challenges. We demonstrate methods for the combination of multiple of indoor mapping generation techniques and discuss their challenges. We introduce an adaptive method for bootstrapping the procedure of indoor mapping in multiple ways through a number of intermediate services. Those services enable us to obtain useful data for this procedure, while they increase the quality of those data. Finally, we discuss the necessary components for such approach and we give an example of a bootstrapping procedure.

6 Acknowledgments

This work is part of the TUM Living Lab Connected Mobility project and has been funded by the Bayerisches Staatsministerium für Wirtschaft und Medien, Energie und Technologie.

References

1. Mautz, R.: Indoor positioning technologies. ETH Zurich, Department of Civil, Environmental and Geomatic Engineering (2012)
2. Alzantot, M., Youssef, M.: CrowdInside: Automatic Construction of Indoor Floorplans. In: SIGSPATIAL '12, ACM (2012) 99–108
3. Gao, R., Zhao, M., Ye, T., Ye, F., Wang, Y., Bian, K., Wang, T., Li, X.: Jigsaw: indoor floor plan reconstruction via mobile crowdsensing. In: Proc. of MobiCom '14, ACM (2014) 249–260
4. El-Hakim, S.F., Boulanger, P.: Mobile system for indoor 3-d mapping and creating virtual environments (December 28 1999) US Patent 6,009,359.
5. Goetz, M., Zipf, A.: Towards crowdsoursing geographic information about indoor spaces; mapping the indoor world. GIM International (2012) 30–34
6. Pipelidis, G., Su, X., Prehofer, C.: Generation of indoor navigable maps with crowdsourcing. In: Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia. MUM '16, New York, NY, USA, ACM (2016) 385–387
7. Pipelidis, G., Prehofer, C., Gerostathopoulos, I.: Adaptive bootstrapping for crowdsourced indoor maps. In: Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1: GISTAM,, INSTICC, SciTePress (2017) 284–289
8. Chen, J., Clarke, K.C.: Modeling standards and file formats for indoor mapping. In: Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1: GISTAM,, INSTICC, SciTePress (2017) 268–275
9. Karlsson, N., Di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., Munich, M.E.: The vslam algorithm for robust localization and mapping. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE (2005) 24–29
10. : ISO 16739:2013 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries (2013)
11. : ISO/TS 12911:2012 - Framework for building information modelling (BIM) guidance (2012)
12. : OGC IndoorGML version 1.0.2 (August 2016) http://www.opengeospatial.org/standards/indoorgml.
13. Liu, H., Shi, R., Zhu, L., Jing, C.: Conversion of model file information from IFC to GML. In: IGARSS'14, IEEE (2014) 3133–3136
14. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. Int. J. Robot. Res. **31**(5) (2012) 647–663
15. Eaglin, T., Subramanian, K., Payton, J.: 3D modeling by the masses: A mobile app for modeling buildings. In: Proc. of PERCOM '13 Workshops, IEEE (March 2013) 315–317

16. Tomlein, M., Bielik, P., Krátky, P., Mitrík, S., Barla, M., Bieliková, M.: Advanced pedometer for smartphone-based activity tracking. In: HEALTHINF. (2012) 401–404
17. Pipelidis, G., Rad, O.R.M., Iwaszczuk, D., Prehofer, C., Hugentobler, U.: A novel approach for dynamic vertical indoor mapping through crowd-sourced smartphone sensor data. In: 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN). (Sept 2017) 1–8
18. Combettes, C., Renaudin, V.: Comparison of misalignment estimation techniques between handheld device and walking directions. In: IPIN'15. (October 2015) 1–8
19. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. SIGKDD Explor. Newsl. **12**(2) (March 2011) 74–82
20. Zhou, P., Zheng, Y., Li, Z., Li, M., Shen, G.: Iodetector: A generic service for indoor outdoor detection. In: SenSys '12. SenSys '12, ACM (2012) 113–126
21. Kourogi, M., Kurata, T.: A method of pedestrian dead reckoning for smartphones using frequency domain analysis on patterns of acceleration and angular velocity. In: Proc. of PLANS '14, IEEE (May 2014) 164–168
22. Roy, N., Wang, H., Roy Choudhury, R.: I am a smartphone and i can tell my user's walking direction, ACM Press (2014) 329–342
23. Nguyen, P., et al.: User-friendly activity recognition using SVM classifier and informative features. In: IPIN'15. (October 2015) 1–8
24. Kaiser, S., Lang, C.: Detecting elevators and escalators in 3d pedestrian indoor navigation. In: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN). (Oct 2016) 1–6

3.2 Dynamic Vertical Mapping with Crowdsourced Smartphone Sensor Data

"Talent hits a target no one else can hit.
Genius hits a target no one else can see"

— Michio Kaku

Georgios Pipelidis
Omid Reza Moslehi Rad
Dorota Iwaszczuk
Christian Prehofer
Urs Hugentoblers

In Special Issue Sensors and Sensing in Indoor Localization, Tracking, Navigation, and Activity Monitoring. Published by MDPI Sensors 2018
Volume: 18
Issue: 2
Article Number: 480
DOI: 10.3390/s18020480
The attached version is the author's manuscript, the original version is available electronically from the publisher's site at: https://www.mdpi.com/1424-8220/18/2/480#

Summary of the Paper

This paper was published as [PRI+17] and an extended version of it was published in [PMRI+18], and it introduces a simplified crowdsourced indoor mapping approach. In particular, for the mapping approach presented in this paper, smartphones were carried around in a set of buildings, and data was collected by users of different profiles. The data was then used to precisely map the vertical dimension of buildings, in great detail, using and even enhancing existing standard models for indoor mapping. We developed and evaluated our method, and we were able to dynamically extract the vertical dimension of buildings. To the best of our knowledge, no other approach for the dynamic vertical mapping using crowdsourced smartphone sensor data has been proposed in the past. Our approach scores an average error of a 0.507 m vertical disposition and has been tested in three different buildings. It is infrastructure-independent and tends to outperform existing approaches, (e.g. [LHG13] and [LIT+14])

Comments on Authorship

My personal contribution to this paper lies in theorizing the approaches implemented during this work. I designed the architecture of the system, proposed the components of this system, such as the combination of the outdoor to indoor transition with barometer for calibration. I developed a strategy for the evaluation of the entire system. Together with the second author of the paper we implemented the system and evaluated its individual components. Moreover, again under the helpful guidance and supervision of the other authors, I authored the majority of the text.

sensors

MDPI

Article

Dynamic Vertical Mapping with Crowdsourced Smartphone Sensor Data

Georgios Pipelidis [1,*], Omid Reza Moslehi Rad [2], Dorota Iwaszczuk [2], Christian Prehofer [1] and Urs Hugentobler [3]

[1] Software and Systems Engineering Research Group, Technical University of Munich, Boltzmannstr. 3, 85748 Garching bei München, Germany; prehofer@fortiss.org
[2] Astronomical and Physical Geodesy, Technical University of Munich, Arcisstr. 21, 80333 Munich, Germany; omidrezamoslehirad2012@gmail.com (O.R.M.R.); dorota.iwaszczuk@tum.de (D.I.)
[3] Satellite Geodesy, Technical University of Munich, Arcisstr. 21, 80333 Munich, Germany; urs.hugentobler@tum.de
* Correspondence: pipelidi@in.tum.de; Tel.: +49-89-289-17840

Received: 31 October 2017; Accepted: 25 January 2018; Published: 6 February 2018

Abstract: In this paper, we present our novel approach for the crowdsourced dynamic vertical mapping of buildings. For achieving this, we use the barometric sensor of smartphones to estimate altitude differences and the moment of the outdoor to indoor transition to extract reference pressure. We have identified the outdoor–indoor transition (OITransition) via the fusion of four different sensors. Our approach has been evaluated extensively over a period of 6 months in different humidity, temperature, and cloud-coverage situations, as well as over different hours of the day, and it is found that it can always predict the correct number of floors, while it can approximate the altitude with an average error of 0.5 m.

Keywords: indoor mapping; outdoor–indoor transition; CityGML; dynamic mapping; vertical mapping

1. Introduction

Indoor maps have become a necessity in robotics, augmented reality, location-based services, mobile ad hoc networks, and search and rescue missions. Because of the high manual effort of generating indoor maps, there have emerged approaches for the dynamic generation of two-dimensional indoor maps through crowdsourced sensor data (e.g., [1,2]). However, these approaches require precise localization. Although many localization providers argue having achieved an average accuracy of 6 m in horizontal localization, none of them provides vertical localization. This has as a result pushed back milestones scheduled by initiatives that are focused on accelerating the research of indoor localization, as these milestones require storey-level localization. Such initiatives are the Enhanced 911 [3] in the United States, and the Enhanced 112 in the European Union [4], as well as the European Accessibility Act [5]. The main reason for the lack of vertical localization providers is the limited information available, for example, the lack of precise altitude indication for every floor in a building in existing maps. To the best of our knowledge, no approach for the dynamic vertical mapping using crowdsourced smartphone sensor data has been proposed.

This paper aims to automate the indoor vertical mapping process, while enriching existing maps with indoor information. In this way, we enable maps to carry information regarding the number of floors in a building and the corresponding altitude of each floor. We achieve this using the novel method we use to fuse the barometric sensor of smartphones with other sensors for the extraction of the ambient reference pressure in locations, which can be used for precise altitude estimation.

More specifically, we first use sensor data extracted from light, proximity, Global Positioning System (GPS), and magnetic sensors to identify the user's transition from the outside to the inside of a building. Once we recognize this transition, we use it as a landmark for the extraction of the reference pressure. We then use this extracted reference pressure to estimate the altitude differences for every step of the user using the barometric formula. For better clustering between altitude values, we filter out vertical transitions (e.g., stairs or elevators), as they do not belong to floors. Because there is no user who is going to visit all the floors of a building, altitude values from multiple users are aggregated for the identification of the number of floors in a building and the height of each floor. Finally, these data are used to generate three-dimensional (3D) models following the standards as defined by the City Geography Markup Language (CityGML) Level of Detail 2, while an enhancement to the standard models is proposed in order to enable it to carry floor information as well as the altitude of each floor. Various studies attempt to vertically localize humans or objects via pressure sensors [6,7]. However, they all assume reference sensor stations permanently installed in the building. Hence, these are highly infrastructure-dependent approaches. Additionally, several studies attempt to vertically localize objects or humans, mostly triangulating them, using the WiFi received signal strength [8], cellular network antennas [9] or Bluetooth Low Energy (BLE) beacons [10]. Unfortunately, every triangulation method highly depends on the assumption of the existence of particular infrastructure, as well as the line of sight. This means that the strength of the signal, and as a consequence the distance estimation, is influenced when the observer is standing in front of the infrastructure (e.g., BLE beacons) or behind it. Finally, approaches for the dynamic generation of vertical maps have also been proposed [11–13]. However, these approaches suggest the use of outdoor characteristics for mapping indoors. This is not feasible, most of the time, as a result of the uniform shape of various buildings, which does not allow any subspace discretization. Additionally, most of the buildings contain underground structures that cannot be recognized through any outdoor model (e.g., subway stations).

Our approach, with an absolute average error of a 0.507 m vertical disposition in three different buildings, although it is infrastructure-independent, performs equally or even outperforms existing approaches, such as in [7], with a 0.8 m vertical disposition, and in [14] with a 0.86 m vertical disposition, which are infrastructure-dependent.

1.1. Background on the Barometric Formula

The atmospheric pressure is the weight exerted by the overhead atmosphere on a unit area of a surface. The barometric formula describes how this atmospheric pressure is reduced when the altitude is increased and vice versa. The unit of pressure is 1 hPa = 1 mbar = 100 Pa.

The barometric formula reads:

$$P = P_b * \left[\frac{T_b}{T_b + L_b * (h - h_b)} \right]^{\frac{g_0 * M}{R * L_b}} \quad (1)$$

where h_b is the reference altitude, T_b and P_b are the temperature and pressure at the reference point, L_b is the standard temperature lapse rate of 6.49 K/km, P is the pressure at the current point at height h, R is the universal gas constant 8.3144621 J/K/mol, g_0 is the earth's gravity acceleration 9.80665 m/s^2 and M is the molar mass of the earth's air 0.0289644 kg/mol.

Equation (1) can be altered for estimating altitude to give the following:

$$h = h_b + \frac{T_b}{L_b} * \left[\left(\frac{P}{P_b} \right)^{-\frac{R * L_b}{g_0 * M}} - 1 \right] \quad (2)$$

The barometer equation is valid within a few kilometers of the earth's surface, within which the lapse rate, gravity acceleration and air composition can be considered constant, given that P_b and T_b consistently refer to the reference height h_b. According to the barometric formula, a 1 mbar difference

in pressure, with a 15 °C ambient temperature, leads to a 8.33 m altitude change, while a 1 m change of altitude leads to a 0.1201 mbar change in pressure.

1.2. Contribution

The contributions of this paper can be summarized as follows:

- We introduce a novel infrastructure-independent method for the dynamic vertical mapping.
- We introduce a novel approach for the reference pressure estimation through the identification of the outdoor–indoor transition (OITransition) of the user through the fusion of three different sensors. In this way, the need of calibration between sensors becomes obsolete.
- We propose an enhancement of the CityGML level of detail two plus (LoD2+) method that provides the indoor geometry of buildings at lower levels of detail.

This paper is an extension of work already presented in [15]. More specifically, in this paper, we have extended the approach, by including an additional sensor for the OITransition discovery. This additional sensor is the magnetic sensor, and more information about it is available in Section 3.3.3. Additionally, the method has been extended and sensor fusion functionality has been added in the reference pressure area component. More information is available in Section 3.3.4. Moreover, the evaluation has been extended with additional collected data over longer period of time, as can be seen in Section 4. Finally, as a result of the above-mentioned extensions of our approach, we have achieved a more accurate identification for the recognition of the OITransition discovery with a true positive score of 99.3% instead of 94.2% in the past. This makes our method more robust against various building characteristics.

1.3. Paper Structure

In this paper, the related work is introduced in Section 2; the approach is described in Section 3; the evaluation is presented in Section 4; the paper concludes in Section 5, where limitations to validity are also presented; the resulting models are presented in Appendix A; and the list of collected data is presented in Appendix B.

2. Related Work

Enhancing CityGML models with indoor geometry has already been discussed in [11]. In this study, the LoD2+ method was introduced. The method is robust and was implemented successfully using Nef Polyhedra. However, the authors used some prior knowledge, such as building facades and available data modeled following the LoD2 format. As a result, this method is not applicable to general cases because not all buildings contain sufficient information that can be used for mapping indoor areas.

Apple holds a patent that focuses on the visualization of information in indoor 3D places [16]. They do not consider altitude estimation, but instead they assume the existence of indoor maps with locations that specify where vertical transitions may occur, annotated on the map, and a two-dimensional localization mechanism. Additionally, they assume that users can be localized in a particular floor using a particle filter-based framework, which is responsible for assessing the probability of a vertical transition. In this framework, the confidence is quantified on the basis of WiFi access points and the receive signal strength.

Kaiser et al. [17] point out the need of detecting vertical transitions because of the limitation of the Zero Velocity Update (ZUPT) algorithm to identify vertical displacements. To solve this problem, they introduce a moving platform detection module. This works by combining accurate sensors, not those available on a smartphone, such as an accelerometer, barometer and magnetometer. These use ZUPT for localization and a Simultaneous Localization and Mapping (SLAM) algorithm for reducing the remaining drift. They estimate altitude using the barometric sensor, while they also use it to identify landmark phases. In addition, they attempt to identify the boundaries of vertical movements. The

intuition for the use of acceleration for the detection of vertical transitions is that the acceleration caused by external factors is weaker than that caused by the pedestrian. However, their approach focuses on correcting real-time localization and assumes the existence of indoor maps.

Li et al. [7] suggest using barometers for 2.5-dimensional (2.5D) (floor-level) localization. They examine how the barometric formula performs for altitude determination. They researched the robustness of altitude estimation on different devices that record differences from 2.1 to 2.5 hPa, which is translated to an offset of multiple floors. They noted that the variation of pressure over 2 h could reach an equivalent of a 10 m height change. They also examined latency robustness as well as stability in the short term, where they noticed changes of 0.1 hPa every 10 min. On the basis of their experiments, they argue that it is impossible to accurately determine height using a barometer in an indoor environment in an absolute manner. They strongly point out the necessity of a reference station. In their study, they used a reference station 5 km away. However, a reference station is not always available, and using other devices such as reference stations requires calibration, which is not realistic in a real-world scenario.

Xia et al. [6] propose the use of multiple barometers as reference points for the floor positioning of smartphones with built-in barometric sensors. This method does not require knowledge of the accurate heights of buildings and storeys. It is robust against temperature and humidity, and it considers the difference in the barometric pressure-change trends and different floors. The intuition is that atmospheric pressure decreases as the altitude increases. Hence, pressure changes that correspond to altitude changes are possible to be calculated using a reference pressure and the barometric formula. As they argue, humidity does not significantly affect the accuracy of the system for indoor altitude estimation; thus, they use the gas constant for dry air and the air molar mass of dry air instead of humid air. On the basis of the barometric formula and using built-in barometric sensors of smartphones as well as information from a local weather station, they are capable of achieving a good discretization between different floor levels. For the current temperature, they consult a local weather station online service. However, this approach is heavily dependent on dense existing infrastructure, while it focuses only on localization and assumes the existence of maps, which describe the location of each sensor.

Bollmeyer et al. [18] use barometers for medical applications in which a precise altitude estimation of the patient's body is needed. A challenge in this case is the disturbances due to macroscopic flow, such as the influence of ventilation, the opening and closing of doors, or the weather. Calibration between sensors is also needed, in order to compensate for the offset between different sensors. In their research, they created a small sensor network, with sensors attached to the patient body, as well as a reference stationary sensor. They measure a maximum error of 21 cm, but they suggest that a second sensor might reduce the maximum error to 10 cm. However, in our application scenario, we do not focus on such accurate vertical localization; we are looking for an infrastructure-independent approach.

Liu et al. [14] argue that the estimation of altitude via GPS is applicable only outdoors, although even there, its error can be 2.5 times the error of the horizontal location. As a result, they suggest barometers for vertical localization. Their main limitation is the lack of reference points, because the only available reference stations are meteorological stations, which are often sparsely located, while they broadcast periodically, usually at 1 h intervals. Therefore, they introduce the concept of ad hoc reference points. They integrate information from multiple points, while they also use forecast models to estimate air pressure on demand. Besides reference meteorological stations, they additionally use other smartphones when the elevation indication is accurate enough. In order to retrieve better accuracy from other phones, first, they take into account all the reference points that are within a specified distance and time period, and then they give higher weights to reference stations that are closer in distance as well as in time. They also assign a different credibility to different reference stations. Hence, a reference station will be more reliable if its location is known and can report better pressure. They score errors of less than 3 m in outdoor walking, 6 m in mountain climbing, and 0.9 m in indoor floor localization. However, an ad hoc reference sensor reading will constantly

have the need of being extracted; it is not clear how this can be achieved, particularly without maps that describe those reference locations.

3. Approach

In this section, we present the main components of our approach. As visualized in Figure 1, the approach is composed of the **Sensor Data Collection** module, which collects the data from smartphone users via an application that has been developed for the purpose of this research and can be found in ref. [19]. After smartphone pressure sensor data are collected, noise is filtered out in the **Signal Filtering** module. The **Reference Pressure Extraction** module has two roles: (1) to filter outdoor data, and (2) to identify locations where pressure readings can be extracted. In the **Stair Removal** module, features that belong to intermediate heights (i.e., stairs or elevators) are rejected. Remaining pressure readings are later used in the barometric formula for **Altitude Estimation**. In the **Data Aggregation** module, we combine data from multiple users, while the **Floor Estimation** module has two roles: (1) to identify the number of floors in a set, and (2) to estimate the altitude of each floor. Finally, in the **CityGML Generator** module, a **CityGML Model** is dynamically generated.

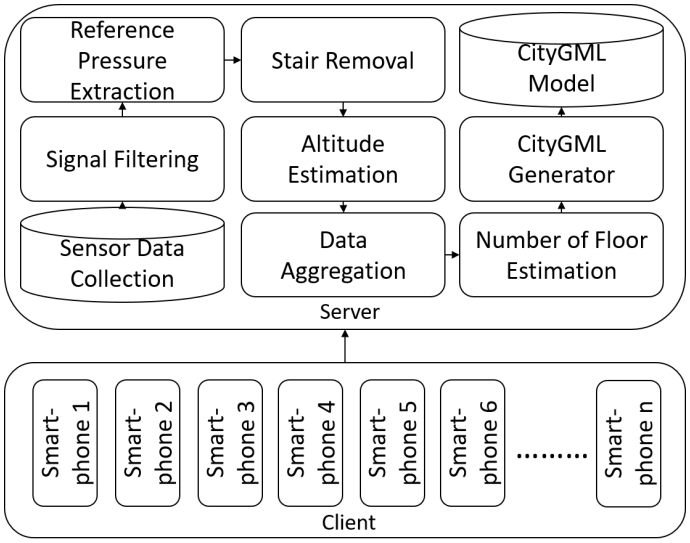

Figure 1. The overall architecture of our system [15].

3.1. Sensor Data Collection

The sensor data collection module collects sensor data from pressure, light, GPS, proximity and magnetic sensors. Data collected during different temperatures, days, times and humidity situations, labeled with a time-stamp and a unique user identifier, are streamed on a server developed for this purpose through a client–server approach via HTTP protocol, in JSON format. Our collected data are openly available in [20].

3.2. Signal Filtering

For smoothing the collected data, the Savitzky–Golay filer [21] is used. Savitzky–Golay is a moving average filter, which applies local regression to a subset of our entire dataset. More specifically, it smooths data by replacing each data point with the average of the neighboring data points within a defined span. This approach is equivalent to

$$y_s(i) = \frac{1}{2N+1} * \Big(y(i+N) + y(i+N-1) + ... + y(i-N) \Big)$$

where $y_s(i)$ is the smoothed value for the ith data point, N is the number of neighboring data points on either side of $y_s(i)$, and $2N+1$ is the span.

3.3. Reference Pressure Extraction

The reference pressure is essential for estimating the altitude differences on the basis of the barometer equation using pressure data. The reference pressure is extracted from areas that fulfil the following preconditions: (1) they are common for all user data of each building, (2) they are located indoors, and (3) the pressure fluctuations are low. Such an area is the one that follows the OITransition, as everyone inside a building was at some point in time outside, while it is located indoors where the pressure disturbances are low.

3.3.1. Light Sensor

As has been already suggested by Zhou et al. [22], the OITransition can be identified by aggregating multiple smartphone sensor data. A very promising sensor for this is the ambient light sensor, considering the fact that there is a difference of the light intensity between indoors and outdoors. For identifying the OITransition, in our research, we fuse light and proximity sensors, with 7 and 25 Hz recording rates, respectively. The first sensor helps us to identify the transition, and the second is used as a supportive sensor, indicating when to trust the data, as it can indicate that an object blocks the light sensor.

As can be seen in Figure 2, the light intensity drops when entering the building during the day and increases during the night, while the proximity sensor indicates whether to trust the light sensor, because of various phone poses (e.g., phone in pocket). Hysteresis thresholding is used for maximizing the margins of the signal that belong outdoors and indoors. Finally binary classification is applied on the basis of the high and low distribution frequency, while the decision of whether the data is collected during day or night taken from the hour angle ω_0 of the sun (negative at sunrise; positive at sunset) is computed with

$$\cos \omega_0 = -\tan \phi \cdot \tan \delta$$

where ϕ is the latitude of the observer on the earth and δ is the sun's declination.

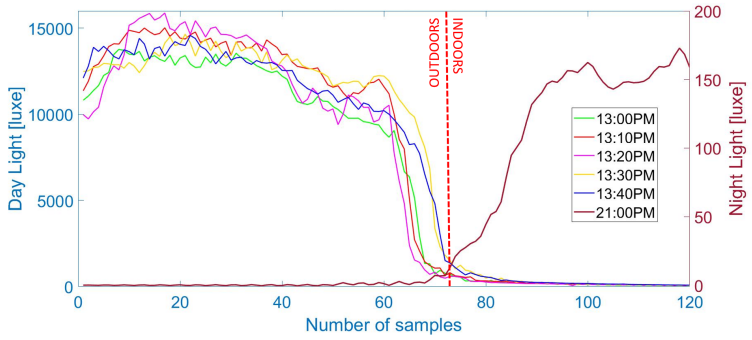

Figure 2. Light data from six outdoor–indoor transitions (OITransitions) collected during the same day, five during day time and one during night. As can be seen, during the OITransition (after the 70th sample), the light intensity rapidly decreases during the day (**left axis**) and increases during the night (**right axis**).

Hysteresis Threshold

The hysteresis thresholding algorithm uses multiple thresholds to find rapid changes in a signal. The algorithm is thus used to discriminate indoor and outdoor locations. Figures 3 and 4 show that it allows the identification of OITransitions with great accuracy. First, we estimate the upper and lower thresholds for the hysteresis thresholding, on the basis of a histogram analysis. In the histogram analysis, we compute frequency distributions of discrete light intensities. We select the upper and lower thresholds on the basis of the pattern of the distribution. If an OITransition exists in the sensor data segment, then the distribution forms a bimodal pattern and the thresholds are selected from the lower and higher peaks of the distribution. Alternatively, if the sensor data segment contains an OITransition, the distribution shows a symmetric pattern and it is not be taken into consideration as a potential OITransition segment. After the upper and lower thresholds have been defined, the upper threshold is used to find the start of a rapid transition. Once a start point is found, then the path is traced from the rapid signal transition through the signal, segment by segment, marking indoors whenever it is above the lower threshold. It stops marking indoors only when the value falls below the lower threshold.

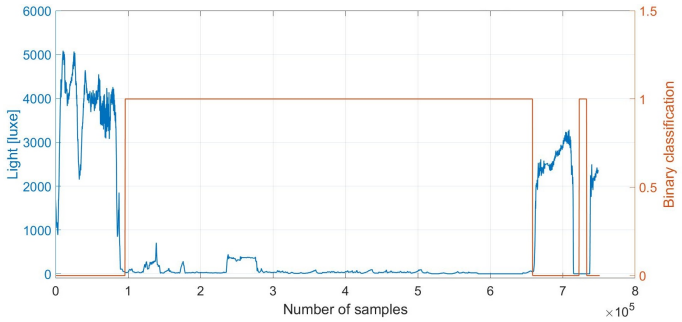

Figure 3. Outdoor–indoor transition (OITransition) classification using light. The binary flag of **1** (orange line and right axis) indicates indoor area. We note that during the period after sample 7×10^5, the smartphone was in a pocket. However, it is wrongly classified as indoors. This demonstrates the need for fusion with the proximity sensor, which can indicate whether the phone is exposed (the light sensor can be trusted) or not.

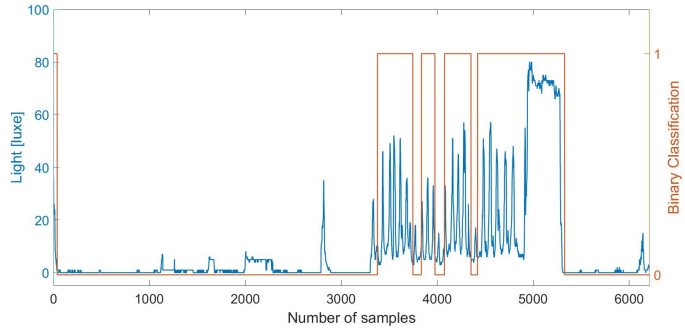

Figure 4. Outdoor–indoor transition (OITransition) classification using light at night. The binary flag of **1** (orange line and right axis) indicates indoor area.

Unfortunately, as can be seen in Figure 4, the accuracy of the indoor classification at night-time is reduced in comparison to during the day time.

3.3.2. GPS Uncertainty

Another characteristic of the OITransition is the rapid increase of the GPS uncertainty. As a result, in our approach, we recorded the GPS uncertainty with a sampling frequency of proximately 1 Hz. As can be seen in Figure 5, at the moment of the transition (after the 80th sample), the GPS uncertainty increased from less than 10 m to almost 60 m. Hysteresis thresholding [23] was applied for the maximization of the margin between low GPS accuracy (indoors) and high-accuracy data (outdoors) for better classification. More specifically, GPS uncertainty was first smoothed via a Gaussian smoothing filter. Then multiple hysteresis thresholding was applied in order to enhance the margin and hence the accuracy of the OITransition classification. The approach can be seen in Figure 6. As can be seen in the figure, raw GPS uncertainty (red line) was first smoothed with a Gaussian filter. Then hysteresis thresholding was applied to the smoothed signal (magenta line).

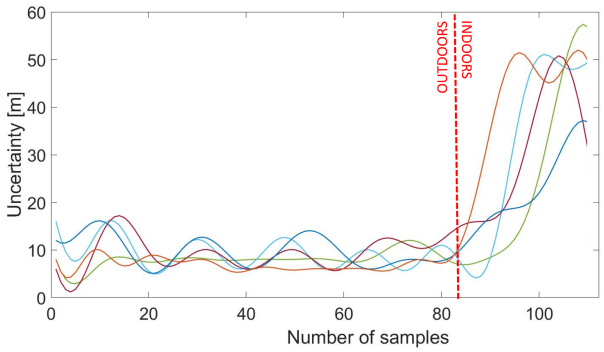

Figure 5. GPS uncertainty data from five outdoor–indoor transitions (OITransitions). As can be seen, at the moment of the transition after the 100th sample, the uncertainty rapidly increased.

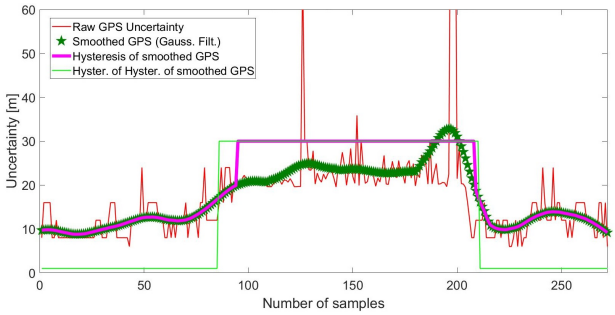

Figure 6. Smoothing and hysteresis thresholding of raw GPS uncertainty signal.

OITransition Detection and Histogram Analysis

Before hysteresis thresholding was applied, the raw GPS uncertainty signal was smoothed via a Gaussian smoothing filter. Then multiple hysteresis thresholding was applied to enhance differences between segments of the signal that belonged outdoors or indoors. This approach is detailed explained in Section 3.3.1 and visualized in Figure 6.

For the identification of an OITransition in the data segment, as well as for the definition of the threshold in the hysteresis thresholding, histogram analysis was applied in the entire GPS uncertainty signal segment. As can be seen in Figure 7, the frequency of different uncertainty radii is visualized in the histograms. As can be seen in Figure 7c, the histogram forms a bimodal pattern when an OITransition occurs. This is a recognizable characteristic of a segment of uncertainty data that contains

an OITransition. Once a transition is identified, the two peaks of the signal are used as the upper and lower thresholds in the hysteresis thresholding algorithm.

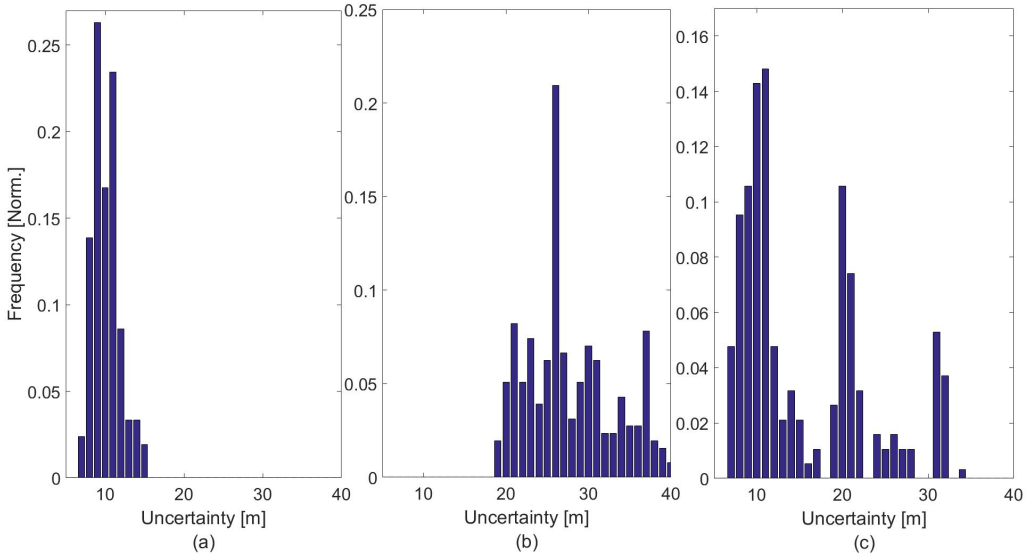

Figure 7. Frequency of GPS uncertainty from data collected from outdoors (a), indoors (b) and during an OITransition.

On the other side, as can be seen in Figure 7a,b, the histograms show a more symmetric pattern, which is an indication that the data are extracted from a single place; this place is either indoors or outdoors. More specifically, as can be seen in Figure 7b, the GPS uncertainty is high—more than 20 m—which is an indication that the particular segment has been extracted from exclusively indoor locations. On the other side, as can be seen in Figure 7a, the GPS scores a low uncertainty—less than 15 m—which is an indication that the data are extracted from exclusively outdoor locations.

3.3.3. Magnetic Signal

The magnetic sensor can detect disturbances of the ambient magnetic field, as a result of steel elements inside the walls of a building. Hence, the intensity of the magnetic field can be used as an indicator for identifying the OITransition [24]. In this section, we introduce a process for the identification of the OITransition by measuring the disturbances of the magnetic field. For the identification of OITransitions, we combine a Gaussian filter and the moving window standard deviation. In the following example, we selected a magnetic dataset from the collected data [20] from four of our buildings. The corresponding magnetic signal is shown in Figure 8:

The route that corresponds to the signal shown in Figure 8 begins outdoors, followed by four indoor transitions and four outdoor transitions. Towards the end of the time interval, the third outdoor transition occurred when exiting the fourth building.

In the first step, disturbances in the signal were found using the moving window standard deviation with a window size of 20 samples along the time axis. The resulting signal is shown in Figure 9 (orange line). Once disturbances were identified, a second moving standard deviation extraction was applied to the new generated signal. This time, the window size corresponded to 200 samples. The result is illustrated in Figure 9 (purple line).

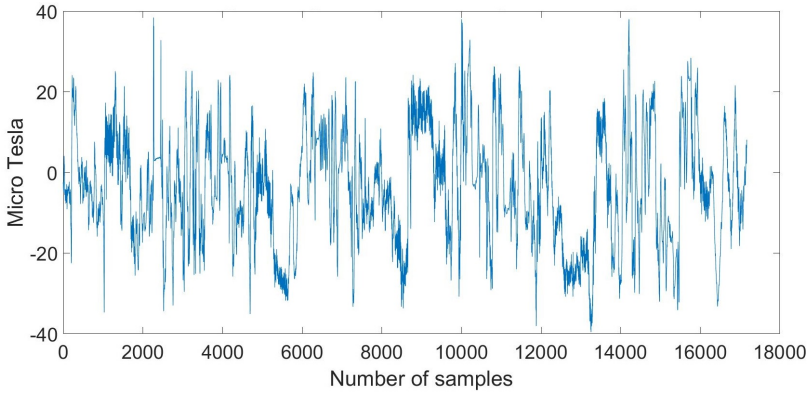

Figure 8. Magnetometer signal from walking into four consecutive buildings.

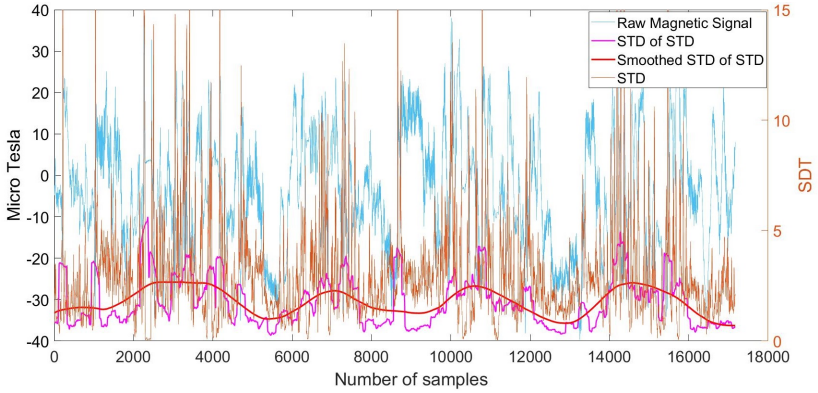

Figure 9. Magnetometer signal from walking into four consecutive buildings and corresponding smoothed moving Standard Deviation (STD) of moving STD with kernel size of 500.

In the third step, a Gaussian filter was applied to the resulting signal in order to smooth it with a kernel of 500 samples. As can be seen in Figure 9 (red line), this contributed to the identification of the four blobs that correspond to the duration—one by one—of the indoor walking activities. They then could be used to distinguish indoors from outdoors.

Finally, in order to enable binary classification between indoor and outdoor areas, a moving STD was performed, followed by another Gaussian filtering step. The resulting signal was then used to determine the start and end of the indoor areas. The final classification can be seen in Figure 10 (black line), where the value 1 corresponds to indoors and 0 corresponds to outdoors.

3.3.4. Fusion

The sensor fusion was made as is described in Table 1. The sensors that have been taken into consideration are the proximity, the light, GPS and the magnetic field sensor. Their decision is fused as follows:

- If the proximity sensor indication is false, this implies that there is no obstacle blocking the light sensor. As a result, three sensors are available. Hence, the result is determined on the basis of the voting fusion. For example, if the light and GPS sensors identify that the particular data segment is extracted from indoors, then the segment is classified as an indoor data segment.
- On the other side, if the proximity sensor indicates "true", then we have only two sensors available. The majority voting can thus not be applied here. Hence, in such a case, the logic

operation *and* is applied. For example, if the magnetic sensor indicates disturbances—and as a result, indoors—but the GPS uncertainty is low, which indicates outdoor space, then the segment is classified as outdoors.

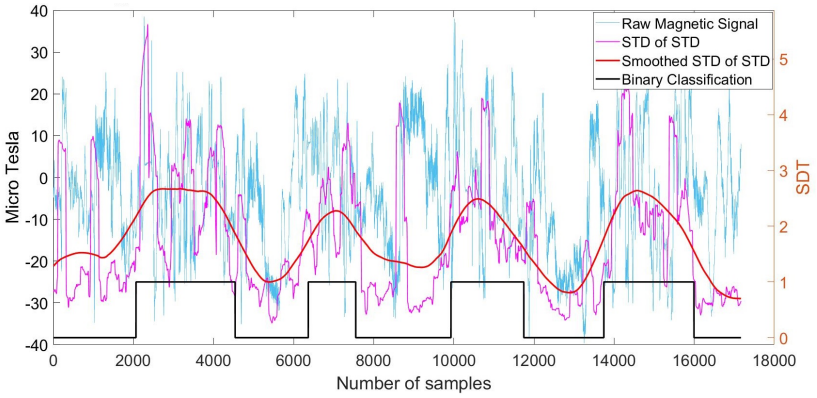

Figure 10. Magnetometer signal from walking into four consecutive buildings and corresponding smoothed moving STD of the disturbance, with kernel size of 200 samples, and the final binary classification.

Table 1. Fusion Rules.

Proximity	Light	GPS	Magnetic	Indoor	Outdoor	Fusion Model
False	0	0	0	F	T	Voting
False	0	0	1	F	T	Voting
False	0	1	0	F	T	Voting
False	0	1	1	T	F	Voting
False	1	0	0	F	T	Voting
False	1	0	1	T	F	Voting
False	1	1	0	T	F	Voting
False	1	1	1	T	F	Voting
True	—	0	0	F	T	*and*
True	—	0	1	F	T	*and*
True	—	1	0	F	T	*and*
True	—	1	1	T	F	*and*

3.4. Stair Removal

In the stair removal phase, sets of features with high disturbances in the pressure readings are rejected, as they mostly correspond either to vertical transitions (e.g., stairs or elevators) or to outliers (e.g., high wind velocities). Such features of high disturbance are identified using the moving window standard deviation.

This approach is equivalent to

$$\sigma = \sqrt{\kappa}$$

where

$$\kappa = \sigma^2 = \frac{1}{N-1}\left(q - \frac{s^2}{N}\right)$$

with

$$q = \sum_{i=1}^{N} x_i^2 \qquad \text{and} \qquad s = \sum_{i=1}^{N} x_i$$

where x_i is the instance of the input signal and N is the number of elements.

3.5. Altitude Estimation

The altitude is estimated on the basis of the barometer Equation (2) as follows:

$$h = \left[\left(\frac{P_0}{P_i} \right)^{\frac{1}{5.25}} - 1 \right] * \frac{T_b + 273.15}{0.0069} \tag{3}$$

where P_0 is the reference pressure extracted from the location where the OITransition was identified, P_i is the current pressure value and T_b is the temperature value in °C, which is extracted via openly available weather stations online.

3.6. Data Aggregation

Data aggregation is essential for identifying all floors inside a building, as not all users are expected to visit all floors. In the data aggregation module, multiple recorded data are fused. Grouped by their GPS coordinates and combined with the building outline, extracted from OpenStreetMap [25], it is ensured that the data always correspond to the same building. More specifically, altitude information estimated from multiple users and labeled by their unique users identifier (UUID) are sorted by their time-stamp and fused together for the classification phase. Because the reference pressure for the altitude estimation is extracted by the same device as that used for estimating it, approximately at the same location for all the users, because of the novel approach for reference altitude extraction on the basis of the identification of the OITransition, there is no need to calibrate any sensor between different phones. In this paper we consider all existing entrances of a building to be at the same altitude. However, in the case of multiple entrances at different altitudes, the entrance altitude, as well as the longitude and latitude, can be extracted from [25], and then the OITransition can be used for the identification of the entrance location. Once the entrance location is identified, the difference between the global altitude of the entrance can be used for locally referring the floor height.

3.7. Number of Floors Estimation

Because the number of floors as well as the label of every floor (i.e., the corresponding altitude) are unknown, for classification, we used a classifier able to cope with unlabeled data. The classifier K-means was selected because of its simplicity and its relatively low processing demand. For estimating K, the elbow method was selected. The classification process is divided into two main steps. The first step is the identification of K, which corresponds to the true number of floors. In the second step, the center of each cluster is recognized, which corresponds to the altitude of every floor.

3.7.1. Identification of K

Because the number of floors is unknown (K), it has to be estimated in the first step. For this purpose, the elbow method [26] was chosen. The elbow method is a clustering analysis method, and it enables the interpretation and validation of the consistency within the cluster analysis. It takes into consideration the percentage of variance explained as a function of the number of clusters: the optimum number of clusters is reached when adding another cluster no longer improves the modeling of the data. If we plot the variance as a function of the number of clusters, the first clusters will add much information, but with an increasing number of clusters, the marginal gain will drop and the graph will flatten out, indicating the optimum number of clusters. Identifying the correct number for K is essential, as it corresponds to the number of floors. A wrong estimation of K can lead to large errors in the estimated altitude of each floor.

3.7.2. The Centroid of the Clusters

After K is identified, the classification is made using K-means, as the cluster label (i.e., the altitude of each floor) is unknown. The input to the algorithm is the computed vector of filtered pressure data

Sensors **2018**, *18*, 480

and the estimated number of floors. The algorithm's output is then a vector with the assigned classes for every input point and the cluster centroids.

3.8. Implementation in CityGML

In our research, we concentrate on the derivation of the floor numbers and their heights. This does not allow us to create a complete LoD4 model. As a result, we enhance the LoD2 model geometry with the hull geometry for each floor. For this purpose, we introduce LoD2+, as visualized in Figure 11.

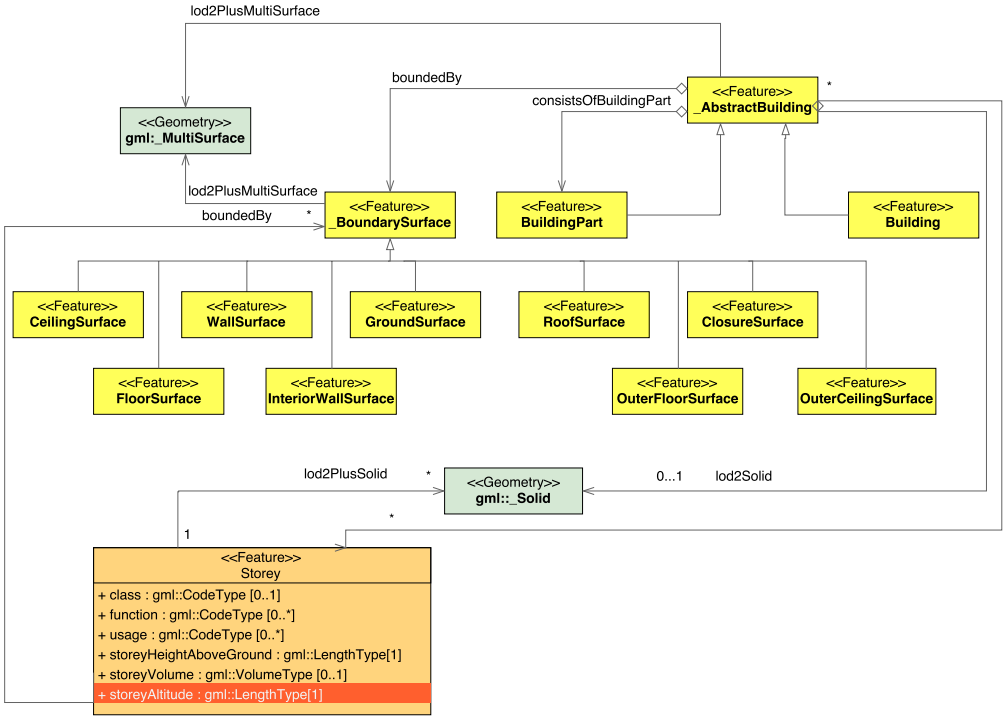

Figure 11. The proposed level of detail two plus (LoD2+) model, which carries information about the number of stores as proposed by [11] and their corresponding altitudes.

In LoD2 and higher LoDs, the outer facade of a building can be modeled semantically by the `_BoundarySurface`. The `_BoundarySurface` is a part of the building's exterior shell with an assigned function such as the wall `WallSurface`, roof `RoofSurface`, ground plate `GroundSurface`, outer floor `OuterFloorSurface`, outer ceiling `OuterCeilingSurface` or `ClosureSurface`. For indoor modeling `FloorSurface`, `InteriorWallSurface`, and `CeilingSurface` can be used [27]. In [11], the authors enhance the CityGML scheme with a new feature class, Storey, which has five attributes: class, function, usage, `storeyHeightAboveGround` and `storeyVolume`.

To model the indoor geometry, we keep the LoD2 representation using `_BoundarySurface` and add indoor geometry for each storey using `FloorSurface`, InteriorWallSurface, and CeilingSurface, as well as the feature class Storey introduced by [11]. In addition, we propose a further attribute of the feature class Storey: `storeyAltitude`. This attribute is necessary for our application, as the output of a navigation device is an altitude and not the height above the ground. This extension is not included in the current version of the CityGML specification, however we suggest to include it in the next release.

For the dynamic generation of the CityGML model, citygml4j [28] was used. This is an open-source library for Java, which binds the XML Schema definitions of CityGML to a Java object model.

4. Evaluation

In this section, we present the evaluation of the proposed method for the dynamic vertical mapping from user smartphone data as shown in Table 2. More specifically, in Section 4.1, the difference in calibration between the two phones used in this experiment is presented. Section 4.2 presents the robustness of our algorithm against various human walking velocities. In Section 4.3, the performance of the identification of OITransitions is evaluated. In Section 4.4, the evaluation of the identification of the number of floors and their altitude estimation during various weather conditions is presented. For the evaluation of the stair removal Section 4.2, data were collected from three different human walking velocities. Finally, a detailed evaluation, with datasets collected over a period of 6 months from three different buildings, is presented in Section 4.4.

Table 2. Collected Data used for evaluation. The table shows the date of collecting the data, the time, the indicated temperature from AccuWeather (T A) and Google (T A) (unit: °C), the relative humidity from the same two sources (H A) and (H G), and the ambient pressure from AccuWeather (P A) (unit: Pa). The buildings belong to the Technical University of Munich (TUM) main campus area and are (1) Agness 27, (2) Adelheid 13A, (3) Agness 33 and (4) TUM main campus.

Date & Time	T A	T G	H A	H B	P A	ID
May 10, 10:20	9	10	70	74	1011	1
May 10, 21:40	11	13	61	45	1006	1
May 12, 18:20	21	19	40	52	1004	1
May 9, 17:00	10	9	49	52	1016	1
May 9, 10:40	8	9	75	72	1017	2
May 9, 17:30	10	11	49	55	1016	2
May 10, 22:00	11	11	61	65	1006	2
May 12, 18:30	21	19	40	45	1004	2
May 9, 10:10	8	9	75	60	1017	3
May 9, 16:40	10	9	49	59	1016	3
May 10, 10:00	9	8	70	40	1011	3
May 12, 17:50	21	19	40	43	1004	3
Feb 11, 14:30	6	2	70	72	1019	3
Feb 12, 19:00	0	1	87	80	1028	3
Feb 21, 21:30	7	0	93	83	1017	3
Mar 21, 13:30	13	8	58	64	1010	3

4.1. Different Phone Calibration

In this section, we discuss the use of our algorithm for two different smartphones. As can be seen in Figure 12, there is an offset between the sensor readings of the two phones. This implies that there cannot be a single point of reference for both sensors and highlights the need for calibration between the two phones. However, as can be seen, the offset between the two sensors is almost stable. As a result, this effect demonstrates the need of self-reference that our approach offers. Hence, considering the fact that each phone will extract reference pressure from its own sensor and the fact that the offset between different phones is stable, our proposed approach will work for any given barometric sensor calibrated under any given circumstances.

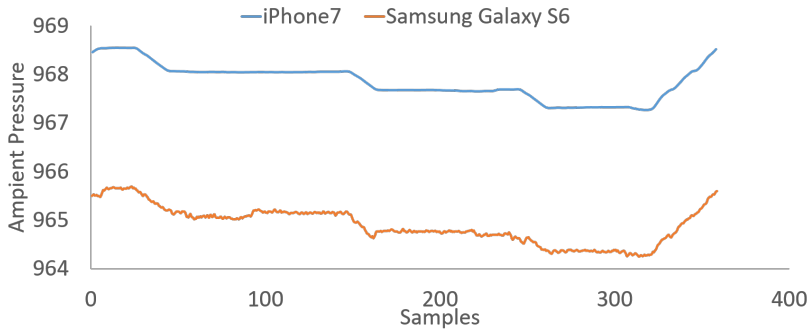

Figure 12. Data collected from an iPhone 7 and a Samsung Galaxy S6, while the user had climbed three floors upwards and the same number of floors downwards.

4.2. Evaluation of Stair Removal

For testing the robustness of our algorithm against different walking velocities in the stair removal component, we recorded data with three different walking velocities, approximately 1×, 1.2× and 1.5×, while climbing five pairs of stairs on a building, as can be seen in Figure 13. As demonstrated in the results (Table 3), the algorithm scored a precision of 94%, recall of 93.8% and F-score of 93.9% on correctly identifying the stairs, with the same sliding window length for all datasets. The sliding window size was 50 samples long or approximately 10 s, while it slid for every sample or approximately every 250 ms.

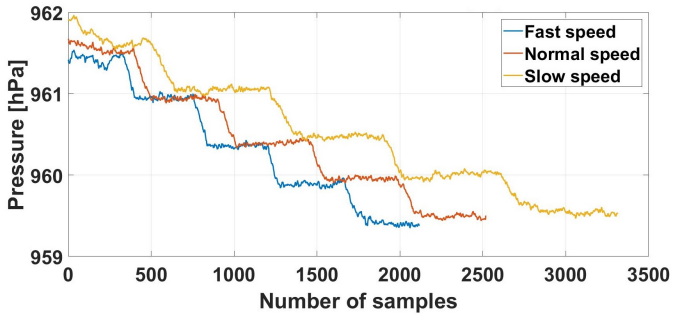

Figure 13. Dataset used for the evaluation of the stair removal method. The data was collected from the same route for three different visits and walking velocities, approximately 1×, 1.5× and 2× [15].

Table 3. Confusion matrix of stair removal.

	Fast		Normal		Slow	
	Floors	**Stairs**	**Floors**	**Stairs**	**Floors**	**Stairs**
Floors	1584	58	2037	0	2683	0
Stairs	179	296	76	404	157	472

4.3. Evaluation of Reference Pressure Extraction

The reference pressure value for the altitude estimation with the barometric formula corresponds to the location that follows the entrance of a building, as detailed described in Section 4.3. As a result, the identification of the OITransition is necessary in order to identify the building entrance. The transition is identified by monitoring peaks and drops by monitoring peaks and drops in the

readings of a number of sensors and their fusion, as suggested by [22] and described in Section 4.3. However, in our scenario, the ambient light, the GPS uncertainty and the disturbances of the magnetic field are taken into consideration, rather than the WiFi Received Signal Strength (RSS) and the Global System for Mobile Communications (GSM) RSS. The approach has been evaluated in three different buildings with four collected datasets for each building, during day and night. Our collected data and the algorithm used for the evaluation are open-source and can be found in [20].

As can be seen in Figure 14, the OITransition (red dots) was successfully identified in all of our datasets. Additionally, the entrance location could also be approximately determined by our approach. This was considered as the place of the transition, for example, the space between the last low GPS uncertainty values and the first high GPS uncertainty values. As a result, we could additionally estimate the spatial error of our approach for the OITransition identification. Hence, the entrance location latitude has been approximated by an average of 1.6 m, while the entrance longitude has been approximated by an average of 5.5 m. This score was lower than the GPS average error outdoors, which was between 10 and 12 m.

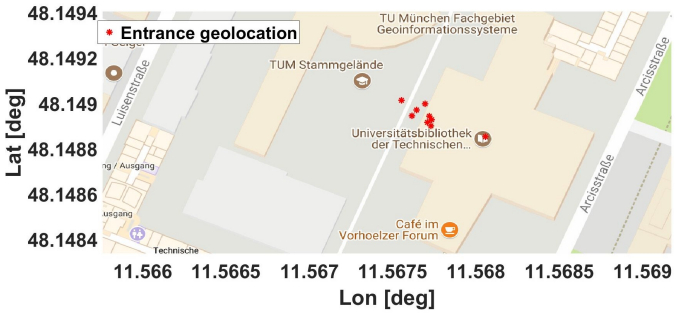

Figure 14. Locations that correspond to the detection of the outdoor–indoor transition (OITransition). The figure includes nine different determined locations for the entrance to the building (red dots) [15].

Furthermore, five out of nine times, the entrance location was identified at the latitude of 48.1489, while two times it was identified at the latitude of 48.14895 and once it was identified at the latitudes of 48.14885 and 48.149. The final latitude was to be decided on the basis of the median, which was 48.14894251, while the true entrance latitude as mapped in the open street maps was 48.1489277. Hence, our algorithm scored an error of 0.00001°, which corresponded to less than 1.64 m. Regarding the longitude, three out of nine times the entrance was localized at the longitude 11.5677, twice at 11.56775 and once at 11.56755, 11.5676, 11.56765 and 11.568. The final entrance location was estimated from the median at 11.568, when the true entrance was located at longitude 11.568. As a result, our algorithm had an error of 0.00004°, which corresponded to approximately 4.614 m.

Finally, in Tables 4–6, a detailed evaluation of the OITransition determination for each sensor and the sensor fusion for all 3 buildings and 12 datasets is presented. According to the tables, our algorithm scored an average of 96.8% for precision, 94.2% for recall and 95.5% for the F-score, for identifying the OITransition using a GPS sensor. It scored 93.6% for precision, 96.3% for recall and 94.9% for the F-score for OITransition detection with a light sensor. It scored 88.8% for precision, 89.2% for recall and 89% for the F-score for OITransition detection with a magnetic sensor. It scored 99.4% for precision, 90.7% for recall and 94.8% for the F-score for the fusion of all sensors on the basis of the voting fusion. When the light sensor was not available or when the proximity sensor indication was true, it scored 99.1% for precision, 97.3% for recall and 98.2% for the F-score.

Table 4. Confusion matrix of Building I.

	GPS		Light		Magnetism		Fusion	
	Indoor	**Outdoor**	**Indoor**	**Outdoor**	**Indoor**	**Outdoor**	**Indoor**	**Outdoor**
Indoor	614	21	6121	210	3323	519	1,162,298	302,269
Outdoor	21	696	163	5526	144	2776	1329	746,993

Table 5. Confusion matrix of Building II.

	GPS		Light		Magnetism		Fusion	
	Indoor	**Outdoor**	**Indoor**	**Outdoor**	**Indoor**	**Outdoor**	**Indoor**	**Outdoor**
Indoor	390	3	5748	1069	2911	460	1,228,507	25,114
Outdoor	20	805	220	4428	0	2383	6915	820,470

Table 6. Confusion matrix of Building III.

	GPS		Light		Magnetism		Fusion	
	Indoor	**Outdoor**	**Indoor**	**Outdoor**	**Indoor**	**Outdoor**	**Indoor**	**Outdoor**
Indoor	127	0	4963	154	7749	179	1,186,784	41,546
Outdoor	29	184	264	3788	1552	3571	13,700	924,258

As a result, we can conclude that the OITransition can be recognized and represents a robust means for the extraction of the reference pressure. Additionally, the GPS sensor scored the lowest false positives, while the light sensor scored the lowest false negatives. Furthermore, the fusion of the three sensors scored the lowest false positive rate, and the false positive rate dropped only by 0.3% when the light and magnetic field sensor were the only sensors that were fused.

4.4. Evaluation

This section presents a long-term evaluation for the number of floors and the floor heights determined for the buildings TUM main campus (library; Section 4.4.1), Building B (Section 4.4.2), and Deutsche Akademie (Section 4.4.3). The ground truth was obtained via a high-precision laser range meter device. We observed these buildings for about 6 months to evaluate the effects of long-term weather conditions on the measurements.

4.4.1. Building 1: TUM Main Campus

TUM main campus has five floors and a ground floor. The true height for each floor is listed in Table 7. Nineteen datasets were collected from TUM main campus, over a 4 month period. The average duration of the datasets collected was 14.2 min, with an average of 3204 samples from the pressure sensor. All the collected datasets are available in [20]. We collected data from various hours during daylight and night; different routes were traveled inside the building, at different temperatures, humidity levels and ambient pressure, and finally with different cloud coverage. After smoothing and clustering the data as explained in Section 3.2, the OITransition was identified as described in Section 3.3. The accuracy of this component is presented in Section 4.3, in Table 4. Once the OITransition was estimated and the reference pressure was extracted, the altitude of every pressure reading that belonged to indoors was computed. Once all the pressure readings were translated into altitude, they were imported to the elbow method for floor number identification.

Table 7. Ground truth, estimated altitude and error for Technical University of Munich (TUM) Main Campus.

Floors	**0**	**1**	**2**	**3**	**4**	**5**
Real floor altitude (m)	0	5.3	10.68	15.05	19.47	24.41
Estimated floor altitude (m)	0	4.81	10.03	14.48	18.86	23.74
Error	0	0.48	0.65	0.57	0.61	0.66

As can be seen in the elbow method results, in Figure 15a, the number of floors (i.e., clusters, K = 6) in our dataset has been identified correctly for the aggregated dataset as well as for the May and June datasets, for which all floors of the building were visited. The threshold selected for all datasets was 99.12% for the distortion percentage, and for the clustering, the K-means algorithm was selected. Additionally, it can be seen that for the February dataset, the number of floors predicted was five ($K_{temp} = 4$), as the fifth floor was not visited during this month. For March, the predicted number of floors was four ($K_{temp} = 3$), as the two highest floors were not visited during that month. For April, the predicted number of floors was three ($K_{temp} = 2$). Finlay, in the datasets extracted during July, the predicted number of floors was four ($K_{temp} = 3$), and the third cluster's distortion fell slightly above the 99.12% threshold.

To demonstrate the performance of the altitude estimation or the label of each class (i.e., the centroid of each cluster), the corresponding estimated floor altitude is visualized together with the ground truth in Figure 16a and is listed in Table 7. As can be seen for the aggregated dataset, the maximum error was at 0.66 m, while the minimum error was at 0.48 m. In Figure 16a, it can also be seen that the fact that some datasets were non-visited floors (i.e., July, April, March and February) did not cause a problem to our database, as these floor altitudes were ignored.

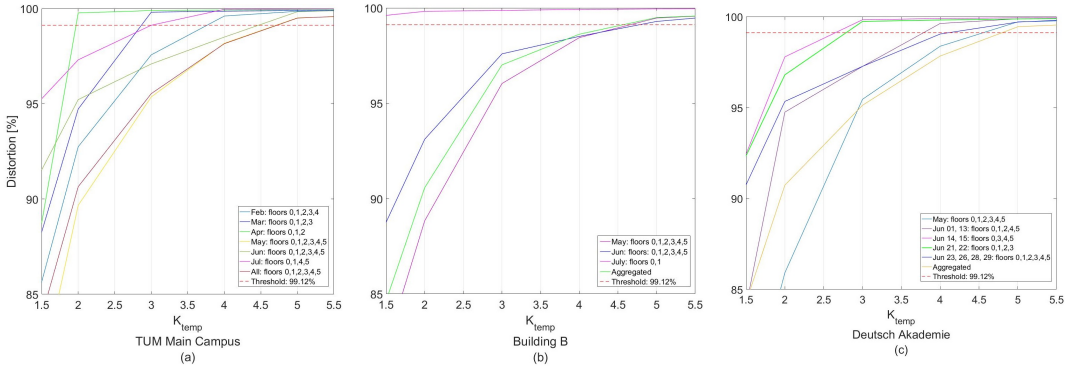

Figure 15. Elbow method result for three test buildings (**a**) TUM Main Campus, (**b**) Building B, (**c**) Deutsche Akademie.

4.4.2. Building 2: Building B

Building B consisted of five floors and an additional ground floor. The true height for each floor is available in Table 8. We have collected data following the same strategy as mentioned above. Twenty-five datasets were collected from Building B in Munich. All collected datasets are available in [20]. After following the procedure described above, for smoothing, clustering and identifying the OITransition, we extracted the reference pressure and then the altitude of every pressure reading that belonged indoors. Finally, once we translated all the pressure readings into altitudes, they were imported to the elbow method for floor number identification.

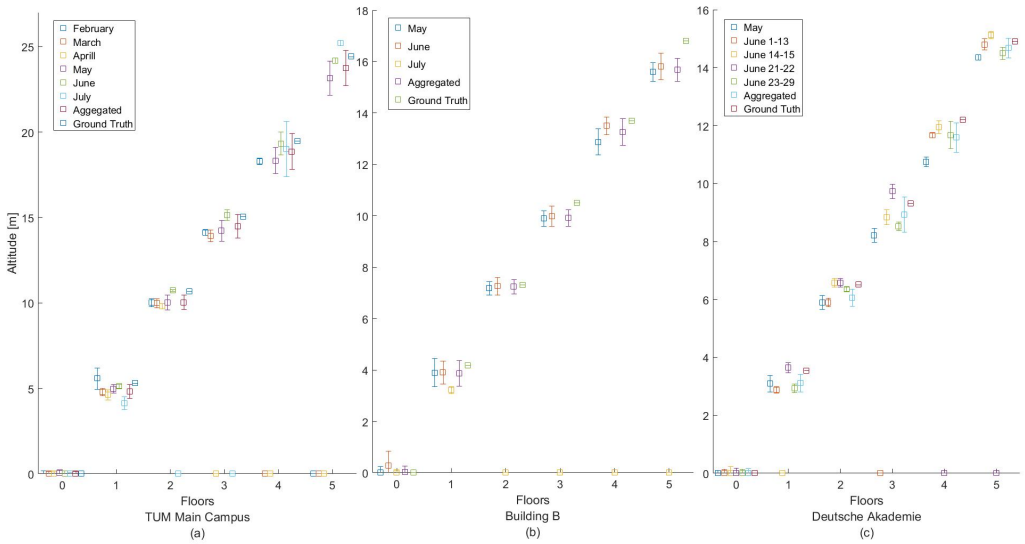

Figure 16. Estimated altitude and ground truth for each floor height for three test buildings (**a**) TUM Main Campus, (**b**) Building B, (**c**) Deutsche Akademie.

Table 8. Ground truth, estimated altitude and error for Building B.

Floors	**0**	**1**	**2**	**3**	**4**	**5**
Real floor altitude (m)	0	4.17	7.31	10.5	13.7	16.8
Estimated floor altitude (m)	0.022	3.86	7.24	9.92	13.26	15.68
Error	0	0.31	0.073	0.585	0.44	1.12

As can be seen in the elbow method results, in Figure 15b, the number of floors (i.e., clusters, $K = 6$) in our dataset has been identified correctly for the aggregated dataset as well as for the May and June datasets, for which all floors of the building were visited. The threshold selected for all datasets was 99.12% for the distortion percentage, and for the clustering algorithm, the K-means algorithm was selected. Additionally, it can be seen that for the July dataset, the number of floors predicted was two ($K_{temp} = 1$), as the four higher floors were not visited during this month.

Regarding the height estimation, the corresponding estimated floor altitude and ground truth are presented together in Figure 16c, as well as in Table 8. As can be seen, in the aggregated dataset, the maximum error was at 1.12 m, while the minimum error was at 0.31 m. In the figure, it can also be seen that for the July dataset, only two floors were visited.

4.4.3. Building 3: Deutsche Akademie

We collected 20 datasets from Deutsch Akademie. This consists of five floors and a ground floor. All collected datasets are available in [20]. The true height for each floor is available in Table 9. Once we estimated the OITransition and extracted the reference pressure, then the altitude of every pressure reading that belonged indoors was computed. Once all the pressure readings were translated into altitude, they were imported to the elbow method for floor number identification.

The ground truth of this building is illustrated in Table 9.

Table 9. Ground truth, estimated altitude and error for DeutschAkademie.

Floors	**0**	**1**	**2**	**3**	**4**	**5**
Real floor altitude (m)	0	3.54	6.51	9.31	12.2	14.9
Estimated floor altitude (m)	0	3.1	6	8.9	11.59	14.67
Error	0	0.4	0.45	0.38	0.61	0.23

As can be seen in the elbow method results, in Figure 15c, the number of floors (i.e., clusters, $K = 6$) in our dataset has been identified correctly for the aggregated as well as the May and June datasets, for which all floors of the building were visited. The threshold selected for all datasets was 99.12% for the distortion percentage and the clustering algorithm was selected for the K-means algorithm. Additionally, it can be seen that for the June 1 and 13 datasets, the number of floors predicted was five ($K_{temp} = 4$), as the third floor was not visited during this period. For June 14 and 15 as well as for June 21 and 22, the predicted number of floors was four ($K_{temp} = 3$), as the two floors were not visited during these period. More specifically, the non-visited floors were the first and second, for the first dataset and the two highest floors for the later dataset.

On the other hand, the corresponding estimated floor altitude and ground truth are visualized together in Figure 16c and Table 9. As can be seen for the aggregated dataset, the maximum error was at 0.61 m, while the minimum error was at 0.23 m. In the figure, it can also be seen that some datasets being non-visited floors (i.e., June 14–15 and 23–29) did not cause a problem to our database, as these floor altitudes were ignored.

5. Conclusions

This paper describes our novel framework for the dynamic mapping of the vertical characteristics of a building. The proposed method makes use of a new sensor available in the latest smartphones (the last from 2017), and the barometric sensor, which indicates the ambient pressure and manages uncertain sensor data collected from crowdsourcing. The method estimates the altitude of the collected data with the use of the barometric formula. For achieving this, we introduce a novel approach for the extraction of the reference pressure at the OITransition of the user, which is identified through sensor fusion. More specifically, the GPS uncertainty, the magnetic disturbances and the ambient light are taken into consideration for identifying the transition, while the proximity sensor is also used as a supportive sensor. We faced an unsupervised classification problem, in which the number of floors—or the number of clusters—as well as the altitude—or the label of each class—for each floor were unknown. To resolve this problem, a clustering analysis technique called the elbow method and the popular K-means clustering algorithm were used. Finally, we propose a way to map these characteristics by enhancing the standards of the CityGML, enabling it to carry information about the vertical characteristics of a building in lower LoDs.

Although it has been demonstrated in the paper that our approach can work with any barometric sensor (Section 4.1), as the offset between different barometric sensors is stable, our approach has been extensively evaluated only for the Samsung Galaxy S6 [29].

Additionally, we noticed that when a significant delay follows the OITrsansition and precedes ascending to different floors, the vertical localization error increases. This is due to the long-term instability of the ambient air pressure. The same happens when there is lack of data from one floor. It is very likely that such data will not be taken into consideration in the clustering analysis and finally in the clustering phase. This will result in a missing floor in the final model.

Acknowledgments: This work is part of the TUM Living Lab Connected Mobility project and has been funded by the Bayerisches Staatsministerium für Wirtschaft und Medien, Energie und Technologie.

Author Contributions: G.P. and C.P. conceived and designed the experiments; O.R.M.R. conducted experiments; D.I. conceived the CityGML extension; G.P. and O.R.M.R. analyzed the data; U.H. provided consultations on physical background; G.P. wrote the paper.

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

Appendix A. Appendix I: Final Models

In this section, the final models together with photographs of the buildings are presented. Those models have been generated dynamically and without the intervention of a user. The outline of the buildings has been extracted from [25]. The CityGML models are available in [20].

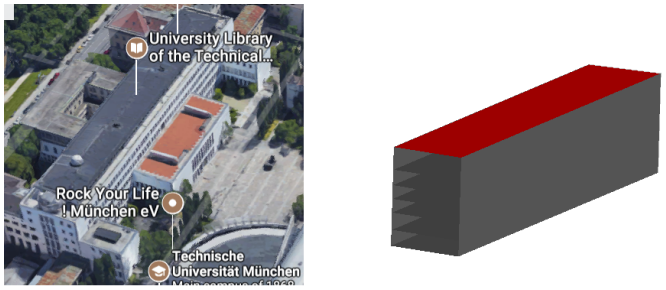

Figure A1. Final CityGML level of detail two plus (LoD2+) model for the TUM main campus building.

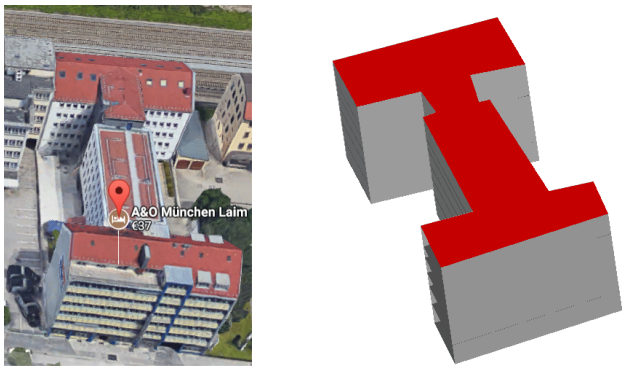

Figure A2. Final CityGML level of detail two plus (LoD2+) model for Building B.

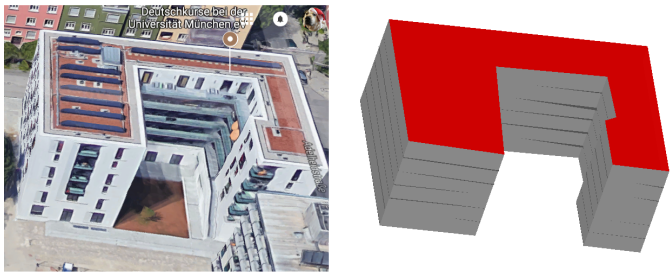

Figure A3. Final CityGML level of detail two plus (LoD2+) model for the Deutsche Akademie building.

Appendix B. Appendix II: Collected Data

In this section, we list all the data that were collected for the evaluation of our model. All the collected data are available in [20], and they include measurements from the following sensors: acceleration, gyroscope, pressure, light, proximity, GPS, magnetometer, pedometer, WiFi, pressure and GSM sensors.

Table A1. Collected data from TUM main campus. The table shows the data acquisition date, the time, the visited floors (V Floors), the indicated temperature from AccuWeather (T A) and Google (T A) (unit: °C), the relative humidity from the same two sources (H A) and (H G), and the ambient pressure from AccuWeather (P A) (unit: Pa).

Date	Time	V Floors	T A	T WC	H A	H WC	P A	P WC	W S	Cloud Cov.
10 Feb	4:30 PM	0, 1	−1	−2	91%	82%	1020	1020	Normal	NG
11 Feb	3:30 PM	0, 1, 3, 4	6	6	87%	84%	1020	1021	Normal	NG
12 Feb	8:00 PM	0, 2	0	1	87%	79%	1028	1027	Normal	NG
27 Feb	4:00 PM	0, 1	14	13	38%	70%	1014	1014	Normal	13%
18 Mar	8:30 PM	0, 3	9	8	81%	64%	1010	1011	Normal	100%
21 Mar	2:30 PM	0, 1, 2	13	13	58%	56%	1010	1010	Normal	90%
26 Mar	12:30 AM	0, 1, 2, 3	4	2	80%	62%	1020	1023	Normal	0%
28 Mar	10:00 AM	0, 1	6	8	60%	69%	1024	1024	Normal	0%
7 Apr	09:30 PM	0, 1, 2	11	9	57%	60%	1022	1023	Normal	20%
11 Apr	11:00 AM	0, 1	9	11	61%	49%	1024	1024	Fast	60%
11 Apr	11:30 AM	0, 2	9	11	61%	49%	1024	1024	Slow	60%
1 May	07:50 PM	0, 3	5	5	86%	90%	1014	1014	Normal	100%
1 May	08:30 PM	0, 1, 2	5	5	86%	90%	1014	1014	Normal	100%
1 May	08:17 PM	0, 1, 2, 3,4, 5	5	5	86%	90%	1014	1014	Normal	100%
1 May	07:35 PM	0, 1	5	5	86%	90%	1014	1014	Normal	100%
17 May	9:00 PM	0, 1	20	21	52%	43%	1018	1020	Normal	0 %
18 May	3:00 PM	0, 1, 3	26	24	44%	55%	1014	1012	Normal	13%
19 May	1:30 PM	0, 1, 4	23	23	56%	53%	1008	1008	Normal	40%
20 May	12:30 PM	0, 1, 5	13	14	58%	49%	1022	1021	Normal	35%
21 May	9:40 PM	0, 2	14	14	71%	65%	1023	1023	Normal	0%
22 May	1:30 PM	0, 2, 3	19	21	59%	48%	1019	1018	Normal	0%
25 May	12:00 AM	0,2,4	16	17	54%	55%	1022	1023	Normal	40%
27 May	11:00 AM	0, 2, 5	20	23	55%	46%	1022	1021	Normal	0%
29 May	6:30 PM	0, 1, 2	29	29	30%	32%	1015	1015	Normal	20%
30 May	6:30 PM	0, 3	29	28	37%	36%	1014	1014	Normal	0%
31 May	6:30 PM	0, 3, 4	25	26	43%	48%	1019	1017	Normal	20%
13 Jun	3:00 PM	0, 3, 5	22	25	40%	43%	1019	1017	Normal	0%
13 Jun	7:40 PM	0, 3, 4	22	24	43%	40%	1017	1017	Normal	20%
15 Jun	5:00 PM	0, 1, 3	28	31	39%	35%	1016	1015	Normal	13%
21 Jun	5:30 PM	0, 2, 3	31	22	73%	37%	1016	1015	Normal	63%
22 Jun	5:30 PM	0, 4	31	22	73%	37%	1016	1015	Normal	63%
28 Jun	7:40 PM	0, 4, 5	23	23	60%	57%	999	999	Normal	20%
1 July	9:30 PM	0, 1, 4	19	19	60%	57%	999	999	Normal	20%
2 July	9:15 PM	0, 1, 5	17	16	77%	84%	1021	1020	Normal	40%

Table A2. Collected data used for evaluation of Building B. The table shows the data acquisition date, the time, the visited floors (V Floors), the indicated temperature from AccuWeather (T A) and the Weather Channel (T WC), the humidity from the same two sources (H A) and (H WC), the ambient pressure from AccuWeather (P A) and from the Weather Channel (P WC), the walking speed (W S) and the cloud coverage.

Date	Time	V Floors	T A	T WC	H A	H WC	P A	P WC	W S	Cloud Cov.
17 May	10:30 PM	0, 1	19	18	55%	45%	1018	1020	Normal	0%
18 May	9:00 AM	0, 1, 2	25	26	59%	62%	1014	1012	Normal	12%
19 May	2:00 PM	0, 1, 3	23	24	56%	53%	1008	1008	Normal	40%
20 May	1:25 PM	0, 1, 4	14	15	54%	44%	1022	1021	Normal	54%
21 May	6:00 AM	0, 1, 5	8	9	87%	88%	1026	1027	Normal	0%
22 May	2:00 PM	0, 2	19	22	59%	46%	1019	1018	Normal	59%
23 May	5:22 AM	0, 2, 3	10	11	93%	84%	1017	1018	Normal	13%
24 May	5:00 AM	0, 2, 4	17	16	67%	71%	1020	1021	Normal	20%
26 May	5:23 AM	0, 2, 5	6	8	100%	92%	1021	1021	Normal	0%

Table A2. *Cont.*

Date	Time	V Floors	T A	T WC	H A	H WC	P A	P WC	W S	Cloud Cov.
27 May	4:00 AM	0, 1, 2	24	26	43%	38%	1021	1020	Normal	0%
28 May	6:00 AM	0, 2, 3	16	17	58%	61%	1019	1019	Normal	20%
29 May	7:30 PM	0, 3	28	28	30%	32%	1014	1011	Normal	20%
30 May	5:00 AM	0, 3, 4	15	19	82%	64%	1015	1014	Normal	0%
31 May	5:00 AM	0, 3, 5	17	17	93%	94%	1019	1019	Normal	100%
12 Jun	5:00 AM	0, 1, 3	15	21	87%	63%	1015	1015	Normal	20%
13 Jun	6:00 AM	0, 2, 3	16	17	58%	61%	1019	1019	Normal	20%
19 Jun	5:00 AM	0, 4	9	13	93%	73%	1023	1022	Normal	20%
21 Jun	6:00 PM	0, 4, 5	28	31	50%	34%	1015	1015	Normal	40%
22 Jun	4:30 PM	0, 1, 4	30	31	34%	34%	1016	1015	Normal	0%
23 Jun	9:00 AM	0,2,4	25	27	60%	52%	1015	1015	Normal	90%
26 Jun	9:00 AM	0, 3, 4	21	21	64%	58%	1016	1015	Normal	0%
27 Jun	7:00 AM	0, 5	17	18	93%	93%	1012	1011	Normal	88%
28 Jun	6:00 AM	0, 1, 5	8	9	87%	88%	1026	1027	Normal	0%
30 Jun	5:00 AM	0, 1	14	13	71%	74%	1007	1007	Normal	20%
4 July	5:00 AM	0, 1	13	15	87%	77%	1022	1022	Normal	20%

Table A3. Collected data used for evaluation of Deutsch Akademie building. The table shows the data acquisition date, the time, the visited floors (V Floors), the indicated temperature from AccuWeather (T A) and the Weather Channel (T WC), the humidity from the same two sources (H A) and (H WC), the ambient pressure from AccuWeather (P A) and from the Weather Channel (P WC), the walking speed (W S) and the cloud coverage.

Date	Time	V Floors	T A	T WC	H A	H WC	P A	P WC	W S	Cloud Cov.
19 May	1:00 PM	0, 1	22	23	60%	56%	1008	1008	Normal	13%
22 May	10:50 AM	0, 1, 2	16	18	67%	56%	1020	1019	Normal	0%
24 May	3:30 PM	0 ,1, 3	17	17	48%	45%	1022	1022	Normal	88%
25 May	11:00 AM	0, 1, 4	16	16	54%	59%	1023	1022	Normal	40%
26 May	6:00 PM	0, 1, 5	22	23	40%	49%	1018	1018	Normal	0%
29 May	5:00 PM	0, 2	29	31	26%	30%	1015	1014	Normal	0%
30 May	6:00 PM	0, 2, 3	28	29	39%	37%	1014	1014	Normal	0%
1 Jun	7:00 PM	0, 2, 4	24	24	46%	46%	1019	1019	Normal	20%
13 Jun	3:30 PM	0, 2, 5	23	26	40%	38%	1018	1016	Normal	0%
13 Jun	7:00 PM	0, 1, 2	23	24	43%	41%	1017	1016	Normal	0%
14 Jun	9:00 AM	0, 3	16	18	67%	57%	1020	1019	Normal	0%
15 Jun	4:30 PM	0, 3, 4	27	30	67%	57%	1020	1019	Normal	0%
15 Jun	5:40 PM	0, 3, 5	28	29	41%	39%	1016	1015	Normal	13%
21 Jun	5:00 PM	0, 1, 3	31	22	73%	37%	1016	1015	Normal	63%
22 Jun	3:00 PM	0, 2, 3	30	33	34%	28%	1016	1015	Normal	13%
22 Jun	8:00 PM	0, 4	29	30	34%	28%	1016	1015	Normal	20%
23 Jun	7:30 PM	0, 4, 5	30	33	34%	28%	1016	1015	Normal	13%
26 Jun	8:30 AM	0, 1, 4	19	21	72%	61%	1016	1015	Normal	13%
28 Jun	7:00 PM	0, 2, 4	23	23	60%	56%	999	999	Normal	95%
29 Jun	2:00 PM	0, 3, 4	19	22	63%	50%	1001	1000	Normal	95%

References

1. Alzantot, M.; Youssef, M. UPTIME: Ubiquitous pedestrian tracking using mobile phones. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Paris, France, 1–4 April 2012; pp. 3204–3209.
2. Grzonka, S.; Dijoux, F.; Karwath, A.; Burgard, W. Mapping indoor environments based on human activity. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA , 3–7 May 2010; pp. 476–481.
3. Wireless E911 Location Accuracy Requirements. Available online: https://www.fcc.gov/document/wireless-e911-location-accuracy-requirements-3 (accessed on 1 February 2018).

4. European Commission—Press Release—Commission Pushes for Rapid Deployment of Location Enhanced 112 Emergency Services. Available online: http://europa.eu/rapid/press-release_IP-03-1122_en.htm?locale=en (accessed on 1 February 2018).
5. European Accessibility Act- Employment, Social Affairs & Inclusion- European Commission. Available online: http://ec.europa.eu/social/main.jsp?catId=1202 (accessed on 1 February 2018).
6. Xia, H.; Wang, X.; Qiao, Y.; Jian, J.; Chang, Y. Using Multiple Barometers to Detect the Floor Location of Smart Phones with Built-in Barometric Sensors for Indoor Positioning. *Sensors* **2015**, *15*, 7857–7877.
7. Li, B.; Harvey, B.; Gallagher, T. Using barometers to determine the height for indoor positioning. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Montbeliard-Belfort, France, 28–31 October 2013; pp. 1–7.
8. Cypriani, M.; Lassabe, F.; Canalda, P.; Spies, F. Wi-Fi-based indoor positioning: Basic techniques, hybrid algorithms and open software platform. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Zurich, Switzerland, 15–17 September 2010; pp. 1–10.
9. Perera, K.; Bhattacharya, T.; Kulik, L.; Bailey, J. Trajectory Inference for Mobile Devices Using Connected Cell Towers. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3–6 November 2015.
10. Thaljaoui, A.; Val, T.; Nasri, N.; Brulin, D. BLE localization using RSSI measurements and iRingLA. In Proceedings of the IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 17–19 March 2015; pp. 2178–2183.
11. Boeters, R.; Ohori, K.A.; Biljecki, F.; Zlatanova, S. Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *Int. J. Geog. Inf. Sci.* **2015**, *29*, 2248–2268.
12. Kim, J.S.; Yoo, S.J.; Li, K.J. Integrating IndoorGML and CityGML for Indoor Space. In Proceedings of the 13th International Symposium on Trichoptera, Seoul, Korea, 29–30 May 2014.
13. Loch-Dehbi, S.; Dehbi, Y.; Plümer, L. Estimation of 3D Indoor Models with Constraint Propagation and Stochastic Reasoning in the Absence of Indoor Measurements. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 90, doi:10.3390/ijgi6030090.
14. Liu, G.; Iwai, M.; Tobe, Y.; Matekenya, D.; Hossain, K.M.A.; Ito, M.; Sezaki, K. Beyond horizontal location context: measuring elevation using smartphone's barometer. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, Seattle, DC, USA, 13–17 September 2014.
15. Pipelidis, G.; Rad, O.R.M.; Iwaszczuk, D.; Hugentobler, U. A Novel Approach for Dynamic Vertical Indoor Mapping through Crowd-sourced Smartphone Sensor Data. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2017), Sapporo, Japan, 18–21 September 2017.
16. Huang, J.D.J.; Tay, D. Floor Level Determination. U.S. Patent 20160356593A1, 8 December 2016.
17. Kaiser, S.; Lang, C. Detecting elevators and escalators in 3D pedestrian indoor navigation. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcala de Henares, Spain, 4–7 October 2016; pp. 1–6.
18. Bollmeyer, C.; Esemann, T.; Gehring, H.; Hellbrück, H. Precise indoor altitude estimation based on differential barometric sensing for wireless medical applications. In Proceedings of the IEEE International Conference on Body Sensor Networks (BSN), Cambridge, MA, USA, 6–9 May 2013; pp. 1–6.
19. G. Pipelidis RecordData. Available online: https://play.google.com/store/apps/details?id=com.recordData.basic&hl=en (accessed on 1 February 2018).
20. Omidreza, M. Open Sourced Datasets, Algorithms and Models. Available online: https://github.com/omidrad2017/Crowdsourced-Vertical-indoor-Mapping (accessed on 1 February 2018).
21. Savitzky, A.; Golay, M.J.E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.* **1964**, *36*, 1627–1639.
22. Zhou, P.; Zheng, Y.; Li, Z.; Li, M.; Shen, G. IODetector: A Generic Service for Indoor Outdoor Detection. In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, Toronto, ON, Canada, 6–9 November 2012.
23. Al Najjar, M.; Ghantous, M.; Bayoumi, M. Hysteresis Thresholding. In *Video Surveillance for Sensor Platforms: Algorithms and Architectures*; Springer: New York, NY, USA, 2014; pp. 147–174.

24. Chung, J.; Donahoe, M.; Schmandt, C.; Kim, I.J.; Razavai, P.; Wiseman, M. Indoor location sensing using geo-magnetism. In Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, Bethesda, MD, USA, 28 June–1 July 2011.
25. Raifer, M. Overpass Turbo—OpenStreetMap Wiki. Available online: http://wiki.openstreetmap.org/wiki/Overpass_turbo (accessed on 1 February 2018).
26. Ketchen, D.J., Jr.; Shook, C.L. The application of cluster analysis in strategic management research: An analysis and critique. *Strateg. Manag. J.* **1996**, *17*, 441–458.
27. Kolbe, T.H.; Gröger, G.; Plümer, L. CityGML: Interoperable Access to 3D City Models. In *Geo-Information for Disaster Management*; van Oosterom, P., Zlatanova, S., Fendel, E.M., Eds.; Springer: Berlin, Germany, 2005; pp. 883–899.
28. CityGML. 3D City Database: Citygml4j. Available online: http://www.3dcitydb.org/3dcitydb/citygml4j/ (accessed on 1 February 2018).
29. Samsung. Samsung Galaxy S6 Edge. Available online: http://www.samsung.com/global/galaxy/galaxy-s6-edge-plus/ (accessed on 1 February 2018).

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

3.3 Extracting Semantics of Indoor Places based on Context Recognition

"Entities should not be multiplied unnecessarily"

— William of Ockham

Georgios Pipelidis
Frederik Fraaz
Christian Prehofer

In 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) Published by IEEE 2018
Electronic ISBN: 978-1-5386-3227-7
USB ISBN: 978-1-5386-3226-0
Print on Demand(PoD) ISBN: 978-1-5386-3228-4
Conference Location: Athens, Greece
INSPEC Accession Number: 18133915
DOI: 10.1109/PERCOMW.2018.8480187
Date of Conference: 19-23 March 2018
Date Added to IEEE Xplore: 08 October 2018
The attached version is the author's manuscript, the original version is available electronically from the publisher's site at: https://ieeexplore.ieee.org/document/8480187

Summary of the Paper

This paper was published as [PFP18a], and it describes our work focused on the extraction of semantics from an indoor area, and in particular a subway station. Hence, we attempted to dynamically extract the semantics of a subway station via the use of the smartphone's Inertial Motion Unit (IMU) data. For achieving this, we used machine learning algorithms for context recognition and fuzzy set theory for the quantification of the uncertainty. We were able to identify from Inertial Motion Unit (IMU) data, seven different activities and we aggregated these activities with others activities, extracted from the same regions, to enable the semantic identification of places.

Comments on Authorship

In this paper, I personally contributed to the definition of the belief functions that can enable the quantification of the uncertainty of the extracted user context. The second author contributed to the implementation of the system. I developed a strategy for the evaluation of the system. I theorized and described the architecture of this system. I and the second author contributed to the evaluation of the system.

PerCom Work in Progress on Pervasive Computing and Communications, 2018

Extracting Semantics of Indoor Places based on Context Recognition

Georgios Pipelidis, Frederik Fraaz and Christian Prehofer
Technische Universität München,
Faculty of Informatics, Research Group for Software &
Systems Engineering, Munich, Germany

***Abstract*—In this paper we present our work in progress for the dynamic extraction of semantics in a subway station, from smart-phone Inertial Motion Unit (IMU) data. For this, we use machine learning for context recognition and fuzzy set theory for quantification of the uncertainty. Our architecture is client-server, where the clients collect data from sensors, extract features and use classification to identify seven different activities, while these activities are then aggregated on the server with other user activities, from the same regions, to enable the semantic identification of places.**

***Index Terms*—Context Recognition, Semantics Annotation of Indoor Places, Crowd-sourcing, Dynamic Mapping**

1. Introduction

Location Based Services (LBS) are widely used in our daily lives. Their main components are a localization method and a map. Even though LBSs are mainly developed for the outdoor world, humans spend approximately 80% of their time indoors [2], where there is a lack of localization technologies and maps. Many alternative approaches for indoor localization have been proposed, based on different technologies [9]. However, not many alternative approaches to indoor mapping are available, with the available approaches to be limited to tools where people are manually labeling the semantics of indoor places such as [12].

Motivation. Mapping places is a challenging process, since it requires the description of geometry, topology and semantics of a place. Unfortunately, not enough means, such as airborne or satellite photography, are available for mapping indoor places. As a result, indoor maps are mostly manually generated. As a result, various approaches for crowd-sourcing such process have been suggested [1], while many products are made for this task [5]. However, those approaches and products are focusing on extracting the geometry and the topology of indoor places and not on extracting the semantics.

Research Goal. Our research goal is to provide a framework for the dynamic, semantic annotation of indoor places based on the user's context. This framework has to incorporate the accumulated uncertainty at this level, while taking decisions on the types of the semantics for each object. Even though user's context is a basic requirement of a smart-device, the current context provided by operating systems is limited to activities such as walking or driving and they do not provide more detailed context required for our research. As a result, in our research we recognize user activities from smart-phone sensor data. For example, the walking activity can be perceived as high disturbances on the smart-phone accelerometer, while sitting as low. Finally, we fuse activities from multiple users and extract higher-level information that can indicate the object's semantics. For example a stair or an escalator.

Contribution. The contribution of this paper can be summarized as: (1) we introduce belief functions for the quantification of the uncertainty of the user context and the place semantics. (2) we have implemented a prototype system and evaluated its components, obtaining promising results. (3) we provide a framework for the dynamic semantic annotation of indoor places from inertial motion unit data from smart-phones. To our knowledge, there is no such framework available. (4) we used probability theory and fuzzy logic for activity recognition and decision making, respectively.

2. Approach

The system architecture, can be seen in Figure 3, consists of intermediate components extended between the client and the server side. In this section, we explain our approach for dynamic semantic annotation of indoor places.

2.1. Client Side

The client side is responsible for collecting data, filtering noise, grouping sensor values; extract feature, classification and streaming data to the server.

Data Collection. This module is collecting sensor data from acceleration, gyroscope and pressure sensors. The sampling frequency is 45Hz, 70Hz and 6Hz respectively.

978-1-5386-3227-7/18/$31.00 ©2018 IEEE

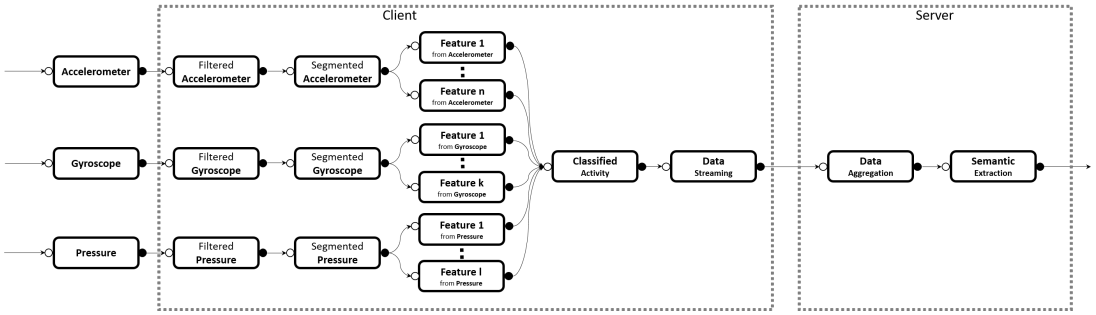

Figure 1. System Architecture

Filtering Module. The filtering module imports data readings and scales them to mean zero and standard deviation, for improving the performance of the classifier [10].

Segmentation Module. After filtering, data readings, we group them into clusters. The segmentation module is a 1.67 seconds sampling window that overlaps every 1 second.

Feature Extraction Module. In the feature extraction module, the absolute values for all axes over the sliding window are used. Additionally, the mean of four different axes is computed from the acceleration and gyroscope sensors, the interquartile range of the gyroscope and the pressure sensors, the covariance between the different axes of acceleration and gyroscope is used and the pressure derivative.

Classification Module. In the activity recognition module, a Support Vector Machine (SVM) algorithm is trained to recognize seven different activities (sitting `(si)`, standing `(st)`, walking `(wa)`, walking upstairs `(wu)`, walking downstairs `(wd)`, using the elevator up `(eu)` and using the elevator down `(ed)`) and twenty-four different features as explained in the feature extraction module. We used LIBSVM [3], since it supports multiclass classification. We created a plugin for using SVM on Android, which uses the Java implementation of LIBSVM. The plugin has been open-sourced and it is available here [4].

2.2. Server Side

The server side is responsible for the aggregation of data from different users and for extracting semantics of an indoor place.

Synthesis Module. In the synthesis module, activities that belong to the same areas are aggregated. The localization algorithm is dead reckoning and has been described here [13].

Semantics Extraction Module. The semantics extraction module maps the aggregated activities to the corresponding semantic annotation of the indoor place. This module follows a rule based model which uses the connectives $\min$ and $\max$ to implement the logical `or` and the logical `and` operation respectively. The input to the module are activities `AT`, while the output gives objects `OT`.

The final mapped objects are following the rules defined by OSM [12]. As a result, an object of the map can be either a `node`, a `way` or a `relation`. Since our framework is focused on subway stations, we take into consideration the particular semantics that correspond to it. Hence, our subway station consists of `platforms`, `corridors`, `stairs`, `escalators`, `elevators` and `entrances`. For the sake of simplicity, we do not take into consideration entrances in our model. The final rules are the following:

- A platform P, is mapped as a `relation`. Activities that indicate a platform are `standing` sta, `sitting` si and `walking` wa.
- An escalator E is mapped as a `way` and is the intersection between `climbing stairs` cs and `using elevator` ue.
- A stair S is a way and the place where people are explicitly `climbing stairs`.
- An elevator is mapped as a node and is place where people are `using elevator`.
- A corridor is mapped as a way and is the place where people are `walking`.

As a result, the rules will be:

IF `AT` IS (`si` AND `st` AND `wa`)
$THEN$ `OT` IS `P`
IF `AT` IS (`cs` AND `ue`) $THEN$ `OT` IS `ES`
IF `AT` IS `cs` $THEN$ `OT` IS `S`
IF `AT` IS `ue` $THEN$ `OT` is `EL`
IF `AT` IS `wa` $THEN$ `OT` IS `CO`

3. Evaluation

In this chapter, we present an evaluation of different individual components of our system.

Segmentation Module Evaluation. The length of the sliding window is selected after evaluating different possible lengths. The evaluation is made based on the pressure sensor, since it is the sensor with the lowest sampling frequency (6 Hz). We decided to use the ROC approach [7] and a subset of our activities, restricting it to activities where vertical transition exist. As can be seen in Figure 2, the AUC is increasing with the increase of the length of the sliding window. As a result the sliding window is fixed at 1.67 seconds, while it slides every 1 second.

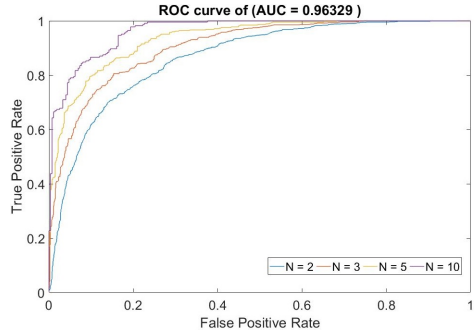

Figure 2. ROC curve and AUC for four different sliding window lengths. Those windows are 2, 3, 5 and 10 samples per window which correspond to 0.3, 0.5, 1.17 and 1.67 seconds.

Feature Evaluation. In the feature selection, two approaches were used. The filter method and the wrapper method; The four best features estimated via the first, are : (1) Pressure Derivative, (2) Interquartile Range of Gyroscope X, (3) Interquartile Range of Acceleration Magnitude and (4) Covariance of Acceleration Y and Z (computed axis). The four best features, as ranked by the second, are : (1) the Pressure Derivative, (2) the interquartile range of the x Gyroscope axes, (3) the interquartile of the acceleration magnitude and the covariance of the computed YZ axis.

Classification Accuracy. The proposed algorithm for the identification of user activities has been evaluated in 10 different users, different from the users used for training, and 7 different activities, in real time. It is found that it can identify more activities than other existing models (e.g. [8]) and it outperforms most of the other frameworks in accuracy and recall. It scores 90.1% accuracy and 92.8% recall. Additionally, it is low resource demand, which is of equal importance since it operates on resource limited smart-phones. More specifically, it uses less than 20% features than other models [11] that perform 2% better, which has a positive impact on the battery consumption.

Semantic Extraction. The semantic recognition of objects follows the rules of subway stations. For its evaluation, we used the results presented in table I. Note that the degree of belief for each activity corresponds to

TABLE 1. CONFUSION MATRIX

	si	sta	wa	wd	wu	eu	ed
si	80	0	2	0	0	0	0
sta	0	152	1	0	0	0	1
wa	0	4	170	17	1	6	0
wd	0	0	11	154	1	0	0
wu	0	1	22	18	216	0	0
eu	0	3	2	1	0	78	0
ed	0	10	1	0	0	0	84

the classification accuracy that each activity achieves:

(sitting) $si \in AT : M_{si} = 0.99$
(standing) $sta \in AT : M_{sta} = 0.89$
(walking) $wa \in AT : M_{wa} = 0.81$
(walking upstairs) $wu \in AT : M_{wu} = 0.99$
(walking downstairs) $wd \in AT : M_{wd} = 0.81$
(elevator up) $eu \in AT : M_{eu} = 0.92$
(elevator down) $ed \in AT : M_{ed} = 0.98$

As a result, the degree of belief in the semantics, following the rules defined in the section *"Semantic Extraction Module"* and they can be assessed as follows:

(Platform) $P \in OT : M_P = M_{si} \vee M_{sta} \vee M_{wa} = = min\{M_{si}, M_{sta}, M_{wa}\} = 0.81$
(Escalator) $ES \in OT : M_{ES} = M_{wd} \vee M_{wu} \vee M_{eu} \vee M_{ed} = = min\{M_{wd}, M_{wu}, M_{eu}, M_{ed}\} = 0.81$
(Stairs) $S \in OT : M_S = M_{wd} \vee M_{wu} = min\{M_{wd}, M_{wu}\} = 0.81$
(Elevator) $EL \in OT : M_{EL} = M_{eu} \vee M_{ed} = min\{M_{eu}, M_{ed}\} = 0.92$
(Corridor) $CO \in OT : M_{CO} = M_{wa} = 0.81$

From the identified objects and their location, we generate the dynamic semantic map depicted in Figure 3(a). As can be seen, the identified objects correspond to real ones from the manually created map, in Figure 3(b). As can be seen, the stair location has been approximated by a minimum of 5 m and a maximum of 10 m, and the semantics of stairs are recognized in a degree of belief 0.81. Similarly, the elevator is recognized with a belief of 0.92, and the escalator, with a belief of 0.81. The escalator is identified due to the fact that activities such as climbing stairs and using the elevator conjunct. On the other hand, the stairs and the elevator have been successfully recognized identified as simple objects. Finally, the location of the elevator was approximated only by 10 meter radius, due to the high magnetic disturbances that its metal structure generates. Finally, the location of the escalator was approximate to 5 m accuracy.

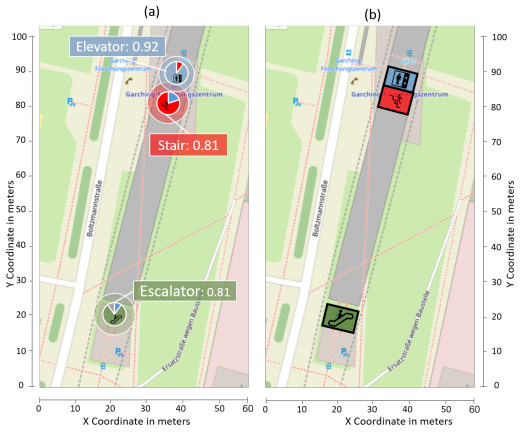

Figure 3. A map of the Garching Forschungszentrum subway station, where (a) has been generated with our approach, while (b) has been manually generated.

4. Related work

Alzantot et. al. [1] suggest to dynamically generate maps, based on activity recognition using IMUs of smart-phones. of an indoor place and not on semantics. Grzonka et. al. [6] used collected data via motion capture suit for estimating traces, which they used to reconstruct a map of the environment. However, those approaches are focusing on geometric and topological maps and not on semantic. Nguyen et. al. [11] use a classifier to recognize moving activities for improving position systems. Kaiser et. al. [8] introduce a moving platform detection module, to recognize vertical transitions. However, in their research, they are using precise sensors and not the noisy off-the-shelf smartphone sensors.

5. Conclusion

In this paper, we present our approach for the dynamic extraction of semantic properties of indoor places from user context, estimated from inertial motion unit sensor data. We use probability theory and fuzzy logic for activity recognition and decision making respectively. We instantiated our model, in a client-server framework, where in the client side data are collected from multiple sensors and are segmented into groups, from which various features are then extracted. Those features are used to train a support vector machine classifier, which is used to identify seven different activities. Those activities are then streamed to a server, where they are aggregated with other user activities using fuzzy logic. This enables a rule-based approach, through which we semantically annotate indoor places. The approach is evaluated and promising results are obtained.

6. Future Work

Our future work focuses to improve our localization approach, making it pose independent which can improve the accuracy of the semantic recognition. Additionally, our focus is to enable the dynamic generation of the rules that will enable us to generalize our model.

References

[1] M. Alzantot and M. Youssef. Uptime: Ubiquitous pedestrian tracking using mobile phones. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 3204–3209. IEEE, 2012.

[2] E. Building. the environment: A statistical summary. *US Environmental Protection Agency*, 2004.

[3] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[4] F. Fraaz. cordova-plugin-classifier - make predictions using libsvm on android. https://github.com/ffraaz/cordova-plugin-classifier. 2017-04-21.

[5] Google. Tango. https://get.google.com/tango/. 2017-06-12.

[6] S. Grzonka, F. Dijoux, A. Karwath, and W. Burgard. Mapping indoor environments based on human activity. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 476–481. IEEE, 2010.

[7] J. A. Handley and B. J. McNeil. The meaning and use of the are under a receiver operating characteristics (roc) curve. *Radiology*, 143(1):29–36, 1982.

[8] S. Kaiser and C. Lang. Detecting elevators and escalators in 3d pedestrian indoor navigation. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, pages 1–6. IEEE, 2016.

[9] R. Mautz. Indoor positioning technologies. In *ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, Institute of Geodesy and Photogrammetry Zurich*, 2012.

[10] G. Melki and V. Kecman. Speeding up online training of l1 support vector machines. In *SoutheastCon, 2016*, pages 1–6. IEEE, 2016.

[11] P. Nguyen, T. Akiyama, H. Ohashi, G. Nakahara, K. Yamasaki, and S. Hikaru. User-friendly activity recognition using svm classifier and informative features. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, pages 1–8. IEEE, 2015.

[12] OpenStreetMap. Josm. https://josm.openstreetmap.de/. 2017-06-20.

[13] G. Pipelidis, O. R. M. Rad, D. Iwaszczuk, C. Prehofer, and U. Hugentobler. A novel approach for dynamic vertical indoor mapping through crowd-sourced smartphone sensor data. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Sept 2017.

3.4 A Novel Lightweight Particle Filter For Indoor Localization

"The most important days of your life are the days you are born, and the day you find why,"

— Ernest T. Campbell

Georgios Pipelidis
Nikolaos Tsiamitros
Christian Gentner
Dina Bousdar Ahmed
Christian Prehofer

In 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN).
Published by IEEE 2019
Date of Conference: 30 Sept.-3 Oct. 2019
Date Added to IEEE Xplore: 28 November 2019
INSPEC Accession Number: 19193062
DOI: 10.1109/IPIN.2019.8911744
Conference Location: Pisa, Italy
The attached version is the author's manuscript, the original version is available electronically from the publisher's site at: https://ieeexplore.ieee.org/document/8911744

This paper was published as [PTG+19], and it describes our invented method for simultaneous localization and mapping using IMU data from off-the-shelf smartphones. This method can be used to support our proposed crowdsourced indoor mapping approach. The algorithm makes use of the geometry, the topology and the semantics of indoor areas. It is infrastructure-independent and can operate in various indoor environments. It introduces a novel particle filter implementation that enables the fusion of inertial motion unit sensors, user context, user gait direction, and map information. It performs localization with up to two orders of magnitude fewer particles than state-of-the-art approaches. It is compatible with existing standards for mapping indoor areas, such as the Open Street Maps and it follows defined standards for map handling.

Comments on Authorship

Overall, the main idea of the paper is of equal authorship. I personally contributed to theorize the idea and design the architecture that describes the technical details. The second author contributed to the realization of the context of the paper and to ensemble the component of the system that has been compared against the related work. I designed the experimental setup for the evaluation, and the second author did the execution. Finally, under the supervision of the other authors, I authored a majority of the text.

2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

A Novel Lightweight Particle Filter for Indoor Localization

Georgios Pipelidis*‡, Nikolaos Tsiamitros*‡, Christian Gentner†, Dina Bousdar Ahmed†
and Christian Prehofer*
*Technical University of Munich, Faculty of Informatics, Research Group for Software & Systems Engineering, Munich, Germany
†, German Aerospace Center, Institute of Communication and Navigation, Oberpfaffenhofen-Wessling, Germany
‡, These two authors contributed equally.

***Abstract*—In this paper, we describe an infrastructure-independent indoor localization approach for various indoor environments. Our method introduces a novel particle filter implementation that enables the fusion of inertial motion unit sensors, user context, user gait direction, and map information. Due to this novel fusion, it performs localization with up to two orders of magnitude fewer particles than state-of-the-art approaches. Additionally, it extracts map information via existing open services, such as the Open Street Maps and it follows defined standards for the map handling. We evaluated all the components of our method in real-time in off-the-shelf smartphones and we find that it performs a median error of 2.3m, while using only 40 particles instead of 400 or up to 4000 particles that other methods require for the same accuracy.**

***Index Terms*—Indoor localizatin, Particle Filter, Sensor Fusion, Context Recognition**

I. Introduction

Location-Based Services (LBS) have become a need in our daily lives. The main components of such services are a localization method and a map [9]. Even though we spend approximately 80% of our time indoors [26], LBSs are developed for the outside world. The main reasons are the lack of indoor localization methods and the lack of maps. However, today there is a series of initiatives, such as the Enhanced 911 [6] in the United States, the Enhanced 112 in the European Union, [2] as well as the European Accessibility Act [1]. These initiatives aim to accelerate the research towards indoor localization.

On the other hand, we are facing an ever-larger plethora of deployed sensing devices in different artifacts of our daily living. (e.g., clothing, cameras, watches, smartphones, electronic appliances, etc.). As an example, it smartphone users will exceed 6.1 billion by 2020. Additionally, progress in Machine Learning and Artificial Intelligence is becoming more and more application-oriented. Localization is essential for making better usage of those devices since localization is an essential aspect of context-awareness [7].

a) ***Problem****:* Indoor localization remains an open challenge, because, unlike outdoor localization, Global Navigation Satelite System (GNSS) signals cannot penetrate the thick walls of buildings. Additionally, energy consumption and computational demand have to be taken into consideration, since they are required to operate in off-the-shelf devices, such as smartphones or other IoT devices. Many alternative approaches to GPS have been proposed [29]. However, all of those approaches are either not accurate enough or computationally expensive.

b) ***Research Goal****:* Our goal is to enable users to find their way inside complex buildings, where limited information is available in off-the-shelf smartphones The necessary information for this is only a crude floor plan of the building, similar to the existing floorplans available in OSM.

c) ***Contribution****:* We present a method that can localize people with high accuracy in buildings for which there is no detailed map and infrastructure available. Our method introduces a novel particle filter implementation that enables the fusion of inertial motion unit sensors, user context, a user walking direction, and map information. Our method uses up to two orders of magnitude fewer particles than state-of-the-art approaches, and it operates following existing map models. Our method performs a median error of 2.3m, in real-time, in off-the-shelf smartphones.

II. Approach

In this section, we explain our approach to infrastructure-less localization. The system architecture, in Figure 2, consists of several components.

A. High Level Architecture

The high-level architecture of our system consists of seven components. The first component is responsible

Manuscript received May 26, 2019; revised July 11, 2019. Corresponding author: G. Pipelidis (email: pipelidi@in.tum.de).

978-1-7281-1788-1/19/$31.00 2019 IEEE © 2019 IEEE

2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

for the user's **context extraction**, the second component is the **Map** component, which is responsible of querying the corresponding **OSM Model** from Open Street Maps (OSM), the third component is the **Pedometer**, responsible for detecting steps, The fourth and fifth components are the **Initial Direction** and the **Direction** estimation, respectively. The sixth component of our architecture is the **Particle Filter**, responsible for the fusion between the aforementioned components.

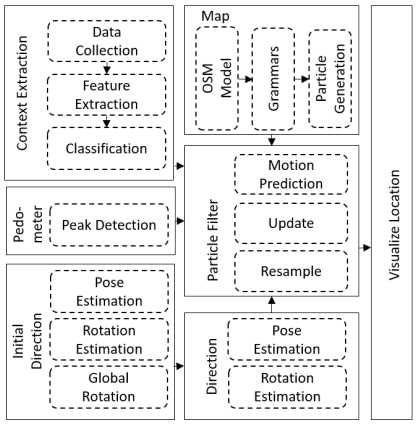

Fig. 1. Localization module architecture.

*1) **Map**:* The first step in our approach is to get the publicly available map of the area (e.g., Open Street Map) of interest, which typically is a building. One well-known source is the Open Street Map (OSM) [33], which provides map data using a data model to represent elements of the physical world, such as buildings, roads, corridors, etc. This data can be dynamically retrieved via the Overpass API [4] and can be used to identify the elements that exist within an area of the physical world as well as their geometry. In our algorithm, we parse this data and identify those elements that can be used for routing, such as stairs, corridors, pathways, which are stored separately and are later used for localization.

*a) **Grammars**:* This module is responsible for parsing and interpreting data queried by OSM, since a map may consist of either a `node`, a `way` or a `relation`, where they indicate a point, a line or collection of lines respectively. Although we used existing semantics defined by the community of OSM, in this module some alternations were necessary. For example the `room` label is interpreted as `building`, due to limitations of visualizing a `room` in the OSM tile server. Additionally, we used the `level` tag to indicate different floors due to the limitations of OSM to visualize different floors. Other tags used were `footway` for corridors, the `steps` for stairs, `elevator`, `entrance`, `escalator`, `travelator`.

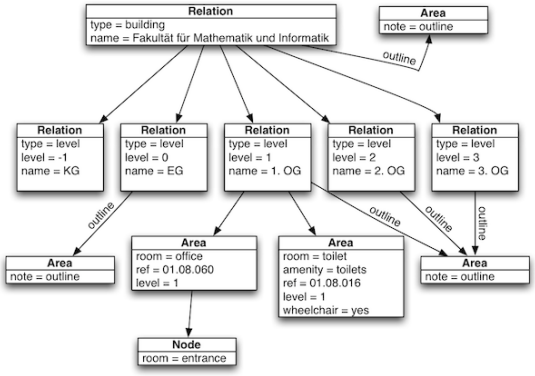

Fig. 2. An example of IndoorOSM schema for the Technical University of Munich, as proposed by [5].

*b) **Particle Generation**:* In this module, the newly constructed map is used to generate a set of randomly positioned particles $S_t^k = [\mathbf{x}_j^k, w_j^k]$, where $\mathbf{x}_j^k$ is the j^{th} particle at time k and w_j^k is the normalized weight of the particle, with uniform prior distribution, as described by the Monte Carlo simulation. Once we have the enhanced map elements, we use them to generate a pool of particles that will be used by the particle filter and the dead reckoning algorithm. In essence, we divide the areas covered by the map elements into tiny rectangles, and we represent each rectangle by its center position characterized by its GPS coordinates. The result is a dense layer of particles that cover the areas of a person's location. In this module, we restructure the newly constructed map as a set of randomly generated particles, with uniform prior distribution, which follows the Monte Carlo simulation description.

*2) **Particle Filter**:* The particle filter consists of two main stages: the prediction stage and the update stage. The proposed filter represents the probability density of the state vector $X_k = [x_k, y_k, \theta_k]^T$ at step k by N_p particles. According to [8], [19], [20] and assuming a first-order hidden Markov model, the posterior filtered density $\mathsf{p}\big(\mathbf{x}_k|\mathbf{z}_{0:k}\big)$ is approximated as

$$\mathsf{p}\big(\mathbf{x}_k|\mathbf{z}_{0:k}\big) \approx \sum_{i=1}^{N_p} w_k^{(i)}\ \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)})\,, \qquad (1)$$

where $\mathbf{z}_{0:k}$ defines the measurement vector for the time steps $0, \ldots, k$, $\delta(\cdot)$ stands for the Dirac distribution, $\mathbf{x}_k^{(j)}$ denotes particle state and $w_k^{(j)}$ denotes the normalized weight. The conditional probability $P(S_t|z_t, S_{t-1}) = P(z_t|S_t, S_{t-1})$ where S_t and S_{t-1} current, and previous state vectors respectively and z_t is the observation, which in our case is $z_t = min(S_{map} - S_t)$, where S_{map} is a vector of all points in the map.

*a) **Motion Prediction**:* We use dead reckoning as our motion prediction mechanism. Hence, arbitrary

motions are modeled as $\begin{bmatrix}\Delta x, & \Delta y\end{bmatrix}^T$, while, if the initial direction is $\hat{\theta}$, then the rotation is modeled as $\delta\hat{\theta} = \hat{\theta}_k - \hat{\theta}$, where $\hat{\theta_k}$ estimation is explained in the **Direction Estimation** section. Hence, our system is:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ z \end{bmatrix} = \begin{bmatrix} x_{k-1} + \rho cos(\hat{\theta}_{k-1}) \\ y_{k-1} + \rho sin(\hat{\theta}_{k-1}) \\ \hat{\theta}_{k-1} - \hat{\theta} \\ \begin{bmatrix} \frac{P_0}{P_i}^{\frac{1}{5.25}} - 1 \end{bmatrix} * \frac{T_b + 273.15}{0.0069} \end{bmatrix}$$

Where x_k, y_k and θ_k are the coordinates and direction at time k, x_{k-1} and y_{k-1} are the previous coordinates, ρ is the distance traveled and θ_k is the direction followed. The ρ is the walking distance and is extracted from the pedometer. The vertical coordinate z is estimated using the barometric formula, where P_0 is the reference pressure, P_i is the current pressure value and T_b is the temperature value. The reference pressure was extracted from the same phone during the outdoor to indoor transitions. More details regarding the reference pressure as well as the altitude estimation are available here [34].

*b) **Update**:* In the update phase, each particle's weight is computed based on the assigned probabilities, a model that describes a set of restrictions and a motion predicted mechanism. Hence the weight of the i^{th} particle at time k, $w_k^{(i)}$ is defined as:

$$w_k^{(i)} = w_{k-1}^{(i)} \mathsf{p}\big(X_k^{(i)}|\delta_k^{(i)}\big)\mathsf{p}\big(X_k^{(i)}|\theta_k^{(i)}\big)\mathsf{p}\big(X_k^{(i)}|\alpha_k^{(i)}\big)$$

where $\mathsf{p}\big(X_k^{(i)}|\delta_k^{(i)}\big)$ describes the conditional probability of the i^{th} particle being at location X at time k given the nearest distance δ to the nearest corridor, $\mathsf{p}\big(X_k^{(i)}|\theta_k^{(i)}\big)$ describes the conditional probability of the particle being at location X, given its current heading direction θ, and finally, $\mathsf{p}\big(X_k^{(i)}|\alpha_k\big)$ the conditional probability of the particle being at location X at time k given the current activity α. By collecting our high level features to a vector such as: $\phi_k^{(i)} = [\delta_k^{(i)}, \theta_k^{(i)}, \alpha_k^{(i)}]$ and given the fact that the probabilities are conditionally interdependent, we can model our system in accordance to Russell's and Norvig's normalization [35] as:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{\mathsf{p}\big(X_k^{(i)}\big)\prod_{j=1}^{m}\mathsf{p}\big(\phi_k^{(j)}|X_k^{(i)}\big)}{\sum_{i=1}^{N_p}\mathsf{p}\big(X_k^{(i)}\big)\prod_{j=1}^{m}\mathsf{p}\big(\phi_k^{(j)}|X_k^{(i)}\big)} \quad (2)$$

where m is the length of ϕ, $\mathsf{p}\big(X_k^{(i)}\big)$ is the probability of the i^{th} particle being at location X at time k, $\mathsf{p}\big(\delta_k|X_k^{(i)}\big)$ defines the likelihood of obtaining a particle with minimum distance to the center of the corridor δ, $\mathsf{p}\big(\theta_k|X_k^{(i)}\big)$ defines the likelihood of obtaining a particle with angle θ, and $\mathsf{p}\big(\alpha_k|X_k^{(i)}\big)$ defines the likelihood of obtaining a particle in place that has been semantically annotated to fit with the activity α

*c) **Resample**:* For proper resample, two additive Gaussian noise models, are used on top of the motion prediction mechanism, which enables us to simulate the uncertainty accumulated due to the step length variation and the noise generated by the gyroscope in the direction estimation. As a result, Gaussian noise is added to simulate the effect of the uncertainty in the step length, $P(M_{dist}, \sigma_{dist})$, where M_{dist} is the median step length and σ_{dist} is the standard deviation between different step sizes. Additionally, normal distribution noise, $N(M_{rot}, \sigma_{rot})$, is introduced to simulate the uncertainty of the direction estimation, where M_{rot} is the median direction over the segment of the window and σ_{rot} is the standard deviation of the direction. The noise is applied separately to the two models since they are considered independent.

*3) **Initial Direction**:* In order to start localization using dead reckoning and the particle filter, we need an initial estimation for the user's absolute walking direction. For extracting the initial direction, we use the phone's accelerometer and magnetometer to calculate the phone's pose and later the rotation matrix, which includes the roll, pitch, and yaw. Using the accelerometer, we can extract the phone pose by estimating the direction of the gravity vector. Once we extract the gravity vector, we can calculate the user heading direction by combining the magnetometer with map information. Even though we can extract the uncertainty of the magnetic sensor, the operating system, this method involves a considerable degree of uncertainty that is difficult to model.

*4) **Direction Estimation**:* The direction estimation module is tracking changes that occur in the current direction in relation to the previous. The heading direction is estimated using the sliding window approach for $1s$ window length, and $500ms$ window stride. We first apply a low pass filter to remove human motion noise, lower than $1Hz$. In order to assign the turning direction in a more generic way, which can be applied independent of the phone pose, we identify the gravity direction first. The gravity direction can be identified based on the maximum value along the three axes of the accelerometer, while the maximum accelerometer values are decided based on the median values over the sliding window length. We estimate the gravity direction by $argmax(\bar{Acc}_x, \bar{Acc}_y, \bar{Acc}_z)$, where $\bar{Acc}_x$, $\bar{Acc}_y$, $\bar{Acc}_z$, are the median accelerometer values across the sliding window axes x, y and z respectively. Finally, the walking direction is estimated by integrating over the estimated gyroscope magnitude G_m axes, where $G_m^{(t)} = \sqrt{G_x^{(t)\ 2} + G_y^{(t)\ 2} + G_z^{(t)\ 2}}$, and the direction $\theta = \theta_0 + \int_1^n G_m dt$. Where θ is the current direction, θ_0 is the previous direction, G_m is the estimated velocity, and n is the number of samples collected in a sliding window period.

*5) **Step Detection**:* In the step detection module, the current number of steps is estimated based on the

repetitive pattern caused in the accelerometer from the human bipedal movement. More specifically, it detects steps via its three major components. The first is the data collection component, which collects data from the accelerometer sensor. The second component is the filtering component, which removes outliers from the data segment and finally is the peak detection component, which detects peaks which are translated in heel strike phases or steps. In the first phase, data are collected from the three axes of the accelerometer $Acc = [X, Y, Z]$ and they are later aggregated into a single magnitude axis $Acc_m = \sqrt{X^2 + Y^2 + Z^2}$. Once Acc_m has been computed for a segment of data, then the average magnitude $\bar{Acc}_m$ of the segment is computed $\bar{Acc}_m = \frac{\Sigma_{i=1}^{n} Acc_m^{(i)}}{n}$, where n is the length of the segment and $Acc_m^{(i)}$ is the i^{th} value of Acc_m. Finally, in the peak detection module, unique patterns, which are caused by the action when heel strikes the ground and are reflected in the acceleration sensor data, are taken into consideration.

*a) **Visualize Location**:* After calculating the new positions of the particles using the dead reckoning algorithm, weighted through a fusion with the map and the user context, the weighted average of the position of the particles is visualized as the final position to the user in our tile server.

6) Context Extraction: This module is responsible for collecting data, filtering noise, grouping sensor values, extracting features, and classification.

***Data Collection**:* This module collects sensor data from the acceleration and pressure sensors. The sampling frequency is 45Hz and 6Hz, respectively. Once data is collected, the filtering module imports data readings and they are normalized to zero mean and standard deviation, for improving the performance of the classifier. After filtering the data readings, we group them into clusters. The segmentation module is 1.67s sampling window that overlaps every 1s. The sampling window is decided at 3s for improving classification results, as can also be seen in the evaluation chapter since most of the activities are periodic activities while the average period of a bipedal movement is estimated at roughly 1.4s.

***Feature Extraction Module**:* This module is responsible for extracting features from the above segments that will be used later for classification. Here, we used the mean of all axes of the acceleration sensor, the interquartile range of the acceleration and the pressure sensors, the covariance between the different axes of acceleration and gyroscope and the pressure derivative.

***Classification Module**:* In the classification module, a Support Vector Machine (SVM) algorithm is trained to recognize seven different activities (sitting `(si)`, standing `(st)`, walking `(wa)`, walking upstairs `(wu)`, walking downstairs `(wd)`, using the elevator up `(eu)` and using the elevator down `(ed)`) and thirteen different features as explained in the feature extraction module. SVM is one of the most popular machine learning models for classification. Its task is to maximize the margin between training samples that belong to different classes while projecting them into higher dimensions. For training the classifier and making live predictions, we used LIBSVM, which is an integrated software for support vector machine classification [10] since it supports multiclass classification. We created a plugin for using SVM on Android, which uses the Java implementation of LIBSVM. The plugin has been open-sourced, and it is available here [17].

III. Evaluation

In this section, we present the evaluation of the components of our system. In particular, we present the evaluation of the context recognition and the localization algorithm. For the latter, we conducted experiments that tested the algorithm for its accuracy using the mean absolute error approach, its resilience to error and noise. The evaluation took place in the Mathematics and Informatics building of the Technical University of Munich in Garching. Carlos Martnez [11] proposed a methodology, which enables to measure both static and moving targets, by creating a predefined path with checkpoints. In our evaluation we follow their introduced guidelines which later inspired established regulations [3] for localization frameworks.

A. Classification Accuracy

The proposed algorithm for the identification of user activities is found that can identify more activities The proposed algorithm for the identification of user activities is found that can identify more activities than other existing models (e.g., [23]) and it outperforms most of the other frameworks in accuracy and recall. It scores 90.1% accuracy and 92.8% recall. Additionally, it has low resource demand, since it uses less than 20% features than other models [30] that perform 2% better, which has a positive impact on battery consumption.The results are presented in Table I

TABLE I
CONFUSION MATRIX [IN %]

	si	sta	wa	wd	wu	eu	ed
si	100.0	0.00	0.96	0.00	0.00	0.00	0.00
sta	0.00	89.4	0.48	0.00	0.00	0.00	1.18
wa	0.00	2.35	81.3	8.95	0.46	7.14	0.00
wd	0.00	0.00	5.26	81.0	0.46	0.00	0.00
wu	0.00	0.59	10.5	9.47	99.1	0.00	0.00
eu	0.00	1.76	0.96	0.53	0.00	92.8	0.00
ed	0.00	5.88	0.48	0.00	0.00	0.00	98.8

B. Localization Accuracy

The results of our localization evaluation are available in this section. Figure 3 presents the Cumulative Distribution Function (CDF) of the error collected

by 13 different experiments, while Figure 4 shows the average error and its standard deviation for each landmark for the same set of experiments.

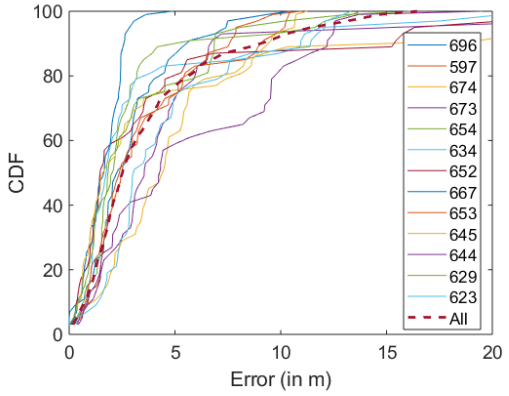

Fig. 3. The Cumulative Distribution Function (CDF) of the error collected from 13 participants of the experiment.

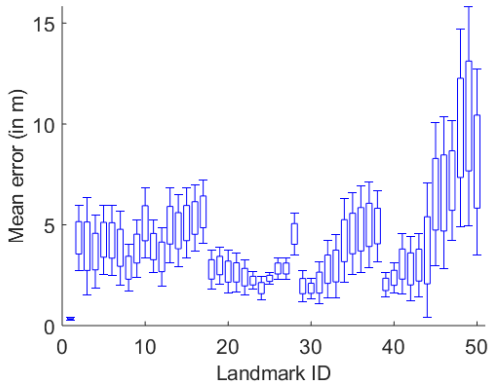

Fig. 4. The mean error with its standard deviation for each landmark collected from 13 repetitions of the experiment

In the CDF of the error, we can see that 81% of the landmarks in the localization have an error of fewer than 5 meters and a median error of 2.3m, which rivals other state-of-the-art particle filter algorithms, as it can be seen in the results of the 2017 Indoor Positioning and Indoor Navigation (IPIN) **off-line** competition [14]. Additionally, a significant advantage of our approach is that it is lightweight enough to be implemented and be used live on a smartphone device. In the same figure, we can see that only approximately 6% of the landmarks have an error greater than 10 meters, with the 95th percentile being close to 10.2m. As seen in Figure 4, the landmarks with the highest maximum and mean error, which constitutes much of that 6% are in the area around 43-49. This figure clearly shows that all the other landmarks have systematically errors less than 12m, for the entirety of the collected dataset, with most of them having maximum errors of less than 10m.

Finally, context recognition can significantly decrease the localization error. As we see in Figure 4, between landmarks 26-27 the error suddenly increases, partly because of the step length error and of their relatively long distance (19m), and partly because, unlike between most other landmarks, there is a considerable freedom of movement between these two, resulting in an additional error in direction. As we see though, the detection of stairs between landmarks 27-28, significantly decreases the allowed particle positions, placing most of them on the stairs, which, consequently, reduces the localization error. Therefore, we can argue that understanding the user's activity can considerably reduce the search space of possible locations, and, thus, increase localization accuracy.

C. Resilience to Error and Noise

1) Resilience to step length error: In our experiments, we used an average step length of 0.67m [22], which becomes visible in Figure 4, where the mean error increases along straight paths, due to the difference that the step length causes. As can be seen in the straight path in Figure 5, between landmarks 0-6,we can see that for some users like user in Path 2, there is a more significant difference in the two-step lengths, resulting in an error of approximately 10m by landmark 5, which is at a distance of 50m from landmark 0, where the error is close to 0, which means that there is approximately 20% error in the calculated length of each step. However, as we see for landmark 7, this error is practically eliminated ($< 2m$) due to three key characteristics of our algorithm that makes it resilient to a certain amount of step length error.

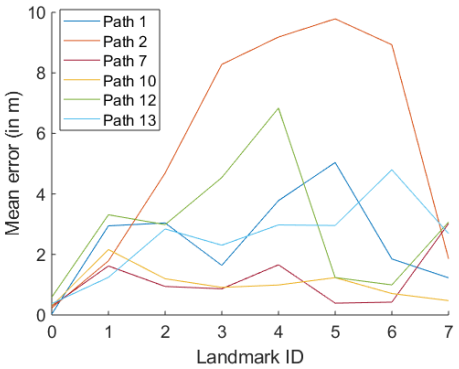

Fig. 5. The difference in displacement for different users

The first characteristic is the step spread of the particles within the uncertainty radius. As more mismatches in the walking direction and the path direction get detected, the uncertainty radius of the position of the weighted average increases and the particles get

2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

spread out to farther positions of the path. The second key characteristic of our algorithm is the fact that we take into consideration the difference between the walking direction and the path direction when weighting the particles The weight of the particle after the update step (normalized to 1 particle) follows a linear relation to the difference between the particle's moving direction and the direction of the path, deriving from the update function: w_k, as described in the approach section. The same relation for the most likely particle (a particle with the highest weight), but normalized for 70 particles. The last key characteristic of our algorithm is to re-sample low weight particles withing the uncertainty radius, which allows new particles to be placed in neighboring areas where the topology matches the movement of the user, and, thus, are more likely to be closer to the correct position of the user.

2) Resilience to direction error and noise: Error in the direction can be caused by noise in the gyroscope data either from the inaccuracy of the sensor or because of small random movements of the user during walking. The particle filter of our algorithm can tolerate a certain amount of noise and error in the direction estimation thanks to three features.

The first feature is the recalibration of direction along straight lines. This mechanism helps particles that more closely follow the direction of the path increase their weight and slowly fix the walking direction error. This is illustrated in Figure 6. In this figure, we see how the walking direction error, starting from different values, slowly but consistently converges to 0^o, even for an initial error of 57^o.

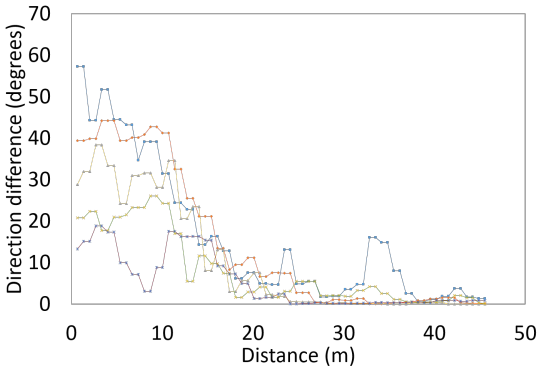

Fig. 6. Correction of direction difference along a straight path

Error influence due to map context recognition: As we can see in Figure 7, the estimated position for landmark 8 lies in the corridor that leads outside of the building and is parallel to the corridor that the user was walking during the experiment. This happened because, as the heatmap visualizes, there are two paths on the right as the user exits the long straight corridor, with 20^o difference in direction. Therefore, an inaccuracy of more than 10^o in the walking direction after taking the turn, can place more particles on the path towards the exit and eventually position the user out of the building, which is what is shown in Figure 7. However, since the difference in the direction of the two paths is relatively small, a few particles are placed and remain on the correct path. When the user eventually turns right for landmark 10, the weight of the particles on the correct path increases, while the weight of the rest decreases and soon they are eliminated. Consequently, after a few meters, the position of the weighted average is corrected and the error remains relatively small ($< 7m$).

Fig. 7. The elimination of direction error because of turns

Error influence due to human context recognition: Semantic recognition, such as the recognition of a stairway in our scenario, can decrease significantly the possible degrees of movement, allowing for a recalibration of the walking direction. For example, by knowing the direction of the stairway, we can correct the walking direction, regardless of the direction error, once we detect that the user is walking up/down the stairs. The existing algorithm can similarly handle elevators, where in most cases there is only one way to exit, and escalators. The problem lies in the handling of false positive floor transition recognitions, as happened in our scenario.

3) Resilience to context noise: As we have already mentioned, a false positive detection of stairs can result in an increase in the localization error, if there is a stairway within the uncertainty radius at that step. In contrast, a failure to detect stairs or a detection of stairs when there is no stairway nearby, does not have a negative effect on the localization, since they result in the normal localization error of the algorithm. Here, we estimate the percentage of false positive detections that affected the localization. To summarize, 80% of the false positive stair detections are ignored and do not affect the results, since the confidence that the user is not close to a stairway is high, while 20% of the times an inflicted localization error occur with $\simeq 42\%$ impact on localization error ($\simeq 4.1m$ from $\simeq 2.9$).

IV. Comparison Against Related work

TABLE II
A COMPARISON OF OUR ALGORITHM (LPF) AGAINST STATE-OF-THE-ART ALGORITHMS.

	[13]	[28]	[25]	[27]	[31]	[24]	LPF
Online		x			x	x	x
Smartph.		x			x		x
Particles	x	x	x	x		x	x
Infrastr.			x			x	
Detailed	x	x	x	x		x	
3D	x			x		x	x
Accur. [in m]	4.69	1.5	1.89	3.23	1.35	2.0	2.3

Jaworski et. al. [21], allows vertical transitions. Fetzar et. al. [16], fused two localization models with particle filter. Klepal et. al. [25], address the problem of skewed paths with Backtracking Particle Filter (BPF). Fox et. al. [12], adapts the number of samples in a particle filter over time. Knauth S. [27] fuses dead reckoning and Wi-Fi RSSI. Kang, W. and Han, Y. [24], use simple PDR. Nurminen Henri et. al. [32], fuses Wi-Fi RSSI and hand-held Inertial Measurement Sensor. F. Ebner et al. [13], enables multiple floor transitions. Fan Li et al. [28], use IMU following a PDR approach. Galcik and Opiela [18] are following a grid-based approach, where they quantifying transition probabilities between cells in a grid stature using Bayesian filters.

In Table II a comparison of our Light-weighted Particle Filter (LPF) algorithm, against state-of-the-art algorithms is presented. As it can be seen, our algorithm is executed in real time (Online) on a smartphone (Smartph.) and follows a particle filter approach (Particles). It is infrastructure independent (Infrastr.), it does not require a detailed map (Detailed) and can localize people in 3D space. Its accuracy when compared against other algorithms that followed the regulations established by [3] for evaluation (Grey background in Accur.), performs even two times better than state of the art algorithms that have been off-line evaluated.

Furthermore, since our evaluation follows the established rules of [3] for indoor localization frameworks, in Figure 8.a we provide a comparison between our framework and all competitors in EvAAL competitions [15]. As can be seen, our system outperforms all online frameworks competed in EvAAL competition between 2015 and 2017. Finally, our system has participated in the 2018 EvAAL competition where it secured the third position among the online competing frameworks, which were executed in a smartphone and the sixth position among all Non-Camera based Positioning frameworks.

V. Conclusion

In this paper we presented a 3D indoor localization algorithm for diverse indoor environments that is based exclusively on the smartphone sensors and crude floor plans, commonly available in open street maps.

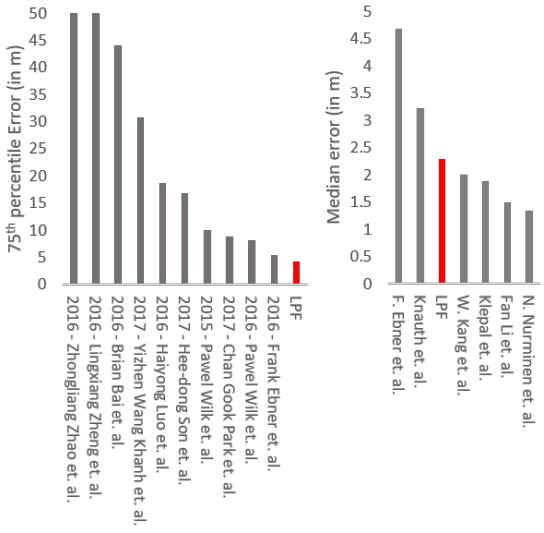

Fig. 8. In (a) we provide a comparison of our system and all systems participated in EVAAL competition since 2016. In (b) we compare our framework against related work

The core module of our algorithm is a particle filter which uses measurements from the gyroscope and the accelerometer sensor, combined with contextual information from the map, to update the states of the particles at each step of the iteration. In our evaluation, we show that our system can achieve twice better accuracy competing with other state-of-the-art solutions, while using an order of magnitude less particles. We demonstrate our algorithm's resilience to a certain degree of sensor errors and we describe the attributes that make it robust. Finally, using updates from our context recognition module, we have illustrated our algorithm's ability to detect floor transitions, which renders it a complete 3D localization and mapping solution.

VI. Future Work

The breadth and extent of this paper provides many possibilities for future work. Although a lot of research has already been done in the general direction without ever reaching an easy and globally applicable solution, there are many heuristics that one could use in order to improve the estimation of highly unreliable sensors, as is the compass. In our approach, for example, we use the compass, along with the allowed path directions of the initial position to identify the best match, which is chosen as the initial direction. However, we have not extensively evaluated the precision of this approach. Concerning the particle filter algorithm itself, one direction for future work could be to identify the optimal number of particles needed to achieve localization and how the number of particles itself affects it. In the same direction, the handling of the

uncertainty radius and the investigation of its deeper effect on the localization could also be part of a future project. Furthermore, the context recognition module could be integrated into the algorithm in a more complete way, eliminating the sensitivity to noisy pressure data and allowing the user to leverage its complete capabilities, which include recognition of whether the user is climbing up or down the stairs, or uses an elevator, or an escalator. In addition, alternative ways to update the weights of the particles could be explored and evaluated against our current approach and measurements from additional sensor data.

VII. Acknowledgments

This work is proudly supported by the European Institute of Innovation and Technology.

References

[1] European Accessibility Act - Employment, Social Affairs & Inclusion - European Commission. http://ec.europa.eu/social/main.jsp?catId=1202.

[2] European Commission - PRESS RELEASES - Press release - Commission pushes for rapid deployment of location enhanced 112 emergency services. http://europa.eu/rapid/press-release_IP-03-1122_en.htm?locale=en.

[3] ISO/IEC 18305:2016 information technology real time locating systems test and evaluation of localization and tracking systems. https://www.iso.org/standard/62090.html. Accessed: 08-08-2018).

[4] Overpass turbo - OpenStreetMap Wiki. http://wiki.openstreetmap.org/wiki/Overpass_turbo.

[5] Proposed features/indoor - OpenStreetMap wiki. https://wiki.openstreetmap.org/wiki/Proposed_features/indoor.

[6] Wireless E911 Location Accuracy Requirements. https://www.fcc.gov/document/wireless-e911-location-accuracy-requirements-3, Dec. 2015.

[7] I. Afyouni, C. Ray, and C. Claramunt. A fine-grained context-dependent model for indoor spaces. In *Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, ISA '10, pages 33–38, New York, NY, USA, 2010. ACM.

[8] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. 50(2):174–188, Feb. 2002.

[9] K. Axel et al. *Location-based services: fundamentals and operation*. John Wiley & Sons, Chichester, England ; Hoboken, NJ, 2005.

[10] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[11] C. M. de la Osa, G. G. Anagnostopoulos, M. Togneri, M. Deriaz, and D. Konstantas. Positioning evaluation and ground truth definition for real life use cases. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE, 2016.

[12] F. Dieter. Kld-sampling: Adaptive particle filters. In *Advances in neural information processing systems*, pages 713–720, 2002.

[13] F. Ebner, T. Fetzer, F. Deinzer, L. Köping, and M. Grzegorzek. Multi sensor 3d indoor localisation. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, pages 1–11. IEEE, 2015.

[14] J. T.-S. et. al. Off-line evaluation of mobile-centric indoor positioning systems: The experiences from the 2017 ipin competition. In *Sensors*, 2018.

[15] EvAAL. EvAAL - the EvAAL framework. http://evaal.aaloa.org/.

[16] T. Fetzer, F. Ebner, F. Deinzer, and M. Grzegorzek. Recovering from sample impoverishment in context of indoor localisation. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Sept 2017.

[17] F. Fraaz. cordova-plugin-classifier - make predictions using libsvm on android. https://github.com/ffraaz/cordova-plugin-classifier. 2017-04-21.

[18] F. Gal?k and M. Opiela. Grid-based indoor localization using smartphones. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Oct 2016.

[19] C. Gentner, S. Zhang, and T. Jost. Log-PF: Particle Filtering in Logarithm Domain. *Journal of Electrical and Computer Engineering*, 2018, 2018.

[20] N. Gordon, D. Salmond, and A. F. M. Smith. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proc. Radar Signal Processing*, 140(2):107–113, 1993.

[21] W. Jaworski, P. Wilk, P. Zborowski, W. Chmielowiec, A. Y. Lee, and A. Kumar. Real-time 3d indoor localization. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Sept 2017.

[22] K. Jordan, J. H. Challis, and K. M. Newell. Walking speed influences on gait cycle variability. *Gait & posture*, 26(1):128–134, 2007.

[23] S. Kaiser and C. Lang. Detecting elevators and escalators in 3d pedestrian indoor navigation. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, pages 1–6. IEEE, 2016.

[24] W. Kang and Y. Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal*, 15(5):2906–2916, 2015.

[25] M. Klepal, S. Beauregard, et al. A backtracking particle filter for fusing building plans with pdr displacement estimates. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pages 207–212. IEEE, 2008.

[26] N. E. e. Klepeis. The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science and Environmental Epidemiology*, 11(3):231, 2001.

[27] S. Knauth. Smartphone pdr positioning in large environments employing wifi, particle filter, and backward optimization. In *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on*, pages 1–6. IEEE, 2017.

[28] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 421–430. ACM, 2012.

[29] R. Mautz. Indoor positioning technologies. In *ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, Institute of Geodesy and Photogrammetry Zurich*, 2012.

[30] P. Nguyen, T. Akiyama, H. Ohashi, G. Nakahara, K. Yamasaki, and S. Hikaru. User-friendly activity recognition using svm classifier and informative features. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, pages 1–8. IEEE, 2015.

[31] H. Nurminen, A. Ristimaki, S. Ali-Loytty, and R. Piché. Particle filter and smoother for indoor localization. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pages 1–10. IEEE, 2013.

[32] H. Nurminen, A. Ristimki, S. Ali-Lytty, and R. t. Pich. Particle filter and smoother for indoor localization. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013.

[33] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

[34] G. Pipelidis, O. R. M. Rad, D. Iwaszczuk, C. Prehofer, and U. Hugentobler. *A novel approach for dynamic vertical indoo mapping through crowd-sourced smartphone sensor data*. In 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 18, Sept 2017, 2017.

[35] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

3.5 Cross-Device Radio Map Generation via Crowdsourcing

"Civilization is a progress from an indefinite, incoherent homogeneity toward a definite, coherent heterogeneity."

— Herbert Spencer

Georgios Pipelidis
Nikolaos Tsiamitros
Efdal Ustaoglu
Romeo Kinzler
Petteri Nurmi
Huber Flores
Christian Prehofer

In 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)
Published by IEEE 2019
Date of Conference: 30 Sept.-3 Oct. 2019
Date Added to IEEE Xplore: 28 November 2019
INSPEC Accession Number: 19193107
DOI: 10.1109/IPIN.2019.8911766
Conference Location: Pisa, Italy
The attached version is the author's manuscript, the original version is available electronically from the publisher's site at: https://ieeexplore.ieee.org/abstract/document/8911766

This paper was published as [PTU+19], and it describes our experience from implementing a crowdsourcing approach for WiFi "landmark" extraction. For achieving this, we made use of our invented localization method to identify "RF landmarks" in an area. These landmarks can later be used to enhance the localization accuracy. Through such approach, objects or humans can be localization without prior knowledge of their precise location. The main challenge that we focused our efforts here was the heterogeneity between different phones since different phones are highly sensitive to variations in devices capturing the measurements. In particular, cross-device variations can decrease performance of crowdsourced bootstrapping approaches up to 70%. As a result, we focused our attention on the developing a framework for the crowdsourcing generation of radio maps, robust against cross-device variations in wireless signals.

Comments on Authorship

Overall, the paper and its main idea of the localization using WiFi probes was conceived by myself. I personally designed on the high-level architecture. I help in the implementation of the approach. In fact I implemented the localization prediction components. Furthermore, I proposed and helped the excecution of the evaluation approach, followed in this paper. Finally, with the valued guidance and supervision of the coauthors, I wrote the majority of the text in this paper.

2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

Cross-Device Radio Map Generation via Crowdsourcing

Georgios Pipelidis[†¶], Nikolaos Tsiamitros[*¶], Efdal Ustaoglu[*], Romeo Kienzler[‡], Petteri Nurmi[§], Huber Flores[§], and Christian Prehofer[*]

[*] Technical University of Munich, Software and Systems Engineering Research Group, Munich, Germany

[†] Ariadne Maps, Munich, Germany

[‡] IBM Watson Internet of Things, Zurich, Switzerland

[§] University of Helsinki, Department of Computer Science, Helsinki

[¶] These two authors contributed equally.

***Abstract*—Crowdsourcing is a powerful technique for bootstrapping sensing systems that are based on wireless signals. For example, wireless sensing systems can ask users to contribute training data and localization systems (such as WiFi fingerprinting) can take advantage of wireless measurements provided by users of the system. Indeed, previous research has demonstrated that crowdsourcing can reduce labor efforts needed to deploy and initialize wireless sensing systems by several orders of magnitude, without compromising on system performance. Despite the many benefits of crowdsourcing, current methods suffer from one significant drawback, namely that they are highly sensitive to variations in devices capturing the measurements. Indeed, as we demonstrate in this paper, cross-device variations can decrease performance of crowdsourced bootstrapping approaches up to 70of radio maps used for localization and to make them robust against cross-device variations in wireless signals. We evaluate our framework by considering WiFi fingerprinting based localization as a representative example of applications that benefit from our approach. Our results demonstrate up to 1.8m localization error, and 18.7**

***Index Terms*—crowdsourcing, indoor mapping, landmark mapping, fingerprint localization, heterogeneity correction, radiomap generation**

I. Introduction

"Organic maps" are focused on continuously updating databases-for-localization or radio-maps. The term implies the organic growth of radio-maps through data provided by the crowd. The first organic maps required users to mark their location manually [21], [14], while it was later suggested to acquire this information following opportunistic approaches [16]. Organic maps are mostly used with fingerprint-based methods, since they rely on databases composed by pairs of <fingerprint, ground-truth location> [21]. When growing organic maps, data may be collected in two ways, either following a "landmark" based approach, where data is collected in a discrete manner, or a SLAM-like approach [3] in a more continues manner. In this paper we will take both approaches into consideration.

978-1-7281-1788-1/19/$31.00 2019 IEEE © 2019 IEEE

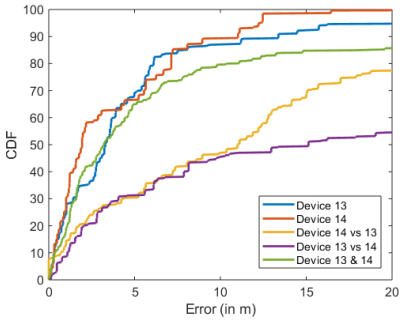

Fig. 1. The error distribution in meters, when machine learning models have been trained with: (a) HTC Wildfire S (Device 13), (b) LT22i (Device 14), (c) LT22i, (d) HTC Wildfire S, (e) both devices, and evaluated on data extracted by: (a) HTC Wildfire S (Device 13), (b) LT22i (Device 14), (c) HTC Wildfire S, (d) LT22i and (e) both devices. As can be seen the error grows rapidly when cross-training occurs.

Problem: Even though growing organic maps can reduce labor efforts by several orders of magnitude, cross-device variations can decrease performance up to 70%. Consider, for example, the UJIIndoorLoc dataset [19], which has been composed by aggregating data from 24 different phones. Data on this dataset are not always collected for every phone by every place. Hence, cross-device-training is inevitable. Furthermore, as can be seen in Figure 1, although the localization error is tolerable when the localization algorithm is evaluated in the same device that was used to collect the data when cross-training between different devices occurs, the error can be increased up to 70%. This effect has been repeatedly noticed in the literature [2], [18], [15], [23], [6], [11], [5] and [1].

Goal: The goal of this paper is to provide a method for improving the generality of organic approaches and to enhance their robustness against cross-device variations.

Contribution: The contribution of this paper can be summarized as follows: we first provide a novel approach for improving the cross-training procedure by integrating state-of-the-art solutions to our archi-

tecture. Then we provide an architecture for growing organic maps following a crowdsourcing approach that enables feature selection, cluster analysis, and cluster selection for enhancing the localization accuracy. We evaluated our method, and our results demonstrate up to 1.8m localization error, and 18.7% improvements in robustness.

Paper Structure The structure of this paper is as follows. In Section II, related work is presented. In Section III, we explain every component of our architecture. In Section IV, we provide a detailed evaluation of our proposed system. The final section concludes the paper and describes our future work.

II. Related work

Many prior studies have focused on crowd-sourcing the generation of radio maps. Among the most popular are Zee [17], UnLoc [24], JustWalk [4], RMapCS [26], as well as [12], [27], [28], [25], [9], [7]. All of this prior work operates as following, a smartphone application enables localization from a known device using the Inertial Motion Unit (IMU) of the phone in a dead-reckoning-like approach or some even more sophisticated introducing a fusion between IMU and either a map or a second localization approach, for example based on geomagnetic landmarks that enables loop closure detection. These methods might even use sophisticated fusion techniques such as particle or Kalman filters. However, the common characteristic of these approaches is that they do not take into consideration the heterogeneity between smartphones. Heterogeneity as we have already demonstrated is a significant challenge when a crowdsourced procedure is followed.

The problem of heterogeneity has also been researched in studies such as the following; Kyle F. Davies et al. [2] use a generative, hierarchical Bayesian approach, which implies that all unknown quantities are treated probabilistically. As a result, the starting point is a hypothesis of a probability function, and the priors are random variables, which enables them to cope with device heterogeneity, by obtaining different parameters for governing the transmission and power or gain of wireless devices. Marios Raspopoulos [18] follows a linear transformation, where data captured from multiple devices are fitted in a model that later uses the corresponding RSSI value for every target device. Jun-Geun Park et al. [15], show that linear transformation is not enough and they compensate for the uncertainty by computing a kernel density for every landmark, which they later use in training as well as in the testing phase. Thorsten Vaupel et al. [23] took into consideration the differences in RSS as static offsets, and they compensate by adding or removing the difference appropriately when compared to others. Shih-Hau Fang et. al. [6], present a comparison of the hyperbolic location fingerprinting (HLF) of Kjrgaard et al. [11], who utilized signal strength ratios to overcome the problem of heterogeneity, against signal strength difference between pairwise APs (DIFF) of F. Dong, et al. [5] and the Signal Strength Differences (SSD) by A. Mahtab Hossain et al. [8] focuses on reducing the computational overhead of DIFF and hence selects an independent subset of it. Shih-Hau Fang et al. [6] shows that although RSS achieves 56.02% localization accuracy, DIFF can achieve up to 67.50%, SSD up to 69.31% and HLF up to 57.06%. Finally, Mustafa Abbas et al. [1] inject random noise to simulate the non-visibility of APs as well as the RSS variations, even in similar phones, while they train for every landmark a Deep Learning model to predict the RSS variations between different phones.

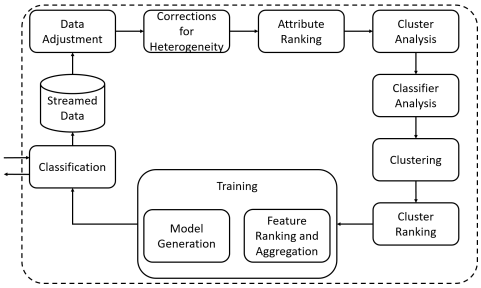

Fig. 2. The Figure describes system architecture.

III. Approach

The proposed system is executed in a distributed environment, where a state-of-the-art SLAM-like algorithm is used for collecting and labeling, with locations, Bluetooth and WiFi data, extracted from smartphones, and streams them to a server. Quality control on the streamed data is performed on the server to identify whether the data is labeled with a location. Data labeled with location are subject to further processing in our method. We then perform an attribute ranking to quantify the contribution of every attribute to the localization method. Once good attributes are selected, a cluster analysis is performed, using the elbow method. Its purpose is to identify the optimum number of clusters in the dataset. When the number of clusters is revealed, the data are clustered in the appropriate amount of clusters. After clustering, the data is labeled with a corresponding ID and is ranked similar to the attributes. Finally, after the best clusters have been selected, a classifier is trained, and location prediction is enabled.

A. Streaming Data

The first component of the proposed system is responsible for parsing the data streamed by the smartphone application from the server (Google Firebase [20]) and then fitting them into data structures. Each distinct Bluetooth/WiFi unique device identifier (i.e., mac address or device's UDID) is stored in a list

2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

together with each distinct latitude and longitude position and its normalized received signal strength. In the streaming component, the REST API call is applied, and the request and response are transferred using the JSON data format. Additionally, the streamed location's latitude and longitude values with the instant timestamp are streamed for further analysis.

B. Data Adjustment

The data adjustment component is responsible for transforming the data into more easily to use segments. As a result, all the attributes are collected in a list, S and hence each attribute is $S = [T, X, Y, Z, A]$ where T is the timestamp, X is the latitude, Y is the longitude, Z is the floor and A is a list of all signals' histogram. Hence, $A = [B_1, B_2, B_3, .., B_m]$, where m is the number of attributes. Moreover, $B = [macAddress, name, signalStrength]$, since every B is described by three characteristics: mac address, name, and signal strength. After collecting all S, we can describe it in a list such as, $M = [S_1, S_2, .., S_n]$, where n is the number of all collected attributes. In order to use M, it has to be in the format of a dictionary, since at this stage, the same access point might be in random positions in A along with the different attributes. Therefore, a function is applied to it: $M_D = CreateDictionary(M)$. This function orders the raw data following a dictionary approach for later processing. The result is M_D, which is equivalent to $M_D = [S_1^D, S_2^D, S_3^D, .., S_n^D]$ and each element of M_D is $S_i^n = [T, X, Y, Z, A^D]$. As can be seen, this step affects only on A (i.e., transformed to A^D, a histogram that describes a list of all received signals). Hence A here becomes $A^D = [B_1^n, B_2^n, B_3^n, .., B_m^n]$. As a result, each element of A^D is $B_{(i)N}^D = [S_1^D, S_2^D, S_3^D, .., S_N^D]$. Where $B_{(i)N}^D$ is the i^{th} element of the newly generated vector of attributes with length N, which is the same $\forall B^D \in A^D$.

C. Heterogeneous Devices Corrections

Since the incoming data can be from various sources heterogeneous devices require correction before sharing fingerprint databases. In our experiment, we use common landmarks to calibrate incoming data based on their known changes. Hence, if the channel follows IEEE 802.11 standards, then the signal strength might vary between ≈ -92 and ≈ -24 and if n measurements x_1 to x_n of the signal strength from device A for landmark K and y_1 and y_1 to y_n of device B from the same landmark K, then if x the RSS of device A in a particular moment, the corresponding RSS for device B can be estimated as:

$$P_B(x) = \sum_{i=1}^{n} y_i \mathcal{L}_{n,i}(x) \tag{1}$$

where $\mathcal{L}_{n,i}$ is the Lagrange coefficient and

$$\mathcal{L}_{n,i}(x) = \frac{\prod_{i=1}^{n}(x - x_i)}{\prod_{i=1}^{n}(x_k - x_i)} \tag{2}$$

D. Attribute Ranking

After the data is adjusted, this component is responsible for ranking each attribute. All the attributes are analyzed, and bad attributes are removed. Hence, assuming that $A_{area} = [min_{Lat}, min_{Lon}, max_{Lat}, max_{Lon}]^T$ defines the area where attributes are non-zero and every attribute is defined from $P_i^M = [Latitute_i^M, Longitude_i^M]$ that contains the i^{th} latitude and longitude from $S_i^D \in M_D$, we chose the attributes for which the following expression holds $P, (P \rightarrow Q) \vDash Q$, where $Q : P_i^D > 0 \wedge inside(P_i^D, A_{area})$. As a matter of fact, we also have $\bar{P}_i^M$ where $\bar{P}, (P \rightarrow \neg Q) \vDash \neg Q$, which is true only if $\neg Q : P_i^D = 0 \wedge inside(P_i^D, A_{area})$.

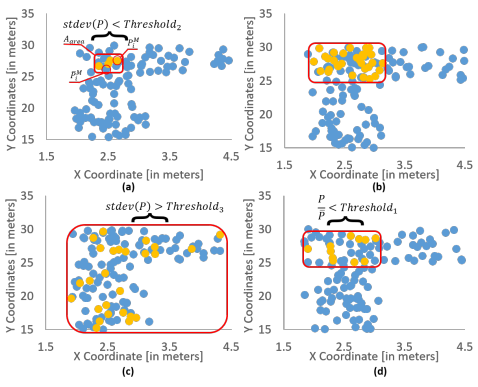

Fig. 3. The four possible scenarios in attribute ranking. a: example where $stdev(P) < Threshold_2$, b: a successful case, c: example where $stdev(P) > Threshold_3$, d: example where $\frac{P}{\bar{P}} < Threshold_1$.

More specifically, in this step, we estimate the frequency of the areas where the attributes are non-zero (i.e., P_i^M) as well as when they are zero (i.e., $\bar{P}_i^M$). Once we obtain P and $\bar{P}$, we rank the attribute based on three different metrics. The first metric is the ratio between zero and non-zero values inside A_{area}, defined as $\frac{P}{\bar{P}} > Threshold_1$, where $Threshold_1$ is defined at 1.0. The second condition is based on their spread, which is estimated by the standard deviation based on the following contrition $Threshold_2 < stdev(P) < Threshold_3$, where $Threshold_2$ is defined at 0.2 meters, $Threshold_3$ is defined at 4.0 meters. A successful example of this ranking can be seen in Figure 3.b. Additionally, Figure 3.a shows an example where $stdev(P) < Threshold_2$, Figure 3.c shows an example where $stdev(P) > Threshold_3$. Finally, in 3.d shows an example where $\frac{P}{\bar{P}} < Threshold_1$.

2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

E. Cluster Analysis

The cluster analysis component is responsible for identifying the optimum number of clusters in every dataset. For this purpose, the elbow method [10] is chosen. The elbow method is designed to help find the appropriate number of clusters in a dataset. It is a method of interpretation and validation of consistency within the cluster analysis. The elbow method looks at the percentage of variance explained as a function of the number of clusters. The optimum number of clusters is achieved such that adding another cluster does not significantly increase the variance. If we plot the variance against the number of clusters, the first clusters will add much information, but at some point the marginal gain will drop, giving an angle to the graph. As a result, the number of clusters is chosen at this point. The input to the elbow method is only the M^D, which is a set of all A^D, and hence, it is not aware of any spatial coordinates or timestamps. The clustering analysis has as an out a single number k, which is the optimal number of clusters.

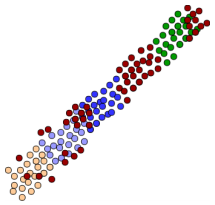

Fig. 4. A clustering example of five different clusters as provided by K-Means. As can be seen in the figure, the brown cluster does not add information, since it is spread along other clusters. As a result, it will be rejected.

F. Clustering

After the number of clusters is identified, the data are clustered into smaller segments using k-Means. The clustering component has as input M^B and the estimated k. The component's output is a vector M^{IND} of length n similar to the length of M^D that carries the assigned classes for every input point and the cluster centroids.

G. Cluster Ranking

This component is responsible for removing bad clusters. A lousy cluster is usually a cluster that does not have uniform spatial relations, which implies that, although their spatial properties have not been taken into consideration, clusters are expected to be spatially uniform. As a result, clusters are ranked similar to attributes but, instead, we consider only the ratio between a cluster and the rest of the clusters, since rejecting clusters according to their spread cannot improve the localization accuracy at this point. An example of a lousy cluster can be seen in Figure 4. As can be seen, the brown cluster is spread along with other clusters, and hence, it will add confusion in the localization method.

H. Training

In this component, a classifier is trained for predicting a series of locations. More specifically, the classifier chosen is the Linear Discriminant Analysis (LDA) [13]. As can be seen in the evaluation, it has many advantages, such as it learns faster, it can score a higher accuracy. It targets the dimensionality reduction; it is faster than most other available classifiers. LDA tries to maximize the separability between different groups in order to make the best decisions. It works by generating a new axis between several features in the dataset according to the number of classes and later projects the features into this new axis in a way that maximizes the separation of categories. The new axis is created based on two criteria. The first is to maximize the distance between the various means μ of our features, while the second criteria targets the minimization of the variance within each category S and hence the problem can be formalized as $\frac{(\mu_1-\mu_2)^2}{S_1^2+S_2^2}$. Ideally, the nominator will be a large number, while the denominator will be minimal and hence we are looking to maximize the ratio.

I. Classification

This component is where the generated classifier is used. The classification is done in real-time and can be explained as follows. First, the user is streaming a data segment of M^D. As already mentioned, the data segment is already tagged with a location estimated through a particle filter algorithm executed on the smartphone. However, explaining the functionality of the smartphone exceeds the focus of this paper. After all, there is no use of the streamed location coordinates here. Instead, the input to the classifier is equivalent to M^B as described in III-E, while the output of the classifier is $N(P^C; P^C_\mu, P^C_\sigma)$, where P^C is the vector of positions that the cluster occupies, P^C_μ is the median location to which the particular received signal strength corresponds and P^C_σ is the standard deviation of the cluster.

IV. Evaluation

In this section, we provide an evaluation of our components. More specifically, we first estimate how successful is the process of turning continues data into desecrate clusters. This process is the clustering analysis and is evaluated based on the standard deviation between the different predictions on the number of clusters with each additional dataset IV-A. Additionally, We provide a comparison between different frameworks that consider heterogeneity IV-C. Furthermore, we provide an evaluation of the entire flow IV-D, where the error is estimated in meters from (a) an existing open-sourced dataset and (b) a dataset generated and open-sourced by us. Finally, we provide an evaluation of six different classifiers and how their accuracy evolves with every additional dataset IV-B.

2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

We project those values in an attempt to identify the optimal number of datasets per classifier.

A. Clustering Analysis

In this section, we provide the evaluation of the clustering analysis. In this section, we provide the evaluation of the clustering analysis. As already mentioned, for the clustering analysis, the elbow method was used. The criteria are applied to the standard deviation between the different predictions of the number of clusters, while the number of datasets was increasing, which is presented in Table I. The threshold used in the elbow method was an average of 90% uniformity between the points inside each cluster. In this paper process of collecting data from a particular area will be referred to as a visit. For instance, the first two visits require at least 17 clusters, in order to reach cluster homogeneity of 0.9. After the third visit, it reduces to 12, which shows that more visits are necessary since the number of clusters is still not satisfied.

TABLE I
THE PREDICTED NUMBER OF CLUSTERS FOR THE ACCUMULATED NUMBER OF VISITS.

Visits	1	2	3	4	5	6	7	8	9
Clusters	18	19	12	10	16	19	17	16	17
Visits	10	11	12	13	14	15	16	17	18
Clusters	16	17	16	17	16	22	22	20	20

The standard deviation during the three first visits is more significant than the four clusters, which can be interpreted as a potential error decrease, between the first and fourth cluster number suggestion, of twenty meters, considering the fact that obtaining a number of twelve clusters in a seventy-nine-meter corridor can lead to an average distance of six meters between the cluster centers, between the resulting clusters. The optimum number of visits is eight, since, in such a case, the number of clusters varies between 21 and 19, and hence the maximum potential error varies between 3.7 and 4.1 meters.

B. Algorithm Comparison

In this section, we provide an evaluation of six different classifiers and how their error evolves with every additional aggregated dataset (i.e., visit). We project those values in an attempt to identify the optimal number of visits per classifier. Those classifiers are: (1) Logistic Regression (LR), (2) Linear Discriminant Analysis (LDA), (3) k-Nearest Neighbors (k-NN), (4) Classification And Regression Tree (CART), (5) Naive Bayes (NB) and (6) Support Vector Machines (SVM). For this evaluation, the same dataset extracted from the subway station is used.

The motivation for this evaluation is to enable us to monitor the increase in accuracy for each collected dataset and thus to approximate the number of minimum required visits to achieve the optimum localization accuracy. In addition to the accuracy, the standard deviation of the error was also monitored. In this way, the rate of decrease on the uncertainty of the classification result can also be quantified, and the appropriate number of datasets necessary for reducing this uncertainty to zero can be estimated.

Logistic regression In LR, by interpolating the classification results, we extract $y = 0.1489{*}ln(x){+}0.3963$, where y is the expected accuracy, and x the number of the required visits. For this classifier, the accuracy is rapidly increased before the fifth visit, while the error deviation drops after the second visit, and the standard deviation at the second visit is 0.07, while the accuracy is 0.64 and 0.73 at visits 3 and 4 respectively. We can conclude that 57 visits would be required for accuracy of almost 100%, while 51 visits would be enough to obtain the minimum possible uncertainty in the error estimation.

Linear Discriminant Analysis: In LDA, by interpolating the classification results, the following formula can be extracted $y = 0.0295 * ln(x) + 0.8945$, where y is the expected accuracy and x the number of the required visits. Here, the accuracy is already significantly increased after the second visit, while it does not highly improve after the addition of more datasets. The computed accuracy and standard deviation of the error in a K-Fold validation at the third visit are 0.98 and 0.02, respectively. From interpolating, we can conclude that 35 visits would be required for accuracy of almost 100%, while 36 visits would be enough to obtain the lower uncertainty on the error.

k-Nearest Neighbors Classifier: In k-NN, by interpolating the classification results, the following formula can be extracted $y = 0.0445 * ln(x) + 0.8281$, where y is the expected accuracy and x the number of the required visits. The k-NN classifier reaches its full potential after the second visit. More specifically, it increases from 0.7 to 0.92, with only one additional dataset while the standard deviation decreases from 0.13 to 0.03. By interpolating, we can conclude that 47 visits would be required for accuracy as closest to 100% as possible, while 27 visits would be enough for obtaining the lowest possible uncertainty of the error.

Classification and Regression Tree Classifier: In CART, by interpolating the classification results, the following formula can be extracted $y = 0.0566 * ln(x) + 0.8009$, where y is the expected accuracy and x the number of the required visits. This classifier's accuracy reaches closer to its maximum potential after the second visit. Later visits improve its accuracy only slightly and hence three visits would be an optimal minimum for reaching an average accuracy of 0.9. However, it requires at least four visits in order to achieve a good standard deviation of the error in a K-Fold validation. Similarly, in accuracy, later visits contribute on gaining a lower level but they do not affect considerably. From the interpolation, we can

conclude that 33 visits would be required for an accuracy close to 100%, while 23 visits would be enough for obtaining the lowest possible uncertainty of the error.

Gaussian Naive Bayes: Naive Bayes is trained and tested similar to the other classifiers. However, the number of aggregated datasets from consequential visits do not seem to significantly influence its accuracy.More specifically, although, the accuracy is varied between 0.6 and 0.8, Here, we can extract the following formula $y = 0.0445 * ln(x) + 0.8281$, where y is the expected accuracy and x the number of the required visits. From the formula, we can conclude that we would need 47 visits for an accuracy that would approximate 100%, while we would need approximately 30 visits for achieving the lowest possible uncertainty.

Support Vector Machines: Finally, the SVM algorithm seems to continuously improve with every additional dataset. However, it can hardly reach a classification accuracy of 0.15 in the 20^{th} visit. By interpolating the classification results, the following formula can be extracted $y = 0.0434 * ln(x) + 0.0523$, where y is the expected accuracy and x the number of the required visits. From the formula, we can conclude that for an accuracy that approximates 100%, we would require tens of thousands of visits. As a result, the SVM algorithm will not be taken further into consideration.

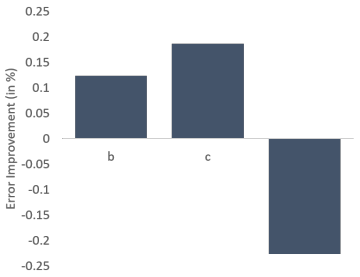

Fig. 5. Improvement of the localization error due to heterogeneity corrections normalized to (a) where plain RSS was used. Here (b) is linear approach as proposed by [18], [23], [5], (c) is hyperbolic approach, proposed by [11] and (d) noise injection method, proposed by [2], [15] and [1].

C. Heterogeneity Compensation

In this section we provide a comparison that a series of approaches, used for heterogeneity compensation, have in the UJIndoorLoc dataset. More specifically, we examine the influence with: (a) no correction, as approached by the vast majority of papers, (b) linear corrections, as proposed by [18], [23], [5] and [8], (c) hyperbolic corrections, as proposed by [11], and finally (d) noise injection, as proposed by [2], [15] and [1]. As can be seen in Figures 5 and 6 linear interpolation can provide 12.3% improvement

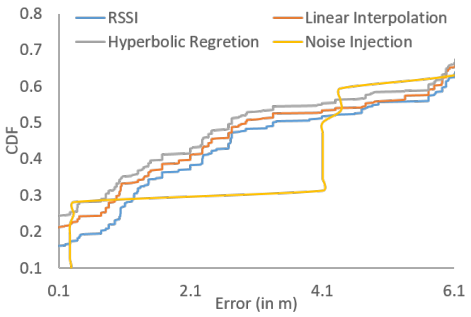

Fig. 6. Improvement of the localization error due to heterogeneity corrections, cumulative distribution function.

to the localization error, while hyperbolic interpolation can even reduce the localization error by 18.7%. Unfortunately, in the UJIndoorLoc, we did not see improvement by the noise injection. The most likely reason is due to lack of enough data to successfully estimate the standard deviation of the kernel that will perform the noise injection.

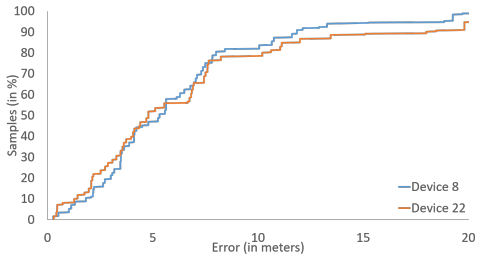

Fig. 7. The error distribution in meters.

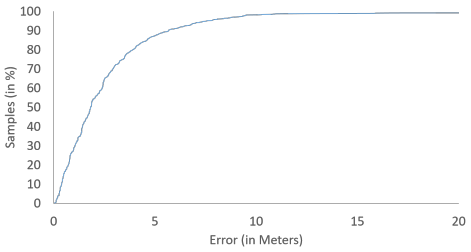

Fig. 8. The error distribution in meters.

D. Error Estimation in Distance

Data Analysis of Building 2: On the floor 0 of building 2, devices 8 (Galaxy Nexus, Android operating system 4.2.2) and 22 (Orange Monte Carlo, Android operating system 2.3.5) are used for the training and validation. The classifier was trained and tested on floor 0 with K-Fold validation. First, the classifier is trained and evaluated with device 8, where 550 instances are collected. As can be seen in Figure 7

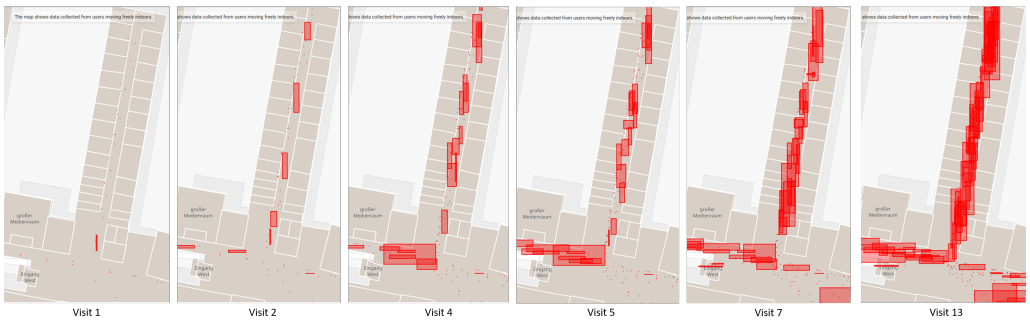

Fig. 9. The evolution of the cluster formation for every aggregated dataset, where Visit 2 is an aggregation of visit 1 and 2.

(blue line), the average distance error is 6.20m, the median is 5.32m, the 75^{th}, 90^{th} and 95^{th} percentiles are 7.50m, 11.85m and 18.84m, respectively. Second, the classifier is trained and evaluated with the device 22, where 527 instances are collected. As can be seen in Figure 7 (red line), the average distance error is 7.28m and the median is 4.79m , the 75^{th}, 90^{th} and 95^{th} percentiles are 7.65m, 18m and 20.74m, respectively.

Our Dataset: As already mentioned in the introduction, we implemented our own tool for collecting data, we have collected data from a subway station environment, we have open-sourced the dataset and is available here [22]. Twenty different datasets were collected along a seventy-nine-meter corridor at the Münchner Freiheit metro station in Munich, Germany. The data was collecting with a Samsung Galaxy S6 Edge+ with OS Android 7.0. As can be seen in Figure 8, the error curve is smoother than the earlier figures. The main reason is because the data was collected following a more continuous approach with an average distance between extracted data at 1.4 meters. Additionally, as can be seen in Figure 8, the average error is at 2.81 meters, the median error is at 1.82 meters and the 75^{th}, 90^{th} and 95^{th} percentiles are 3.48, 5.68 and 7.55 meters, respectively.

E. Influence of Visits in the Localization Error

In this section, we present the influence that aggregated datasets or visits have in the localization accuracy. This evaluation was contacted in a dataset collected in the Informatics-Mathematics building of the Technical University of Munich. The data was collected by 13 different people while walking along four different corridors and two hallways and climbing a staircase. The dataset is available here [22]. The error is estimated in meters, and the accuracy has been monitored for each additional dataset. The evolution of cluster formation is presented in Figure 9. As depicted here, every additional dataset introduces new clusters, while the cluster size, or the area that each cluster occupies, increases with additional datasets until a specific size, and later the size becomes stable. Although the algorithm obtains a small number of clusters, even during the sixth visit, and almost no surface is occupied by clusters, by the 13^{th} aggregated visit, the algorithm generates tens of clusters and almost the entire area is covered by cluster surfaces. Additionally, as can be seen in the Cumulative Distribution Function (CDF) Figure 10, the accuracy is improved with each additional dataset, reaching its maximum accuracy with the 13^{th} aggregated dataset. More precisely, in Figure 11, we monitor the median error, which begins from over 30 meters in the six aggregated visits and reaches the 4.3 meters in the 13^{th} dataset. Additionally, in the figure, we can see the 75^{th} percentile, which begins from an error that reaches 50m offset and later is decreased to an error of less than 10m.

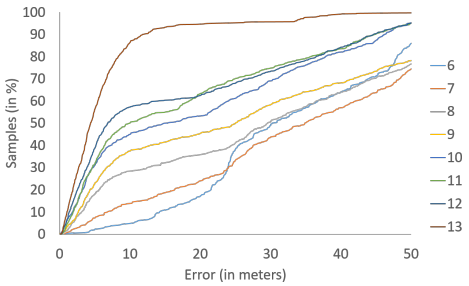

Fig. 10. Cumulative Distribution Function (CDF) for each dataset, where 6 implies six aggregated datasets.

V. Conclusions and Future Work

In conclusion, this paper provides a novel fingerprint technique for indoor localization, which is keeping itself continuously up-to-date, while its localization accuracy continuously improves. We achieve this by introducing a novel approach for clustering the collected data in real-time. Once data are collected and then clustered, they are ranked, and classifiers are trained for enabling indoor localization. Additionally, this paper provides a detailed study of how classification accuracy increases based on the number of datasets collected, and we evaluated different

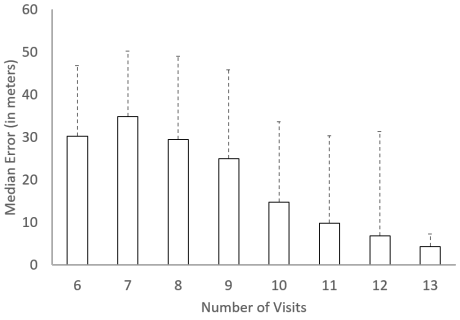

Fig. 11. The accuracy of Support Vector Machines over the number of visits based on the median error in meters. The error bars are extended to the 75^{th} percentile.

classifiers that can be used for localization. The best classifier for a small amount of data is Linear Discriminant Analysis, while for more data, others perform good as well. Finally, in our evaluation, we achieve a median error of 1.8 meters purely through fingerprint technology. In our future work, we want to extend our architecture to enable it to dynamically select the classifier to train based on the number of collected datasets.

VI. Acknowledgments

This work is part of the IOTA ecosystem and has been supported by the IOTA Ecosystem Development Fund.

References

[1] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and M. Youssef. Wideep: Wifi-based accurate and robust indoor localization system using deep learning. 03 2019.

[2] K. F. Davies, I. G. Jones, and J. L. Shapiro. A bayesian approach to dealing with device heterogeneity in an indoor positioning system. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2018.

[3] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics Automation Magazine*, 13:99–110, June 2006.

[4] M. M. Elhamshary, M. F. Alzantot, and M. Youssef. Justwalk: A crowdsourcing approach for the automatic construction of indoor floorplans. *IEEE Transactions on Mobile Computing*, pages 1–1, 2018.

[5] J. L. Q. N. F. Dong, Y. Chen and S. Piao. A calibration-free localization solution for handling signal strength variance. In *MELT (Berlin, Heidelberg)*, page 79?90. Springer-Verlag, 2009.

[6] S.-H. Fang, C.-H. Wang, S.-M. Chiou, and P. Lin. Calibration-free approaches for robust wi-fi positioning against device diversity: A performance comparison. In *2012 IEEE 75th vehicular technology conference (VTC Spring)*, pages 1–5. IEEE, 2012.

[7] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2004.

[8] A. K. M. M. Hossain, H. N. Van, Y. Jin, and W. Soh. Indoor localization using multiple wireless technologies. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–8, Oct 2007.

[9] S.-H. Jung and D. Han. Automated construction and maintenance of wi-fi radio maps for crowdsourcing-based indoor positioning systems. *IEEE Access*, 6:1764–1777, 2018.

[10] D. J. Ketchen Jr and C. L. Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, pages 441–458, 1996.

[11] M. B. Kjærgaard. Indoor location fingerprinting with heterogeneous clients. *Pervasive and Mobile Computing*, 7(1):31–43, 2011.

[12] Z. Li, X. Zhao, and H. Liang. Automatic construction of radio maps by crowdsourcing pdr traces for indoor positioning. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2018.

[13] G. McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004.

[14] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. page 271. ACM Press, 2010.

[15] J.-g. Park, D. Curtis, S. Teller, and J. Ledlie. Implications of device diversity for organic localization. In *2011 Proceedings IEEE INFOCOM*, pages 3182–3190. IEEE, 2011.

[16] L. Radaelli and C. S. Jensen. Towards fully organic indoor positioning. pages 16–20. ACM Press, 2013.

[17] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 293–304. ACM, 2012.

[18] M. Raspopoulos. Multidevice map-constrained fingerprint-based indoor positioning using 3-d ray tracing. *IEEE Transactions on Instrumentation and Measurement*, 67(2):466–476, Feb 2018.

[19] U. M. L. Repository. Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems in proceedings of the fifth international conference on indoor positioning and indoor navigation, 2014, Accessed: 2018-03-21.

[20] S. T. Sonam Khedkar. Real time databases for applications. International Research Journal of Engineering and Technology (IRJET), June-2017.

[21] S. Teller, J. Battat, B. Charrow, D. Curtis, R. Ryan, J. Ledlie, and J. Hicks. Organic indoor location discovery. *Computer Science and Artificial Intelligence Laboratory Technical Report*, 75:16, 2008.

[22] E. Ustaoglu. Muenchner freiheit bluetooth fingerprint. https://github.com/efdalustaoglu/wifi-localization-data, Accessed: 2018-04-29.

[23] T. Vaupel, J. Seitz, F. Kiefer, S. Haimerl, and J. Thielecke. Wi-fi positioning: System considerations and device calibration. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7. IEEE, 2010.

[24] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 197–210. ACM, 2012.

[25] Y. Ye and B. Wang. Indoor radio map construction based on crowdsourced fingerprint splitting and fitting. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, March 2017.

[26] Y. Ye and B. Wang. Rmapcs: Radio map construction from crowdsourced samples for indoor localization. *IEEE Access*, 6:24224–24238, 2018.

[27] M. Zhang, L. Pei, and X. Deng. Graphslam-based crowdsourcing framework for indoor wi-fi fingerprinting. In *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, pages 61–67, Nov 2016.

[28] B. Zhou, Q. Li, Q. Mao, and W. Tu. A robust crowdsourcing-based indoor localization system. *Sensors*, 17(4), 2017.

4 Conclusion and Open Challenges

"I may not have gone where I intended to go, but I think I have ended up where I needed to be."

— Douglas Adams

This thesis discussed our view on the future of techniques for indoor mapping. We proposed customized, crowdsourced and scalable approaches. We introduced an adaptive method for bootstrapping the procedure of indoor mapping. We enabled a novel sensor fusion framework, for dynamically mapping vertical characteristics of buildings with uncertain sensor data collected via crowdsourcing. Our method identifies the outdoor-to-indoor-transition of users, while walking in a building, and uses this information for the estimation of reference altitude and a reference pressure. We faced an unsupervised classification problem, in which the number of floors and the altitude for each floor was unknown. We used clustering analysis techniques and clustering algorithms to face this problem. We proposed ways to map characteristics and we proposed enhancements of existing standards. We introduced approaches for the dynamic extraction of semantic properties of indoor places from a user context. The user context was estimated from inertial motion unit sensor data. We used probability theory and fuzzy logic for activity recognition and decision making respectively.

Additionally, we proposed and developed a 3D indoor localization system for diverse indoor environments that is based exclusively on the smartphone sensor data and crude floor plans, commonly available in open street maps. Its module is a particle filter that fuses gyroscope and the accelerometer sensor measurements with contextual data. Our system can achieve twice better accuracy when compared to state-of-the-art solutions while using an order of magnitude fewer particles. Furthermore, we provided an organic fingerprint technique for indoor localization, that continuously updates and improves autonomously. Once data is collected, it is clustered, ranked, and used to train a classifier that enables indoor localization predictions. We found that our method can achieve a median error of 1.8 meters purely through fingerprint technology. We enabled passive spatial human analytics using WiFi probes, by the use of "surveyor" devices, i.e. devices that can monitor signals emitted by smart devices. We achieved this by deploying specialized devices in the area, and by using our novel localization method for labeling RSS from probes requests to locations. We then combined WiFi probes to locations and generated a radio map that enabled tracking of the majority of devices in an area. Our median error is 1.56m, the 75^{th} percentile is 4.22, while the 95^{th} percentile is 10.25m.

We envision that all the developed frameworks will play a key role in future research

of indoor localization and indoor mapping problems. In particular, in today's privacy concerned world [Bru], and considering the fact that passive localization approaches are widely available, research on privacy-preserving methods are essential. We believe that our system will be used as a "playground" for anonymization solutions in the future. Additionally, scalability is a crucial issue for the survival of indoor LBS. As a result, solutions that can be rapidly deployed, can make use of existing infrastructure, and do not require high installation effort, have to be researched. Lack of synchronization methods is the reason for the lack of anonymization methods. Hence, research on wireless nanosecond-based synchronization has to be further researched. Visualization of spatial data is crucial. Today there is a large number of projects that enable the visualization of sensor data. Although, these projects provide great detail outdoors, unfortunately, there is a lack of tools for providing such services indoors. Visualizing floodplains does not provide a seamless user experience yet. Further research is needed and standards for visualization have to be established.

Finally, with roaming being increasingly available between countries fewer and fewer people are using WiFi data, which fact when combined to the advancements on 5G technologies, implies that localization and tracking will be dominated by telecommunication antennas. Hence, research on localization and tracking with cellular technologies have to be researched.

Bibliography

[29] Adopted Specifications | Bluetooth Technology Website. https://www.bluetooth.com/specifications/adopted-specifications. accessed online 05/2016.

[A+05] Küpper Axel et al. *Location-based services: fundamentals and operation*. John Wiley & Sons, Chichester, England ; Hoboken, NJ, 2005.

[ARC10] Imad Afyouni, Cyril Ray, and Christophe Claramunt. A fine-grained context-dependent model for indoor spaces. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, pages 33–38. ACM, 2010.

[AY12a] M. Alzantot and M. Youssef. UPTIME: Ubiquitous pedestrian tracking using mobile phones. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 3204–3209, April 2012.

[AY12b] Moustafa Alzantot and Moustafa Youssef. Uptime: Ubiquitous pedestrian tracking using mobile phones. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 3204–3209. IEEE, 2012.

[BAOBZ15] Roeland Boeters, Ken Arroyo Ohori, Filip Biljecki, and Sisi Zlatanova. Automatically enhancing citygml lod2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science*, 29(12):2248–2268, 2015.

[BBM87] Robert C. Bolles, H. Harlyn Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[BD06] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics Automation Magazine*, 13(3):108–117, 2006.

[BEGH13] C. Bollmeyer, T. Esemann, H. Gehring, and H. HellbrÃ¼ck. Precise indoor altitude estimation based on differential barometric sensing for wireless medical applications. In *2013 IEEE International Conference on Body Sensor Networks*, pages 1–6, May 2013.

[Ber] April Berthene. Why one airport chooses Wi-Fi for mobile navigation over beacons. https://goo.gl/V3koXv. accessed online 05/2016.

[BOBZ15] Roeland Boeters, Ken Arroyo Ohori, Filip Biljecki, and Sisi Zlatanova.

Automatically enhancing citygml lod2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science*, 29(12):2248–2268, 2015.

[Bru] 11 June 2015 Brussels. Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation) - preparation of a general approach.

[bui04] Buildings and the Environment: A Statistical Summary. Technical report, U.S. Environmental Protection Agency Green Building Workgroup, December 2004.

[C62] H.P.V. C. Method and means for recognizing complex patterns, December 18 1962. https://www.google.com/patents/US3069654.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.

[CCC+16] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[CDS+11] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman. Indoor location sensing using geo-magnetism. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 141–154. ACM, 2011.

[CKKC15] Hang Chu, Dong Ki Kim, and Tsuhan Chen. You Are Here: Mimicking the Human Thinking Process in Reading Floor-Plans. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2210–2218, 2015.

[CLCS10a] M. Cypriani, F. Lassabe, P. Canalda, and F. Spies. Wi-fi-based indoor positioning: Basic techniques, hybrid algorithms and open software platform. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–10, Sept 2010.

[CLCS10b] Matteo Cypriani, FrÃ©dÃ©ric Lassabe, Philippe Canalda, and FranÃ§ois Spies. Wi-Fi-based indoor positioning: Basic techniques, hybrid algorithms and open software platform. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–10. IEEE, 2010.

[CR15] C. Combettes and V. Renaudin. Comparison of misalignment estimation techniques between handheld device and walking directions. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, October 2015.

[Dai13] J Christopher Dainty. *Laser speckle and related phenomena*, volume 9. Springer Science & Business Media, 2013.

[DB06] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, 2006.

[DM96] Pierre Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581, 1996.

[DM97] Pierre Del Moral. Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 325(6):653–658, 1997.

[DWB06] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics Automation Magazine*, 13(2):99–110, June 2006.

[EKS13] Thomas Eiter, Thomas Krennwallner, and Patrik Schneider. Lightweight spatial conjunctive query answering using keywords. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data*, pages 243–258, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[EW99] G. A. Einicke and L. B. White. Robust extended kalman filtering. *IEEE Transactions on Signal Processing*, 47(9):2596–2599, Sep 1999.

[EY15] Moustafa Elhamshary and Moustafa Youssef. SemSense: Automatic construction of semantic indoor floorplans. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, pages 1–11. IEEE, 2015.

[FAVT12] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan. Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing. *IEEE Transactions on Mobile Computing*, 11(12):1983–1993, December 2012.

[Fis36] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[GDKB10] Slawomir Grzonka, Frederic Dijoux, Andreas Karwath, and Wolfram Burgard. Mapping indoor environments based on human activity. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 476–481. IEEE, 2010.

[GFB05] Wenyu Guo, Nick P. Filer, and Stephen K. Barton. 2d indoor mapping and location-sensing using an impulse radio network. In *Ultra-Wideband, 2005. ICU 2005. 2005 IEEE International Conference on*, pages 296–301. IEEE, 2005.

[GP] Christian Prehofer Georgios Pipelidis, Nikolaos Tsiamitros. Method and system for tracking a mobile device. In *WO2020144353, Technical University of Munich*, page 10.01.2020.

[GTG+05] S. Gezici, Zhi Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84, July 2005.

[GZ11] Marcus Goetz and Alexander Zipf. Extending openstreetmap to indoor environments: bringing volunteered geographic information to the next level. *Urban and Regional Data Management: Udms Annual 2011*, pages 47–58, 2011.

[GZ12] M Goetz and A Zipf. Towards crowdsoursing geographic information about indoor spaces; mapping the indoor world. *GIM International*, pages 30–34, 2012.

[GZY+14] Ruipeng Gao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, Kaigui Bian, Tao Wang, and Xiaoming Li. Jigsaw: indoor floor plan reconstruction via mobile crowdsensing. pages 249–260. ACM Press, 2014.

[Hen12] Robin Henniges. Current approches of Wifi Positioning.pdf, 2012.

[HGTH14] Seyed Amir Hoseinitabatabaei, Alexander Gluhak, Rahim Tafazolli, and William Headley. Design, Realization, and Evaluation of uDirect-An Approach for Pervasive Observation of User Facing Direction on Mobile Phones. *IEEE Transactions on Mobile Computing*, 13(9):1981–1994, September 2014.

[HHT+02] Matti Hamalainen, Veikko Hovinen, Raffaello Tesi, Jari HJ Iinatti, and Matti Latva-aho. On the uwb system coexistence with gsm900, umts/wcdma, and gps. *IEEE Journal on Selected areas in Communications*, 20(9):1712–1721, 2002.

[KK14] M. Kourogi and T. Kurata. A method of pedestrian dead reckoning for smartphones using frequency domain analysis on patterns of acceleration and angular velocity. pages 164–168, May 2014.

[KKB14] Clemens Nylandsted Klokmose, Matthias Korn, and Henrik Blunck. WiFi proximity detection in mobile web applications. pages 123–128. ACM Press, 2014.

[KNO+01] Neil E Klepeis, William C Nelson, Wayne R Ott, John P Robinson, Andy M Tsang, Paul Switzer, Joseph V Behar, Stephen C Hern, and William H Engelmann. The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology*, 11(3):231–252, 2001.

[Kue05] Axel Kuepper. *Location-based services: fundamentals and operation*. John Wiley, Chichester, England ; Hoboken, NJ, 2005.

[KYL14a] Joon-Seok Kim, Sung-Jae Yoo, and Ki-Joune Li. Integrating IndoorGML and

CityGML for Indoor Space. In Dieter Pfoser and Ki-Joune Li, editors, *Web and Wireless Geographical Information Systems*, number 8470 in Lecture Notes in Computer Science, pages 184–196. Springer Berlin Heidelberg, May 2014. DOI: 10.1007/978-3-642-55334-9_12.

[KYL14b] Joon-Seok Kim, Sung-Jae Yoo, and Ki-Joune Li. *Integrating IndoorGML and CityGML for Indoor Space*, pages 184–196. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[LCBA08] M. Llombart, M. Ciurana, and F. Barcelo-Arroyo. On the scalability of a novel WLAN positioning system based on time of arrival measurements. In *5th Workshop on Positioning, Navigation and Communication, 2008. WPNC 2008*, pages 15–21, March 2008.

[LCS11] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *011 International Conference on Computer Vision*, pages 2548–2555, Nov 2011.

[LDDP17] Sandra Loch-Dehbi, Youness Dehbi, and Lutz PlÃ¼mer. Estimation of 3d indoor models with constraint propagation and stochastic reasoning in the absence of indoor measurements. *ISPRS International Journal of Geo-Information*, 6(3), 2017.

[LHG13] Binghao Li, B. Harvey, and T. Gallagher. Using barometers to determine the height for indoor positioning. In *International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7, Oct 2013.

[LIT+14] Guangwen Liu, Masayuki Iwai, Yoshito Tobe, Dunstan Matekenya, Khan Muhammad Asif Hossain, Masaki Ito, and Kaoru Sezaki. Beyond horizontal location context: measuring elevation using smartphone's barometer. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 459–468. ACM, 2014.

[LL10] Ki-Joune Li and Jiyeong Lee. Indoor spatial awareness initiative and standard for indoor spatial data. In *Proceedings of IROS 2010 Workshop on Standardization for Service Robot*, volume 18, 2010.

[LV10] U. Lipowezky and I. Vol. Indoor-outdoor detector for mobile phone cameras using gentle boosting. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 31–38, June 2010.

[MF09] Guoqiang Mao and Baris Fidan, editors. *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking*. IGI Global, 2009.

[NB08] Zdenek NEMEC and Pavel BEZOUSEK. The Time Difference of Arrival Estimation of Wi-Fi Signals. *Radioengineering*, 17(4):51, 2008.

[NO20] Sharareh Naghdi and Kyle O'Keefe. Detecting and correcting for human

obstacles in ble trilateration using artificial intelligence. *Sensors*, 20(5):1350, 2020.

[Ope] OpenStreetMap. Josm. https://josm.openstreetmap.de/. 2017-06-20.

[PBKB15] Kushani Perera, Tanusri Bhattacharya, Lars Kulik, and James Bailey. Trajectory inference for mobile devices using connected cell towers. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '15, pages 23:1–23:10, New York, NY, USA, 2015. ACM.

[PFL+95] Scott Pace, G Frost, I Lachow, D Frelinger, D Fossum, DK Wassem, and M Pinto. Gps history chronology and budgets. *The global positioning system, Santa Monica, RAND Corporation*, 1995.

[PFP18a] G. Pipelidis, F. Fraaz, and C. Prehofer. Extracting semantics of indoor places based on context recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 464–467, March 2018.

[PFP18b] Georgios Pipelidis, Frederik Fraaz, and Christian Prehofer. Extracting semantics of indoor places based on context recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 464–467. IEEE, 2018.

[Pip17] Georgios Pipelidis. Opensourced IndoorMapping project. https://github.com/pipelidis/IndoorMapping, June 2017. original-date: 2017-06-12T11:19:45Z.

[PMRI+18] Georgios Pipelidis, Omid Moslehi Rad, Dorota Iwaszczuk, Christian Prehofer, and Urs Hugentobler. Dynamic vertical mapping with crowdsourced smartphone sensor data. *Sensors*, 18(2):480, 2018.

[PPC+12] Jun-geun Park, Ami Patel, Dorothy Curtis, Seth Teller, and Jonathan Ledlie. Online pose classification and walking speed estimation using handheld devices. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 113–122. ACM, 2012.

[PPG17] Georgios Pipelidis, Christian Prehofer, and Ilias Gerostathopoulos. Adaptive bootstrapping for crowdsourced indoor maps. In *Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1: GISTAM,*, pages 284–289. INSTICC, ScitePress, 2017.

[PPG19] Georgios Pipelidis, Christian Prehofer, and Ilias Gerostathopoulos. Bootstrapping the dynamic generation of indoor maps with crowdsourced smartphone sensor data. In Lemonia Ragia, Robert Laurini, and Jorge Gustavo Rocha, editors, *Geographical Information Systems Theory, Applications and Management*, pages 70–84, Cham, 2019. Springer International Publishing.

[PRI+17] G. Pipelidis, O. R. M. Rad, D. Iwaszczuk, C. Prehofer, and U. Hugentobler. A novel approach for dynamic vertical indoor mapping through crowdsourced smartphone sensor data. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Sept 2017.

[PTG+19] G. Pipelidis, N. Tsiamitros, C. Gentner, D. B. Ahmed, and C. Prehofer. A novel lightweight particle filter for indoor localization. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Sep. 2019.

[PTU+19] G. Pipelidis, N. Tsiamitros, E. Ustaoglu, R. Kienzler, P. Nurmi, H. Flores, and C. Prehofer. Cross-device radio map generation via crowdsourcing. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2019.

[RAK14] Patrick Robertson, Michael Angermann, and Bernhard Krach. Method for creating a map relating to location-related data on the probability of future movement of a person, January 7 2014. US Patent 8,626,443.

[RAL] Valérie Renaudin, Muhammad Haris Afzal, and Gérard Lachapelle. Complete triaxis magnetometer calibration in the magnetic domain. *Journal of sensors*, 2010.

[RKSM14] Valentin Radu, Panagiota Katsikouli, Rik Sarkar, and Mahesh K. Marina. A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys '14, pages 280–294, New York, NY, USA, 2014. ACM.

[RM09] Mahnaz Roshanaei and Mina Maleki. Dynamic-KNN: A novel locating method in WLAN based on Angle of Arrival. In *IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, volume 2, pages 722–726, 2009.

[RNBM11] Lenin Ravindranath, Calvin Newport, Hari Balakrishnan, and Samuel Madden. Improving wireless network performance using sensor hints. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 281–294, Berkeley, CA, USA, 2011. USENIX Association.

[ROP+19] Valerie Renaudin, Miguel Ortiz, Johan Perul, Joaquín Torres-Sospedra, Antonio Ramón Jiménez, Antoni Pérez-Navarro, Germán Martín Mendoza-Silva, Fernando Seco, Yael Landau, Revital Marbel, et al. Evaluating indoor positioning systems in a shopping mall: The lessons learned from the ipin 2018 competition. *IEEE Access*, 7:148594–148628, 2019.

[RSL12] Valérie Renaudin, Melania Susi, and Gérard Lachapelle. Step length estimation using handheld inertial sensors. *Sensors*, 12(7):8507–8525, 2012.

[SAW94] Bill N. Schilit, Norman Adams, and Roy Want. Context-aware computing ap-

plications. In *IN PROCEEDINGS OF THE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, pages 85–90. IEEE Computer Society, 1994.

[SRL13] Melania Susi, Valérie Renaudin, and Gérard Lachapelle. Motion mode recognition and step detection algorithms for mobile phone users. *Sensors*, 13(2):1539–1562, 2013.

[TVNB15] A. Thaljaoui, T. Val, N. Nasri, and D. Brulin. Ble localization using rssi measurements and iringla. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 2178–2183, March 2015.

[WCS10] Yik-Chung Wu, Qasim Chaudhari, and Erchin Serpedin. Clock synchronization of wireless sensor networks. *IEEE Signal Processing Magazine*, 28(1):124–138, 2010.

[WSE+12] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 197–210, New York, NY, USA, 2012. ACM.

[XWQ+15] Hao Xia, Xiaogang Wang, Yanyou Qiao, Jun Jian, and Yuanfei Chang. Using multiple barometers to detect the floor location of smart phones with built-in barometric sensors for indoor positioning. *Sensors*, 15(4):7857–7877, 2015.

[ZD02] Manfred Zimmermann and Klaus Dostert. A multipath model for the powerline channel. *IEEE Transactions on communications*, 50(4):553–559, 2002.

[Zha04] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):892–899, July 2004.

[ZSN+13] Sisi Zlatanova, George Sithole, Masafumi Nakagawa, Qing Zhu, and A Gist. Problems in indoor mapping and modelling. volume XL-4/W4, 12 2013.

[ZXQ+20] Wei Zhao, Liangjie Xu, Bozhao Qi, Jia Hu, Teng Wang, and Troy Runge. Vivid: Augmenting vision-based indoor navigation system with edge computing. *IEEE Access*, 8:42909–42923, 2020.

[ZZL+12] Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, Mo Li, and Guobin Shen. Iodetector: A generic service for indoor outdoor detection. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 113–126, New York, NY, USA, 2012. ACM.

A Glossary

AC Alternative Current
AoA Angle of Arrival
AP Access Point
BCSK Binary Color Shift Keying
BIM Building Information Modeling
BLE Bluetooth Low Energy
CV Computer Vision
GIS Geographic Information System
GML Geographic Markup Language
GNSS Global Navigation Satellite System
GPS Global Position System
GSM Global System for Mobile Communications
IEEE Institute of Electrical and Electronics Engineers
IMU Inertial Motion Unit
InLocation Indoor Location
IR Infra Red **LAN** Local Area Network
LBS Location Based System
LED Light-emitting diode
LOD Level Of Detail
LS Location System
LTE Long-Term Evolution
OITransition Outdoor to Indoor Transition
ORD Optical Rotatory Dispersion
OS Operating System
OSM Open Street Maps
PDR Pedestrian Dead Reckoning
RF Radio Frequency
RSS Received Signal Strength
RSSI Received Signal Strength Intensity
RTT Round Trip Time
SLAM Simultaneous Localization And Mapping
TDoA Time Difference of Arrival
ToA Time of Arrival
VGI Volunteered Geographic Information
VLP Visual Light Positioning
UWB Ultra Wide Band

Crowdsource obtain (information or input into a particular task or project) by enlisting the services of a large number of people, either paid or unpaid, typically via the Internet. "she crowdsourced advice on album art and even posted an early version of the song so fans could vote for their favorite chorus"
Geomagnetism the magnetic properties of the earth.
LiDAR is a surveying method that measures the distance to a target by illuminating the target with laser light and measuring the reflected light with a sensor.
smartphone a mobile phone that performs many of the functions of a computer, typically having a touchscreen interface, Internet access, and an operating system capable of running downloaded apps.