Exploiting embedded sensing in outdoor localization : a data-driven perspective

Wang, Jin

2019

Wang, J. (2019). Exploiting embedded sensing in outdoor localization : a data-driven perspective. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/106002>

<https://doi.org/10.32657/10220/47852>

EXPLOITING EMBEDDED SENSING IN OUTDOOR LOCALIZATION: A DATA-DRIVEN PERSPECTIVE

A thesis submitted to
School of Computer Science and Engineering
Nanyang Technological University

by

Wang Jin

in fulfilment of the requirements
for the degree of Doctor of Philosophy

March 2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

2019.03.01

Date

A handwritten signature in black ink, appearing to be 'Wang Jin', written on a light gray background.

Wang Jin

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

2019.03.01

Date

Luo Jun

Authorship Attribution Statement

This thesis contains material from 2 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as **Wang, J., Tan, N., Luo, J., & Pan, S. J. (2017, May). WOLoc: WiFi-only outdoor localization using crowdsensed hotspot labels. In IEEE INFOCOM 2017-IEEE Conference on Computer Communications (pp. 1-9). IEEE.**

The contributions of the co-authors are as follows:

- Prof. Luo and Prof. Pan provided the initial project direction.
- I prepared the manuscript drafts. Prof. Luo edited the manuscript. All authors contributed to revision.
- I designed and refined the algorithm and system, and implemented all the system codes. Mr. Tan contributed in collecting data and conducting onsite experiments.

Chapter 4 is published as **Wang, J., Luo, J., Pan, S. J., & Sun, A. (2018). Learning-Based Outdoor Localization Exploiting Crowd-Labeled WiFi Hotspots. IEEE Transactions on Mobile Computing.**

The contributions of the co-authors are as follows:

- Prof. Sun and Prof. Luo provided the initial project direction.
- I prepared the manuscript drafts. Prof. Luo and Prof. Pan edited the manuscript. All authors contributed to revision.

- I designed the algorithm and experiments, implemented all the relevant codes and conducted the experiments.

2019.03.01

Date

A handwritten signature in black ink, appearing to be 'Wang Jin', written on a light gray rectangular background.

Wang Jin

Acknowledgments

First of all, I would like to express my sincere gratitude towards my supervisor Prof. Jun Luo for his patient guidance, encouragement, immense knowledge and advice. Under his elaborative supervision, I have not only learned the state-of-the-art mobile sensing techniques, but also acquired a conscientious research attitude through critical thinking. The knowledge and methodology I learnt are invaluable to my research, and even to my life. I feel very grateful to my co-supervisor Prof. Ying He for his continuous support and patient guidance. Many thanks for sharing his extensive knowledge and experience in research. His advice on both research and my career have been invaluable.

I would like to express my deep gratitude to all my coauthors: Prof Sinno Jialin Pan, Prof Aixin Sun, Dr. Hui Liang, Dr. Qian Sun, Dr. Chen Wang, Qingmei Zhao, Nicolas Tan. I appreciate all their efforts on the research work and I enjoy the experience of the collaborations with them all the time. Especially for Prof Pan, I appreciate a lot for all his insightful discussions with me, as well as his kind guidance and encouragement for all my thesis works.

I would also like to appreciate all my fellow friends and colleagues from my company and my school lab for their friendly support and help. They include Haodan Yang, Wenya Wang, Xin Zheng, Qing Guo, Bingshui Da, Peng Ni, Wei Cui, Ruidan He, Jianshu Li, Kun Ouyang, Jianglei Han, Yanbing Yang, Shuya Ding, Zheng Fang, Dr. Xiaoqi Yan, Dr. Haiyun Lu, Dr. Yangyang Xiang, Dr. Daniel Dahlmeier, Steve Lee, and Pheobe Poon. I am truly thankful for all the convenient space, excellent facilities and financial support provided by SAP Innovation Center and Computer Networks and Communications Lab. I also want to express a big thank you to my friend, Haoyuan Xing, for his encouragement and his greeting through postcards during the dark moments along the journey.

I would like to give my special thanks to my friends Boyang Che and Yue Wang for our 10-year precious friendship. With their encouragement and constant companion, my Ph.D life has been much more delightful and full of hope.

I would like to express my sincere thanks to my friend, Shiyan Xu. Without his enduring support, I would not be able to finish my Ph.D career.

Last but not least, I would like to sincerely thank my father and mother, for their support, encouragement and love all the time. Greatest thanks for all the selfless sacrifices that they have made for me throughout my life.

Abstract

Although research community have put lots of effort in improving localization in indoor environment through various of sensing techniques, outdoor localization still heavily relies on GPS solely. Despite GPS performs successfully in perfect outdoor scenario (e.g., large open air zone), many other outdoor scenarios in metropolitan area make GPS fail to achieve a satisfactory performance due to signal blocking or multi-path effect. Therefore, to improve localization performance in metropolitan area, an integrated outdoor localization solution is desired, which takes GPS as one part and other sources as supplementary. For designing such an integrated solution, we need to consider two questions: (1) what are the proper supplementary sources? (2) when are the proper moments to trigger a switch between GPS and other sources?

To find a proper complementary source for outdoor localization in urban area, we have to consider the potential large-scale infrastructure and human effort in collecting and labelling data. The ever-expanding scale of WiFi deployments in urban area has made accurate GPS-free outdoor localization become possible. Additionally, there are many existing crowdsensing database available that collect WiFi signal of users' surrounding hotspots and users' locations.

Leveraging the existing infrastructure and data source, we propose WOLoc (WiFi-only Outdoor Localization) as an outdoor localization solution using only WiFi hotspots labelled by crowdsensing. On one hand, we do not take these labels as fingerprints as it is almost impossible to extend indoor localization mechanisms by fingerprinting urban areas. On the other hand, we avoid the over-simplified local synthesis methods (e.g., centroid) that significantly lose the information contained in the labels. Instead, WOLoc adopts a semi-supervised manifold learning approach that accommodates all the labeled and unlabeled data for a given area, and the output concerning the unlabeled part will

become the estimated locations for both unknown users and unknown WiFi hotspots. Empirical results show that WOLoc can achieve a meter-level localization accuracy in many urban areas.

Despite the success of WOLoc in many cases, due to the lack of labelled data for hotspot location, WOLoc may suffer from an imbalanced alignment between the user manifold and hotspot manifold. Such misalignment may lead to extreme cases which are meaningless in some scenarios. To reduce such failures, we apply text mining techniques to analyze the SSIDs of hotspots, and develop a pipeline to incorporate SSID analysis in both pre-processing and hotspot labelling process of WOLoc. We make use of existing crowdsourced location services databases (e.g. Foursquare, Google Places) to give more accurate hotspot location labels. Empirical results show that the hotspot SSID analysis pipeline has successfully reduced extreme failure cases for WOLoc and lead it to a more consistent performance

To find a proper trigger between GPS and WiFi, we need to understand how GPS performs in different urban outdoor scenarios. The latest developments in Android have granted us access to raw GNSS measurements, which contain far more information than commonly derived GPS location indicators. We explore these newly available measurements in order to better characterize the diversified urban scenarios and propose a deep learning model to identify representations for respective location contexts. With the deep profiling, we offer a more fine-grained semantic classification than binary indoor-outdoor detection and derive a GPS error indicator more meaningful than that provided by Google Maps. These results are all corroborated by our extensive data collection and trace-driven evaluations.

To sum up, in this dissertation, we propose a series of data-driven approaches that aim for an integrated outdoor localization by exploiting WiFi RSSI data, WiFi SSID labels and GNSS raw measurements. We choose WiFi as supplementary sources and design a full version of system based on manifold learning technique to achieve WiFi-only outdoor localization in urban areas. We further develop the SSID-based text-mining techniques to improve both accuracy and robustness. We finally propose a GNSS profiling technique to evaluate GPS performance and differentiate different location context, which can be used as a trigger between GPS and WiFi in an integrated solution.

Contents

Acknowledgements	vii
Abstract	ix
List of Figures	xv
List of Tables	xviii
1 Introduction	1
1.1 Research Scope	1
1.2 Research Challenges	4
1.2.1 WiFi-based Outdoor Localization	4
1.2.2 GPS-based Location Context Profiling	6
1.3 Research Approaches	9
1.3.1 RSSI-based Localization Approach	9
1.3.2 SSID-based Localization Improvement	10
1.3.3 GPS-based Location Context Profiling	10
1.4 Research Contributions	11
1.5 Dissertation Organization	13
2 Literature Review	15
2.1 GPS-based Localization	15
2.1.1 GPS Fundamentals	15
2.1.2 GPS Enhancement in Accuracy	16
2.1.3 GPS Enhancement in Efficiency	18
2.2 GPS-free Outdoor Localization	18
2.2.1 Range-based Localization Methods	19
2.2.2 Range-Free Localization Methods	21

2.2.3	Current Practices of Outdoor GPS-free Localization	22
2.3	SSID-based Analysis	24
2.4	Indoor-Outdoor Detection	25
2.5	Mining Mobile Sensing Data	26
2.6	Summary	27
3	WOLoc: WiFi-only Outdoor Localization	29
3.1	Manifold Learning in WiFi-based Localization	29
3.2	WOLoc: A Manifold Perspective in Localization	32
3.2.1	Pre-Processing of Crowdsensed Data	32
3.2.2	Problem Formulation	34
3.2.3	Manifold Construction	34
3.2.4	Offline Learning for Location Estimations	35
3.2.5	Online Location Query Processing	36
3.3	System Evaluation	37
3.3.1	Experiment Setting	37
3.3.2	Statistics on Hotspots	39
3.3.3	Time Efficiency of WOLoc Localization	41
3.3.4	Accuracy of User Localization	43
3.3.5	Comparison with Other Systems	44
3.4	Summary	46
4	SSID-based Text Analysis for Manifold-based Localization	49
4.1	Limitation of WOLoc with User Location Indicators Only	49
4.2	SSID and Location Service Database as Additional Location Indicators .	50
4.3	SSID-based Text Analysis	53
4.3.1	Problem Formulation	53
4.3.2	Pre-processing on SSID Text	54
4.3.3	POI Database Query	55
4.3.4	Hotspot-POI Matching	55
4.3.5	SSID-WOLoc: Extended Solution with SSID Analysis	57
4.4	Experiments and Evaluations	61

4.4.1	Experiment Setup	61
4.4.2	Performance Analysis on Individual Component of SSID-WOLoc	62
4.4.3	Sensitivity Analysis	67
4.5	Summary	71
5	A Deep Profiling of GNSS-based Location Contexts	73
5.1	From Understanding GNSS Data to Problem Definition	74
5.1.1	Case Study of GPS-based Localization Performance	74
5.1.2	Problem Definition	76
5.2	GNSS-based Location Context Profiling	77
5.2.1	Feature Engineering Pipeline	77
5.2.2	Location Context Autoencoder	79
5.2.3	Applications of Location Context Profiling	82
5.3	Evaluation	85
5.3.1	System Implementation and Data Preparations	85
5.3.2	Autoencoder Model Training	86
5.3.3	Localization Error Estimator	88
5.3.4	Location Contexts Semantic Analysis	90
5.3.5	Energy Consumption	91
5.4	Summary	92
6	Conclusions and Future Work	95
6.1	Summary of Contributions	95
6.2	Future Work	98
	List of Publications	101
	References	103

List of Figures

2.1	A two-stage localization approach: Hotspots Localization (Left) and User Localization (Right). We mark known locations in black and estimated locations in red. Hotspots Localization aims to locate hotspots (AP1 to AP5) given several user locations (U1 to U4) along with corresponding hotspots RSSIs. User Localization aims to estimate a new user's (User X) location based on previously estimated locations and their respective RSSIs. [Best viewed in color.]	23
3.1	Comparing localization based on Weighted Centroid Method (Left) and Manifold-based Learning (Right). We consider a target hotspot of which the true location is shown as the black star. Black dots show locations of users that discover it. Blue stars are its neighbouring hotspots in the constructed manifold. Red star indicated its estimated location in each method based on which the red a potential propagation range is shown as the red circle. Phone icon indicates a new user location, which may just fall in the estimated propagation range and can be better predicted in the manifold-based learning scenario. [Best viewed in color.]	31
3.2	WOLoc system architecture.	32
3.3	Maps provided by Google Map for all areas concerned in our experiments. (a) Downtown. (b) Campus. (c) Hybrid Residential Area. (d) Residential Blocks. (e) Community Area. (f) Downtown Entertainment Area.	38
3.4	Hotspots density for all areas in our experiments.	40
3.5	Query processing time using all hotspots in a query and their neighbouring hotspots.	41
3.6	Error statistics as a function of the number of candidate hotspots.	42

3.7	Processing time using only hotspots in a query.	43
3.8	Error in meters for estimating user location using WOLoc.	44
3.9	Median error comparisons between WOLoc, OpenBMap, Skyhook and Google for all 6 areas.	45
3.10	Location error distributions for 3 commercial systems: (a) to (e) for OpenBMap, (f) to (j) for Skyhook, and (k) to (o) for Google.	48
4.1	SSID-WOLoc system architecture.	57
4.2	Accuracy comparison between SSID-WOLoc without pre-processing and SSID-WOLoc with pre-processing in terms of median error and mean error.	63
4.3	Accuracy comparison between “Cut-and-Pin” and “SSID text analysis” in terms of median error and mean error.	64
4.4	CDF of error for “Cut-and-Pin” and “SSID text analysis”.	65
4.5	Example cases on how SSID text analysis improves localization error for extreme cases. Yellow dots: the estimated location of users; Red dots: fixed hotspot locations; Green dots: estimated hotspot locations. [Best viewed in color.]	66
4.6	Comparison between “Cut-and-Pin” and “SSID text analysis” in processing time of online queries.	67
4.7	Localization accuracy over different testing ratio.	68
4.8	Accuracy comparison between WOLoc without pre-processing and SSID-WOLoc with pre-processing in terms of median error and mean error.	69
4.9	Performance analysis for different levels of hotspots label granularity.	70
5.1	Preliminary studies on GNSS data. (a) outdoor long wide street, (b) inside glass building and on void deck, (c) sheltered street, (d) high-rise enclosed area, (e) building surrounded street, and (f) the same as (e) but at a different time. We put ground-truth and GPS traces on Google Maps snapshots in the first row, satellite distribution patterns (with black dot indicating the position of receiver) in the second row, and GPS error distributions in the last row. [Best viewed in colors.]	75
5.2	Autoencoder structure. [Best viewed in colors.]	80

5.3	Snapshots and Corresponding representations. [Best viewed in colors.] . .	81
5.4	Context semantic analysis. [Best viewed in colors.]	83
5.5	Autoencoder parameter vs reconstruction error.	87
5.6	Deviation from true localization error.	88
5.7	Trace samples with error estimation. [Best viewed in colors.]	89
5.8	Performance of localization accuracy indicator.	90
5.9	Clustering performance with various threshold and k.	91
5.10	Trace samples with context label results. [Best viewed in colors.]	92

List of Tables

1.1	Raw GNSS data structure.	7
3.1	Hotspots density and number of hotspots per record	39
4.1	Hotspot SSID examples for public POIs.	52
4.2	Sample results of tokenization.	52
5.1	Battery consumption comparison for different sensor set.	92

Chapter 1

Introduction

1.1 Research Scope

Localization has been a widely-studied topic for decades due to the potential wide range of services it can provide by leveraging Internet of Things (IoT), and ubiquitous connectivity. Existing localization/navigation systems are clearly separated into two categories: outdoor [44, 35] versus indoor [3, 100]. Although indoor systems leverage pervasively available signal sources, such as WiFi [3, 100, 99], visible light [47, 105], and geomagnetism [94, 103], outdoor systems are mostly dominated by Global Positioning System (referred as GPS hereafter) only given its excellent localization performance in perfect outdoor scenarios (e.g. open-air zone like large car park outside shopping mall). Nevertheless, the landscape of outdoor (user) localization is shifting due to two reasons. On one hand, embedded GPS sensors (in smartphones, for example) have been criticized for its high energy consumption and its cold start in getting a location fix in many applications [108, 73, 11]. On the other hand, the frequent loss of signal in “urban canyon” [87, 86] may make GPS fail to localize users in metropolitan areas. Therefore, an integrated solution, which makes GPS as one part and other sources as supplementary, is desired for outdoor localization in metropolitan areas.

In order to achieve such an integrated solution, it is natural to consider two things. The first thing is to identify a proper supplementary source and design a solution based

on the source. The source should be practical to use and scalable for outdoor scenarios in metropolitan areas, and the solution should be able to overcome the instability and inaccuracy resulted from GPS-based localization. In addition, considering the large geographical scale and the data-driven approach, the labelling and collection of such data source should not be labor-intensive. The second thing is to design a proper trigger between GPS and other sources. The trigger should be highly correlated to the performance of GPS and able to differentiate various location contexts to decide when to incorporate or switch to the supplementary data sources.

To identify such a supplementary source, it is natural to consider the sources which have been used in existing indoor localization techniques. Whereas many location indicators, namely general RF signal [91, 86, 12], light [2], sound [85], and magnetic field [104], can be explored indoors, they either lose their location discriminability (e.g., light, sound, and magnetic field) or offer very low localization accuracy due to the sparse deployment of signal sources (Cellular and FM). As a result, the pervasively available WiFi infrastructure in metropolitan areas appears to be a promising choice for us to explore further.

While the majority of the research efforts are still dwelling in indoor localization, quite a few industrial practices have already started to provide GPS-free outdoor localization services based on WiFi infrastructure [96, 65, 79, 29, 21]. These services are backed up by one fact: since one WiFi scan may discover up to hundreds of WiFi hotspots in a common metropolitan area, crowdsensing by a large number of smartphone users has already labeled those hotspots without the need for wardriving. Consequently, even a small database in such a system (e.g., OpenBMap [65]) may have thousands of WiFi hotspots recorded for one metropolitan area, with each one getting several hundreds of labels. If we can properly exploit such “big data”, GPS-free localization in metropolitan areas can be made very accurate.

Although we have identified that the ever-expanding scale of WiFi deployments in metropolitan areas would make it possible to overcome the potential failure of GPS-only

outdoor localization, in order to incorporate such methods to achieve better localization performance, we still need to identify the cases that GPS-only outdoor localization generally fails. The question about how much trust we may put on each source (in particular to use GPS or not) for localization remains open.

To identify such a trigger between GPS and other sources, previous studies has made a clear-cut separation between indoor and outdoor, so as to trigger timely switches between GPS-based localization for outdoor and WiFi-based localization for indoor generally; this has provoked several research proposals on detecting the transition from outdoor to indoor or vice versa (a.k.a. *IO-detection*) [108, 73, 11].

However, though perfect indoor scenarios (wrapped by concrete walls and roof entirely) do exist, GPS signal can grow very weak while walking on downtown areas, while users inside a room with big windows may still obtain very good GPS location indicators. Apparently, the outcome of IO-detection can be insufficient given the diversified urban environments, so we should better look into a holistic profiling (i.e., identifying semantic representations) of various location contexts. In other words, there exist several location contexts lying between perfect outdoor areas (e.g., large parking areas out of shopping malls) and perfect indoor spaces (e.g., windowless research laboratories surrounded by thick walls). An adequate profiling of these location contexts may eventually allow us to fuse various signal sources for achieving ideal urban localization.

In summary, in this dissertation, we aim to explore an integrated solution for outdoor localization in metropolitan areas by firstly developing novel WiFi-based outdoor localization approaches as the supplementary source beyond GPS and then developing an adequate profiling of GPS-based localization performance as a potential trigger between GPS and WiFi.

1.2 Research Challenges

According to the research scope illustrated in Section 1.1, two problems are defined and investigated in this dissertation, namely, WiFi-based outdoor localization, and GPS-based location context profiling. Next, we will respectively demonstrate the research challenges of the two problems.

1.2.1 WiFi-based Outdoor Localization

Given the large geographical scale of metropolitan areas, existing methods for indoor localization may not be able to directly applied in outdoor scenarios. For methods that require distance measurements between transmitters and receives (ToF/ToA/TDoA) [55, 92, 57], there have to be non-RF signal sources involved so that the time can last long enough to be measurable. Such signal sources are quite unstable and easily interfered by noises, which makes them not suitable for outdoor scenarios. Methods such as dead reckoning [50] are useful to track a user in a small scale space, but will heavily suffer from error accumulation for a large space in metropolitan areas. WiFi fingerprinting methods (e.g., [4]) have been proven effective in indoor localization, and some academic proposals have tried to migrate them to a metropolitan area (e.g., [14, 89]). However, fingerprinting such a huge area through wardriving is extremely difficult (if not impossible), as it requires a huge amount of human input to carefully design and execute the wardriving process. Some other academic proposals (e.g., [87]) along with most industrial practices take a simpler approach that involves a WiFi hotspot localization phase using the labels and a user localization phase based on the estimated hotspot locations. Whereas this method avoids the weakness of the fingerprinting method and also delivers the WiFi hotspot locations as a byproduct, it cannot achieve a good localization accuracy because the synthesizing methods in both phases (e.g., centroid [87, 65]) are oversimplified and they process data only in a localized (in topological sense) manner, so that they i) may not

handle the label errors well enough to avoid error accumulation across the two phases, and ii) can cause a significant information loss to hamper the crowdsensed labels from fully contributing to the user localization.

1.2.1.1 RSSI-based Localization Approach

As all the other methods either require complicated deployment or oversimplify the process, the error-prone received signal strength indicator (referred as RSSI hereafter) model-based ranging [15] seems to be a reasonable solution. However, it is impractical to estimate a certain propagation model for all hotspots in outdoor scenarios. As a result, there will be potentially large errors introduced in ranging based on RSSI model. Nevertheless, existing approaches handle these potentially very large errors through a “brute-force” dimension reduction conducted by minimizing the mean errors between the error-twisted high dimensional structure and its 2D projection [15, 55]. We need to design an approach which can fully utilize existing crowdsensed labels (the location of a user and the RSSI values it receives from its surrounding hotspots) and meanwhile can reasonably reduce the potential error introduced by RSSI-based ranging.

1.2.1.2 SSID-enabled Localization Improvement

Additionally, existing WiFi crowdsensed database is based on user locations, which means the database is contributed by individual users and only location indicators of these users are available. Therefore, location indicators for hotspots are not directly available. Besides the basic service sets identifiers (BSSIDs), the service set identifiers (referred as SSID hereafter) of hotspots which serve as “network names” are also recorded in many crowdsensed databases. We observe that SSIDs have certain patterns for personal hotspots and public hotspots, so that they can be used to remove unuseful hotspots during the localization process.

Moreover, with the increasing popularity of crowdsourced social venue check-in databases (e.g. *FourSquare*, *Yelp*) and industry-maintained venue databases (e.g. *Google Places*, *Baidu Map*), more information regarding public places in metropolitan areas are publicly available, including name and geo-location. Since a part of hotspots in urban areas is from public areas for food, leisure or services, it is highly possible that the places that maintain the hotspots have been discovered and socially checked-in by mobile users to crowdsource venue databases. Another part of hotspots is from areas for companies and agencies, which are mostly maintained in industry-maintained venue database. By analyzing the text information in the SSIDs of collected hotspots, venue information is revealed to facilitate the labeling process for parts of the hotspots whose locations are unknown. Therefore, how to utilize the information hidden in SSIDs to improve the localization process become another interesting topic for us to explore.

1.2.2 GPS-based Location Context Profiling

As illustrated in Section 1.1, an adequate profiling of different location context is critical and necessary to fuse different data sources for better localization. In the past, such an analysis is nearly impossible for users with commercial devices (e.g., smartphones), because users can only access the GPS location indicators distilled through sophisticated filtering processes.

Fortunately, new devices [24] supported by Android 7 open up the access to raw GNSS measurements. (Global Navigation Satellite System (GNSS) is a generic term referring to satellite navigation systems such as GPS and Galileo. However, here raw GNSS measurements denote the standard data format shared by all such systems, though we actually obtain only GPS-GNSS measurements.) Although the GPS location indicators are derived from these GNSS measurements, substantial information lost may have been caused by the significant dimension reduction resulting from convention signal processing

Table 1.1: Raw GNSS data structure.

Fields Name	Description
Binary Fields	
STATE_SYMBOL_SYNC	whether tracking is synchronized at symbol level
STATE_MSEC_ambiguous	whether the measurement's tracking state contains millisecond ambiguity.
STATE_TOW_DECODED	whether time-of-week is known for tracking
STATE_SUBFRAME_SYNC	whether the measurement's tracking is synchronized at subframe level.
STATE_BIT_SYNC	whether tracking synchronized is synchronized at bit level
MultipathIndicator	has multipath been detected?
HardwareClockDiscontinuity	whether the hardware clock discontinues from last measurement.
...	...
Number Fields	
ReceivedSvTimeNanos	the received GNSS satellite time in ns.
BiasUncertaintyNanos	the clock's bias uncertainty in ns.
ReceivedSvTimeUncertaintyNanos	the error estimate (1-sigma) for the received GNSS time, in nanoseconds.
PseudorangeRateUncertainty...	rate uncertainty (1-sigma) in m/s.
Cn0DbHz	the Carrier-to-noise density in dB-Hz.
...	...

pipelines. Therefore, it is reasonable to believe that effectively mining a large volume of these raw measurements could end up with a much more holistic profiling on urban environments than the binary IO-detection.

However, analysing such high volume of GNSS data is not straight-forward. There are generally 2 main challenges:

- **Beyond dimensionality:** The involved data structure is fairly complicated, as shown in Table 1.1 for one satellite. One may deem it as a high dimensional vector and simply stack 32 of them (for all GPS satellites) into a matrix, yet, such an input data structure is inefficient: many fields are highly correlated and such redundancy needs to be sorted out before proceeding to further analysis. In other words, a more effective data representation is hence necessary to efficiently capture location contexts to reduce the future computation complexity.
- **Inherent misalignment:** No matter what data structure is used, a proper indexing is a key to align relevant features. Whereas conventional mobile sensing has the sensor IDs or sensor axes as a natural indexing, using satellites IDs for the same purpose can severely misalign the feature. It is because the satellites are moving all the time, thus the relative position among them and to the receiver varies from time to time. There is no fixed relationship between pairs of these satellites. The same location may witness very different satellite coverage at different points in time, while even the same pattern of satellites coverage may result from a completely different set of satellites. As a result, satellite IDs are not suitable for indexing, and we need a more suitable mechanism to pack the measurements from different satellites.

1.3 Research Approaches

1.3.1 RSSI-based Localization Approach

In order to fully exert the strength of WiFi-based localization outdoors, we firstly propose an integrated solution, namely WOLoc (WiFi-based Outdoor Localization), to better utilize the crowdsensed WiFi labels for improving the localization accuracy.

Equipped with a large amount of label data, WOLoc takes a holistic view on all such data collected within a metropolitan area (or a sub-area) and it processes the label based on semi-supervised manifold-learning techniques. The rationale behind our design is the following: assuming all labels are perfect (with each label produced by a mobile device δ for a hotspot Θ containing a tuple of {location of δ , RSSI from Θ to δ }), the locations of all mobile devices and hotspots should lie on a low dimensional Euclidean space (normally 2D or at most 3D). Although imperfect labels (in terms of both location and RSSI) may “bend” the original space into a much higher dimension, it is highly possible that those locations still lie on some manifold structure of low dimension [68]. Therefore, our design of WOLoc aims to discover this manifold structure so as to recover the true locations of both users and WiFi hotspots.

Specifically, we design an end-to-end solution from data processing to query processing, which includes mainly 4 parts: (1) preprocessing of hotspots labels from crowdsensed database, (2) generating global manifold of hotspot and users, (3) offline learning for location estimation on existing label data, and (4) online user location query processing. Such design not only fully utilize existing crowdsensed labels to achieve better outdoor localization performance, but also consider the computational complexity and load at different stages to enable online localization service in the future.

1.3.2 SSID-based Localization Improvement

Despite the success of WOLoc in many cases, failure cases sometimes occur due to a suboptimal alignment between user manifold and hotspot manifold. That is due to the imbalance labelled data in the crowdsensed database. As discussed earlier, the only location indicators available in crowdsensed database are user locations, thus the unknown user location can be well estimated by known user locations from the database given the user manifold structure generated from the RSSI values. However, considering the manifold constructed by hotspots, there are no location indicators, which is labelled data in the context of manifold learning, and the range scale within hotspots and between users and hotspots will not have a reference. As a result, larger errors may be introduced for areas with fewer labelled data, and make some cases end with meaningless locations. Besides, in the preprocessing stage of WOLoc, although we have introduced mechanism such as DBSCAN to remove most of mobile hotspots, certain mobile hotspots which are static during crowdsensing but essentially mobile cannot be removed by analysing the RSSI only.

Therefore, we extend existing WOLoc to SSID-WOLOc which introduces an SSID-based text analysis mechanism to facilitate the preprocessing and hotspot labelling process. By analysing certain pattern appearing in the SSID, we are able to identify some mobile hotspots in the pre-processing stage and get rid of them in the manifold construction. We also create a comprehensive matching scheme to match certain hotspots with popular venues in publicly available databases by considering the SSID similarity, RSSI pattern and venue source credibility.

1.3.3 GPS-based Location Context Profiling

To understand how GPS performs in different location contexts, we conduct a thorough GNSS data collection in a major metropolitan city, targeting at representative scenarios

such as sheltered walkways, urban canyons, buildings with an open ground level, and so on, and conduct an analysis on the performance of GPS in different location contexts. Preliminary results show that GPS performance is not only determined by location structure itself which directly affect the number of satellites visible, but also affected by the satellites pattern and corresponding state of satellites. Therefore, we aim to process these data using effective packing mechanism and learning strategy, so as to distill latent representations out of these useful features from raw GNSS measurement data.

Specifically, we make satellite location as the index to pack raw measurement data from different satellites, and transfer useful features into an image set. Then we develop an autoencoder to learn a latent representation of the image set, which can best represent the original features. We further develop two applications based on these representations. On one hand, are used to construct semantic profiles for characterizing various urban location contexts, and on the other than, they can be used to provide a better accuracy indicator than, for example, the “blue disk” offered by Google Maps. Both applications can help in the fusion of various signal for ideal outdoor localization in the future.

1.4 Research Contributions

Our major contributions in this dissertation are summarized as follows:

- We define and investigate two main directions in research for an integrated outdoor localization solution which aim at improving accuracy and robustness of outdoor localization performance while reducing the energy consumption of embedded sensing system. They are respectively switching to WiFi signal as supplementary sources to solve the potential failure of GPS and profiling GPS performance in different location contexts to trigger such switches.

- To use WiFi for outdoor localization and make it suitable for the large scale of the metropolitan area, we develop a unified framework of WiFi-based outdoor localization, namely WOLoc, which utilize existing crowdsensed databases. We specifically design a manifold-learning scheme to holistically synthesize all the labels belonging to a certain metropolitan area so as to locate both user (with unknown locations) and all WiFi hotspots. We conduct extensive experiments in several metropolitan areas to validate the effectiveness of our WOLoc system. The efficiency of WOLoc also shows its potential as an online localization service.
- To further improve preprocessing stage of WOLoc system and solve the problem of potential suboptimal alignment between user manifold and hotspot manifold, we introduce SSID-WOLOc which incorporates SSID text analysis into preprocessing stage and hotspot labelling process. We design a unified text analysis pipeline to retrieve venue information from hotspot SSID and query venue-related database for positioning part of the unlabelled hotspots in the manifold. Empirical results show that SSID-WOLOc outperforms WOLoc in most cases and successfully reduce the number of failure cases. To the best of our knowledge, we are the first to consider SSID analysis into the process of user localization.
- To understand GPS performance under different location contexts, we, for the first time, propose to profile location contexts via mining raw GNSS measurements. We innovate in the feature engineering pipeline to handle both high-dimensionality and misalignment and apply a deep autoencoder to distill compressed representations out of the GNSS data sets. We further develop two preliminary applications for the profiling and they deliver semantic representations and better location error estimations respectively. Extensive data collection and experiments are performed to validate our proposals.

1.5 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, we provide an overview of previous studies related to our research problems. Chapter 3 presents our approach to use WiFi signal for outdoor localization and introduce WOLoc as the unified solution. To further improve the performance of WOLoc in term of robustness and accuracy, Chapter 4 introduces SSID-based text analysis mechanism and demonstrated the SSID-WOLOc with its performance evaluation. In Chapter 5, we demonstrate the proposed compressed representation on GNSS data to profile different location contexts, together with two applications based on the designed representation. Finally, Chapter 6 concludes this dissertation and points out several promising directions for future work.

Chapter 2

Literature Review

In this chapter, we provide an overview of state-of-the-art works that are related to our study based on the two problems and our approaches defined in Chapter 1: localization problem, location context profiling problem. For the localization problem, we firstly make a brief introduction about localization based on GPS and relevant works that improve GPS performance. Then we move on to GPS-free localization methods, including academic proposals and current industry practices. We lastly survey literature that utilizes SSID information for data mining. We discuss the strength and limitations of these work in the light of localization performance in outdoor scenarios. For location context profiling problem, we survey literature relevant to indoor-outdoor detection and existing literature that utilize data mining on mobile sensing data. This motivates us to consider mining GNSS data while suggesting a few major challenges in achieving so.

2.1 GPS-based Localization

2.1.1 GPS Fundamentals

As the earliest and most representative global navigation satellite system (GNSS), Global Positioning System (GPS) owned by the United States government has been in operation for more than 20 years. GPS provides geolocation and time information to a GPS receiver

(a user in our context) anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.[44]

A users position is estimated by PVT solution (position, velocity and time) using the distance measurements (namely *pseudoranges*) between the users receiver antenna and the position of at least four satellites. To determine the pseudoranges, all GPS signals are modulated with a specific Pseudo-Random Noise (PRN) code, and such PRN uniquely identifies each satellite. The receiver continuously compares and aligns a local copy of the PRN code with the received satellite signal to determine the particular satellites source. The measured delay of the received PRN code is equal to the transmission time if the transmitter and receiver clocks are perfectly synchronised. Given one satellite position, the receiver position lies somewhere on a sphere around the satellite with a radius equal to the range. If the range measurements of three satellites are available and perfectly measured, the three spheres should intersect at two points of which only one point will lie on the surface of the earth and is the true position of the receiver. However, due to the impact of the receiver clock, the three spheres do not intersect at a common point, thus additional satellites (at least four visible satellites totally) are required to determine the user location[61].

2.1.2 GPS Enhancement in Accuracy

Primary GPS error sources are satellite-based (orbits and clock), atmosphere (ionospheric and tropospheric delays), noise and local effects (including multipath)[41]. There have been many research proposals that focus on reducing these errors. Differential GPS (DGPS)[75] uses multiple receivers to increase GPS accuracy. In DGPS, mobile receivers can locate themselves with higher accuracy by correcting their received satellite measurements based on corrections sent out by one or more static base stations whose own positions and clock biases have been well-calibrated to a high degree of accuracy.

Although DGPS can provide absolute instead of relative coordinates, it requires the preliminary setup of expensive, stationary base stations which precludes it from being used in everyday mobile system deployments.

Another class of methods to increase localization accuracy is called Real Time Kinematic (RTK) satellite navigation [101, 84]. RTK determine the location of a receiver using carrier phase measurements with respect to a dedicated sensing station. The sensing station, of which the location is pre-defined and well-known, transmits its own position and raw carrier phase measurements to surrounding receivers. Participating receivers can calculate their coordinates relative to a base receiver with a high degree of accuracy. The main drawback of RTK is that carrier phase measurements requires knowledge of an unknown (but constant) number of carrier signal wavelengths between a satellite and receiver at anytime whenever an absolute positioning is required, so the system has to either maintain extremely accurate knowledge of base station position or spend a large amount of time and resource to solve this ambiguity, which is commonly expensive for consumer GPS receiver.

Hedgecock et al. [37, 38] propose a series of ideas similar to the methodology of DGPS and RKT, which use multiple GPS receivers to build up a network instead of involving a stationary station. Nodes in the networks can share their raw satellite measurements and use this data to track the relative motions of neighbour nodes as opposed to computing their own absolute coordinates. Their works are able to achieve centimeter-level accuracy in unobstructed environments, but it cannot mitigate errors resulted from local effects such as multipath or interference, which is quite common in urban scenarios. As the local effect cannot be easily mitigated with GPS receivers solely, we may need other complementary sources to improve the accuracy.

2.1.3 GPS Enhancement in Efficiency

GPS-based localization has been continuously optimized at both higher and lower levels since its operation. In order to improve energy efficiency, both EnTracked [45] and RAPS [66] add adaptivity by leveraging the estimation/prediction from historical GPS readings. Jurdak et al. [42] and Paek et al.[67] explore various duty cycling strategies. Others (e.g., A-Loc [52]) also suggest fusing other sensing information to assist GPS so that to reduce its energy consumption. Thanks to these proposals and related engineering efforts such as A-GPS [48], latest GPS modules in our smartphones have become more energy efficient than ever, especially when only GNSS measurements are retrieved.

Optimization at lower level focuses on signal processing and computation. In particular, signal sparsity has been exploited in both [35] and [60] to speed up computation and thus saving energy. Further computation reduction can be achieved by cloud offloading [54]: the GPS receiver collects only a small number of raw GPS readings while moving the location estimation to the cloud. In fact, the raw GPS readings contain a fairly high amount of information, simply summarizing them into 3-axis location indicators using model-based methods almost surely causes information loss. Our attempt to process these raw data using machine learning schemes is the first step towards their more holistic utilization.

2.2 GPS-free Outdoor Localization

Whereas most user localization systems are designed for indoor scenarios, GPS-free outdoor localization has a long history under the topic of wireless sensor network (WSN) localization but very few of them are dedicated to user localization. Existing node self-localization methods[13] can be divided into two main categories: i) range-based methods and ii) range-free methods.

2.2.1 Range-based Localization Methods

Range-based methods normally require pairwise distance measurements among all or part of the devices (or among various locations of the same device). The distance measurements are normally obtained through time-of-flight (ToF)/time-of-arrival (ToA), time-difference-of-arrival (TDoA), angle-of-arrival (AoA), dead reckoning and received signal strength (RSS) with a certain propagation model.

ToF/ToA and TDoA: ToF/ToA method measures travel times of signals between transmitter and receiver [55, 92, 57, 72, 76, 83] when they are perfectly synchronized. TDoA is introduced for cases that synchronization whose precision can match the signal speed is hard to achieve. TDoA locates by measuring the signals arrival time difference between transmitters and receivers. Since it requires only the difference between arrival times observed at several receivers, TDoA removes the need for synchronization between the transmitters and receivers, but the receivers must still be synchronized. TDoA can also refer to the difference between arrival times of two signals, which are mainly radio frequency (RF) and ultrasound in particular [72, 76]. This mechanism measures the propagation time of one signal and makes use of another signal to perform time synchronization. Measuring distance through ToF/ToA/TDoA sometimes requires non-RF signal sources so that the time can last long enough to be measurable. For example, Liu et al. [55] propose Guoguo which consists of an anchor network with a coordination protocol to transmit modulated localization beacons using high-band acoustic signals. Alternatively, it would require a sophisticated design for RF signal [92], which would not be usable for outdoor localization any sooner.

Dead Reckoning: Dead reckoning is the process of calculating one's current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course. Dead reckoning is useful

for assisting user tracking in a small scale indoor space [109, 50, 43], otherwise the accumulated errors can render the results unusable. Thus locating a user in a metropolitan area cannot solely rely on dead reckoning.

Angle of Arrival: AoA is initially used in cellular networks and later used for localization in WSN [63, 74, 46, 88, 102]. AoA requires additional gears (e.g., an antenna array) for each receiver to detect the propagation direction of an RF wave from the transmitter. Given the array geometry, the angle information can be derived based on differences in arrival times of an incoming signal at different antenna elements. However, AoA-based methods are highly sensitive to multi-path and non-light-of-sight. In addition, the localization accuracy with AoA decreases when the distance between the transmitter and the receiver increases. Both make AoA-based solutions not suitable in outdoor urban scenarios.

RSS-based Approach: RSS has established the mathematical model on the basis of path loss attenuation with distance. RSS-based methods are cheaper and simpler compared to time-based methods in terms of available system infrastructure and more suitable in large-scale scenario compared to dead reckoning and AoA. As a result, the error-prone RSSI model-based ranging seems to be a reasonable solution. RSS values have been proven to be largely affected by path loss and shadowing effect[25], therefore existing RSS-based methods focus on reducing these potentially very large errors. [15] use pair-distance constraints obtained between hotspots and users to infer an RF model. However, in an outdoor scenario, we lack knowledge of hotspot location so we cannot directly apply a learned RF model. [51] firstly introduces collaborative localization through a “brute-force” dimension reduction conducted by minimizing the mean errors iteratively between the error-twisted high dimensional structure and its 2D projection. Many following works [93, 90, 1] improve its efficiency through new iterative approaches or by defining the problem into a new optimization problem. However, all these methods are used in

WSN and only consider relations between transmitters and receivers derived by RSS. As a result, they fail to consider relations within transmitters or within receivers. The approach of manifold-alignment [68] can be deemed as an implicit range-based method: it does not directly convert RSSI readings into distances, but it rather considers those readings as distance metrics in a certain manifold structure. This approach has been applied to indoor tracking [69] which incorporated temporal relationship into consideration, but it is still an open question whether it works for localization with crowdsensed labels in the absence of sequential timestamps.

2.2.2 Range-Free Localization Methods

Range-free methods have two different manifestations, namely beacon-enabled methods for multi-hop networks and fingerprinting method for indoor localization.

Beacon-enabled Methods: The beacon-enabled methods [9, 64, 36] only require a node/user to hear from a few beacons with known locations, and then use simple computations [9] or logical reasoning [64, 36] to obtain a coarse-grained location estimation. In outdoor scenarios, the locations for the beacons are mostly not available. We will explain how industry applies a similar two-stage approach for outdoor localization soon in Section 2.2.3.

Fingerprinting Methods: The fingerprint-based localization involves two phases. During the first phase, the received signals at selected locations are recorded in an offline database called radio map. During the second phase, also called the online stage, the pattern matching algorithms are used to infer the location of an unknown node by matching the current observed signal features to the prerecorded values on the map. Fingerprint can be obtained on various type of data source, such as visual images [77, 106], motion data from IMU (inertial measurement unit) [104, 17] and RF signal [14, 58]. Visual fingerprinting methods require dense labelled data to achieve an accuracy about tens of

meters in outdoor cases. Motion fingerprinting requires additional information such as path and map structure. These two types of fingerprinting data both require huge labour resource to collect data or label map information. WiFi fingerprinting method [4, 100] take RSSIs not as a distance indicator but rather as an observed pattern, so indicating locations by pattern matching has the potential to achieve a fine-grained localization if a certain area is fully labeled with the observable patterns (or fingerprints). However, whereas certain efforts have been made to migrate the fingerprinting methods from indoor scenarios to outdoor environment [14, 89], it is now well accepted that i) fingerprinting an area (even a very small one) through wardriving is a major bottleneck even for indoor localization, and ii) the localization ability is confined to only the region that has been fingerprinted. As a result, practical deployments for outdoor localization are mainly using the computationally light beacon-enabled methods by taking WiFi hotspots as beacons [87, 65].

2.2.3 Current Practices of Outdoor GPS-free Localization

Most of current commercial or open-source WiFi localization systems can be clearly divided into two stages: Hotspot Localization (HL) and User Localization (UL), as illustrated by Fig. 2.1. Hotspots localization is often regarded as the offline pre-processing stage, where the locations of WiFi hotspots are estimated based on crowdsensed labels collected and stored in a database. These estimations stored in the database are regularly updated as new labels become available. To the best of our knowledge, although WiGLE [96] and Skyhook [79] are closed source for the implementation, they have published that they employ the weighted centroid method to estimate hotspot locations based on the crowdsensed labels. In particular, each label contains a GPS location indicating where the concerned hotspot is heard, as well as the RSSI from that hotspot indicating the receiver's relative distance to the hotspot. As a result, a hotspot location is esti-

Figure 2.1: A two-stage localization approach: Hotspots Localization (Left) and User Localization (Right). We mark known locations in black and estimated locations in red. Hotspots Localization aims to locate hotspots (AP1 to AP5) given several user locations (U1 to U4) along with corresponding hotspots RSSIs. User Localization aims to estimate a new user's (User X) location based on previously estimated locations and their respective RSSIs. [Best viewed in color.]

mated as the centroid of all labels (their GPS locations) concerning it but weighted by the respective RSSIs.

User localization is regarded as the online localization stage when a user location is calculated based on the observed hotspots whose positions have been estimated and stored at the first stage, as well as their RSSI readings. The weighted centroid method is again used in this stage, which is a reversed process of getting the hotspots locations: the estimated hotspot locations are used to compute the centroid that indicates the user location, with RSSIs serving as the weights. OpenBMap [65] is open-source and its offline localization algorithm applies a Kalman Filter to sequentially process the hotspot labels during this stage, this seemingly more sophisticated method essentially yields the same (unsatisfactory) localization accuracy.

Although a two-stage approach may work in an ideal case, it is prone to error accumulation across the two stages because the information contained in the original labels do not get fully propagated to the UL stage. Moreover, a two-stage approach treats each estimation (in both stages) in a localized manner, neglecting the spatial relationship among hotspots and users; losing such information can be fatal to the final location estimation result.

2.3 SSID-based Analysis

Although WiFi signal has been used a lot for data mining, SSID as part of WiFi information has not drawn much attention from the research community. Existing works about SSID focus on 3 type of problems: (1) analysing semantic contexts of places; (2) analysing user profiles from WiFi logs; (3) mapping hotspots to known places.

For **semantic analysis of places**, Wong et al. [98] mark wireless space in cities as public and private by identifying certain name pattern in the SSID. Chang et al. [10] extract a temporal feature from WiFi log to get information like hour distribution for each day, weekday distribution for each week, stay duration and visits number, and classify a place into a pre-defined type based on activity for a person (e.g. working, dining, education, etc). The latest work from the same group [19] further introduce spatial correlation feature of a hotspot extracted from WiFi logs for classification. Their works are based on data evenly collected over a long period of time, thus temporal feature are representative. However, they are not suitable for our problem since the collection time of crowdsensed data do not follow a uniform distribution generally. Besides, the works for venue classification are orthogonal to our problem of hotspot-venue matching, and can be a part of a more robust system in the future. Additional information such as venue type can be further added as a constraint for matching.

For **user profile analysis**, Fan et al. [20] predict a user's preference for leisure activities based on the user's WiFi log over a long period of time. Cunche et al. [18] study the relation between WiFi fingerprints from different devices and infer the social relation of device owners. They also study the mobile device vendor preference based on SSID analysis. Macro V. et al. [5] study the WiFi probe requests received at various events and analyse the international nature of the event, the density of foreign participants and socio-economic characteristics of the participants. These works are either based on data collected from particular users to analyse their preference or based on data collected over an event or a venue to study the social relation or status of various users. Although such information is marginally relevant to our current problem, the analysis approach can help in the future if user profiles of crowdsensed participants are known.

For **hotspot place mapping**, [78] tends to match hotspots to business venues in Yelp dataset, but only considers the text similarity and no other information. [33] designs a shop profiling system and tends to locate a hotspot into a shop by matching SSID to brand name and formulating the WiFi heatmap. Their search domain is quite small which is only within the shopping mall, while our problem is on a larger scale. Design a matching mechanism for a much larger geographical scale and multiple larger crowdsourced place databases with duplicated information is non-trivial. In our problem, we need to consider not only SSID text and WiFi RSS but also information source and its credibility in order to find the most suitable match among a huge amount of candidate venues.

2.4 Indoor-Outdoor Detection

Differentiating indoor and outdoor scenarios has been the first application of the data analytic approach to GPS readings. In fact, earlier approaches of IO-detection [108, 73] criticize simply data analytics on the number of visible satellites as being slow and energy inefficient. As a result, they apply various machine learning techniques to fuse non-GPS

sensor readings: while [108] applies a naive decision tree, [73] has tried out several semi-supervised learning techniques. Nevertheless, classical machine learning techniques are too sensitive to feature selection, which can be an obstacle when facing sophisticated data inputs. Recently, SatProbe [11] revisits the number of visible satellites for IO-detection, and it proposes efficient algorithms to extract this feature. However, the access to raw GNSS measurement granted by Android 7 provides users with much more information (e.g., the spatial relation among visible satellites) than just this counter-based feature. There are some works which use other sensor indicators on the device or additional sensors as infrastructure to detect indoor and outdoor scenario. For example, [81] analyze sound to create a seamless positioning system to detect indoor and outdoor, and [110] uses i-Beacon as additional sensors in the environment to better distinguish outdoor environment from the semi-outdoor environment. However, these works either require complicated infrastructure setup or are sensitive to changes in the context environment. Moreover, existing approaches all categorize urban scenarios into 3 classes: outdoor, semi-outdoor, and indoor; this is apparently oversimplifying our real-life location contexts.

2.5 Mining Mobile Sensing Data

Thanks to the latest mobile devices equipped with numerous sensors, we can obtain a huge volume of sensing data, mining which has enabled quite a few applications including localization (e.g., [3, 100, 103]) and activity recognition (e.g., [40, 59, 71, 53]). Most of these proposals follow a classical machine learning pipeline, starting with empirically selecting a few features and ending with a mapping from these features to certain known patterns. In order to relieve the reliance on pre-knowledge, Lasagna [53] is among the first to apply deep learning technique for activity recognition. Nevertheless, the data inputs considered so far are simple time series (e.g., gyroscope delivers intensity readings

on 3 axes), whereas the GNSS data we shall look into have a much more complicated structure as illustrated in Section 1.2.2, rendering existing proposals inapplicable.

2.6 Summary

This section provides a systematic summary of the current research related to our study. We first briefly introduce the dominant localization strategy for outdoor scenarios: GPS-based localization. Then we survey some research works on improving its performance in terms of accuracy and efficiency, and discuss the limitation and opportunities of these work which together drive us to an integrated outdoor solution. The limitation of works on improving GPS accuracy leads us to look for a supplementary source for the outdoor solution, while the opportunities provided by recent optimization on GPS efficiency enable us to utilize GNSS raw measurements for profiling context as a potential trigger between different data sources in the integrated solution.

To address the two problems mentioned in Chapter 1, we further discuss works about GPS-free localization, SSID-based analysis, and context-aware mobile computing. In terms of methodology, we also survey some state-of-the-art works on mining mobile sensing data, which motivates us to develop novel data-driven solutions. By analysing the strength and weakness of these existing works, we propose a series of strategies to alleviate their respective limitations and drive to an integrated outdoor localization solution with data-driven approaches: (1) we propose WOLoc, a WiFi-based localization method with manifold learning technique on crowdsensed data, in Chapter 3 to solve the potential failure caused by two-stage localization currently used as industry practice; (2) we propose SSID-WOLOc, an extended wifi-based localization solution, which utilize SSID-based text analysis technique to locate hotspot by leveraging on multi-source public venue database, in Chapter 4 to mitigate potential failure caused by insufficient labelled data; (3) we propose a novel deep-learning method to learn compact representation from

raw GNSS measurements and apply such representation to profile GPS performance in Chapter 5.

Chapter 3

WOLoc: WiFi-only Outdoor Localization

In this chapter, we firstly introduce manifold learning and its application to WiFi-based localization, then elaborate our WOLoc system in details, and finally evaluate its performance in both efficiency and accuracy.

3.1 Manifold Learning in WiFi-based Localization

Manifold learning is essentially a non-linear dimensionality reduction method. It is based on a basic observation that dimensionality of many data sets is only artificially high. Algorithms relevant to manifold learning tends to learn the manifold structure underlying in the dataset. When a part of the data is labelled, semi-supervised manifold alignment can be applied [34] to predict the unlabelled data. If we represent each data as a vertex and construct a graph based on their neighbourhood relation, a general objective cost function of semi-supervised manifold alignment problem is defined as:

$$C(\mathbf{f}) = \sum_i |f_i - y_i|^2 + \gamma \mathbf{f}^T L \mathbf{f}, \quad (\text{Eq. 3.1})$$

where $\mathbf{f}$ is a mapping function defined on the vertices of the graph that matches labelled vertices to the target values, y_i represents each labelled data value, L is the graph Laplacian for the underlying manifold, and γ controls the relative weights among terms. The

first term is the fitting error, and the second term is the regularization term for graph Laplacian which ensures nearer points on the manifold have more similar values, thus it enforces the smoothness along the manifold.

In the context of WiFi-based localization, if we consider the signal received for all hotspots from one location as a data point, the dimension of the data is high given that hundreds of hotspots can be observed at that location. Fortunately, as two close-by locations should have similar signal readings, the distance between data points in the high-dimensional space intrinsically preserves the geometry between locations. If every signal is received perfectly and follows the Path Loss Model based on the distance between transmitter and receiver, the data is only artificially high-dimensional and should lie on a 2D manifold. However, due to the errors inherent to RSSIs, the manifold created based on them would be bend to a space with dimension higher than 2. Given some of the locations are labeled by crowdsensing participants, a semi-supervised manifold regularization aims to learn the graph structure in the low-dimensional space that can best fit all the signal data while preserving their geometry. The unlabeled locations are thus estimated through the low-dimensional structure [6]. Different from the two-stage method (introduced in Section 2 as “centroid-based” methods) that focuses locally on a single hotspot or user, manifold learning takes a more holistic view over all crowdsensed data. It not only uses RSSI as distance metrics between user and hotspots but also reconstructs the topological relations among hotspots and users. User manifold is constructed under the observation that close-by locations observe similar RSSIs from all hotspots, while hotspot manifold is constructed under the observation that two close-by hotspots cause similar RSSI readings to all receivers. Furthermore, these two manifolds (for users and hotspots, respectively) are unified into one large manifold (more details in Section 3.2.3).

Figure 3.1 shows an illustration of a comparison between the weighted centroid method and manifold learning. One main limitation of the centroid-based method in

Figure 3.1: Comparing localization based on Weighted Centroid Method (Left) and Manifold-based Learning (Right). We consider a target hotspot of which the true location is shown as the black star. Black dots show locations of users that discover it. Blue stars are its neighbouring hotspots in the constructed manifold. Red star indicated its estimated location in each method based on which the red a potential propagation range is shown as the red circle. Phone icon indicates a new user location, which may just fall in the estimated propagation range and can be better predicted in the manifold-based learning scenario. [Best viewed in color.]

estimating a hotspot location is that it treats the hotspot independently from other hotspots. Therefore, no matter how RSSIs are factors as weights, the estimated hotspot location (red star) is always inside the convex hull induced by the observing user locations (black dots). When the collected data are mainly on the road, the weighted centroid method also gives the estimated location of a hotspot very close to the road. Apparently, such a large error may seriously jeopardize the user localization later: if we simply estimate a user requesting location (the mobile phone) as within the red circle centered around the estimated hotspot location, it can be seriously biased. Comparatively, manifold learning takes a more holistic view. As shown in Figure 3.1 (right side), within the constraint of the large manifold, the target hotspot (red star) is not independently estimated by the surrounding users' observations (black dots) but rather together with its surrounding hotspots (blue stars). Obviously, constructing a manifold to represent the relations among hotspots and users preserves the label information to the maximum extent, hence it has the potential to obtain a higher localization accuracy.

Figure 3.2: WOLOC system architecture.

3.2 WOLOC: A Manifold Perspective in Localization

To overcome the potential problem inherent to the current practices, we proposed WOLOC as an outdoor localization system driven by manifold-based learning techniques. The system architecture comprises of three parts shown in Figure 3.2: pre-processing of crowdsensed data, offline manifold learning exploiting existing crowdsensed labels, and online location query processing.

3.2.1 Pre-Processing of Crowdsensed Data

Many crowdsensing applications available in the market share a similar mechanism to obtain crowdsensed hotspot location data. The application starts a hotspot discovery according to various schedules (e.g., triggered by a significant location change). It records, for each discovered hotspot, the BSSID, SSID, RSSI. It also obtains its own location (latitude, longitude) along with GPS signal statistics (accuracy, represented by confidence range, and the number of visible satellites), and this location and the corresponding timestamp are associated with every discovered hotspot. All these information for a given hotspot constitute a *label*. A *record* contains a set of labels collected by a user at a

given position. Crowdsensing data include two types: i) sequential data with timestamps and ii) single data at any position.

For sequential data with timestamps, we first mark the records with a very small number of visible satellites or a large confidence range as “suspicious records”, which mostly occur among high-rises, under shelters or at the beginning of a trip when GPS is still searching for visible satellites. Then we eliminate, out of these suspicious records, those with fewer than 5 satellites or a confidence range larger than 20 meters. For data logged sequentially (in timestamp), we also remove those with a huge jump in distance and velocity to avoid potential errors caused by inaccurate GPS location; this is done by calculating the distance between consecutive records and average velocity inside a sliding window of 3 records. We set a distance threshold as 100 meters and velocity threshold as 80 m/s given a sampling rate of 1Hz.

Among all the detected hotspots, two types of mobile hotspots should be eliminated: i) personal hotspots and ii) public transport hotspots. Normally, a fixed hotspot has a signal range of about 100 meters, so we apply the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering algorithm on all label locations for each hotspot. Assume there are k labels available for one hotspot, we set the minimum points of a cluster as $0.8k$ and the maximum distance as 200 meters. If all the points are finally labelled as “noise” after DBSCAN, it means the heard locations for the hotspot are too sparsely distributed, and the hotspot is highly possible to be a mobile hotspot. We keep a record for all the mobile hotspots discovered and avoid using them in the following processing. Such information is maintained in the database as a reference for data processing in the future.

As we want to limit the size of the database to achieve efficient computation in the following process, labels with same locations are combined into one by averaging the RSSI for each hotspot, where the “same” is defined by limiting the precision of recorded geo-locations (i.e. latitude and longitude). The number of combined labels is recorded in the

database, and will be used for a future combination when new crowdsensed data is available. For any new label inserted into the database, a same-location check/combination is performed to minimize the size of the database.

3.2.2 Problem Formulation

After filtering processing, we can construct a signal matrix S for all the remaining labels. Assume that we have n hotspots detected in m records, S will be a $m \times n$ matrix, and $S = \begin{bmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{m1} & \cdots & s_{mn} \end{bmatrix}$ where s_{ij} is the RSSI for the j -th hotspot in the i -th record. Each column represents one hotspot, and each row represents one record. We fill all the blank cells with a small default value $s_{\min}$. Locations of records are maintained using a $m \times 2$ matrix $\mathbf{u} = [u_1, \dots, u_m]'$ where $u_i = [u_{ix}, u_{iy}]'$. Given the signal matrix S generated from pre-processing stage and the corresponding labelled user locations $\mathbf{u}$, our goal is, for any new record $\mathbf{s}_{m+1} \in \mathbb{R}^{1 \times n}$, to estimate the user location u_{m+1} . It turns out that, as a byproduct, we will obtain the hotspot locations $\mathbf{h} = [h_1, \dots, h_n]'$ simultaneously, where $h_i = [h_{ix}, h_{iy}]'$.

3.2.3 Manifold Construction

The construction of manifold is based on three facts: i) two near locations receive similar signal strengths from surrounding hotspots, ii) a user receives similar signal strength from two hotspots near to each other, and iii) the nearer a user is to a hotspot, the stronger the signal received will be [69]. In our context, these translate to: i) if each row of S is represented as a point in n -dimensional space, two locations, u_i and u_j , spatially near in real-world should be close to each other in the n -dimensional space, ii) if each column of S is represented as a point in m -dimensional space, two hotspots, h_i and h_j , spatially near in real-world should be close to each other in the m -dimensional space, and iii) the larger s_{ij} is, the nearer j -th hotspot is to the location of the i -th record.

Therefore, we construct two separated manifolds first: user location manifold and hotspot location manifold, and the neighbourhood relationship is given by k-Nearest-Neighbour (KNN) method. Since the RSSI and distance is not linearly related, we first convert the RSSI values to weights using a non-linear transformation to get the normalized signal matrix $\tilde{S}_N$: $\tilde{s}_{ij} = \exp\left(-\frac{(s_{ij} - s_{\max})^2}{2\sigma^2}\right)$, where $s_{\max}$ is the maximum RSSI a user can receive in an outdoor environment, which indicates a significantly small distance between user and hotspot. σ is known as the Gaussian kernel width. Empirically, we set $s_{\max} = -30\text{dBm}$ and $\sigma = 12$ based on the crowdsensed data. Note that σ affects the spatial density of hotspots: the larger the σ is, the more sparsely hotspots are distributed. Given users' geo-locations, we directly use great-circle distance as the metric for user location manifold between pairs in $\mathbf{u}$ for user location manifold. For hotspots location manifold, we use the Euclidean distance between column vectors in $\tilde{S}_N$ as the metric.

For each manifold, we define a weighted adjacency matrix A_* where $a_{ij} = \exp\left(-\frac{\|\tilde{\mathbf{s}}_i - \tilde{\mathbf{s}}_j\|^2}{2\sigma^2}\right)$ if i and j are neighbours in the manifold; otherwise 0. Let A_u be the $m \times m$ matrix for the user location manifold and A_h be the $n \times n$ matrix for the hotspot location manifold. To align the two manifolds into one, we define a unified adjacency matrix $A = \begin{bmatrix} r_u A_u & r_s \tilde{S}_N \\ r_s \tilde{S}_N' & r_h A_h \end{bmatrix}$ where parameters r_u, r_s, r_h are set to be small positive values induced by harmonic functions on the graph. A clearly represents the relative distances and connectivity among users and hotspots based on the three aforementioned facts.

3.2.4 Offline Learning for Location Estimations

To solve the hotspot locations and unknown user locations at one time, we apply a semi-supervised learning approach. Given the relative locations of users and hotspots represented by A , known locations denoted by $\mathbf{y} = [\mathbf{u}', \mathbf{h}']'$, and indication matrix $K = \text{diag}(k_1, \dots, k_{m+n})$ where $k_i = 1$ if the location of user or hotspot is given in $\mathbf{y}$, otherwise

$k_i = 0$, our objective is to find a set of locations $\mathbf{p}$ best fit current relative patterns and has the minimum fitting errors compared to known locations. Therefore, the objective is:

$$\mathbf{p}^* = \arg \min_{\mathbf{p} \in \mathbb{R}^{(m+n) \times 2}} (\mathbf{p} - \mathbf{y})' K (\mathbf{p} - \mathbf{y}) + \gamma \mathbf{p}' L \mathbf{p}, \quad (\text{Eq. 3.2})$$

where L is the graph Laplacian: $L = D - A$ where $D = \text{diag}(d_1, d_2, \dots, d_{m+n})$ with $d_i = \sum_{k=1}^{m+n} A_{ik}$. The second term is the regularization term, where $\gamma > 0$ controls the smoothness of the coordinates along the manifold. The problem has a closed-form solution:

$$\mathbf{p}^* = (K + \gamma L)^{-1} K \mathbf{y}, \quad (\text{Eq. 3.3})$$

where $\mathbf{p}^* = [\mathbf{u}^*, \mathbf{h}^*]'$ yields estimated locations for both users and hotspots.

3.2.5 Online Location Query Processing

When processing the online location queries, involving all records in a database (hence the full manifold) can be avoided for efficiency purpose if the queries are geographically confined in a small region. In the WOLOC system, the hotspot manifold is constructed offline and stored in the database. Upon receiving a user location query (i.e., a record with unknown locations, s_u), the WOLOC server searches among the hotspots in the query record, and retrieves a subset of relevant hotspots from the database. This *candidate set* concerns all the hotspots in the query, as well as their neighbouring hotspots in global hotspots manifold.

Then WOLOC selects a subset of records from the database to formulate $\hat{\tilde{S}}$ along with the query record s_u ; a record is selected if it contains an RSSI value significant enough for any hotspot in the candidate set. $\hat{A}_h$ is computed based on $\hat{\tilde{S}}$ and sub-manifold retrieved from the global hotspot manifold computed offline. Based on the location $\hat{\mathbf{u}}$ from the selected records, WOLOC creates a user location manifold online

and inserts query record using KNN with the Euclidean distance between row vectors in $\hat{S}$ as distance metrics, and then computes $\hat{A}_u$. The query is sent in a JSON format, which includes a list of key-value pairs for BSSID and RSSI. An example is shown as follows: `{"WolocQuery":[{"bssid":"12-34-56-78-91-00", "rssi":-57}, {"bssid":"98-76-54-33-22-11", "rssi":-87}]}`.

After obtaining $\hat{A}_h$ and $\hat{A}_u$, the WOLOC server applies the learning solver (Eq. 3.3) to obtain the optimal solution for these local structures and returns the queried location back to the user. The location is also returned as JSON, with a format of `{"location":{"lat": 48.85702, "lng": 2.29520}}`. By processing a much smaller set of records, the processing time is significantly reduced and WOLOC can respond to the query more promptly, as we shall demonstrate in Section 3.3.3.

3.3 System Evaluation

3.3.1 Experiment Setting

We conducted experiments in the following 6 outdoor areas:

- **Downtown:** central business district filled with commercial and business buildings as shown in Figure 3.3.a.
- **Campus:** educational institute district with buildings in open area as shown in Figure 3.3.b.
- **Hybrid Residential Area (Hybrid R.A.):** medium-density residential neighbourhood with a few shops and a community center as shown in Figure 3.3.c.
- **Residential Blocks (R.B.):** high-density residential neighbourhood filled with high-rises as shown in Figure 3.3.d.

Figure 3.3: Maps provided by Google Map for all areas concerned in our experiments. (a) Downtown. (b) Campus. (c) Hybrid Residential Area. (d) Residential Blocks. (e) Community Area. (f) Downtown Entertainment Area.

- **Community Area (C.A.):** mixture of residential high-rises, private houses, markets, shopping malls and community centers as shown in Figure 3.3.e.
- **Downtown Entertainment Area (D.E.):** high-density of business high-rises, shopping malls, restaurants, and entertainment facilities along riverside as shown in Figure 3.3.f.

As the commercial platforms either do not open their database [79, 29] or have very limited coverage in our city [65], we have limited open data from online sources for our evaluation. We construct the cases (e) and (f) from OpenBMap database that has in total 26 traces from 2010 to 2016 covering some of these 2 areas. To further extend our evaluation cases, we develop an Android application to collect WiFi and location data through walking and cycling. The Android application continuously detects user location

Table 3.1: Hotspots density and number of hotspots per record

Area	Hotspots Density (APs/km ²)	# Hotspots per record		
		Mean	Standard Deviation	Median
Downtown	30400	51.32	32.99	41
Campus	32900	88.42	36.08	91
Hybrid R.A.	27300	32.17	6.95	31
R.B.	29800	38.77	12.21	38
C.A.	18800	35.90	15.89	32
D.E.	26100	48.21	31.14	41

using GPS module and scans surrounding WiFi hotspots at 1Hz. For each hotspots scan, we record all the standard information as discussed in Section 3.2.1.

All the complementary data are collected over a 2-month period at various times in a day (30% in the morning, 53% in the afternoon, 17% in the evening). 3 Android phones with different brands (HTC One M8, Xiaomi Redmi Note 4 and Samsung Galaxy S4) are used. In each area, 2 traces are collected by each of the 3 phones, thus in total 6 traces are collected to cover each of the areas. Data in cases (a)-(d) are collected by walking, while data in cases (e) and (f) are complemented by cycling given the larger area.

We have a full-implementation for WOLOC server in Java on a PC with 16GB RAM. For each evaluation purpose, we select part of records as testing data and use the remaining records as training data. For each area, the server firstly builds up a database and constructs manifolds offline based on the training data, then it processes location queries in JSON format (generated from testing data) and returns user locations.

3.3.2 Statistics on Hotspots

Figure 3.4 shows the distribution of the number of hotspots detected per record for each of the 6 areas. Table 3.1 shows the statistics for hotspots per record for different areas. As expected, downtown and campus have higher hotspot density than residential zones, where the number of hotspots per record can reach more than 100 in some areas.

Figure 3.4: Hotspots density for all areas in our experiments.

Downtown area also has a high variance in the number of hotspots per record as a result of the various height of buildings and unevenly distributed buildings in the zone. Campus has generally more hotspots detected per record and highest density, as the hotspots are densely located to achieve high accessibility for all users in the campus. Residential blocks have a bit denser hotspots distribution as the blocks have more levels and more residents compared with private semi-detached houses in the hybrid residential area. Community area, as a larger scale of residential areas, shares similar properties as hybrid residential area and residential blocks. Most of the records in this case contain about 15 to 45 hotspots. Downtown entertainment area has almost the same distribution as downtown case, which shows not only streets and pedestrian streets but also riverside streets have sufficient hotspots equipped. However, the reported hotspots density at the two large

3.5.a: Impact of number of hotspots/records.

3.5.b: Processing time distribution.

Figure 3.5: Query processing time using all hotspots in a query and their neighbouring hotspots.

areas is lower than the first 4 areas as we cannot cover the entire large space in details due to the lack of manpower. In summary, nowadays metropolitan areas have sufficient WiFi infrastructure to help outdoor localization if we use them properly.

3.3.3 Time Efficiency of WOLOC Localization

We verify the time efficiency of the system before evaluating its performance in term of accuracy. WOLOC has two separated processes, namely offline process and online process. During the offline process, logs submitted to the server are pre-processed and global manifolds are pre-computed in the server. It only happens when there are a sufficient number of new user logs received. An online process is invoked in response to a user location query. This process involves local manifold construction and location computation. Time to accomplish the online process is the *processing time* for the server to return location back to a user, so this is what we are evaluating here. We implement a full-version of WOLOC and arbitrarily select 100 records as testing data and build the global manifold with the remaining records. We record the time that WOLOC takes to accomplish online processing for each query. We plot the processing time as a function of

Figure 3.6: Error statistics as a function of the number of candidate hotspots.

the number of hotspots involved in the online processing in Figure 3.5.a; it is exponentially increased with both the number of hotspots and number of records. If we retrieve all the surrounding hotspots concerned by a location query, 70% of the queries in the experiment can be finished within 5 seconds as shown in Figure 3.5.b. The mean processing time is 4.22 seconds. To further reduce the processing time, we test the performance by involving only those hotspots in the query and even a subset of it. We select the subset based on the RSSI value, and we only take the hotspots with strong RSSI values for further processing. Figure 3.6 shows the accuracy when processing with different numbers of hotspots. We observed that the location accuracy is largely insensitive to this number as long as it is sufficiently large (≥ 6). Figure 3.7.a and 3.7.b show that, after reducing the number of candidate hotspots, the processing time can be reduced to 0.5s in most cases. The mean processing time is 158.12 ms with a standard deviation of 146.98 ms. Therefore, for the following experiments, we only take the most important hotspots contained in a query and their neighbours as candidates. As it is impossible to tell the processing time from the Internet delay for public web services, we have to omit the comparison of processing time with other systems at this stage.

3.3.4 Accuracy of User Localization

To evaluate the accuracy of WOLoc in user localization, we conduct 50 experiments for each area. For each experiment, we first randomly select 100 records with high accuracy level (≤ 10 meters) and sufficient number of satellites (≥ 8) as the testing set. The locations contained in these records are treated as “ground truth” for the evaluation purpose; they are temporarily removed from the records so that they can emulate the location queries issued to WOLoc. We then use the remaining records as the crowdsensed data set to emulate the database; they are used by WOLoc to construct the manifolds.

In Figure 3.8, we only report the results for 10 experiments in each area due to space limit. WOLoc yields median errors less than 7 meters for all testing cases in first 4 areas (a)-(d), as well as third quartile of errors all less than 20 meters. Normally, an error less than 10 meters can be achieved if the number of hotspots per record is high (e.g., in Campus case), whereas large errors are often due to an insufficient number of hotspots in a record (e.g., in Downtown case). For Community Area, it has a higher median of 13 meters compared with all other areas, and both Figure 3.8.e and Figure 3.8.f have higher variances. These stem from the low WiFi coverage given the much larger areas. Note

Figure 3.7: Processing time using only hotspots in a query.

that the median errors yielded by WOloc are quite comparable to the accuracy level of GPS, which is about 3 to 7 meters if there is a sufficient number of satellites.

3.3.5 Comparison with Other Systems

We also compare WOloc’s user localization accuracy against 3 open-source or commercial systems available in the market: OpenBMap Offline Localization System [65], Skyhook Precision Location Service [79], and Google Location Service [31]. We issue the same location queries to the 3 systems mentioned earlier. Though each of them has its own database, the open-source nature of OpenBMap [65] allows us to compensate its sparse WiFi labels: it has only about 5,000 hotspots available in their database for the areas

Figure 3.8: Error in meters for estimating user location using WOloc.

that we conduct the experiments, so we add more hotspots labels from WiGLE [96] to enlarge the database to over 25,000 hotspots.

Skyhook [79] provides a Python API for us to submit online location queries, but we have no details about its database. A similar situation applies to Google Location Service [31], but it by default requires GPS to achieve an accurate localization, though WiFi-based localization is used to complement the GPS. To have a fair comparison, we disable GPS when issuing queries to Google in JSON format through Google Maps Geolocation API [31]. OpenBMap returns a location containing only latitude and longitude, but both Skyhook and Google return a JSON response, in which besides the estimated location, there is an “accuracy indicator” of the estimated location represented as the radius of a circle around the given location.

Figure 3.9: Median error comparisons between WOloc, OpenBMap, Skyhook and Google for all 6 areas.

Figure 3.9 shows a comparison between 4 different systems, and it is very clear that WOloc outperforms all of them. Detailed error distributions are shown in Figure 3.10 for all the 3 commercial systems with 10 test rounds for each of the 5 areas (1 area is omitted due to space limit). Generally, all 4 systems perform better in smaller areas (the first 4) than larger areas (the last 2), but WOloc significantly improves the performance (in both

statistics and distributions) compared with others. OpenBMap’s algorithm with weighted centroid and Kalman filter performs worse given the same database as WOLoc, which shows the ineffectiveness of its oversimplified method. It is a bit of a surprise that Google performs worse than WOLoc, which is probably because that Google relies too heavily on GPS localization without making a lot of efforts in improving its localization algorithm using WiFi. Given the similar performance between Google and OpenBMap, we suspect that Google most probably applies similar algorithms to what OpenBMap claims to have used, due to their simplicity to achieve high computation efficiency. Skyhook’s performance has a higher variance compared with the other two; this may attribute to the lack of sufficient labels in its self-maintained database for certain areas.

3.4 Summary

To leverage on the existing WiFi infrastructure in metropolitan area and “big data” available in crowdsensed hotspot label database, we develop WOLoc as a WiFi-only outdoor localization system that relies solely on crowdsensed hotspot labels. We apply a semi-supervised manifold learning techniques to estimate a queried location based on its connection to the labeled manifold structure. We have conducted experiments in 6 metropolitan areas, and our results show that WOLoc yields localization errors between 5 to 15 meters in most cases. This result is significantly better than 3 systems currently available in the market, namely OpenBMap, Skyhook, and Google, in terms of WiFi-only outdoor localization, suggesting its effectiveness in outdoor localization. We have also figured out that the density of WiFi labels is a key, as WOLoc can have a larger localization error if the label density is low. Finally, the average processing time after our optimization is less than 200ms, demonstrating WOLoc’s capability in responding to realtime location queries.

Our work indeed has several limitations. Firstly, even manifold construction is part of offline processing which may not affect the responsiveness of online query service, there is indeed heavy computational cost when we need to generate the global manifold for a much larger area. Our current approach is to divide the area manually by functions of different subareas as in our experiments. However, such manual partitioning is not scalable for an automatic localization service in the future. For automatic partition of large areas into subareas, we leave it as future work for a more complete and robust localization service. A future solution to this problem can be a clustering algorithm to find cliques of hotspots and partition the areas based on the cliques. Secondly, current manifold alignment is mainly relying on user locations, and very little information from hotspots have been used. As a result, the algorithm will be quite sensitive to hyperparameters when constructing the manifold due to the lack of labelled data. We aim to incorporate more information from hotspots in this problem and solve it in our next work.

Figure 3.10: Location error distributions for 3 commercial systems: (a) to (e) for OpenBMap, (f) to (j) for Skyhook, and (k) to (o) for Google.

Chapter 4

SSID-based Text Analysis for Manifold-based Localization

In this chapter, we will first discuss the limitation of our previous work WOLoc. Then we discuss the usage of SSID and location service database as the potential location indicators. We elaborate the SSID-based text analysis techniques in details and demonstrate extended solution SSID-WOLoc which incorporates SSID-based text analysis to existing WOLoc framework. Lastly, we evaluate it by comparing with WOLoc and examining the overall performance of SSID-WOLoc.

4.1 Limitation of WOLoc with User Location Indicators Only

Despite the success of WOLoc in many urban scenarios elaborated in Section 3, it indeed has some limitations due to the format of crowdsensed data. As discussed earlier, the crowdsensed data is contributed by individual users, and each user submits a series of records, each of which consists of a location indicator and RSSI values of surrounding available hotspots. The location indicators submitted by individual contributors represent the user locations, which is collected by GPS and can be considered as the ground truth location of a user after filtering. However, when WOLoc tries to conduct semi-supervised learning with the global manifold, known location indicator (a.k.a labelled

data) for both users and hotspots are required to achieve a balanced distribution and better estimate the unknown position of users or hotspots.

In existing version of WOLoc which only utilize RSSI value of surrounding hotspots, the location indicator for hotspots is labelled based on an inference from the location indicators for users. Specifically, we adopt a “cut-and-pin” method. Since the nearer a user is to a hotspot, the larger the received signal strength (RSSI) will be, WOLoc sets a high threshold $s_{\maxloc}$ to pick out those user-hotspot pairs which are quite close to each other, then locate the hotspots with the user location labels as a rough estimation. For each hotspot Θ and its label set $\{(\mathbf{l}_{\Theta i}, r_{\Theta i})_{i=1,2,\dots}\}$,

$$\mathbf{l}(\Theta) = \begin{cases} \mathbf{l}_{\Theta k}, k = \arg \max_i r_{\Theta i} & \text{if } r_{\Theta k} > s_{\maxloc} \\ \perp & \text{otherwise} \end{cases}$$

where $\mathbf{l}_{\Theta i}$ represents the location of the i -th user, $r_{\Theta i}$ denotes the RSSI from Θ to that user, and $\perp$ means undefined.

Although this “cut-and-pin” method is easy to implement, it suffers from low accuracy given the signal vanishing and fading effect in outdoor scenarios. With this method, very few hotspots satisfy such criteria due to signal loss and they are labelled on the street with sub-optimal accuracy due to the nature of inferring user location as hotspot location. Especially when no hotspots are located with this method, there is an extreme imbalance for labels between users and hotspots. Such extreme imbalance will increase the negative effect introduced by wrongly choosing hyperparameters when constructing the manifold, and therefore introduce larger errors. Therefore, we need additional information to help us label more hotspots with higher accuracy when constructing the manifold.

4.2 SSID and Location Service Database as Additional Location Indicators

Fortunately, for any hotspot we discover through our smartphones, besides its BSSID and RSSI value, the SSID (service set identifier) of the hotspot is also available. An

SSID is a primary name associated with an 802.11 wireless local area network (WLAN) including home networks and public hotspots. Routers can broadcast the SSID to help wireless clients find the network, and client devices use this name to identify and join wireless networks.

Different from BSSID which can uniquely identify a hotspot, the same SSID name can be shared by different hotspots. The SSID is a case-sensitive text string that can be as long as 32 characters consisting of letters, numbers and/or symbols. Within those rules, the SSID can be anything. In most cases, each hotspot has a default SSID name which is stored by a broadband router or broadband modem, and this default SSID is normally set by the router manufacturers, such as Linksys, Singtel, DLink, etc. However, the SSID is allowed to be changed by administrators and most public hotspots rename their SSID which can better represent the owner and be easily identified by their users.

Although SSID of the hotspot may show a strong inference on the name of its owner, we still need another data source which can provide a location indicator based on the name of its owner. There have been several public POI (point of interests) databases which are crowdsourced from users or maintained by companies, such as *Foursquare* and *Google Place*. *Foursquare* [21] allows individual users to submit a discovery of new POI and check-in on existing POIs. *Google Places* is a service enabled by *Google Maps Platform*, which consists of over 100 million POIs and has frequent and in-time updates on the information and real-time location [30]. Both databases enable location queries by the name and region of POIs.

We assume that many public venues name their own hotspot with a relevant name to their services. To validate our assumptions, we collect several traces of hotspot labels in downtown areas and analyze the hotspot SSID and their location. For public hotspot with specific SSID and known location, we search their records inside *Google Place* database and *Foursquare* database to get the POI name.

Table 4.1: Hotspot SSID examples for public POIs.

SSID	POI Name	POI Category	POI Source
Guest@Truefitt&Hill	Truefitt & Hill	Salon/Fashion	FourSquare
myhelper	MyHelper Pte Ltd	Agency	Google Places
keckseng-wlan2	Keck Seng (s) Pte Ltd	Company	Google Places
brotzeit_2.4	Brotzeit	Food	FourSquare
chanhampe2 (5 ghz)	Chan Hampe Galleries	Art Gallery	FourSquare
iptv@south.african	South African High Commission	Government office	Google Places
www.homemart247.com (5g)	HomeMart	Home Services	FourSquare
smu_visitor	Singapore Management University	University	Google Places
leica-store	The Leica Store	Store	FourSquare
sunnyhills@raffles-2g	Sunnyhills	Confectionery	Google Places
fairmont_meeting	Fairmont Singapore	Hotel	FourSquare

Table 4.2: Sample results of tokenization.

SSID	Keyword Tokens	N gram
Guest@Truefitt&Hill	truefitt, hill	truefitt, hill ,truefitt hill
myhelper	myhelper	myhelper
keckseng-wlan2	keckseng	keckseng
brotzeit_2.4	brotzeit	brotzeit
chanhampe2 (5 ghz)	chanhampe	chanhampe
iptv@south.african	iptv, south, african	iptv, south, african, iptv south, south african, iptv south african
www.homemart247.com (5g)	homemart	homemart
smu_visitor	smu	smu
leica-store	leica, store	leica, store, leica store
sunnyhills@raffles-2g	sunnyhills, raffles	sunnyhills, raffles, sunnyhills raffles
fairmont_meeting	fairmont, meeting	fairmont, meetin, fairmont meeting

Table 4.1 shows some examples of hotspot SSID for public places. We find that *Foursquare* maintain more POIs relevant to consumer activities (such as restaurants, entertainment and fashions) while POI relevant to business and public services (such as government offices, hospitals and universities) are better maintained by *Google Places*. Generally, we find there is a similarity in the text between SSID name of hotspots and the name of the POIs that they belong to. Such similarity is sufficiently high which enable us a natural mapping solution: firstly mapping the SSID of the hotspot to the name of a POI and then retrieving location indicator of the POI as the location label for the hotspot from location service databases.

4.3 SSID-based Text Analysis

Given the strong correlation between SSID name and POI name, we develop a mapping mechanism between the public hotspot and POI based to locate the hotspot based on the problem formulation. The whole process involves 3 parts: pre-processing of SSID text, POI database query, and hotspot-POI matching.

4.3.1 Problem Formulation

We define any hotspot in the crowdsensed database as $\Theta = (n_\Theta, \mathbf{p}_\Theta, \boldsymbol{\tau}_\Theta)$, where n_Θ is the SSID of hotspot, $\mathbf{p}_\Theta$ are locations of all users that hear from θ and $\boldsymbol{\tau}_\Theta$ are corresponding RSSI value. The location of a hotspot Θ are denoted as $\mathbf{l}_\Theta$. We also define any POI venue in location service database as $\rho = (n_\rho, \mathbf{l}_\rho)$, where n_ρ is the name of the POI ρ and $\mathbf{l}_\rho$ is the location of ρ . For any pair of hotspot θ_i and POI ρ_j , we define the geodistance between them as $d(\Theta_i, \rho_j) = \text{geodistance}(\mathbf{l}_{\Theta_i}, \mathbf{l}_{\rho_j})$.

Given a set of n hotspots $H = \{\Theta_i : i = 1 \dots n\}$ and a set of m POIs $P = \{\rho_j : j = 0 \dots m\}$, where ρ_0 represent a default *NULL* POI that $d(\Theta_i, \rho_0)$ is set to a predefined cost σ for all $i = 1 \dots m$, for any $\Theta \in H$ there exist a mapping to $\rho \in P$ with the cost $d(\Theta_i, \rho_j)$,

our goal is to find the mapping for all elements in H to P such that the overall mapping cost is minimized.

4.3.2 Pre-processing on SSID Text

During the pre-processing stage, we firstly filter out those hotspots that do not belong to public services or contain no useful information that can identify the owner. As router brand is always used as the default SSID, we firstly assign hotspots with router brand names (such as Singtel, dLink, etc) to the default POI (ρ_0). Similarly, for personal hotspots and dashcams on cars, the brand name is always a part of the default SSID. By assigning these hotspots to default POI, we exclude these hotspots for further mapping process and make these hotspots unlabelled in the end.

We then tokenize the SSID text string by non-alphabet characters, so the generated tokens are all made up by alphabet characters. After tokenizing, we remove high-frequency words as they appear quite commonly in SSID as a convention and not useful in identifying a particular place. Such high-frequency words include but do not limit to: *wifi, free, guest, visitor, ghz, g, www, com, public*. The remaining tokens are kept in sequence as *keyword tokens* for a particular hotspot. Table 4.2 shows the keyword tokens for SSID in Table 4.1. As these keywords are used for querying POI databases, individual tokens may not best represent the SSID while querying all tokens may introduce extra noises or give too strong constraint. For example, for the SSID “*iptv@south_african*”, the keyword tokens are “iptv”, “south”, “african”. Obviously, “south african” is the most representative 2-gram within this SSID, while individual tokens such as “south” or “african” do not capture the right information, and entire token sequence will include “iptv” which is essentially a noise term. Therefore, based on the keyword tokens, we generate a set of N-grams which include all unigrams, bigram, up to N-gram from the keyword tokens, and make it as the potential query set to POI databases. Given the limited length of

SSID, we set $N = 3$ empirically to capture most of the relevant information. Example keyword N-grams generated from keyword tokens are shown in Table 4.2.

By the end of preprocessing, each hotspot will have several keyword N-grams and one N-gram may be shared by several hotspots since each place may have more than one hotspot with similar SSIDs.

4.3.3 POI Database Query

After obtaining keyword N-grams for each hotspot, we can query the POI database with the N-grams as search terms. Both *Foursquare* and *Google Place* support POI query with a search text and a specific region (center with a radius or simply “nearby” in real-time location service), and return a list of relevant POIs within the region. For each hotspot Θ , we submit queries with each N-gram as search text, the user location with the maximum RSSI for Θ as the location center and a default radius. Then we combine all returned POIs together as the candidate POI set P , and each POI will have properties like unique ID (for their corresponding database), name, geo-location (latitude and longitude) and category. For *Foursquare*, POI is also associated with user-related statistics, like total counts of check-ins and total counts of check-in users, and a boolean term “verified” which indicates whether the owner of this venue has claimed it and verified the information. We keep all these information for each POI and proceed to the hotspot-POI matching stage.

4.3.4 Hotspot-POI Matching

We further conduct a scoring mechanism for matching among all candidate POIs ρ for each hotspot Θ to determine the most suitable one. Given each returned POI ρ and its corresponding hotspot Θ (its SSID represented by n_Θ and all corresponding labels represented by $\{(\mathbf{p}_{\Theta i}, \boldsymbol{\tau}_{\Theta i})_{i=1,2,\dots}\}$), we compute an overall similarity score Φ between ρ

and Θ as:

$$\Phi = w_n \phi_n + w_l \phi_l + w_c \phi_c,$$

where the individual scores are defined as follows, and w_n , w_l , w_c are corresponding weights. There are totally 3 parts considered in the matching scheme:

- **Name similarity:** ϕ_n is defined by adding several string similarity metrics including Jaccard Similarity, Normalized Levenshtein Distance, JaroWinkler Distance, Long Common Subsequence, Cosine Similarity and N-gram Similarity [26], such that $\phi_n = \sum_i \alpha_i \phi_{ni}(n_\rho, n_\Theta)$ where each $\phi_{ni}(n_\rho, n_\Theta)$ indicates a kind of text similarity metric and α_i is the corresponding weight. We select these text similarity metrics because they cover both character-level similarity and term-level similarity [26], so the scoring mechanism will not be biased to a particular pattern or term.
- **Location closeness:** $\phi_l = -\text{corr}(\mathbf{d}, \boldsymbol{\tau})$ is calculated as a negative correlation between the distance sequence of labels to the estimated location of POI and the normalized signal strength sequence, where $\mathbf{d} = [d_1, d_2, \dots, d_n]$, $d_i = \text{geodistance}(\mathbf{p}_{\Theta i}, \mathbf{l}_\rho)$ and $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_n]$, $\tau_i = \exp(-\frac{(r_{\Theta i} - s_{\max})^2}{2\sigma^2})$. It is based on the assumption that the distance from a user location to a hotspot location should be inversely proportional to the received signal strength. The higher the negative correlation is, the closer the location of POI should be to the true location of hotspot.
- **Source credibility:** $\phi_c = \beta_d c_d + \beta_p c_p$ is a credibility score for the data source, where c_d represents a credibility score for the database and c_p is a popularity score for a POI. Generally, we assign POI from *Google Places* with the highest value of c_d since the location is maintained by *Google Maps Platform* and more accurate than *Foursquare*. Within POIs from *Foursquare*, we assign a higher c_d for a POI with

its “verified” property set to *TRUE* as the information is verified by the business. c_p is a term for POI from *Foursquare* only, which corresponds to the popularity of a POI. A POI with a higher number of check-in users enjoys a higher value of c_p , which is based on the assumption that a popular place will have a more accurate location contributed and corrected by more users. With the credibility term, our scoring mechanism tends to favor results from more reliable sources.

Among all the candidate places for a hotspot Θ , we select the most suitable place ρ^* with the highest overall score, and l_{ρ^*} will be assigned to Θ as a location indicator.

4.3.5 SSID-WOLoc: Extended Solution with SSID Analysis

As described in Chapter 3, WOLoc is a system based on hotspot crowdsensed database for Wifi-only outdoor localization. It comprised of 3 main components: pre-processing, manifold learning, and online location query processing, as shown in Figure 3.2. The previous method for hotspot labelling adopts a “cut-and-pin” methods as described in

Figure 4.1: SSID-WOLoc system architecture.

Section 4.1 and it is a part of manifold learning. With the SSID-based text analysis mechanism, we further extend the WOLoc system and introduce a new component of “Label Text Processor” which takes care of all relevant processing on SSID, as shown in Figure 4.1. The new component will be mainly in charge of two things: (1) filter out more mobile hotspots based on SSID; (2) assign location indicators to unknown hotspots based on SSID and location service databases.

4.3.5.1 Preprocessing: Mobile Hotspot Removal based on SSID

In Section 3.2.1, we introduce the pre-processing part of WOLoc which applies DBSCAN to identify mobile hotspots based on its heard location labels. Besides the mobile hotspots that can be identified based on their location labels, some hotspots are essentially mobile but may not be easily identified using locations if the carriers are static when the logs are collected, such as the dashcams on vehicles parked nearby or the personal hotspots on mobile phones from users who work nearby. The location labels of these hotspots may infer them as static hotspots, as they may be not moving during the data collection. We want to further eliminate these essentially mobile hotspots by analysing their SSIDs.

By observing the SSIDs in our data sample, we find that there are several typical patterns for personal hotspots enabled by personal mobile phones and hotspots enabled by dashcams on vehicles. Many personal hotspots have the user’s name and the phone’s brand name as a default SSID, such as “Alice’s iPhone 6” or “Ben’s Samsung Galaxy”. Similarly, we find that many dashcams share the same pattern which starts with a brand name and ends with a 4-digit or 6-digit random number and letters, such as “DR650GW-F0BF62” and “IROAD_AEV_077865”.

Based on the crowdsensed dataset, we extract the SSID and build up a brand corpus for smartphones and dashcams. During preprocessing, we use the “label text processor” to search over the SSIDs of all hotspots and check whether SSID contains these brand

names or match these patterns. Then we mark those with brand names or similar patterns as “mobile hotspots” and remove all the hotspot labels associated with them from the database to avoid involving potential mobile hotspots into the following computation.

4.3.5.2 Online Hotspot Labelling

As described in Section 4.3, SSID-based text analysis pipeline can locate hotspots through matching hotspot to POI in location service databases. To incorporate the technique into WOLoc, we need to consider several more things regarding efficiency and accuracy.

The first thing is to optimize queries to location service databases. As too many queries to web services may incur additional charges and increase network burden, we should try to minimize the total number of queries to location service databases. We observe that it is common that several hotspots have similar names for one place. For example, a restaurant may have different hotspots for employees and customers, and both contain the name of the restaurant but a different prefix or suffix. Therefore, instead of querying by text-location pair for each keyword n-grams for each hotspot, we can aggregate all hotspots with the same n-gram together to get a larger coverage for the query text. The coverage is defined by the bounding box of all heard user location labels of the hotspots with a particular n-gram text. In this way, only one query will be issued for each n-gram so that the overall number of queries can be heavily reduced. However, given a larger coverage, we may need more POI returned as candidates. The limit number of returned POI per response is another parameter that needs to be further monitored.

In some special cases where one place from the location service databases is associated with a large number of different hotspots, it is highly possible that the place covers a large region, which all hotspots belong to share the same SSID, like outdoor parks or universities. Location service databases normally only return one point inside the region as the location indicator for the POI, and it is obviously not appropriate to locate

all the relevant hotspots to the same location. Therefore, we set a threshold for the maximum number of hotspots that an n-gram can be associated with, and skip hotspot-POI matching of that particular n-gram.

Another practical issue to consider is the parameter setting for the weights defined in Section 4.3.4. The parameter can be learned through two different approaches. To consider the entire system as a unified solution, they can be empirically set through a grid search and cross-validation in terms of the overall localization performance of SSID-WOLoc. Alternatively, the parameters can be learned as weights for a series of features in a binary classification problem. As we do not have the ground truth for actual mapping between hotspot and POI, we can only label the mapping based on the best of our knowledge, and assume such mapping achieve the minimum distance cost. By defining each string similarity in name similarity, location closeness and source credibility as a feature set for any pair of POI ρ and hotspot Θ , we label the pair of POI and hotspot as positive if there is a match and negative for the remaining pairs. Then the problem can be trained as a binary classification problem. To avoid imbalance between classes caused by too many negative pairs, we sample the negative pairs by selecting those with higher rank order in the returned list of POIs.

In summary, although the “cut-and-pin” method does not have as high accuracy as the SSID text analysis method, it does not require any online query and will not suffer from potential large error due to wrong matches or inaccurate database information. However, SSID text analysis provides us with more references for localization, and generally improves the accuracy of localization by avoiding unexpected large errors due to imbalanced manifold labelling.

4.4 Experiments and Evaluations

As we do not have ground truth for hotspot-POI matching and the ground truth location for hotspots, we cannot evaluate the performance of SSID-based hotspot labelling independently. Therefore, in this section, we evaluate the component by examining the overall performance of SSID-WOLoc, and compare it with WOLoc to verify its effectiveness. Besides, we conduct several sensitivity tests on SSID-WOLoc as additional evaluation.

4.4.1 Experiment Setup

We use the same dataset as for evaluation of WOLoc. There are totally 6 cases involved, namely downtown, campus, hybrid residential area, residential block, community area, and downtown entertainment. Detailed description and statistics for the 6 cases can be found in Section 3.3.1 and Section 3.3.2.

We implemented the overall system of SSID-WOLoc which includes preprocessing with SSID analysis and SSID-based online hotspot labelling in Java on a PC with 16GB RAM. For hotspot-POI matching, we use *Foursquare* and *Google Place* as location service databases. All crowdsensed hotspot labels and the returned POIs are maintained in a PostgreSQL database.

The training and testing datasets used in the following evaluations are the same as WOLoc in Section 3.3. As we do not have ground truth label for hotspot-POI matching, we omit the implementation of machine learning approach for setting the weights. Instead, we split training set into 10 fold and use grid-search for setting weights parameters in hotspot-POI matching through cross-validation.

4.4.2 Performance Analysis on Individual Component of SSID-WOLoc

We verify the effectiveness of two main components of SSID-WOLoc: pre-processing with SSID and hotspot location labelling. The testing set contains 100 records for each case, the rest records are used as the training set to simulate crowdsensed databases.

4.4.2.1 Pre-Processing with SSID

The pre-processing module of SSID-WOLoc includes removing records with inaccurate GPS data and removing mobile hotspots by DBSCAN and SSID text analysis. To evaluate the effectiveness of the pre-processing module, we implement 2 versions for the SSID-WOLoc system: one without the pre-processing module and one with the pre-processing module. The same training set and testing set are used to test the two systems. Since the pre-processing are in the offline process, we omit the evaluation of online processing time for two systems.

Figure 4.2.a and Figure 4.2.b show a comparison between SSID-WOLoc without pre-processing and a full-version SSID-WOLoc. Results show that pre-processing improves the localization accuracy in all cases. There is no significant improvement in campus case. It is probably because that (1) testing areas in the campus are open and have no shelters or blocking, so GPS can work properly; (2) no vehicles are parked at the testing zone and few students are outdoor during the testing period, so there are few meaningless hotspots detected. As a result, the pre-processing module does not improve much for the results in the campus case. However, in other more crowded cases where GPS fails to work, the pre-processing module is proven to successfully remove inaccurate and irrelevant data from the database and finally improves the localization accuracy.

4.2.a: Mean error comparison.

4.2.b: Median error comparison.

Figure 4.2: Accuracy comparison between SSID-WOLoc without pre-processing and SSID-WOLoc with pre-processing in terms of median error and mean error.

4.4.2.2 SSID Text Analysis for Hotspot Localization

As presented in Section 4.3.5.2, we use different methods to label hotspot locations in WOLoc and SSID-WOLoc. WOLoc adopts the “Cut-and-Pin” method which uses an RSSI threshold to locate all hotspots to its nearest user location label, while SSID-WOLoc uses “SSID text analysis” to extract useful venue information from SSIDs of hotspots and label hotspots’ positions with the help of online venue database. To compare the performance of the two methods, we implement each method in two versions of systems and test them with the same queries to compare their performance. For “Cut-and-Pin” method, we set the $s_{\max\text{loc}}$ to -50dBm. For SSID text analysis, we connect it to both *FourSquare API* and *Google Places API* for POI queries. Both systems are implemented

4.3.a: Mean error comparison.

4.3.b: Median error comparison.

Figure 4.3: Accuracy comparison between “Cut-and-Pin” and “SSID text analysis” in terms of median error and mean error.

with pre-processing module with SSID analysis for a fair comparison. The same 100 queries selected earlier are issued to both systems.

Figure 4.3.a and Figure 4.3.b show a comparison between “Cut-and-Pin” and “SSID text analysis” in mean error and median error for different cases. It is observed that SSID text analysis significantly improves the mean error as it helps in reducing errors for extreme cases. The average localization errors can be bounded within 30 meters for 6 different cases. Except for the last 2 cases with a larger area, the first 4 cases have mean errors less than 10 meters. SSID text analysis helps to reduce the median error in all 6 cases, which is further validated by Figure 4.4. SSID text analysis not only constrains

Figure 4.4: CDF of error for “Cut-and-Pin” and “SSID text analysis”.

the error within a boundary but also further improves the accuracy for sufficient small errors, leading to an overall better performance.

Figure 4.5 shows a case that using SSID text analysis significantly improves the accuracy. Fixed hotspots by these 2 different methods are shown in red dots. Estimated positions of hotspots are shown as green dots. The yellow dot is estimated user location, while the cyan dot is ground truth location. Observe that SSID text analysis not only gives better estimation in user location, but also helps to locate the unknown hotspots inside the buildings instead of on the street.

Since SSID text analysis mainly happens during the offline training process, the online process only needs to query the local database to check whether a certain hotspot has been located based on SSID before. We compare the online query performance time for both methods in Figure 4.6. The CDF of processing time for both methods are almost the same, and both methods are able to process nearly 80% of queries within 200ms. A detailed comparison in mean processing time and median processing time, Figure 4.6.b, shows that SSID text analysis may result in slightly longer processing time, but given the better error control and higher localization accuracy, “SSID text analysis” method outperforms “Cut-and-Pin” method generally. Therefore, we suggest incorporating the SSID text analysis if the system allows online queries to third-party venue databases

4.5.a: Case with “Cut-and-Pin”

4.5.b: Case with “SSID text analysis”

Figure 4.5: Example cases on how SSID text analysis improves localization error for extreme cases. Yellow dots: the estimated location of users; Red dots: fixed hotspot locations; Green dots: estimated hotspot locations. [Best viewed in color.]

4.6.a: CDF comparison.

4.6.b: Mean and Median comparison.

Figure 4.6: Comparison between “Cut-and-Pin” and “SSID text analysis” in processing time of online queries.

during the offline training process.

4.4.3 Sensitivity Analysis

4.4.3.1 Training vs Testing Data Proportion

For the results presented earlier, the testing proportion within the total dataset is about 5% to 10% given each case has about 1500-2000 records. We evaluate the performance of the system by choosing different training/testing ratio. We firstly randomly select 5%, 10%, 20%, 50%, 60%, 70% 80%, 90% out of the entire dataset for each of cases (a)-(d), and use the remaining data as training data to build a global manifold. Then we test

on all the selected queries and report the mean and median error in Figure 4.7. Results show that the testing ratio has no significant impact on localization accuracy generally. The median error remains about 7 meters for a testing ratio below 60% and gradually increase with the testing ratio from 60%. Similarly, the mean error only shows an increase from 60%. It shows as long as the training data are evenly distributed within the zone, SSID-WOLoc does not rely on a high-volume of training data to achieve a satisfactory accuracy.

4.4.3.2 Number of Hotspots Per Query

We further analyze the effect of the number of hotspots involved in each query on localization accuracy. We collect all the testing results in Section 4.4.3.1 for all the testing ratio, and group them by the number of hotspots involved in each query. Figure 4.8.a shows the distribution for the number of hotspots involved per query. Over half of queries are with 20-60 hotspots per query. Figure 4.8.b shows localization error for different groups of queries with various numbers of hotspots. Queries with fewer than 10 hotspots suffer from a large mean and median error, which is because there is too limited information involved to infer an accurate location. For queries with larger than 10 hotspots, median error drops below 10 meters and the performance does not vary much with the increase in

Figure 4.7: Localization accuracy over different testing ratio.

the number of hotspots. It shows the performance of SSID-WOLoc is not very sensitive to the number of hotspots per query.

4.4.3.3 Sampling Frequency

To evaluate how sampling frequency will affect the result, we re-sample the collected data with a varying sample rate, i.e., only keep one record for every N records with $N = 1, 5, 10, 15$. We only conduct this evaluation on cases (a) - (d) as all data collected for these cases are at the same sampling rate, because data on cases (e)-(f) are sampled at unknown frequencies. Given the original sampling frequency is 1Hz, such re-sampling corresponds

4.8.a: Distribution for number of hotspots involved per query.

4.8.b: Localization error for queries with different number of hotspots.

Figure 4.8: Accuracy comparison between WOLoc without pre-processing and SSID-WOLoc with pre-processing in terms of median error and mean error.

4.9.a: Median errors for the first 4 areas under different sampling rates.

4.9.b: Mean and standard deviation of distance between two consecutive records for different sampling rates.

Figure 4.9: Performance analysis for different levels of hotspots label granularity.

to different sampling rate at 1/5Hz, 1/10Hz, 1/15Hz. This emulates a crowdsensing database at various granularity.

The median error at different sampling rate is shown in Figure 4.9.a, and the statistics on the distance between two consecutive records in down-sampled databases are reported in Figure 4.9.b. The median errors for $N \leq 10$ are all below 10 meters, and the increase in median error for $N = 15$ suggests that the WiFi labels may be too sparse for localization purposes. Re-sampling at $N = 10$ and $N = 15$ also helps us to simulate data from crowdsensed participants who contributed by driving a vehicle. When $N = 10$ or 15, the

average distance between every two consecutive records in the training sampling data is about 15 meters and 20 meters, respectively 54km/h and 72km/h when sampling at 1Hz, which is faster than normal driving speed in the city street and results in quite sparse crowdsensed data point. The results show that our system can also work on data collected by users when driving.

4.5 Summary

To reduce the potential error introduced by wrongly labelled hotspots in WOLoc, we propose a new method to locate hotspot in the manifold. By leveraging the existing location service database, we extend WOLoc solution to SSID-WOLOC which incorporated preprocessing and online hotspot labelling by analysis the text in SSID. We develop a scoring mechanism for matching hotspots to potential POIs in location service databases, and discuss the potential optimization for query management and parameter setting. We have conducted experiments in 6 metropolitan areas with the SSID-WOLOc system, and our results show that SSID-WOLOc can effectively manage the failure cases resulted from WOLoc and yields a much lower average error. By making hotspot matching components offline, SSID-WOLOc can process 80% of queries in 200ms and achieve a comparable efficiency with WOLoc, which makes it applicable as an online localization service. We also conducted several sensitivity tests on SSID-WOLOc to show its insensitivity to several hyperparameters, which validates the robustness of SSID-WOLOc.

Chapter 5

A Deep Profiling of GNSS-based Location Contexts

To achieve an integrated solution of outdoor localization, we have identified two main components in Chapter 1: (1) a proper supplementary source and (2) a trigger to switch between GPS and the supplementary source. Our previous work of WOLoc and SSID-WOLOc have identified WiFi hotspots in the metropolitan area as the supplementary sources and propose a series of approaches to achieve meter-level accuracy in outdoor scenarios with the help of crowdsensed database and location service database. We are left with the second problem of designing a proper trigger to switch between different data sources. In Section 1.2.2, we have identified the newly available GNSS data in Android as a starting point to explore GPS performance in various location contexts.

In this chapter, we will elaborate in more details on our approach of analysing the GNSS data. We will first discuss a preliminary case study on various urban scenarios, followed by a problem definition. We then demonstrate on our deep profiling on GNSS data and the two applications on the profiling. Lastly, we evaluate the effectiveness of our deep GNSS profiling by examining the performance of two applications.

5.1 From Understanding GNSS Data to Problem Definition

In order to get a better understanding of the problem and its solution feasibility, we have performed extensive studies on the GNSS data and their corresponding location contexts (including GPS localization performance). In this section, we will first present a few key observations, which motivates a concrete problem definition, as well as later solution methods.

5.1.1 Case Study of GPS-based Localization Performance

During our extensive survey in a city, we have collected the following data: i) raw GNSS measurements, ii) Android GPS location indicators, iii) ground truth location indicators (human labelled), and iv) Google Maps snapshots for various location scenarios.

We briefly report a few typical results in Figure 5.1, where we choose to put the overlay of ii) and iii) onto iv) in the first row, i) alone in the second row, and the errors distribution derived from ii) and iii) in the last row. As it is impossible to exhibit all GNSS data in one figure, we show the derived location patterns for visible satellites, as inspired by constellation-based astronomical localization [32]. The following are a few key observations on these results.

- **Satellite pattern matters:** a wide street 5.1.a enjoys a balanced satellite pattern and thus an almost perfect (for commercial GPS) average localization error of 5-10 meters, whereas semi-exposed indoor areas 5.1.b (glass-covered building or void deck) cause a biased pattern and larger variations in GPS errors. A balanced pattern obtained for sheltered street 5.1.c again leads consistent GPS performance, though the location scenario is quite close to indoor in common sense.

Figure 5.1: Preliminary studies on GNSS data. (a) outdoor long wide street, (b) inside glass building and on void deck, (c) sheltered street, (d) high-rise enclosed area, (e) building surrounded street, and (f) the same as (e) but at a different time. We put ground-truth and GPS traces on Google Maps snapshots in the first row, satellite distribution patterns (with black dot indicating the position of receiver) in the second row, and GPS error distributions in the last row. [Best viewed in colors.]

- Correlations between GNSS data and locations exist:** besides the discussions in the previous bullet, high-rise surrounded square or narrow street 5.1.d (scenarios often termed “building well” or “urban canyon”) lead to a more skewed pattern, as well as a more severe multipath effect (another GNSS field not shown in the figure). Suppose we may properly differentiate the location contexts 5.1.a to 5.1.d from the GNSS data (which GPS location indicators may fail to do), adequate corrections by other complementary signal sources can be put into action.
- Other GNSS features may further help:** differentiating 5.1.a from 5.1.d may sometimes require more features than the satellite pattern itself. In addition, even when the location remains roughly the same, 5.1.e and 5.1.f exhibit very different

GPS tracking performance during two distinct time periods, indicating the change of location context (not location). Apparently, the patterns (albeit being different) might not solely explain such a change; other GNSS features must have an impact here.

Essentially, our preliminary studies have demonstrated the necessity and benefit of profiling location contexts. Moreover, the importance of satellite pattern (and its independence of satellite IDs) may have suggested an effective way to re-organize the GNSS data so as to enable an effective mining.

5.1.2 Problem Definition

By now, we can clearly define the problem we aim to tackle in this paper. The major task is to profile location contexts based on raw GNSS measurements. In particular, we consider a large volume of GNSS data records collected by various GPS receivers (of capable smartphones), each *record* containing *samples* obtained within a time segment $\mathcal{T}$ but from one of $\mathcal{N}$ visible satellites and each sample includes $\mathcal{M}$ fields. Our goal is to first process and pack all the records of dimension $\mathcal{T} \times \mathcal{M} \times \mathcal{N}$ into valid feature map, then we identify a vectorized representation through unsupervised learning, targeting at compactly characterizing these seemingly sophisticated data structures.

The outcome of this profiling process should satisfy the following criteria to ensure its effectiveness:

- (i) It can be further derived and used to estimate an accuracy level of GPS localization performance.
- (ii) It has specific semantic meaning that can be further clustered or classified to represent various urban location contexts.

- (iii) Its dimension should be sufficiently reduced, so as to be computationally efficient for other applications.

Although the ultimate purpose of this profiling is to assist an integrated localization service, we would refrain from being over-ambitious in this preliminary study. Instead, we use the above requirements 1) and 2) to act as two immediate applications, in order to demonstrate the usefulness of our proposed GNSS profiling.

5.2 GNSS-based Location Context Profiling

In this section, we present a specially designed pipeline to transform raw GNSS measurements into compact data structures adapted to deep learning models, then we engineer an autoencoder model to perform location context profiling based on these data. We further evaluate the effectiveness of this profiling with a supervised task, i.e., localization accuracy estimation, and an unsupervised task, i.e., semantic analysis on location contexts.

5.2.1 Feature Engineering Pipeline

5.2.1.1 GNSS Data from Android 7.0

Previously, GPS modules handle all the tasks from receiving and processing signals to calculating pseudoranges and thus estimating locations; they only return estimated locations and corresponding accuracy level to front-end applications. Starting from Android 7.0, raw GNSS measurements become available to users via a number of devices [24]. Each sample of a GNSS measurement include 3 general types of fields: i) state indicators which describe clock synchronization, multipath effect and hardware clock continuity, ii) raw measurements, such as carrier frequency, cycles and phases, and iii) computed measurements, such as uncertainty of some raw measurements. A detailed list of fields can be found in [23]. By invoking the GNSS Raw Measurement API, a sample including

all these fields is returned from every visible satellite. In order to facilitate our learning purpose, we regroup them into two categories of features (part of them shown in Table 1.1) and also derive a few new features. We briefly explain them in the following:

- **Binary features** take only boolean values and they are related to the clock status, carrier phase validity, and multipath effect. Some of these features may not be directly applicable to the profiling, yet they indicate the validity of other features.
- **Number features** take float number values and mainly include measurements concerning information quality. For example, there are measures concerning SNR and Doppler shift, along with their respective uncertainty fields providing $1\text{-}\sigma$ values. Although these features are often used to infer localization accuracy given a model-based approach, we intend to use them in a non-parametric manner to avoid losing information.
- **Derived features** are not directly obtained from Android API; we compute them mainly to facilitate reshaping the input to deep learning. In particular, *SvPosition* is the position (under spherical coordinate system) of a satellite when a GNSS sample is received, derived using the GPS ephemeris [62] and the receiving timestamps.

5.2.1.2 Satellite Position based Feature Packing

As we have elaborated in Section 1.2.2, packing samples for all satellites into one compact and meaningful representation is highly non-trivial. Fortunately, the preliminary studies in Section 5.1.1 indicate that the satellite pattern might be a good way of indexing the complex data structure. Further inspired by the fact that most deep learning models take images (i.e., 2D pixel maps) as a common input, we design a pixel-position based packing mechanism. Essentially, we represent each feature of all visible satellites as a *snapshot*, i.e., an image with certain pixels representing the satellites and the pixels

taking their values from the features. Each snapshot is centered at the receiver’s location and the relative satellite positions derived from *SvPosition* are normalized to determine the corresponding pixel positions. As a result, each sample (for all satellites) contains $\mathcal{M}$ snapshots and a record has $\mathcal{T} \times \mathcal{M}$ of them. We could further pack the temporal dimension into snapshots, but that would require a much higher resolution and may also lose track of temporal correlation. We showcase a few snapshot samples in Figure 5.3.a.

To ensure the scale and trends of features are consistent, normalization and re-ranking are done before packing them into snapshots. The normalization phase transforms feature values within the range $[0, 1]$, and it should produce a “scattered” distribution for all features. In particular, most uncertainty features follow a log-normal distribution, so we convert them logarithmically to obtain a quasi-normal distribution. The re-ranking phase makes sure all the features at each pixel has a consistent trend in physical meaning. Since pixel values are in greyscale (i.e., non-negative integers) and zero value indicates no valid data presented, smaller values should correspond to higher uncertainty. Therefore all the uncertainty and ambiguity features are reversely ordered to maintain consistency.

5.2.2 Location Context Autoencoder

To learn a compact representation from the packed features (i.e., image sequences), we naturally choose an autoencoder for embedding [39], and the compact representations are extracted from the central layer. Sequence inputs are handled by incorporating recurrent neural network (RNN) as the fundamental structure [82]. Given a sequence of inputs and a desired output sequence, the goal of the RNN is to estimate the conditional probability of output sequence given the input sequence. Incorporating the two structures, we design our autoencoder as a sequence-to-sequence structure with images as input. Both the encoder and the decoder are implemented using gated recurrent unit (GRU) cells. As shown in Figure 5.2, given the snapshots from time 0 to $\mathcal{T}$, denoted as

Figure 5.2: Autoencoder structure. [Best viewed in colors.]

$\mathbf{x} = (x_0, \dots, x_\tau, \dots, x_T)$, where $x_\tau \in \mathbb{R}^{n_i \times n_i \times \mathcal{M}}$, we want to get a compact representation $\mathbf{v} \in \mathbb{R}^\ell$ (shown as green in Figure 5.2). The encoder and decoder are defined as transitions Θ and Φ . Here, $\Theta : \mathbf{x} \rightarrow \mathbf{v}$, $\Phi : \mathbf{v} \rightarrow \mathbf{x}'$, where $\mathbf{x}'$ denotes the reconstructed sequence of $\mathbf{x}$ with the same dimension. The training is conducted with back-propagation algorithm to find transition Θ^* and Φ^* based on the following objective:

$$\{\Theta^*, \Phi^*\} = \arg \min_{\Theta, \Phi} \sum_{\tau=0}^T \|x_\tau - x'_\tau\|_2^2$$

where n_i is the size of a snapshot, $\mathcal{M}$ is number of GNSS features, ℓ is the latent dimension of GRU.

Encoder first applies a convolution layer with filter size (m_f, m_f, ℓ) to each x_t and flattens it into a vector as input for GRU cell. At the decoder side, the context vector $\mathbf{v}$ is set as the hidden state, the output generated by RNN is reshaped and applied a deconvolutional layer with the same filter size as that applied at the encoder side to obtain

5.3.a: Snapshots at $\tau = 1, 2, 3$.

5.3.b: Representation $\mathbf{v}$.

Figure 5.3: Snapshots and Corresponding representations. [Best viewed in colors.]

the final output x'_t . Parameters setting for m_f , ℓ and $\mathcal{T}$ will be studied in Section 5.3.2.1. Figure 5.3 shows examples of snapshots and generated representations, based on the 6 scenarios in Figure 5.1. Due to space limit, we only visualize the first 3 steps in a sequence of length 5, and select 3 features for RGB channels of an image.

5.2.3 Applications of Location Context Profiling

5.2.3.1 Localization Error Estimator

As shown in Section 5.1.1, different contexts will result in various error distributions. So the profiling vector $\mathcal{V}$ can be used to estimate localization performance. Such information can yield a confidence level of GPS localization similar to the “blue disk” provided by Google Maps. At any time t , given the representation $\mathbf{v}_t$ generated from GNSS measurements during previous $\mathcal{T}$ time steps, an error estimation function Λ and true GPS error e_t , and the predicted error $\hat{e}_t = \Lambda(\mathbf{v}_t)$, we aim to find $\Lambda^* = \arg \min_{\Lambda} \|e_t - \hat{e}_t\|_2^2$. Using this objective as the loss function, we design a 4-layer feed-forward deep neural network structure with a single neuron as last layer as our error estimator model. We denote it by DNN-R hereafter.

We also implement two other deep learning structures, namely DNN-C and RNN, to compare with DNN-R. DNN-C converts the regression problem into a classification problem, in which we adopt a classification layer before the regression neuron layer, and make the final layer as an expected value based on class distribution. RNN implements GRU cells to include estimation in previous steps; it adapts to the filtering effect inside GPS that results in relatively smooth changes in location estimations and corresponding errors. Detailed settings are presented in Section 5.3.3.

5.2.3.2 Context Semantic Analysis

Our studies in Section 5.1.1 also suggest that GNSS measurements can represent the context of environments, which can be used to trigger sensor fusion for better localization.

We aim to create a context database containing labelled training representations and allow a new representation to query for a context label. Since our representation is learned from records, a semantic label should be assigned to a trip segment rather than a spot sample. However, manually labelling representations can be meaningless given the great variety in human perception, so we adopt a bottom-up clustering mechanism to build up the labelled database. Each training representation has 2-level of labels.

5.4.a: Detailed context for trace 1.

5.4.b: Detailed context for trace 2.

5.4.c: General Context.

Figure 5.4: Context semantic analysis. [Best viewed in colors.]

- **Lower-level clustering:** For a set of representations, we use DBSCAN to cluster them into small clusters, assign each cluster with a predefined label as the *detailed context* and calculate the center of the cluster. Detected noises are just labelled as “noise”. Lower level clustering captures information of scenario such as “on street”, “blocked by 1-side building”, “building enclosing”, and “on void deck”. Figure 5.4.a and Figure 5.4.b demonstrate two traces after lower-level clustering.
- **Higher-level clustering:** Cluster centers and “noises” from lower-level clustering are further combined into the upper level of clusters, and assigned a unified label as *general context*. Higher-level clustering captures information of satellite patterns in general. The noise vectors detected from the upper level of clustering are treated as a single cluster since it may capture an infrequent satellite pattern at the moment, but can be further clustered into other groups when the database is enlarged. To illustrate the correspondence with lower-level clustering, Figure 5.4.c shows the representations with the color label of upper-level clusters.

At query stage, each representation will be firstly matched to the nearest upper-level clusters to determine its general context, and then compared with all representations inside the cluster to find its k nearest neighbours and assign the final label by majority voting. We set a threshold for finding the kNN that queries with the smallest distance larger than the threshold will not be assigned with any label since we expect it as a new context representation that is not included in the training database.

Remark: All the training procedures presented in Section 5.1 are performed offline, making the online computations (basic algebraic operations) in smartphones very lightweight.

5.3 Evaluation

In this section, we firstly provide details on our system implementation, data preparation, and model parameter settings, then we evaluate the effectiveness of context profiling by examining the performance of the two derived applications.

5.3.1 System Implementation and Data Preparations

We develop applications in Android with Java, using 4 smartphones for data collection: Huawei P10, P10 Plus, Mate 9, and Samsung S8. The entire data processing and model training pipeline is implemented in Python, and set up on a PC with 2.6 GHz Intel Core i7 CPU and a 16 GB 1600 MHz DDR3 memory.

5.3.1.1 Data Collection

Our Android application collects raw GNSS measurements and locations returned by GPS. The GPS location indicator has 2 sources in Android: AndroidLoc obtained through Android LocationManager API [27] and GoogleLoc provided by Google FusedLocationProviderApi [28]. We record location and its accuracy (associated with Location class) for both sources at a frequency of 1 Hz. The ground truth is manually labelled by marking critical points along the way, and the locations between the critical points are linearly interpolated by time assuming users moving at a constant speed. We collect 30 traces at different times covering about 9k typical location points in the urban area, roughly summarized into the following 9 scenarios:

On street	On void deck
Blocked by 1-side buildings	Blocked by 2-side buildings
Building enclosing	Inside glass-covered building
On sheltered street	In tunnel
Inside room with heavy walls	

We randomly select 20 traces for the training purpose and make the remaining traces as testing traces, resulting in a training dataset with about 6600 location points and a testing dataset with about 2300 location points.

5.3.1.2 Data Preprocessing

We match the corresponding raw GNSS measurements, GPS locations, accuracy (from two sources), and ground truth locations by their timestamps. We further calculate the true GPS localization errors as the great circle distance between GPS locations and their respective ground truth locations. As explained in Section 5.2.1.2, we use SvPosition as index to pack features into snapshot. We set the snapshot (image) resolution as 18×18 . The following 18 features are chosen; their details can be found in [23]:

ADR.STATE_CYCLE_SLIP	HardwareClockDiscontinuityCount
ADR.STATE_RESET	BiasUncertaintyNanos
ADR.STATE_VALID	ReceivedSvTimeUncertaintyNanos
STATE.SYMBOL_SYNC	Cn0DbHz
STATE.MSEC_AMBIGUOUS	PseudorangeRateMetersPerSecond
STATE.TOW_DECODED	PseudorangeRateUncertaintyMetersPerSecond
STATE.SUBFRAME_SYNC	AccumulatedDeltaRangeUncertaintyMeters
STATE.BIT_SYNC	DeltaRange
STATE.CODE_LOCK	MultipathIndicator

5.3.2 Autoencoder Model Training

5.3.2.1 Parameter Setting

As presented in Section 5.2.2, there are 3 groups of hyperparameters to be determined: (1) input sequence steps for sequence-to-sequence model; (2) latent dimension for RNN GRU cells; (3) filter size for convolutional layer. For time steps, we set it to 5 thus the model includes measurements from past 5 seconds. We set such a relatively small number as we tend to make the context relatively stable within the period so that the resulted representation can better characterize a stable context. As the loss for our model is the

Figure 5.5: Autoencoder parameter vs reconstruction error.

reconstruction error, we evaluate the effect of these parameters through reconstruction error as shown in Figure 5.5. The latent dimension directly affects the dimension of our representation. There is a trade-off in determining the kernel number. On one hand, a larger latent dimension allows representations to encode more information. On the other hand, a larger latent dimension will bring the extra cost for storage, model training and application at a later stage. To achieve a balance, we select 60 when the error starts to drop down slowly. For convolutional filter size, we set it as (12,12) so that each patch can capture sufficient image while achieving an acceptable reconstruction performance. We leverage Adam Optimizer and ReLU activation function during the training procedure.

5.3.2.2 Comparison with Other Embedding Mechanism

As a comparison, we also build a Convolutional Restricted Boltzmann Machine (CRBM) as presented in [53]. To incorporate the same structure, we treat satellites identifier as their dimensions (D) and various features as the channel (C). Since the feature packing is different, we omit the comparison in reconstruction error and compare the effectiveness of representation through application in Section 5.2.3.

5.3.3 Localization Error Estimator

To evaluate the effectiveness of representation, we firstly examine its performance in estimating localization error. We build 3 types of location error estimators as described in Section 5.2.3.1: DNN-R, DNN-C, RNN. DNN-R consists of 4 layers with 32, 16, 8, 1 neurons respectively and ReLU as the activation function. Loss function is the mean squared error between regression result and true GPS error. DNN-C has the similar structure as DNN-R with 4 layers, and the layer with 8 neurons has an additional activation layer of soft-max which aims to output the error range classification for 8 classes. The last layer neuron is an expected value for all error bin. The errors are divided into 8 bins, of which the bin size is 8 meters. RNN shares a similar structure as DNN-R except the first 3 layers are recurrent neural layers with GRU cells.

We train the 3 models with representations generated from our autoencoder with the training traces, and test with those generated with testing traces. To compare with CRBM embedding shown in Section 5.3.2.2, we conduct the same procedure with the CRBM embedding model and report the results. The mean and median error for all testing data is shown in Figure 5.6. DNN-R perform best out of 3 models, which can achieve a median difference of 4.16 meters and a mean difference of 5.96 meters with the true errors. Our model generally performs better than CRBM except for RNN, both

5.6.a: Mean error.

5.6.b: Median error.

Figure 5.6: Deviation from true localization error.

5.7.a: Error estimation for trace 1.

5.7.b: Error estimation for trace 2.

Figure 5.7: Trace samples with error estimation. [Best viewed in colors.]

models perform worst as the model tends to over-fit the sequence pattern. To show the change in error for a trace, 2 sample traces are also shown in Figure 5.7. Both sequences generally capture the error trends, though the magnitude is not perfectly matched.

Android system does not provide an absolute estimation on localization error, instead it provides an accuracy indicator as the radius of 68% confidence. To compare our estimation with this accuracy range, we transform our estimated error into an accuracy indicator by getting the 68% percentile over error distribution within past 10 seconds. To evaluate the effectiveness of accuracy indicator, we compute the probability of error falling within the accuracy range. A probability close to 68% implies an adequate accuracy indicator for localization performance. Larger than 68% implies an over-conservative estimation, while smaller than 68% implies an over-confident estimation. Figure 5.8 shows a comparison between the accuracy indicator derived from our error estimator with the one from Android [27] and Google [28]. We measure the probability with sliding

window of different window sizes. Our model achieves almost around 68%, while the other two indicators always give over-confident estimation. A relatively stable median and mean for all models show that the estimators perform in a consistent manner.

5.3.4 Location Contexts Semantic Analysis

To test the effectiveness of representation on characterizing different contexts, we firstly build up a context database by applying DBSCAN for the bottom-level clustering on all training trips. For 20 training trips, we finally get 86 lower-level clusters and 7 upper-level clusters and 14 small clusters remained as noise at the upper level given their unique satellite patterns. Then we manually assign predefined labels as described in Section 5.3.1.1 to all the representations in each cluster. For noise in lower-level, we label them as “Noise”.

At the query phase, we conduct the top-down search and assign each representation with the same detailed context label of its k nearest neighbors using cosine distance as the metric, if the nearest distance is larger than our defined threshold, we record it as a fail case. We manually examine the assigned label with its corresponding location for validation. As shown in Figure 5.9.b, we first evaluate the success rate given different

5.8.a: Mean probability of error falling within the confidence range at different time resolutions.

5.8.b: Median probability of error falling within the confidence range at different time resolutions.

Figure 5.8: Performance of localization accuracy indicator.

5.9.a: Success rate vs threshold.

5.9.b: Accuracy vs k for kNN search.

Figure 5.9: Clustering performance with various threshold and k.

thresholds. The success rate of queries increases when threshold increases, and a threshold of 0.5 can cover 98.6% of queries. We then evaluate classification accuracy under the threshold of 0.5 for different k values, and achieve the highest classification accuracy of 91.62% when $k = 20$. All k value gives an average performance about 90%.

To better showcase the performance in context semantic analysis, we show the result of two testing traces with a snapshot of Google Map in Figure 5.10. Most of the location points are correctly labelled except some noise points and a mismatch between “on void deck” and “on sheltered street” in both traces given that two scenarios share quite a similar context and hard to be distinguished.

5.3.5 Energy Consumption

To evaluate the energy consumption of GNSS raw measurement collection, we make two Apps: one only collects GNSS raw measurement data, the other one collects magnetic field data, cellular data and light sensor data as a comparison. We measure the battery life when running each App with the screen set to the minimum brightness. As shown in Table 5.1, the energy consumption is comparable between the two sets of sensors, so our method is energy efficient while achieving more functionality compared with existing IO detection proposals [108, 73].

Table 5.1: Battery consumption comparison for different sensor set.

Sensor Set	Battery Life (Hours)
Magnetometer + Cellular + Light	29.2
GNSS Raw Measurement Listener	23.3

5.4 Summary

In this work, we have gone beyond well-studied indoor-outdoor detection and proposed to holistically profile the various location contexts in urban area. Based on extensive studies on raw GNSS measurements recently made available by Android 7, we have innovated in a new method to organize this unconventional data structure so that effective mining techniques can be applied. We have then engineered an autoencoder module to extract compact representations out of GNSS traces. In order to demonstrate the efficacy of this context profiling, we have showcased two applications: a localization error estimator

5.10.a: Label result for sample trace 1.

5.10.b: Label result for sample trace 2.

Figure 5.10: Trace samples with context label results. [Best viewed in colors.]

better than that provided by Google Maps, and a context semantic database potential extensible to assist other applications, including the seamless integration of indoor and outdoor localization schemes.

Chapter 6

Conclusions and Future Work

In this chapter, we first summarize the contributions of this dissertation, and then point out several promising directions for future work.

6.1 Summary of Contributions

Localization with embedded system is a fundamental task for many applications enabled by smartphones. Existing academic proposals focus more on embedded sensing in indoor localization, while outdoor localization is still mostly relying on GPS. Although GPS is performing well in an open air environment, it often fails to achieve satisfying performance in urban scenarios due to signal loss or blocking. Therefore, in this dissertation, we aim to exploit an integrated solution of outdoor localization based on GPS and other supplementary sources to eliminate the potential failure of GPS in urban areas. We identify two main problems to achieve the integrated solution, which are firstly to identify a proper supplementary source and secondly to design a proper trigger between GPS and the supplementary source.

For the first problem, WiFi has been widely used in indoor scenarios as a source for localization, and existing WiFi infrastructure in urban scenarios enables us to choose it as the supplementary source. Existing crowdsensed WiFi databases also motivate us to develop a data-driven approach. However, existing solutions on WiFi mainly focus

on the relationship between users and hotspot (RSSI-based modelling) when the location of hotspots are known, or between user and user (fingerprinting) when the location of hotspots are unknown. Industry practices mostly adopt a two-stage solution based on the weighted centroid method, which is over-simplified and may suffer from error accumulation.

In Chapter 3, we present WOLoc as a WiFi-only outdoor localization system that relies solely on crowdsensed hotspot labels. We apply a semi-supervised manifold learning technique to consider the relationship between users and hotspots, among users and among hotspots. We design a full version solution from pre-processing to query processing based on the crowdsensed hotspot labels. Extensive experiments in 6 metropolitan areas show that WOLoc can achieve meter-level accuracy for most cases and outperform 3 systems currently available in the market. The processing time for each query is about 200ms and make it applicable to answer realtime location queries.

In Chapter 4, we further discuss the performance of WOLoc by examining its potential failure when hotspot manifold is wrongly or merely labelled. To avoid potential errors introduced by an insufficient number of labelled hotspots, we develop an SSID-based text analysis technique which processes SSID of hotspots and maps them to the relevant POIs in location service databases. We incorporate the component of SSID analysis into existing WOLoc solution to get the unified solution, namely SSID-WOLOc. We conduct experiments on the same dataset as WOLoc. Results show that SSID-WOLOc successfully avoids extreme failure cases in WOLoc, and achieves a much lower mean localization error. SSID-WOLOc process the SSID analysis mostly offline and will not affect its efficiency for online query processing. We also evaluate the SSID-WOLOc with several sensitivity tests to validate its robustness across different parameter settings.

For the second problem, existing solutions mainly focus on a clear-cut between indoor and outdoor, and enable WiFi for indoor scenarios and use GPS for outdoor scenarios,

or introduce a transition state namely semi-outdoor, which is always not clearly defined. However, in current urban areas, there are no perfect outdoor scenarios anymore. A trigger simply differentiates indoor and outdoor is not enough any more. A more comprehensive profiling which can differentiate more cases in outdoor scenarios is desired. Recently available raw GNSS measurement data in Android gives us far more information compared with traditional location indicators returned by GPS, which motivates us to design a profiling approach to distinguish different urban scenarios.

In Chapter 5, we go beyond the well-studied indoor-outdoor detection problem and propose to holistically profile the various location contexts in urban areas. We have innovated a new method to organize the raw GNSS measurements, and engineered an autoencoder module to extract compact representations from the GNSS measurement traces. We also design 2 applications to demonstrate the potential use of the GNSS profiling. One is to estimate GPS localization error which can evaluate the performance of GPS, the other one is a context semantic database potentially extensible to assist other applications. Evaluations on these 2 applications demonstrate the effectiveness of our GNSS profiling and its potential usages in other applications.

To sum up, in this dissertation, we contribute a series of novel approaches to achieve an integrated solution of outdoor localization for urban scenarios. We achieve this goal from two different perspectives: 1) for 2 components for the solution namely source and trigger, we develop a robust WiFi-based outdoor localization methods suitable for metropolitan areas, and a GNSS-based context profiling to serve as a potential trigger; 2) for data and techniques, we explore various types of data, including WiFi signal data, SSID text label data and GNSS raw measurement data, and various data-driven approaches, including manifold-learning, rule-based approaches, and deep learning techniques.

6.2 Future Work

There are multiple potential directions for future works. In this dissertation, we present 4 potential and promising directions:

- **Manifold-based Localization with Transfer Learning:** Although WOLoc can provide a satisfactory accuracy in many cases, the performance highly relies on the amount of location-labelled WiFi data in the queried area. Even though we have used SSID-based hotspot labelling to increase the amount of labelled data, it may not help much for areas with fewer popular POIs. In addition, voluntary crowdsensing projects (e.g., [65]) always result in unevenly collected WiFi data since unpopular areas attract fewer voluntary contributors. Finally, large-scale sensing experiments collect WiFi data without location labels [97, 7], and WiFi-based localization services [79, 31] receive users' queries that are all unlabelled data. Consequently, crowdsensing actually provides us with very unevenly distributed WiFi data: abundant and labelled in some areas, while partially labelled or even sparsely collected in some others. Thus, how to transfer the knowledge of densely labelled areas to sparsely labelled areas is a critical question in order to scale our outdoor localization solution. Transfer learning [pan2010survey] has been widely used in various domain, such as NLP [16], recommendation [49] and sentiment analysis [8]. In the localization domain, transfer learning also has been applied to improve the performance [56, 80, 107]. Considering transfer over spaces, [70] and [95] have considered transfers with two overlapped areas where at least some hotspots from densely labelled areas can be observed in sparsely labelled areas. Transfer learning between two non-overlapped areas become a promising direction to explore.

- **POI-Hotspot Matching with Deep Learning and Graph Matching:** Although our existing SSID-based POI-hotspot matching have successfully reduced the mean localization errors, the methods still suffer from two limitations. Firstly, the method requires a heavy process of feature engineering. We design the 3 types of similarities which can best represent the relationship between two entities. However, it is the best inference based on human knowledge. If more matching ground truth and data are available, deep learning techniques can be applied to extract more useful information from raw data, thus to achieve higher accuracy. Secondly, current matching is on a one-to-one basis, which does not consider surrounding hotspots when mapping one particular hotspot to a POI. However, the surrounding context can provide more constraints to the mapping, thus may achieve better mapping results. Therefore, a many-to-many mapping can be further exploited, which maps a group of hotspots to a group of POIs by considering the relationships among entities. Such mapping can be achieved by applying methods in graph matching [22].
- **GNSS Deep Profiling for Multi-constellation with More Measurement Data:** Our existing solution for profiling GNSS data is based on single constellations (GPS). However, there has been more constellations in operations, including GLONASS, BeiDou and Galileo which are all accessible from Android 7.0. As multi-constellation based localization and even dual-frequency localization is a general trend in GNSS-based localization, incorporating more measurements from more sources for a better profiling seems promising. However, more constellations essentially provide more information but potentially introduce more noises. How to pack raw measurements from various constellations while preserving the most useful information is an open question to explore. Additionally, the features used in the current solution are all uncertainty terms, but the absolute measurements

relevant to pseudorange and clock are not fully used yet. Absolute measurement values are not easily normalized to the same scale for different satellites, therefore, a more suitable packing mechanism which captures both the absolute spatial and temporal features and relative relations among satellites is desired.

- **Full-fledged Urban Localization Service** Our current work on GNSS deep profiling is still rather preliminary at this stage, mostly due to the fact that our GNSS data are only gathered by a few users. Therefore, the work aims mainly to raise the attention of our community: if we gather such data in a more pervasive manner (e.g., crowdsensing incentivized by cloud offloading [54]), we will be in a much better position towards an integrated outdoor localization solution. A preliminary solution of an integrated outdoor service can be a binary switch trigger from GPS to WiFi when GPS localization error is larger than a threshold, then WiFi-based localization should be able to improve the localization performance as expected. However, given the complexity of the urban scenarios and different levels of GPS performance in terms of localization errors and location contexts, it remains open that whether we should incorporating such a component in the unified solution as a trigger between different approaches or as a control for fusing different sources. It requires more experiments to get a robust and efficient full-fledged urban localization services, which could be another future direction worthy to be investigated in.

List of Publications

- **Jin Wang**, Jun Luo, Sinno Jialin Pan, and Aixin Sun. Learning-Based Outdoor Localization Exploiting Crowd-Labeled WiFi Hotspots. *IEEE Transactions on Mobile Computing (TMC)*, 2018.
- **Jin Wang**, Qingmei Zhao, Chen Wang. DeepTravel: Online transport mode identification based on low-rate sampling sensors through deep neural network. In *Proceedings of 4th International Conference on Systems and Informatics (ICSAI'17)*, 2017.
- **Jin Wang**, Nicholas Tan, Jun Luo, and Sinno Jialin Pan. WOLoc: WiFi-only outdoor localization using crowdsensed hotspot labels. In *Proceedings of the 36th IEEE Conference on Computer Communications (INFOCOM'17)*, 2017.
- Hui Liang, **Jin Wang**, Qian Sun, Yong-Jin Liu, Junsong Yuan, Jun Luo, Ying He. Barehanded music: real-time hand interaction for virtual piano. *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (i3D'16)*, 2016. (Equal contribution for first 2 authors).

References

- [1] A Alhasanat, B Sharif, Charalampos Tsimenidis, and J Neasham. Efficient rss-based collaborative localisation in wireless sensor networks. *International Journal of Sensor Networks*, 22(1):27–36, 2016.
- [2] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Proc. of the 15th ACM MobiCom*, pages 261–272, 2009.
- [3] P. Bahl and V.N. Padmanabhan. RADAR: an In-building RF-based User Location and Tracking System. In *Proc. of the 19th IEEE INFOCOM*, pages 775–784, 2000.
- [4] P. Bahl and V.N. Padmanabhan. RADAR: an In-building RF-based User Location and Tracking System. In *Proc. of the 19th IEEE INFOCOM*, pages 775–784, 2000.
- [5] Marco V Barbera, Alessandro Epasto, Alessandro Mei, Vasile C Perta, and Julinda Stefa. Signals from the crowd: uncovering social relationships through smartphone probes. In *Proc. of the 2013 Conference on Internet Measurement*, pages 265–276. ACM, 2013.
- [6] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, 2003.
- [7] Gergely Biczok, Santiago Díez Martínez, Thomas Jelle, and John Krogstie. Navigating mazemap: indoor human mobility, spatio-logical ties and future potential. In *Proc. of IEEE PerCom 2014*, pages 266–271. IEEE, 2014.
- [8] John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

- [9] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less Low-Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications*, 7(5):28–34, 2000.
- [10] Chih-Wei Chang, Yao-Chung Fan, Kuo-Chen Wu, and Arbee LP Chen. On the semantic annotation of daily places: A machine-learning approach. In *Proc. of the 4th International Workshop on Location and the Web*, pages 3–8. ACM, 2014.
- [11] K. Chen and G. Tan. SatProbe: Low-Energy and Fast Indoor/Outdoor Detection based on Raw GPS Processing. In *Proc. of the 36th IEEE INFOCOM*, pages 1–9, 2017.
- [12] Yin Chen, Dimitrios Lymberopoulos, Jie Liu, and Bodhi Priyantha. FM-based Indoor Localization. In *Proc. of the 10th ACM MobiSys*, pages 169–182, 2012.
- [13] Long Cheng, Chengdong Wu, Yunzhou Zhang, Hao Wu, Mengxin Li, and Carsten Maple. *International Journal of Distributed Sensor Networks*, 2012, 2012.
- [14] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm. Accuracy Characterization for Metropolitan-scale Wi-Fi Localization. In *Proc. of the 3rd ACM MobiSys*, pages 233–245, 2005.
- [15] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. Indoor Localization Without the Pain. In *Proc. of the 16th ACM MobiCom*, pages 173–184, 2010.
- [16] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of the 25th ICML*, pages 160–167. ACM, 2008.
- [17] Ionut Constandache, Romit Roy Choudhury, and Injong Rhee. Towards mobile phone localization without war-driving. In *Proc. of the 29th IEEE INFOCOM*, pages 1–9. IEEE, 2010.
- [18] Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. Linking wireless devices using information contained in wi-fi probe requests. *Pervasive and Mobile Computing*, 11:56–69, 2014.

- [19] Yao-Chung Fan, Chih-Wei Chang, Wang-Chien Lee, Kuo-Chen Wu, and Arbee LP Chen. On the semantic annotation of wi-fi ssid logs in mobile applications. *Pervasive and Mobile Computing*, 43:131–145, 2018.
- [20] Yao-Chung Fan, Yu-Chi Chen, Kuan-Chieh Tung, Kuo-Chen Wu, and Arbee LP Chen. A framework for enabling user preference profiling through wi-fi logs. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):592–603, 2016.
- [21] FourSquare. FourSquare - About Us. <https://foursquare.com/about>. Accessed: 2016-05-18.
- [22] Brian Gallagher. Matching structure and semantics: A survey on graph-based pattern matching. *AAAI FS*, 6:45–53, 2006.
- [23] GNSS-Android. Android Developers Documentations. <https://developer.android.com/reference/android/location/GnssMeasurement>. Accessed: 2018-07-18.
- [24] GNSS-Android. Android Raw GNSS Measurements. <https://developer.android.com/guide/topics/sensors/gnss>. Accessed: 2018-07-18.
- [25] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.
- [26] Wael H Gomaa and Aly A Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.
- [27] Google. Android Developers Documentations: LocationManager. <https://developer.android.com/reference/android/location/LocationManager>. Accessed: 2018-07-18.
- [28] Google. Google APIs for Android: FusedLocationProviderApi. <https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi>. Accessed: 2018-07-18.
- [29] Google. Google Maps Mobile. <https://www.google.com/mobile/maps/>. Accessed: 2016-05-18.

- [30] Google. Google Maps Platform: Places. <https://cloud.google.com/maps-platform/places/>. Accessed: 2018-08-04.
- [31] Google. The Google Maps Geolocation API. <https://developers.google.com/maps/documentation/geolocation/intro>. Accessed: 2016-05-18.
- [32] E.J. Groth. A Pattern-Matching Algorithm for Two-Dimensional Coordinate Lists. *The Astronomical Journal*, 91(5):1244–1248, 1986.
- [33] Xiaonan Guo, Eddie CL Chan, Ce Liu, Kaishun Wu, Siyuan Liu, and Lionel M Ni. Shopprofiler: Profiling shops with crowdsourcing data. In *Proc. of the 33rd IEEE INFOCOM*, pages 1240–1248. IEEE, 2014.
- [34] Jihun Ham, Daniel D Lee, and Lawrence K Saul. Semisupervised alignment of manifolds. In *AISTATS*, pages 120–127, 2005.
- [35] H. Hassanieh, F. Adib, D. Katabi, and P. Indyk. Faster GPS via the Sparse Fourier Transform. In *Proc. of the 18th ACM MobiCom*, pages 353–364, 2012.
- [36] Tian He, Chengdu Huang, Brian M Blum, John A Stankovic, and Tarek Abdelzaher. Range-Free Localization Schemes for Large Scale Sensor Networks. In *Proc. of the 9th ACM MobiCom*, pages 81–95, 2003.
- [37] Will Hedgecock, Miklos Maroti, Akos Ledeczi, Peter Volgyesi, and Rueben Banalagay. Accurate real-time relative localization using single-frequency gps. In *Proc. of the 12th ACM SenSys*, pages 206–220. ACM, 2014.
- [38] Will Hedgecock, Miklos Maroti, Janos Sallai, Peter Volgyesi, and Akos Ledeczi. High-accuracy differential tracking of low-cost gps receivers. In *Proc. of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, pages 221–234. ACM, 2013.
- [39] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *AAAS Science*, 313(5786):504–507, 2006.
- [40] Tâm Huynh, Mario Fritz, and Bernt Schiele. Discovery of Activity Patterns Using Topic Models. In *Proc. of the 10th ACM UbiComp*, pages 10–19, 2008.

- [41] J.M. Juan Zornoza J. Sanz Subirana and M. Hernndez-Pajares. *GNSS Data Processing: Fundamentals and Algorithms*. European Space Agency, 2013.
- [42] Raja Jurdak, Peter Corke, Dhinesh Dharman, and Guillaume Salagnac. Adaptive gps duty cycling and radio ranging for energy-efficient localization. In *Proc. of the 8th ACM SenSys*, pages 57–70. ACM, 2010.
- [43] Wonho Kang and Youngnam Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal*, 15(5):2906–2916, 2015.
- [44] E. Kaplan and C. Hegarty. *Understanding GPS Principles and Applications*. Artech House, 2 edition, 2005.
- [45] Mikkel Baun Kjærgaard, Jakob Langdal, Torben Godsk, and Thomas Toftkjær. EnTracked: Energy-efficient Robust Position Tracking for Mobile Devices. In *Proc. of the 7th ACM MobiSys*, pages 221–234, 2009.
- [46] Paweł Kułakowski, Javier Vales-Alonso, Esteban Egea-López, Wiesław Ludwin, and Joan García-Haro. Angle-of-arrival localization based on antenna arrays for wireless sensor networks. *Computers & Electrical Engineering*, 36(6):1181–1186, 2010.
- [47] Y. Kuo, P. Pannuto, K. Hsiao, and P. Dutta. Luxapose: Indoor Positioning with Mobile Phones and Visible Light. In *Proc. of the 20th ACM MobiCom*, pages 447–458, 2014.
- [48] J. LaMance, Javier DeSalas, and Jani Järvinen. Assisted GPS: A Low-Infrastructure Approach. <http://gpsworld.com/innovation-assisted-gps-a-low-infrastructure-approach/>, 2002. Accessed: 2018-07-18.
- [49] Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate?: cross-domain collaborative filtering for sparsity reduction. In *Proc. of the 21st IJCAI*, pages 2052–2057, 2009.

- [50] Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng Zhao. A Reliable and Accurate Indoor Localization Method Using Phone inertial Sensors. In *Proc. of the 14th ACM UbiComp*, pages 421–430, 2012.
- [51] Xinrong Li. Collaborative localization with received-signal strength in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 56(6):3807–3817, 2007.
- [52] Kaisen Lin, Aman Kansal, Dimitrios Lymberopoulos, and Feng Zhao. Energy-accuracy Trade-off for Continuous Mobile Device Location. In *Proc. of the 7th ACM MobiSys*, pages 285–298, 2010.
- [53] Cihang Liu, Lan Zhang, Zongqian Liu, Kebin Liu, Xiangyang Li, and Yunhao Liu. Lasagna: Towards Deep Hierarchical Understanding and Searching over Mobile Sensing Data. In *Proc. of the 22nd ACM MobiCom*, pages 334–347, 2016.
- [54] Jie Liu, Bodhi Priyantha, Ted Hart, Heitor S. Ramos, Antonio A. F. Loureiro, and Qiang Wang. Energy Efficient GPS Sensing with Cloud Offloading. In *Proc. of the 10th ACM SenSys*, pages 85–98, 2012.
- [55] Kaikai Liu, Xinxin Liu, and Xiaolin Li. Guoguo: Enabling Fine-grained Indoor Localization via Dmartphone. In *Proc. of the 11th ACM MobiSys*, pages 235–248, 2013.
- [56] Guoyu Lu, Yan Yan, Li Ren, Philip Saponaro, Nicu Sebe, and Chandra Kambhampettu. Where am i in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging. *Neurocomputing*, 173:83–92, 2016.
- [57] J. Luo, H.V. Shukla, and J.-P. Hubaux. Non-Interactive Location Surveying for Sensor Networks with Mobility-Differentiated ToA. In *Proc. of the 25th IEEE INFOCOM*, pages 1241–1252, 2006.
- [58] Robert Margolies, Richard Becker, Simon Byers, Supratim Deb, Rittwik Jana, Simon Urbanek, and Chris Volinsky. Can you find me now? evaluation of network-based localization in a 4g lte network. In *Proc. of the 36th IEEE INFOCOM*, pages 1–9. IEEE, 2017.

- [59] Yasuko Matsubara, Yasushi Sakurai, and Christos Faloutsos. AutoPlait: Automatic Mining of Co-evolving Time Sequences. In *Proc. of ACM SIGMOD*, pages 193–204, 2014.
- [60] P. Misra, W. Hu, Y. Jin, J. Liu, A. S. de Paula, N. Wirstm, and T. Voigt. Energy efficient GPS acquisition with Sparse-GPS. In *Proc. of the 13th ACM/IEEE IPSN*, pages 155–166, 2014.
- [61] Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements and Performance*. Massachusetts: Ganga-Jamuna Press, 2 edition, 2006.
- [62] NASA. CDDIS: NASA’s Archive Space Geodesy Data. https://cddis.nasa.gov/Data/_and/_Derived/_Products/GNSS/broadcast/_ephemeris/_data.html. Accessed: 2018-07-18.
- [63] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps) using aoa. In *Proc. of the 22nd IEEE INFOCOM*, volume 3, pages 1734–1743. Ieee, 2003.
- [64] Dragoş Niculescu and Badri Nath. DV Based Positioning in Ad Hoc Networks. *Telecommunication Systems*, 22(1-4):267–280, 2003.
- [65] OpenBMap. OpenBMap Project. <https://radiocells.org/>. Accessed: 2016-05-18.
- [66] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient Rate-adaptive GPS-based Positioning for Smartphones. In *Proc. of the 8th MobiSys*, pages 299–314, 2010.
- [67] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proc. of the 8th ACM MobiSys*, pages 299–314. ACM, 2010.
- [68] J.J. Pan, Q. Yang, H. Chang, and D.Y. Yeung. A Manifold Regularization Approach to Calibration Reduction for Sensor-Network Based Tracking. In *Proc. of the 21st AAAI*, pages 988–993, 2006.

- [69] J.J. Pan, Q. Yang, and S.J. Pan. Online Co-localization in Indoor Wireless Networks by Dimension Reduction. In *Proc. of the 22nd AAAI*, pages 1102–1107, 2007.
- [70] Sinno Jialin Pan, Dou Shen, Qiang Yang, and James T Kwok. Transferring localization models across space. In *Proc. of 2008 AAAI*, pages 1383–1388, 2008.
- [71] Abhinav Parate, Meng-Chieh Chiu, Chaniel Chadowitz, Deepak Ganesan, and Evangelos Kalogerakis. RisQ: Recognizing Smoking Gestures with Inertial Sensors on a Wristband. In *Proc. of the 12th ACM MobiSys*, pages 149–161, 2014.
- [72] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proc. of the 6th ACM MobiCom*, pages 32–43. ACM, 2000.
- [73] V. Radu, P. Katsikouli, R. Sarkar, and M. Marina. A Semi-Supervised Learning Approach for Robust Indoor-Outdoor Detection with Smartphones. In *Proc. of the 12th ACM SenSys*, pages 280–294, 2014.
- [74] Peng Rong and Mihail L Sichitiu. Angle of arrival localization for wireless sensor networks. In *Proc. of the 3rd IEEE SECON*, volume 1, pages 374–382. IEEE, 2006.
- [75] RoseIndia.Net. What is Differential GPS? <https://www.roseindia.net/technology/gps/what-is-Differential-GPS.shtml>. Accessed: 2018-08-04.
- [76] Andreas Savvides, Chih-Chieh Han, and Mani B Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of the 7th ACM MobiCom*, pages 166–179. ACM, 2001.
- [77] Georg Schroth, Robert Huitl, David Chen, Mohammad Abu-Alqumsan, Anas Al-Nuaimi, and Eckehard Steinbach. Mobile visual location recognition. *IEEE Signal Processing Magazine*, 28(4):77–89, 2011.
- [78] Suranga Seneviratne, Fangzhou Jiang, Mathieu Cunche, and Aruna Seneviratne. Ssids in the wild: Extracting semantic information from wifi ssids. In *Proc. of the 40th IEEE Conference on Local Computer Networks*, pages 494–497. IEEE, 2015.

- [79] Skyhook. Skyhook Precision Location. <http://www.skyhookwireless.com/products/precision-location>. Accessed: 2016-05-18.
- [80] Zhuo Sun, Yiqiang Chen, Juan Qi, and Junfa Liu. Adaptive localization through transfer learning in indoor wi-fi environment. In *Proc. of 8th International Conference on Machine Learning and Applications*, pages 331–336. IEEE, 2008.
- [81] Rakmin Sung, Suk-hoon Jung, and Dongsoo Han. Sound based indoor and outdoor environment detection for seamless positioning handover. *ICT Express*, 1(3):106–109, 2015.
- [82] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Proc. of the 27th NIPS*, pages 3104–3112, 2014.
- [83] Yuki Takabayashi, Takashi Matsuzaki, Hiroshi Kameda, and Masayoshi Ito. Target tracking using tdoa/fdoa measurements in the distributed sensor network. In *SICE Annual Conference*, pages 3441–3446. IEEE, 2008.
- [84] T Takasu. Rtklib: An open source program package for gnss positioning, 2011.
- [85] Stephen P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan Memik. Indoor Localization without Infrastructure using the Acoustic Background Spectrum. In *Proc. of the 9th ACM MobiSys*, pages 155–168, 2011.
- [86] Arvind Thiagarajan, Lenin S. Ravindranath, Hari Balakrishnan, Samuel Madden, and Lewis Girod. Accurate, Low-Energy Trajectory Mapping for Mobile Devices. In *Proc. of the 8th USENIX NSDI*, pages 267–280, 2011.
- [87] Arvind Thiagarajan, Lenin S. Ravindranath, Katrina LaCurts, Sivan Toledo, Jakob Eriksson, Samuel Madden, and Hari Balakrishnan. VTrack: Accurate, Energy-Aware Traffic Delay Estimation Using Mobile Phones. In *Proc. of the 7th ACM SenSys*, pages 85–98, 2009.
- [88] Hui Tian, Shuang Wang, and Huaiyao Xie. Localization using cooperative aoa approach. In *Proc. of International Conference on Wireless Communications, Networking and Mobile Computing, 2007*, pages 2416–2419. IEEE, 2007.

- [89] Arvin Wen Tsui, Wei-Cheng Lin, Wei-Ju Chen, Polly Huang, and Hao-Hua Chu. Accuracy Performance Analysis between War Driving and War Walking in Metropolitan Wi-Fi Localization. *IEEE Transactions on Mobile Computing*, 9(11):1551–1562, 2010.
- [90] Reza Monir Vaghefi, Mohammad Reza Gholami, R Michael Buehrer, and Erik G Strom. Cooperative received signal strength-based sensor localization with unknown transmit powers. *IEEE Transactions on Signal Processing*, 61(6):1389–1403, 2013.
- [91] Alex Varshavsky, Anthony LaMarca, Jeffrey Hightower, and Eyal de Lara. The SkyLoc Floor Localization System. In *Proc. of the 5th IEEE PerCom*, pages 125–134, 2007.
- [92] D. Vasisht, S. Kumar, and D. Katabi. Decimeter-Level Localization with a Single WiFi Access Point. In *Proc. of the 13th USENIX NSDI*, pages 165–178, 2016.
- [93] Gang Wang and Kehu Yang. A new approach to sensor node localization using rss measurements in wireless sensor networks. *IEEE transactions on Wireless Communications*, 10(5):1389–1395, 2011.
- [94] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No Need to War-drive: Unsupervised Indoor Localization. In *Proc. of the 10th ACM MobiSys*, pages 197–210, 2012.
- [95] Hua-Yan Wang, Vincent W Zheng, Junhui Zhao, and Qiang Yang. Indoor localization in multi-floor environments with reduced effort. In *Proc. of 2010 IEEE PerCom*, pages 244–252. IEEE, 2010.
- [96] WiGLE. WiGLE: Wireless Network Mapping. <https://wigle.net/>. Accessed: 2016-05-18.
- [97] Erik Wilhelm, Sandra Siby, Yuren Zhou, Xavier Jayaraj Siddarth Ashok, Melani Jayasuriya, Shaohui Foong, Jacksheeng Kee, Kristin L Wood, and Nils Ole Tippenhauer. Wearable environmental sensors and infrastructure for mobile large-scale urban deployment. *IEEE Sensors Journal*, 16(22):8111–8123, 2016.

- [98] Matthew Wong, Alison Powell, and Andrew Clement. Reading service set identifiers (ssids): Marking and locating public and private wireless spaces. In *Proc. of the International Conference, Internet Research 8.0, in Association of the Internet Researchers (AoIR)*, 2007.
- [99] J. Xiong and K. Jamieson. ArrayTrack: A Fine-grained Indoor Location System. In *Proc. of the 10th USENIX NSDI*, pages 71–84, 2013.
- [100] Moustafa Youssef and Ashok Agrawala. The Horus WLAN Location Determination System. In *Proc. of the 3rd ACM MobiSys*, pages 205–218, 2005.
- [101] Xiao-long YU and Xue-kui HU. Gps rtk technique: Merits, demerits and perspective. *Bulletin of Surveying and Mapping*, 10:015, 2007.
- [102] C. Zhang, F. Li, J. Luo, and Y. He. iLocScan: Harnessing Multipath for Simultaneous Indoor Source Localization and Space Scanning. In *Proc. of the 12th ACM SenSys*, pages 91–104, 2014.
- [103] C. Zhang, K.P. Subbu, J. Luo, and J. Wu. GROPING: Geomagnetism and cROwd-sensing Powered Indoor NaviGation. *IEEE Transactions on Mobile Computing*, 14(2):387–400, 2015.
- [104] C. Zhang, K.P. Subbu, J. Luo, and J. Wu. GROPING: Geomagnetism and cROwd-sensing Powered Indoor NaviGation. *IEEE Transactions on Mobile Computing*, 14(2):387–400, 2015.
- [105] C. Zhang and X. Zhang. Pulsar: Towards Ubiquitous Visible Light Localization. In *Proc. of the 23rd ACM MobiCom*, pages 208–221, 2017.
- [106] Jerry Zhang, Aaron Hallquist, Eric Liang, and Avidesh Zakhori. Location-based image retrieval for urban environments. In *Proc. of the 18th IEEE International Conference on Image Processing*, pages 3677–3680. IEEE, 2011.
- [107] Vincent Wenchen Zheng, Evan Wei Xiang, Qiang Yang, and Dou Shen. Transferring localization models over time. In *Proc. of the 2008 AAAI*, pages 1421–1426, 2008.

- [108] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen. IODetector: A Generic Service for Indoor Outdoor Detection. In *Proc. of the 10th ACM SenSys*, pages 113–126, 2012.
- [109] Xiaojun Zhu, Qun Li, and Guihai Chen. Apt: Accurate outdoor pedestrian tracking with smartphones. In *Proc. of the 32nd IEEE INFOCOM*, pages 2508–2516. IEEE, 2013.
- [110] Han Zou, Hao Jiang, Yiwen Luo, Jianjie Zhu, Xiaoxuan Lu, and Lihua Xie. Bluede-tect: An ibeacon-enabled scheme for accurate and energy-efficient indoor-outdoor detection and seamless location-based service. *Sensors*, 16(2):268, 2016.