

**Locatietracking in binnen- en buitenomgevingen
gebaseerd op het viterbi-principe**

**Location Tracking in Indoor and Outdoor Environments
Based on the Viterbi Principle**

Jens Trog

**Promotoren: prof. dr. ir. W. Joseph, prof. dr. ir. D. Plets
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: elektrotechniek**



**UNIVERSITEIT
GENT**

**Vakgroep Informatietechnologie
Voorzitter: prof. dr. ir. B. Dhoedt
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2019 - 2020**

ISBN 978-94-6355-279-0
NUR 986
Wettelijk depot: D/2019/10.500/87

**Supervisors**

prof. dr. ir. Wout Joseph
prof. dr. ir. David Plets

Chair

prof. dr. ir. Filip De Turck

Other members of the examination board

prof. dr. ir. Eli De Poorter
prof. dr. ir. Heidi Steendam
prof. dr. ir. Maarten Weyn
dr. ir. Joaquín Torres Sospedra

Acknowledgments

“Keep your word of thanks concise.”

— William Strunk Jr.

I would like to thank all my colleagues, friends, and family.

Ghent, September 2019
Jens Trog

Table of Contents

Acknowledgments	i
Nederlandstalige samenvatting	xxv
English summary	xxxi
1 Introduction	1
1.1 Context	1
1.2 Localization techniques	2
1.3 Ranging	3
1.3.1 Time of arrival	3
1.3.2 Time difference of arrival	4
1.3.3 Angle of arrival	4
1.3.4 Phase difference of arrival	5
1.3.5 Signal strength	5
1.4 Fingerprinting	5
1.5 Enabling technologies for localization	6
1.5.1 Indoor	6
1.5.1.1 RFID	6
1.5.1.2 WiFi	7
1.5.1.3 Bluetooth Low Energy	7
1.5.1.4 Ultra-Wideband	8
1.5.2 Light	8
1.5.3 Outdoor	9
1.5.3.1 GPS	9
1.5.3.2 Cellular networks	9
1.5.3.3 LPWAN: LoRa, Sigfox, NB-IoT	10
1.6 Signals of opportunity	10
1.6.1 Magnetic	10
1.6.2 Acoustic	11
1.6.3 FM	11
1.7 Pedestrian dead reckoning	12
1.8 Sensor fusion	12

1.9	Filtering and post-processing methods	13
1.9.1	Hidden Markov model	13
1.9.2	Kalman filter	13
1.9.3	Particle filter	14
1.10	Cooperative localization	15
1.11	User anonymity	15
1.12	Main research contributions and outline	15
1.13	Publications	16
1.13.1	Publications in international journals (A1)	16
1.13.1.1	As first author	16
1.13.1.2	As co-author	17
1.13.2	Publications in international conferences (P1)	17
1.13.2.1	As first author	17
1.13.2.2	As co-author	18
	References	20

I Indoor localization 35

2	Indoor location tracking based on signal strength data 37
2.1	Introduction 37
2.2	Related work 38
2.3	RSS fingerprinting 39
2.4	Location tracking algorithm 40
2.4.1	Viterbi principle 40
2.4.2	Map information and motion model 41
2.4.3	Robust start position 42
2.5	Experimental configuration 44
2.5.1	Algorithm settings 46
2.5.2	Evaluation metrics 47
2.6	Simulations 48
2.6.1	Influence of noise level 49
2.7	Experimental validation 52
2.7.1	ZigBee 52
2.7.2	WiFi 52
2.8	Sensitivity analysis 53
2.8.1	Node density 54
2.8.2	Grid size 54
2.8.3	Execution time 55
2.8.4	Path loss model 57
2.8.5	Semantic data 57
2.8.6	Future measurement data 58
2.9	Conclusion 59
	References 60

3	Compensation of human body shadowing	63
3.1	Introduction	63
3.2	Related work	64
3.3	Characterization of human body influence	65
3.4	System overview	67
3.5	Location tracking algorithm	68
3.6	Compensation techniques	69
3.6.1	Combining multiple tags	69
3.6.2	Compensating user orientation	71
3.6.2.1	Orientation estimator	71
3.6.2.2	Compensation models	74
3.7	Experimental validation	78
3.7.1	Configuration	78
3.7.2	Results	78
3.7.3	Impact of multiple mobile tags	79
3.7.4	Impact of body shadowing compensation	81
3.7.5	Comparison of compensation models	82
3.8	Conclusion	83
	References	85
4	Radio map optimization through unsupervised learning	89
4.1	Introduction	89
4.2	Related work	90
4.3	Methodology	91
4.3.1	Experimental configuration	93
4.3.2	Radio map	94
4.3.3	Self-calibration	96
4.4	Unsupervised learning	97
4.4.1	Motivation	97
4.4.2	Location tracking algorithm	102
4.4.3	Radio map update step	103
4.5	Simulations	103
4.5.1	Settings	104
4.5.2	Results	107
4.5.2.1	Influence of room and local deviation	107
4.5.2.2	Influence of additional noise	108
4.6	Experimental validation	108
4.7	Conclusions	112
	References	114
II	Outdoor localization	117
5	Map matching and lane detection	119
5.1	Introduction	119

5.2	Related work	121
5.2.1	Map matching	121
5.2.2	Lane detection	123
5.3	Map matching	124
5.3.1	Grid and road segments	124
5.3.2	Algorithm	126
5.3.2.1	General	126
5.3.2.2	Geographic data selection	127
5.3.2.3	Mapping and interpolation	128
5.4	Lane detection	130
5.4.1	Lane changes	130
5.4.2	Driving lane allocation	136
5.5	Experimental validation	137
5.5.1	Trajectories	137
5.5.2	Performance metrics	138
5.5.3	Map matching accuracy	140
5.5.3.1	GPS sample interval	140
5.5.3.2	GPS noise	142
5.5.3.3	Sensitivity analysis	144
5.5.4	Lane detection accuracy	146
5.5.4.1	Lane changes	146
5.5.4.2	Driving lane	146
5.5.4.3	Accelerometer sample rate	147
5.6	Conclusion	147
	References	149
6	Outdoor location tracking based on cellular data	155
6.1	Introduction	155
6.2	Related work	156
6.3	Methodology	158
6.3.1	Cellular network	158
6.3.2	Grid	158
6.3.3	Experimental data	161
6.3.4	Measurement format	161
6.3.5	Propagation delay	167
6.3.6	Measurement report	168
6.4	Data availability	170
6.5	Tracking algorithms	171
6.5.1	Cell-ID	171
6.5.2	Centroid	171
6.5.3	AMT: antenna, map, and timing information based tracking	172
6.5.3.1	Phase I: temporary estimation	172
6.5.3.2	Phase II: route mapping filter	176
6.5.4	Mode of transportation estimator	178

6.5.5	Particle Filter	179
6.6	Experimental validation	180
6.6.1	General	180
6.6.2	Comparison with other algorithms	182
6.6.3	Non-time-critical vs. real-time	183
6.6.4	Impact of environment	184
6.6.5	Impact of smartphone usage	184
6.6.6	Impact of mode of transportation	185
6.7	Conclusion	185
	References	187
7	Conclusions and future work	191
7.1	Conclusions	191
7.2	Future work	193

List of Figures

2.1	Reconstructed trajectory with and without semantic data taken into account.	42
2.2	Example of additional starting positions with 2 <i>levels</i> and an <i>inter distance</i> of 2 m for visibility.	44
2.3	Mobile nodes and indoor environment.	45
2.4	Floor plan with indication of the access points and the material of all walls and doors.	45
2.5	Floor plan with indication of the walls, doors, and three test trajectories shown in dashed orange, dotted green, and solid red.	46
2.6	Reconstruction of a test trajectory by the basic (a) and the proposed location tracking algorithm (b)	48
2.7	Location of the fixed nodes for a sparse, normal, dense, and very dense network.	49
2.8	Influence of noise for various node densities (simulation). RT: real-time, KF: Kalman filter, PF: particle filter, and PFS: particle filter with smoother.	50
2.9	Average execution time to calculate one location update for different grid sizes (solid line: proposed algorithm, dotted line: basic algorithm).	56
2.10	Influence of the amount of future measurement data on the four metrics (in a normal node density).	58
3.1	The effect of a full rotation and human body shadowing in an indoor environment.	66
3.2	Floor plan of testbed with indication of walls, doors, access points, near access point, far access point, and location of the rotating user.	66
3.3	Flow graph of the location tracking system with human body compensation.	67
3.4	Body-worn tags.	70
3.5	Red and blue trajectory for testing the orientation estimator (simulation).	72

3.6	Performance of orientation estimator for the two trajectories and three noise levels. Solid line: median error, dashed line: standard deviation.	73
3.7	Angles between body-worn tag and AP.	74
3.8	Graphical representation of the two compensation models. . .	75
3.9	Simulations in S4L.	77
3.10	Floor plan of testbed with indication of test trajectory and start point (red dot).	79
3.11	Mean accuracy as a function of the number of used mobile tags for the basic and advanced positioning algorithm.	80
3.12	Impact of body shadowing compensation on the tracking accuracy.	82
3.13	Comparison of different compensation models on the tracking accuracy.	83
4.1	Flow graph of the proposed radio map construction and optimization technique. N_{IT} is the current training iteration. .	92
4.2	Floor plan of the office building with indication of walls, doors, and access points.	93
4.3	Tripod with battery-powered mobile node and ceiling-mounted fixed access points in corridor.	94
4.4	Floor plan of the office building with indication of walls, doors, access points, and static validation locations where a mobile node broadcasted packets at 5 Hz.	97
4.5	Average difference between model-based radio map and measurements, grouped per room, for one access point (green dot), before (a–c) and after the unsupervised learning (d–f). Unvisited rooms are indicated in yellow.	98
4.6	Location of the access points for the simulated scenarios with 35, 15, and 9 fixed access points: dense (a), normal (b), and sparse (c) configuration. The relative improvement in median accuracy after unsupervised learning with 1 h of unlabeled training data, evaluated on 1000 uniformly spread locations for the three scenario's (d–f). The median accuracy before (g–i) and after learning (j–l) for the three scenario's. The room deviation (x -axis) and local deviation (y -axis) vary from 0 dB to 16 dB in steps of 2 dB.	106
4.7	Accuracy before and after unsupervised learning for a varying level of additional noise with the normal access point configuration.	109
4.8	Floor plan with rough indication of the training data trajectory.	109
4.9	Cumulative distribution function of the positioning accuracy before (dashed line) and after training (solid line), for the three access point configurations and the WHIPP path loss model.	111

5.1	Map matching problem: nearest grid point	120
5.2	Map matching problem: minimizing a distance-based metric .	121
5.3	Grid and road segments based on OpenStreetMap data and a grid size of 10 m.	125
5.4	Flow graph of the map matching algorithm. The dashed lines are only executed at the end if all timestamped geographic data is processed. SGC: selected geographic coordinates, PIM: paths in memory, RS: road segments, DC: direction changes, and CC: code changes.	126
5.5	Available driving lanes per road segment based on open map data.	131
5.6	Alignment of sensor and car coordinate systems.	131
5.7	Sample of raw and low-pass filtered sensor values for the lat- eral axis of the accelerometer with indication of left and right lane changes for both trajectories.	133
5.8	Sample of the driving lane at each road segment for both trajectories.	134
5.9	Raw GPS input data with ground truth construction.	139
5.10	F_1^{mm} score and median error as a function of the sample interval per mode of transportation.	141
5.11	F_1^{mm} score and median error as a function of the sample interval per mode of transportation with interpolation at 1 Hz.	142
5.12	F_1^{mm} score and median error as a function of the GPS noise per mode of transportation for a sample interval T of 1 s (a– b) and 1 min (c–d). Overall F_1^{mm} score and median error as a function of the GPS noise per sample interval T (e–f) . . .	143
5.13	F_1^{mm} score and median error as a function of the update distance (a–b), distance weight α (c–d), direction change weight β (e–f), and code change weight γ (g–h).	145
5.14	Precision, recall, and F_1^{ld} score as a function of the accelerom- eter sample rate over both trajectories.	148
6.1	NodeB density in urban and rural environments (NodeBs are indicated by blue triangles).	159
6.2	Antenna directions and configuration.	160
6.3	Grid based on OpenStreetMap data with grid size 50 m.	160
6.4	GPS trajectories on foot in the city center of Ghent (black lines), estimated positions (blue dots), error between estima- tion and ground truth (blue lines), and NodeBs (gray triangles).	162
6.5	GPS trajectories by bike in the city center of Ghent (black lines), estimated positions (blue dots), error between estima- tion and ground truth (blue lines), and NodeBs (gray triangles).	163

6.6	GPS trajectories by car around the city center of Ghent (black lines), estimated positions (blue dots), error between estimation and ground truth (blue lines), and NodeBs (gray triangles).	164
6.7	GPS trajectories on foot and by bike in a rural area (black lines), estimated positions (blue dots), error between estimation and ground truth (blue lines), and NodeBs (gray triangles).	165
6.8	GPS trajectories by car in a rural area (black lines), estimated positions (blue dots), error between estimation and ground truth (blue lines), and NodeBs (gray triangles).	166
6.9	Propagation delay granularity and accuracy.	168
6.10	Measured path loss as a function of distance between mobile user and NodeBs (blue dots). A fitted path loss model is plotted as a red line.	169
6.11	Average number of measurement reports (solid orange line) and propagation delays (dotted blue line) per user, per hour for more than a million distinct active users during one week in Belgium.	170
6.12	Flow graph of the proposed location tracking algorithm: phase I and phase II.	172
6.13	Working principle of proposed location tracking algorithm: phase I. Mobile user (yellow square), NodeBs (gray triangles), serving NodeB (green star), NodeBs with measurements (red triangles), antenna orientation (red lines), black cross sign \times (centroid estimation), black plus sign $+$ (AMT estimation), CA_{sc} (green arc), and CS_{nb} (blue circle sectors).	174
6.14	Detail of the estimated locations before and after the route mapping filter: temporary estimations (green crosses) and final estimated trajectory (blue dots). The road network and GPS trajectory are indicated in gray and black lines.	178
6.15	Median accuracy per scenario with the <i>TE</i> , <i>PF</i> , <i>AMT-RT</i> , and <i>AMT-NTC</i> technique.	182

List of Tables

2.1	Settings of the proposed indoor location tracking algorithm. <i>MP</i> : maximum number of paths retained in memory with every location update.)	47
2.2	Improvement of the proposed location tracking algorithm (Viterbi) compared to the reference and real-time (RT) positioning techniques for four node densities and a noise level of 10 dB.	51
2.3	Average performance with the ZigBee node. <i>RT</i> : real-time. . .	52
2.4	Average performance with the WiFi node. <i>RT</i> : real-time. . .	53
2.5	Performance for different node densities. IAPD: inter access point distance	54
2.6	Performance for different grid resolutions.	55
2.7	Influence of used path loss model.	57
2.8	Influence of semantic data.	58
3.1	Positioning accuracy as a function of the used mobile tags for the basic and advanced algorithm, and the improvement when using multiple tags (on the bottom) as well as the improvement between both algorithms (on the right).	80
3.2	Positioning accuracy as a function of the used mobile tags with and without compensation, and the improvement between both.	81
4.1	Experimentally measured differences and deviations compared to the theoretical path loss models after self-calibration. . . .	101
4.2	Accuracy of experimental validation test set per access point configuration (sparse, normal, and dense) and path loss model. The first and second value are the accuracy before and after training and the third value is the relative improvement. . . .	111
5.1	Lane change peak statistics based on lateral accelerometer data. LC: amount of lane change.	135
5.2	GPS dataset details per mode of transportation (MoT). . . .	137

5.3	Driving lane details per trajectory based on dashcam video recordings. LC: amount of lane changes, LCPM: average number of lane changes per minute, 1-av: only one lane available, and number of accelerometer samples.	138
5.4	Precision, recall, and F_1^{ld} score for the lane change detection. tp: true positives, fp: false positives, fn: false negatives. . . .	146
5.5	Accuracy of the lane detection algorithm as a percentage of the complete trajectory. 1-av indicates that there is only one lane available, and 1-off or 2-off that the estimated lane is one or two lanes next to the correct lane.	147
6.1	Trajectory details.	161
6.2	Accuracy, number of positioning updates, and average time and distance between two consecutive location updates, per scenario and algorithm. TE: temporary estimation (phase I), AMT-RT: real-time route mapping filter (phase II), AMT-NTC: non-time-critical route mapping filter (phase II), PF: particle filter.	181

List of Algorithms

1	Indoor location tracking system.	43
2	Map matching technique.	129
3	Calculating the temporary estimation.	173
4	Route mapping filter.	177

List of Acronyms

A

ABS	Acoustic Background Spectrum
AMT	Antenna, Map, and Timing information
AoA	Angle of Arrival
AoD	Angle of Departure
AP	Access Point

B

BAN	Body Area Network
BLE	Bluetooth Low Energy
BMI	Body Mass Index

C

CAGR	Compound Annual Growth Rate
CCD	Charge-Coupled Device
CDF	Cumulative Distribution Function
CRLB	Cramér–Rao Lower Bound
CSI	Channel State Information
CSS	Chirp Spread Spectrum

D

D2D	Device-to-Device
dB	Decibel
DBPSK	Differential Binary Phase-Shift Keying
DFL	Device-Free Localization
DSSS	Direct-Sequence Spread Spectrum

E

EKF	Extended Kalman Filter
-----	------------------------

F

FCC	Federal Communications Commission
FCD	Floating Car Data
FDTD	Finite-Difference Time-Domain
FFBS	Forward Filtering-Backward Smoothing
FM	Frequency Modulation
FSPL	Free-Space Path Loss

G

GFSK	Gaussian Frequency Shift Keying
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile communications

H

HMM	Hidden Markov Model
-----	---------------------

I

i.i.d.	independent and identically distributed
IEEE	Institute of Electrical and Electronics Engineers
IMSI	International Mobile Subscriber Identity
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IoT	Internet of Things
IR	Infrared
ISM	Industrial, Scientific and Medical
ITS	Intelligent Transportation Systems

K

KF	Kalman Filter
----	---------------

L

LBS	Location-Based Service
LED	Light-Emitting Diode
LIDAR	Light Detection and Ranging
LPWAN	Low-Power Wide-Area Network
LoRa	Long Range
LoS	Line-of-Sight
LQE	Linear Quadratic Estimation
LSE	Least Squares Estimation
LTE	Long-Term Evolution

M

MAP	Maximum a Posteriori
MDI	Motion Dynamics Information
MEMS	Microelectromechanical Systems
MIMO	Multiple-Input Multiple-Output
MLE	Maximum Likelihood Estimation

mmWave	millimeter wave
MoT	Mode of Transportation
MPC	Multipath Component
MSE	Mean-Square-Errors

N

NB-IoT	Narrowband Internet of Things
NIC	Network Interface Card
NLoS	Non Line-of-Sight

O

OFDM	Orthogonal Frequency-Division Multiplexing
OLoS	Obstructed Line-of-Sight

P

PDF	Probability Density Function
PDoA	Phase Difference of Arrival
PDR	Pedestrian Dead Reckoning
PF	Particle Filter
PIFA	Planar Inverted-F Antenna
PL	Path Loss

R

RF	Radio-Frequency
RFID	Radio-Frequency Identification
RMS	Root-Mean-Square
RMSE	Root-Mean-Square Error
RSCP	Received Signal Code Power
RSS	Received Single Strength
RSSI	Received Single Strength Indication
RT	Real-Time
RTLS	Real-Time Locating System

RTT Round-Trip Time

S

S4L	Sim4Life
SA	Spectrum Analyzer
SC-FDMA	Single-Carrier Frequency-Division Multiple Access
SLAM	Simultaneous Localization and Mapping
SMC	Sequential Monte Carlo
SNR	Signal-to-Noise Ratio
SoOP	Signal of Opportunity

T

TDoA	Time Difference of Arrival
TE	Temporary Estimation
TGmc	Task Group mc
TMSI	Temporary Mobile Subscriber Identity
ToA	Time of Arrival
ToF	Time of Flight
TTFF	Time to First Fix
TWR	Two-Way Ranging

U

UKF	Unscented Kalman Filter
UMTS	Universal Mobile Telecommunications System
USA	United States of America
UTM	Universal Transverse Mercator
UWB	Ultra-Wideband

V

VFM	Virtual Family Male
VNA	Vector Network Analyzer

W

WBAN	Wireless Body Area Network
WHIPP	WiCa Heuristic Indoor Propagation Prediction
WiCa	Wireless & Cable
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WLS	Weighted Least Squares
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

Nederlandstalige samenvatting –Dutch summary–

Locatietracking in binnen- en buitenomgevingen is van groot belang geweest in de afgelopen decennia en wordt nog steeds belangrijker. De alsmaar toenemende hoeveelheid beschikbare locatiedata door de alomtegenwoordige connectiviteit van mensen en het internet der dingen heeft geleid tot vele location-based services (LBS), ook wel locatiegebaseerde diensten genoemd. Deze verzameling van diensten maakt gebruik van geografische locatiedata van mobiele toestellen en bevindt zich in meerdere domeinen, bijvoorbeeld de gezondheidszorg, overheid of openbare diensten. Voorbeelden van deze LBS-diensten zijn persoonlijke navigatie, het optimaliseren van de productiviteit in de industrie en het analyseren van verkeerspatronen. De wereldwijde LBS-markt wordt momenteel geschat op 30 miljard dollar en zal naar alle waarschijnlijkheid de kaap van 133 miljard dollar bereiken tegen 2023. De grootste uitdagingen voor de verdere ontwikkeling en acceptatie van LBS zijn problemen die verband houden met de privacy van locatiegegevens, beschikbaarheid van up-to-date kaarten en de betrouwbaarheid, nauwkeurigheid en kosten van lokalisatiesystemen.

Lokalisatietechnieken kunnen op verschillende manieren van elkaar worden onderscheiden. Deze systemen kunnen actief of passief zijn, naargelang het object of de persoon die wordt getraceerd al dan niet is uitgerust met een lokalisatiedevice. Veel positioneringssystemen zijn actief en in dit geval is het object of de persoon die wordt getraceerd, voorzien van een actieve tag. Bij passieve positioneringssystemen, ook wel device-free lokalisatie (DFL) genoemd, neemt het object of persoon niet actief deel aan het positioneringsproces. De positie wordt geschat op basis van veranderingen in de omgeving, die op hun beurt worden veroorzaakt door de aanwezigheid en beweging van het object of de persoon die wordt gevolgd. Dit is interessant voor niet-coöperatieve lokalisatie zoals inbraakdetectie, opvolgen van wilde dieren of het beschermen van goederen. Een ander onderscheid kan worden gemaakt op basis van de gebruikte communicatietechnologie als drager van

het informatiesignaal, bijvoorbeeld WiFi, Bluetooth Low Energy (BLE), GSM of GPS, en de gebruikte rangingtechniek om de locatie te bepalen via triangulatie of multilateratie. Een bekend alternatief is fingerprinting op basis van signaalsterkte. Deze techniek schat de locatie door te zoeken naar de beste overeenkomst in een fingerprintkaart (een databank met signaalsignaturen per locatie). Het grootste voordeel van deze fingerprintingtechniek is dat het voordeel haalt uit multipath, terwijl dit voor rangingtechnieken altijd een nadeel is. Dit komt omdat multipath zorgt voor variatie in de signaalsignaturen wat resulteert in unieke en goed onderscheidbare fingerprints terwijl deze variatie wordt gezien als extra ruis op de metingen, hetgeen nadelig is voor rangingtechnieken.

Het eerste deel van dit proefschrift (Hoofdstuk 2–4) gaat over lokalisatie in binnenomgevingen met een focus op signaalsterkte gebaseerde systemen, invloed van het menselijk lichaam en optimalisatie van modelgebaseerde fingerprintkaarten. Het tweede deel van dit proefschrift (Hoofdstuk 5–6) richt zich op het mappen van GPS-data op een wegennetwerk met ondersteuning voor rijstrookdetectie en op lokalisatie in buitenomgevingen op basis van mobiele gegevens.

Hoofdstuk 2 beschrijft een real-time locatietrackingsysteem voor binnenomgevingen op basis van het Viterbi-principe en semantische gegevens. Het Viterbi-algoritme is een dynamisch programmeeralgoritme dat wordt gebruikt in de context van hidden Markov modellen (HMM) om de meest waarschijnlijke sequentie van verborgen toestanden (het Viterbi pad) te bepalen op basis van waarneembare observaties. Om deze techniek toe te passen op lokalisatie, worden de toestanden geïnterpreteerd als echte locaties op een grondplan en komt dit principe neer op het bepalen van de meest waarschijnlijke sequentie van posities in plaats van enkel de meest waarschijnlijke huidige positie. Bovendien wordt om het aantal mogelijke overgangen tussen twee posities te beperken gebruik gemaakt van semantische data, dit is de omgeving van het object dat wordt getraceerd en een bewegingsmodel. Op deze manier wordt er een realistisch en fysiek mogelijk pad gereconstrueerd dat geen muren kruist of onmogelijke bewegingssnelheden impliceert. Simulaties bevestigden dat het vooropgestelde locatietrackingsysteem robuust was tegen ruis, vooral voor netwerken met een lage node densiteit. Een gesimuleerd ruisniveau van 10 dB resulteerde in verbeteringen van 70.9%, 52.8%, 54.1%, en 42.1% in gemiddelde nauwkeurigheid ten opzichte van een fingerprinting techniek, Kalman filter, particle filter en particle smoother. Een experimentele validatie uitgevoerd in een echte kantooromgeving resulteerde in een verbetering van 26.1% in vergelijking met een standaard fingerprintingtechniek. In een gevoeligheidsanalyse wordt verder aangetoond dat de densiteit van de access points, semantische gegevens, padverliesmodel en rastergrootte een aanzienlijke invloed hebben op de nauwkeurigheid, het vereiste rekenvermogen en het geheugengebruik. Deze laatste zijn belangrijk om in real-time te kunnen werken op goedkope mobiele toestellen.

Veel positioneringssystemen proberen al om te gaan met de signaalvariaties die worden veroorzaakt door multipath fading en diffractie. Een andere belangrijke factor is de invloed die wordt veroorzaakt door de gebruiker zelf, die wordt getraceerd, de zogenaamde human body shadowing (signaalvariaties veroorzaakt door het menselijk lichaam). Deze factor wordt vaak genegeerd door de prestatie van positioneringssystemen te verifiëren door stapsgewijs een mobiele node op een statief te verplaatsen, waardoor de invloed van het menselijk lichaam er expliciet wordt uitgehaald, terwijl praktische lokalisatietechnieken altijd de aanwezigheid van een persoon impliceren. De aanwezigheid van een persoon beïnvloedt de radiofrequente signaalpaden tussen een mobiele tag op of nabij het lichaam en de ontvangende access points, en kunnen naargelang de oriëntatie en locatie van de gebruiker de line-of-sight (gezichtslijn) blokkeren. Dit veroorzaakt extra propagatieverliezen die typisch niet in rekening worden gebracht en leiden tot een lagere nauwkeurigheid van op signaalsterkte gebaseerde positioneringssystemen. Hoofdstuk 3 beschrijft twee compensatiemethoden, die zijn geïntegreerd in het trackingalgoritme van Hoofdstuk 2, om het effect van human body shadowing te verminderen. De eerste methode combineert de gemeten signaalsterkte van meerdere mobiele tags, geplaatst op verschillende delen van een menselijk lichaam; gecombineerd vertonen deze minder variatie, wat de invloed beperkt van het lichaam van de persoon die wordt getraceerd. De tweede methode houdt rekening met de oriëntatie van deze persoon ten opzichte van de access points en de relatieve positie van de persoonlijke tag, en gebruikt een model om de invloed van het menselijk lichaam expliciet te compenseren. Beide methoden kunnen onafhankelijk worden gecombineerd en verminderen de invloed die wordt veroorzaakt door human body shadowing, wat de gemiddelde nauwkeurigheid van het positioneringssysteem verbetert van 3.76 m naar 2.33 m, of een verbetering van 38.1% betekent.

Het grootste nadeel van op signaalsterkte gebaseerde fingerprinting voor lokalisatie is het aanmaken en onderhouden van de fingerprintkaart. Een kaart op basis van een theoretisch model kan veel sneller worden gegenereerd dan op metingen gebaseerde kaarten maar zijn over het algemeen niet nauwkeurig genoeg. Hoofdstuk 4 beschrijft een unsupervised learning techniek om modelgebaseerde fingerprintkaarten te optimaliseren voor lokalisatie, bijvoorbeeld om de kaart nauwkeuriger te maken of om automatisch rekening te houden met wijzigingen in een kantoorindeling. De input voor deze unsupervised learning techniek zijn willekeurige wandelingen waarvoor de exacte locatie onbekend is, een initiële fingerprintkaart gebaseerd op een theoretisch padverliesmodel, een zelfkalibratiemethode en een geavanceerde routeringsfilter, namelijk het trackingalgoritme van Hoofdstuk 2. Deze aanpak vereist geen meetcampagne, kalibratie of terreinonderzoek, die arbeidsintensief en tijdrovend zijn, of een inertiael navigatiesysteem, dat vaak niet beschikbaar is en extra energie verbruikt. Het uitgangspunt van deze techniek is dat de verschillen tussen echte metingen en referentiewaarden, af-

geleid van een modelgebaseerde fingerprintkaart, meestal gecorreleerd zijn per kamer en per access point. Metingen en simulaties tonen aan dat deze verschillen tussen referentiewaarden en echte metingen kunnen worden geleerd in verschillende scenario's, gebaseerd op de willekeurige wandelingen die een typische persoon aflegt. Dit resulteert in referentiewaarden die beter overeenkomen met de echte metingen en die daarom leiden tot een betere fingerprintkaart en nauwkeurigheden. Een experimentele validatie op een testbed in een groot kantoorgebouw bevestigde deze simulaties met relatieve verbeteringen tot 28.6% na unsupervised learning met slechts een kwartier aan trainingsdata zonder label.

Hoofdstuk 5 beschrijft een snel, geheugenefficiënt en wereldwijd toepasbaar map matching algoritme gebaseerd op geografische coördinaten en open kaartinformatie. Het voorgestelde algoritme combineert een Markovketen en een kortste pad algoritme, terwijl er rekening wordt gehouden met het type en de richting van alle wegsegmenten, informatie over eenrichtingsverkeer, maximaal toegestane snelheid per wegsegment en het rijgedrag. Daarnaast is er een rijstrookdetectie-algoritme op basis van accelerometer metingen en rijstrookinformatie uit de open kaartgegevens, dat zichzelf aanpast aan het rijgedrag, toegevoegd aan het map matching algoritme. Een experimentele validatie bestaande uit 12 trajecten te voet, per fiets en per auto, toonde de efficiëntie en nauwkeurigheid aan van de voorgestelde algoritmen, met een gemiddelde F1-score en mediaanfout van 99.5% en 2.09 m voor de map matching en een gemiddelde F1-score van 86.7% voor de detectie van rijstrookveranderingen, wat in 93.0% van de tijd resulteerde in de correct geschatte rijstrook.

Veel technieken voor locatietracking via mobiele netwerken worden gevalideerd in optimale of gecontroleerde omgevingen op een kleine dataset of louter via simulaties. Hoofdstuk 6 beschrijft een techniek om alle gebruikers op een mobiel netwerk efficiënt te lokaliseren, die niet afhankelijk is van voorgaande trainingsdata, ingrepen aan de mobiele kant vereist of zelfs hardware- en softwareveranderingen aan de netwerkinfrastructuur van de telecomoperator. Dit is nuttig voor toepassingen waar er typisch geen samenwerking of overeenkomst is met de mobiele gebruiker, bijvoorbeeld verkeersmonitoring, schatting van de bevolkingsstromen of detectie van criminele activiteiten. Dit maakt het ook onmiddellijk toepasbaar voor telecommunicatiebedrijven. De input voor deze lokalisatietechniek zijn de topologie en metingen van het mobiele netwerk, open kaartinformatie, een methode om het vervoersmiddel te detecteren en het routeringsfilter van Hoofdstuk 5. De techniek werd geëvalueerd op een land omvattende dataset die werd verzameld en verwerkt in real-time, in samenwerking met een telecommunicatiebedrijf in België. De experimentele validatie toonde aan dat de voorgestelde techniek robuust is en dat het vervoersmiddel, het smartphone gebruik en de omgeving een grote invloed hebben op de nauwkeurigheid. De prestaties van een bestaande particle-filter met kaartinformatie werden overtroffen met 31.0% in mediaan nauwkeurigheid.

Ten slotte besluit Hoofdstuk 7 dit proefschrift met een samenvatting van het voltooide werk en bespreekt een aantal mogelijke richtingen voor toekomstig werk.

English summary

Location tracking in indoor and outdoor environments has been of great importance in the last decades and is still gaining attraction. The growing amount of available positioning data due to ubiquitous connectivity and the Internet of Things (IoT) has led to many location-based services (LBS). These are a collection of applications that use geographical location data of mobile devices and are situated in many domains, e.g., health care, government, or public service. Examples of these location-aware applications are personal navigation, optimizing productivity in manufacturing, and analyzing traffic patterns. The worldwide LBS market is currently valued at around USD 30 billion and is expected to reach USD 133 billion by 2023. The largest challenges related to the further development and widespread adoption of LBS are privacy concerns associated with location data, availability of up-to-date maps, and the reliability, accuracy, and cost of indoor and outdoor positioning technologies.

Localization techniques, can be distinguished from each other in multiple ways. These systems can be either active or passive depending on the participation or cooperation of the tracked object or person. A lot of positioning systems are active and in this case, the object that is being tracked, is equipped with an active tag. With passive positioning systems, also known as device-free localization (DFL), the moving object is not actively participating in the positioning process. The position is estimated based on changes in the environment, which are in turn caused by the presence and movement of the entity that is being tracked. This is interesting for non-cooperative localization like intrusion detection, wildlife monitoring, or protecting outdoor assets. Another distinction can be made based on the used communication technology to carry the information signal, e.g., WiFi, Bluetooth Low Energy (BLE), GSM, or GPS, and the used ranging technique to determine the location via triangulation or multilateration. A well known alternative is signal strength based fingerprinting. This technique estimates the location by looking for the closest match in a pretrained fingerprint database, also known as radio or coverage map. The largest advantage of this fingerprinting technique is that it exploits multipath whereas ranging based techniques suffer from it, i.e., multipath ensures unique signal signatures, which is beneficial for fingerprinting whereas it is seen as addi-

tional noise on the measurements, which is disadvantageous for ranging.

The first part of this work (Chapter 2–4) is about indoor location tracking with a focus on signal strength based systems, influence of the human body, and the optimization of model-based radio maps for indoor positioning. The second part (Chapter 5–6) focuses on GPS based map matching with support for lane detection and outdoor positioning based on cellular network data.

Chapter 2 presents a real-time indoor location tracking system based on the Viterbi principle and semantic data. The Viterbi algorithm is a dynamic programming algorithm used in the context of hidden Markov models (HMM) to determine the most likely sequence of hidden states, called the Viterbi path, resulting in the sequence of observed events. To apply this technique for positioning, the states are interpreted as real locations on a floor plan and hence this principle comes down to determining the most likely sequence of positions instead of only the most likely current position. Furthermore, the number of allowed transitions between two grid points is restricted by using semantic data, i.e., the environment of the object that is being tracked and a motion model. This ensures that the reconstructed paths are realistic and physically possible, i.e., no walls are crossed and no unrealistically large distances are traveled within a given time frame. Simulations confirmed that the proposed location tracking system was more robust against measurement noise, especially for networks with smaller node densities, e.g., a simulated noise level of 10 dB resulted in improvements of 70.9%, 52.8%, 54.1%, and 42.1% in mean accuracy compared to a fingerprinting technique, Kalman filter, particle filter, and particle smoother, respectively. An experimental validation conducted in a real office environment resulted in an improvement of 26.1% compared to a basic fingerprinting technique. In a sensitivity analysis it is shown that the access point density, semantic data, path loss model, and grid size have a major impact on the accuracy, required computational power, and memory usage, which is important to work in real-time on low-cost portable devices.

Many positioning systems already try to cope with signal deterioration caused by multipath fading and diffraction. However, another important factor is the influence caused by the user being tracked itself, so-called human body shadowing. This factor is often neglected by verifying the performance of positioning systems by stepwise moving a node placed on a tripod, hereby explicitly removing the human from the equation, while practical human tracking applications always imply the presence of a user's body. The presence of such a user will influence the radio-frequency signal paths between a body-worn tag and the receiving nodes, and can block the line-of-sight depending on the user's orientation and location. This causes additional propagation losses that are currently not accounted for and will generally decrease the accuracy of signal strength based positioning systems. Chapter 3 presents two compensation methods, built on top of the tracking algorithm of Chapter 2, to eliminate the effect of human body shadowing.

The first method combines the measured signal strengths from multiple mobile nodes, placed on different parts of a human body, which show less variation caused by the user's body and hence limits its influence. The second method takes into account the user's orientation towards the fixed infrastructure and the body-worn tag's relative position, and uses a human body loss model to explicitly compensate for the user's influence. Both approaches can be independently combined and reduce the influence caused by body shadowing, improving the mean accuracy from 3.76 m to 2.33 m, corresponding to an improvement of 38.1%.

A major burden of signal strength-based fingerprinting for indoor positioning is the generation and maintenance of a radio map. Model-based radio maps are generated much faster than measurement-based radio maps but are generally not accurate enough. Chapter 4 presents an unsupervised learning technique to construct and optimize model-based radio maps for indoor positioning systems, e.g., to make the radio map more accurate or to automatically cope with changes in an office layout. The input for this unsupervised learning technique are random walks for which the ground truth locations are unknown along with an initial radio map based on a theoretical path loss model, a self-calibration method, and a route mapping filter, i.e., the location tracking algorithm of Chapter 2. No measurement campaign, device calibration, or site survey, which are labor-intensive and time-consuming, or inertial sensor measurements, which are often not available and consume additional power, are needed for this approach. The premise of this technique is that the differences between real measurements and reference values, derived from a model-based radio map, tend to be correlated per room and access point. It was shown by measurements and simulations that the discrepancies between reference fingerprints and real measurements could be learned in various scenarios, based on the random walks that a typical person does. This results in reference fingerprints that match the real measurements more closely and hence will lead to better radio maps and location accuracies. An experimental validation on a testbed in a large office building confirmed the simulations with relative improvements up to 28.6% after unsupervised learning with only 15 min of unlabeled training data.

Chapter 5 presents a fast, memory-efficient, and worldwide map matching algorithm based on raw geographic coordinates and enriched open map data. The proposed algorithm combines the Markovian behavior and the shortest path aspect while taking into account the type and direction of all road segments, information about one-way traffic, maximum allowed speed per road segment, and driving behavior. Furthermore, a lane detection algorithm based on accelerometer readings and traffic lane information from the open map data, that self-adapts to different driving behaviors, is added on top of the map matching algorithm. An experimental validation consisting of 12 trajectories on foot, by bike, and by car, showed the efficiency and accuracy of the proposed algorithms, with an average F1-score and median

error of 99.5 % and 2.09 m for the map matching algorithm and an average F1-score of 86.7% for the lane detection algorithm, which resulted in the correctly estimated lane 93.0% of the time.

Most outdoor tracking algorithms for cellular networks are validated in optimal or controlled environments on a small dataset or are merely validated by simulations. Chapter 6 presents a technique to efficiently track all mobile users residing on a cellular network. It does not depend on prior training data and does not require any cooperation on the mobile side or changes to the network side. The latter is useful for applications where there is typically no cooperation at the mobile side, e.g., traffic monitoring, population movement estimation, or criminal activity detection, and makes it immediately applicable for mobile network operators. This location tracking algorithm exploits enriched open map data, a mode of transportation estimator, and the route filtering of Chapter 5 on top of the cellular network topology and measurements to track the movement and locations of mobile devices. Furthermore, it was evaluated on a nationwide dataset collected and processed in real-time, in cooperation with a major network operator in Belgium. The experimental validation showed that the mode of transportation, smartphone usage, and environment impact the accuracy and that the proposed technique is more robust and outperforms an existing particle filter with map information by 31.0% in median accuracy.

Finally, Chapter 7 concludes this book with a summary of the accomplished work and discusses some directions for future work.

Chapter 1

Introduction

1.1 Context

Location tracking in indoor and outdoor environments has been of great importance in the last decades and is still gaining attraction. The growing amount of available positioning data due to ubiquitous connectivity and the Internet of Things (IoT) has led to many location-based services (LBS). These are a collection of applications that use geographical location data of mobile devices provided by, e.g., WiFi, Bluetooth Low Energy (BLE), Global Positioning System (GPS), or cellular networks, and are situated in many domains, e.g., health care, government, public service, industrial, military, automotive, advertising, retail, or cultural sector. Examples of these location-aware applications are personal navigation, museum guidance, intrusion detection, finding your car in a parking garage, wayfinding in a large shopping mall or hospital, location-based gaming, personalized advertising, security, blue force tracking, emergency 911 services, surveillance, asset tracking, fleet and inventory management, optimizing productivity in manufacturing or distribution, transportation planning, and analyzing traffic patterns [1, 2].

According to studies [3, 4], the worldwide LBS market is valued at around USD 30 billion in 2019 and is expected to reach USD 133 billion by 2023, with a compound annual growth rate (CAGR) of 36.55% during the forecast period. The largest challenges related to the further development and widespread adoption of LBS are privacy concerns associated with location data, availability of up-to-date maps, and the reliability, accuracy, and cost of indoor and outdoor localization or positioning technologies [5].

This chapter provides an overview of the existing localization techniques and technologies, and introduces the available filtering and post-processing

methods, to situate the proposed techniques in this work. Finally, the aim and contributions of this dissertation are presented.

1.2 Localization techniques

Location tracking systems, also known as localization techniques, can be distinguished from each other in multiple ways [6]. These systems can be either active or passive depending on the participation or cooperation of the tracked object or person. A lot of location tracking systems are active and in this case, the object that is being tracked, is equipped with an active tag. This tag sends or receives packets that are received or sent by a fixed infrastructure consisting of access points, also known as fixed nodes or base stations, that form the wireless network [7–9]. The processing of the measurements takes place centrally on a server if the tag is sending packets and takes place locally if the tag is receiving packets from the fixed infrastructure. With passive location tracking systems, also known as device-free localization (DFL), the moving object is not actively participating in the localization process [10–12]. The position is estimated based on changes in the environment, which are in turn caused by the presence and movement of the entity that is being tracked. In contrast to active localization these passive systems benefit from rich multipath channels because the perturbations become the signal and not the noise [13]. This is interesting for non-cooperative localization like intrusion detection, wildlife monitoring, or protecting outdoor assets, such as pipelines, railroad tracks, and perimeters [14].

Another distinction can be made based on the used ranging technique. The best known are: Angle of Arrival (AoA) [15, 16], Angle of Departure (AoD) [17], Time of Arrival (ToA) [18, 19], Time Difference of Arrival (TDoA) [7, 20], Phase Difference of Arrival (PDoA) [21, 22], and Received Signal Strength (RSS) [11, 23–25]. Section 1.3 discusses these technologies in detail.

Besides differentiation on the used signal, another distinction of localization systems can be made, based on the used communication technology. A lot of localization methods use GPS [26], cellular networks [27], or WiFi [28] as information carrying signal. Other location tracking systems use special-purpose hardware and infrastructure like radio-frequency identification (RFID) [29], Bluetooth Low Energy (BLE) [7], or ultra-wideband (UWB) [30, 31]. Section 1.5 discusses these technologies in detail.

Knowledge of the network topology to estimate the distances between a mobile user and a set of base stations, reduces the localization to a triangulation [32, 33] or multilateration [34] problem for all ranging techniques. An

alternative to triangulation or multilateration is pattern matching with, e.g., RSS fingerprints [28]. This technique is also called fingerprinting and has the advantage that most devices are already capable of measuring the RSS, whereas AoA or TDoA systems require dedicated hardware, i.e., directional antennas for AoA and clock synchronization between the receiving nodes for TDoA [35]. The ranging and fingerprinting techniques are discussed in Section 1.3 and Section 1.4.

Besides the signals from existing GPS, cellular networks, WiFi, RFID, BLE, or UWB infrastructures, there are other signals that can be used for positioning or location tracking. These so-called signals of opportunity (SoOPs) are already present in the area of interest, e.g., magnetic field [36], visible light [37], acoustic background noise [38], or FM radio signals [39]. Section 1.6 discusses the SoOPs in detail.

The improved performance of optical systems due to the recent advances in computing power and CCD sensors chips (charge-coupled devices) has triggered image or video based positioning systems to become an attractive alternative for RF-based systems [40]. Typical applications of these vision-based systems are visual navigation for autonomous vehicles and Augmented Reality (AR) [41]. The two main approaches for vision-based positioning are fixed and mobile camera systems [42]. The first one uses fixed cameras in the area of interest to locate a target based on its position within the captured images. In the second one uses a target equipped with a mobile camera and landmarks at known location or extracted features to estimate the camera pose, i.e., location and orientation from which the image is taken [43]. The largest advantage of optical positioning systems is that they are not susceptible to interference or multipath but they become unusable in situations with limited visibility like smoke or fire.

1.3 Ranging

1.3.1 Time of arrival

Time of Arrival (ToA), also known as Time of Flight (ToF) techniques measure the time the radio signals travel between a single transmitter and multiple receivers [35, 44]. The transmitter and receiver are interchangeable and can be a mobile object or user at an unknown position, fixed access points, sensor nodes, or base stations at known locations. ToA measurements are directly correlated to the distance between a transmitter and a receiver because the radio propagation velocity, i.e., the speed of light ¹, is well known. A disadvantage is that the receivers must know the start time

¹speed of light: $2.9979 \cdot 10^8 ms^{-1}$

of source transmission, which requires cooperation and synchronization between the transmitter and receiver, an assumption that severely limits the applicability of ToA-based location tracking systems. Moreover, after the initial synchronization, the clocks can drift away from each other due to the imperfections of clock oscillators. This clock drift causes errors in the measured ToA and hence positioning inaccuracies. A possible solution is Two-Way Ranging (TWR), in its simplest form the round-trip time (RTT) is measured from the start of the transmission until the reception of the replied signal and the ToA is calculated based on this RTT and the reply time [45]. Disadvantages of TWR are that it introduces an additional delay and processing cost, and doubles the amount of transmissions and hence energy consumption.

1.3.2 Time difference of arrival

Another solution to the clock synchronization problem, and hence clock drift between transmitter and receiver, is Time Difference of Arrival (TDoA). In the TDoA technique, time differences between the ToA of multiple radio signals are measured at the clock-synchronized sensor nodes. Note that the subtraction of pairwise ToA measurements leads to correlated noise in TDoA and strengthens the measurement noise by 3 dB [35]. The maximum-likelihood (ML) approach is optimal for TDoA-based location tracking but due to the non-linear and non-convex nature, the performance will strongly depend on the initial estimate. Therefore, the most widely used approach to process TDoA measurements is the two-step weighted least squares (WLS), which is an accurate and computationally efficient technique to minimize the spherical least squares error [46].

1.3.3 Angle of arrival

The Angle of Arrival (AoA) technique determines the propagation direction of a radio-frequency (RF) wave and requires a directional antenna or antenna array at the side of the incoming wave [17]. The position of an unknown mobile device is determined based on the intersection of multiple AoA lines (triangulation) [32]. This technique performs especially well in line-of-sight (LoS) conditions but in practice the location accuracy will drastically deteriorate due to the finite number of antennas in the array, noise, and multipath [47]. Another challenge is the difficulty to obtain accurate results while keeping the system small to enough to be implemented in pocket-size devices due to the typically large dimensions of directional antennas [48].

1.3.4 Phase difference of arrival

The basic Phase Difference of Arrival (PDoA) technique uses two signals with different carrier frequencies that are sent between a tag and a receiver. The phase delays are proportional to their respective carrier frequencies and the phase difference observed at the two frequencies can be used to estimate the distance between a tag and a receiver [22]. A disadvantage of PDoA-based on dual-frequency signaling is the trade-off between the maximum unambiguous range and the sensitivity of range estimation to additive noise [21]. A large separation between the two frequencies reduces this noise sensitivity but severely limits the unambiguous range. A possible solution is to use more carrier frequencies and average the range estimates over multiple frequency pairs.

1.3.5 Signal strength

A lot of location tracking systems in GPS-denied environments rely on signal strength measurements from existing wireless network infrastructures due to their simplicity and availability, e.g., WiFi, ZigBee, or BLE compatible devices. These signal strength measurements can be translated to a location by making use of the fingerprinting technique (discussed in Section 1.4) or the well-known multilateration method after converting the RSS measurement to a distance [49]. Note that a signal propagation or path loss (PL) model is used for the RSS to distance conversion and that these models depend on the considered environment [50].

1.4 Fingerprinting

An alternative for the amplitude-based technique, is estimating the location by looking for the closest match in a pretrained fingerprint database, also known as radio or coverage map. The largest advantage of this fingerprinting technique is that it exploits multipath whereas ranging based techniques suffer from it, i.e., multipath ensures unique signal signatures, which is beneficial for fingerprinting whereas it is seen as additional noise on the measurements, which is disadvantageous for ranging. The fingerprint database is essentially a look-up table that maps possible positions with a vector of associated values, e.g., RSS, CSI, or cell-ID values [51–53]. The fingerprinting technique generally uses reference values of a single type of measurements but to increase the robustness a hybrid approach can be considered, e.g., ToF, AoA, and RSS [54] (Section 1.8). The reference values are collected in an offline phase and can be measurement-based by drive-testing the area

of interest [55–57], an indoor measurement campaign [28, 58], simulation-based by using a propagation model [59], ray tracing [60], or a hybrid approach [61]. Drive-testing, also known as wardriving, is labor-intensive and needs to be redone each time the wireless network or even the environment undergoes changes, e.g., new buildings, renovations, or office lay-out modifications [62]. The simulation-based approach is much faster but will generally lead to less accurate location estimations. Alternatively, a crowd-sourced measurement campaign can be used instead of drive-testing [63].

1.5 Enabling technologies for localization

1.5.1 Indoor

1.5.1.1 RFID

Just as most location tracking systems, RFID comes in two classes: active and passive [64]. Active RFID requires a power source that can be a connection to a powered infrastructure but is usually an integrated battery. In the latter case, the active RFID tag’s lifetime is limited by the stored energy and balanced against the required number of read operations the device must undergo. One of the first location systems that used this technology was Active Badge [65]. Each badge has a globally unique code that is periodically broadcast through an infrared interface. A network of sensors installed in the building picks up these infrared signals that are reflected on furniture and walls. The location of the badge’s owners are estimated based on which badges were seen by which sensors. This real-time sensing makes it possible to automate office tasks and provide authentication for entering a secure area.

Passive RFID is interesting for applications that require a small tag and an indefinite operational time, e.g., in retail where the tags needs to be small enough to fit into a practical adhesive label. Passive RFID tags consist of a conductor chip attached to an antenna and some form of encapsulation [66]. Here, the tag reader is responsible for communication and powering through either magnetic induction (near-field) or electromagnetic wave capture and backscattering (far-field).

These passive RFID systems can be outfitted with additional sensors to extend the range of applications, e.g., to monitor temperature in a cold food chain [67], and are called semi-passive RFID.

Disadvantages of the RFID technology are the limited capabilities, scanning range, and capacity (in terms of simultaneously supported tags). Furthermore, the initial deployment cost is typically higher as it requires a separate infrastructure and it is somewhat outdated. There are also some

security issues with RFID, as unauthorized devices may be able to read and even change data on tags without the knowledge of owner. In a side-channel attack, RFID data can be picked up as it is passed from a tag to a reader, which could give an attacker access to sensitive information or the ability to copy a card [68].

1.5.1.2 WiFi

Most wireless local area networks (WLANs) are based on IEEE 802.11 standards and are marketed under the WiFi brand name. WLANs have become popular for in-home usage, due to their ease of installation and operation. They are widely deployed, which makes them suitable for localization with negligible overhead and without expensive additional hardware costs [24].

One of the first systems to suggest that a large class of location-aware services can be built over a WLAN network is RADAR [28]. This system uses signal strength information from WiFi signals at multiple base stations, positioned to provide overlapping coverage in the area of interest, to estimate the location of a user by multilateration (Section 1.3.5) or fingerprinting (Section 1.4). Recent approaches try to improve the accuracy of WLAN-based systems by using more advanced filtering (Section 1.9), sensor fusion with an inertial measurement unit (IMU) and/or pedestrian dead reckoning (PDR) technique (Section 1.8 and Section 1.7), or channel state information (CSI) [69]. Some advanced WiFi network interface cards (NIC) are able to obtain CSI measurements, which can be used as input to improve the performance of indoor location tracking systems. For example, the Intel WiFi Link 5300 NIC implements an OFDM system with 48 subcarriers and the device driver is able to read the amplitude and phase response of 30 subcarriers, which can reveal completely different properties than RSS. Recently, the Task Group mc (TGmc) added support for fine timing measurements using RTT with the release of IEEE 802.11mc [70].

1.5.1.3 Bluetooth Low Energy

New opportunities are provided with the introduction of the BLE radio protocol [71]. This wireless personal area network (WPAN) technology is designed by the Bluetooth Special Interest Group and is aimed at applications in, e.g., the health care, security, and home entertainment sector. As its name suggests, it is intended to provide a considerably reduced power consumption and a low cost while maintaining a similar communication range compared to Classic Bluetooth. These BLE systems estimate the location of a user by multilateration (Section 1.3.5) or fingerprinting (Section 1.4). A distinct advantage over WiFi is that the technology supports portable

battery-powered beacons, which makes it possible to form a wireless sensor network (WSN) of distributed beacons at a low cost. These WSNs can be used for detecting spatial variations in environmental conditions, such as temperature or pollution [72]. Recently, a long-range mode was added in BLE version 5.0, which has the capability of extending the communication range by using error correcting coding and higher transmitting power [73].

1.5.1.4 Ultra-Wideband

The UWB technology is inherently well suited for location tracking systems in challenging environments due to the large available bandwidth, which results in robust communication in dense multipath or GPS-denied scenarios, a simple implementation for multiple-access communications, and a fine temporal resolution and hence accurate ranging [74]. Localization with UWB systems generally relies on ToA-based (Section 1.3.1) or TDoA-based ranging (Section 1.3.1). It has the ability to penetrate walls and to resolve subnanosecond delays, which result in centimeter-level positioning accuracy. However, the maximum range and location accuracy are in practice significantly degraded due to the non line-of-sight (NLoS) effect when operating indoors, e.g., in office or apartment type spaces [75]. The NLoS effect deteriorates the range measurements with larger dispersion, outliers, and a reduced measurable range to less than 10 m, as a result [76], which makes the UWB technology more suited for short- and mid-range applications. Furthermore, the initial deployment cost will generally be much higher compared to WLAN or BLE.

1.5.2 Light

Visible light from LEDs (light-emitting diodes) offers many advantages compared to conventional light sources, e.g., energy efficiency, life expectancy, and the ability for high speed modulation imperceptible for the human eye [77, 78]. The latter makes it possible to combine illumination and communication, and to assign different codes to LEDs with a known position, enabling receivers to estimate their locations with centimeter-level accuracies based on the received signal strengths [79] (Section 1.3.5). Advantages of these optical positioning systems over the traditional RF-based approaches are that it is free of interference from systems in other rooms and multipath is less challenging because reflected components, if present, will have much lower power than the LoS path [37]. Furthermore, most positions in a room have LoS because the lights are typically placed at regular interval on the ceiling. Disadvantages are the limited range, confined illumination angle, and becoming unusable in situations with limited visibility

like smoke or fire.

1.5.3 Outdoor

1.5.3.1 GPS

The Global Positioning System (GPS) is a satellite-based navigation technique that is ubiquitous due to the worldwide coverage, widespread use, decent accuracy, and adoption in all modern smartphones [80]. The GPS technology is mostly used for outdoor location tracking because the accuracy degrades in indoor environments due to the inability to penetrate most building materials, which makes GPS signals not suited for indoor location tracking. However, even in outdoor environments GPS outages can occur caused by the unavailability of GPS signals from sufficient satellites due to, e.g., mountains, tall buildings, or multi-level overpasses. Possible solutions are geometric intersection [81] or inertial navigation [82–84] (discussed in Section 1.7).

1.5.3.2 Cellular networks

Cellular or network-based localization algorithms locate a mobile user based on measured radio signals from base stations in its vicinity [85]. A well known approach to locate a mobile device with telecommunication data from a network infrastructure is cell-ID based [27]. The mobile user is mapped to the location of its serving base station, i.e., the cell a mobile device is currently connected to. It has a low cost, a short response time, and is easy to implement and applicable in all places with cellular coverage but has a low accuracy for high cell ranges.

The location tracking accuracy can be improved if signal strength measurements or timing information from the surrounding base stations are available [86]. To infer the mobile user's location, the signal strength measurements can be combined with multilateration and a path loss model [87], or a fingerprinting technique [88] (discussed in Section 1.4).

Note that with every new generation of the cellular network technology, the location precision improves due to the higher base station density, adoption of recent advances, e.g., millimeter wave (mmWave) signals, and more accurate timing information, e.g., 3G's propagation delay estimations have a time granularity of 780 ns (corresponding to 234 m) and those of 4G have a time granularity of 260 ns (corresponding to 78 m). In 5G, the main breakthrough will be to the usage of millimeter wave (mmWave) signals [89]. The employment of mmWave has a two-fold advantage: a large available bandwidth and a small signal wavelength. The larger bandwidth

results in reduced latency due to shorter symbol times, increased robustness to multipath, and a higher accuracy of time-based measurements due to the better time resolution [90]. The smaller signal wavelength makes it possible to pack a large number of antenna elements on a small surface, e.g., in a smartphone. This enables massive multiple-input multiple-output (MIMO), giving the possibility of realizing highly directional beamforming capabilities [91].

1.5.3.3 LPWAN: LoRa, Sigfox, NB-IoT

Low-power wide-area networks (LPWANs) are designed to allow long range communications at a low bit rate among connected things, e.g., battery-powered sensors. LoRa (Long Range), Sigfox, and narrowband IoT (NB-IoT) are wireless communication protocols that compete against other [92].

LoRa uses a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology to enable communication on the license-free sub-gigahertz radio frequency bands like 433 MHz, 868 MHz (Europe) and 915 MHz (North America). Sigfox employs the differential binary phase-shift keying (DBPSK) and the Gaussian frequency shift keying (GFSK) to enable communication on 868 MHz (Europe) and 902 MHz (US). NB-IoT uses a subset of the Long-Term Evolution (LTE) standard, but limits the bandwidth to a single narrow-band of 200 kHz. The used modulation techniques are orthogonal frequency-division multiplexing (OFDM) for the downlink communications and single-carrier frequency-division multiple access (SC-FDMA) for the uplink communications.

The low power consumption enables devices to operate for years without replacing the battery [93]. The extremely long range connectivity, typically up to 10 km depending on the environment, is achieved by trading off data rate, i.e., below 50 kbps. Due to this low data rate and limited duty cycle, these systems are only suitable for delay-tolerant applications [94].

1.6 Signals of opportunity

1.6.1 Magnetic

Localization systems that use the magnetic field as a SoOP, exploit the local disturbances of the geomagnetic field that are typical for an indoor environment [36]. Besides the earth's magnetic field, there are other sources that have an influence on the magnetic measurements as well, e.g., electrical machines or magnetic speakers. The local anomalies are caused by metal in the construction materials, e.g., structural steel in a building, and give rise to unique magnetic fingerprints [95], which can be used in a similar way as

RSS measurements in the traditional fingerprinting technique (Section 1.4). Note that the magnetic field intensity data only consists of three components whereas in RSS fingerprinting this depends on the number of visible access points. A disadvantage of geomagnetic-based indoor location tracking systems is that outliers could be placed at any location in the fingerprint map because there are no local references as is the case with the fixed access points of WLAN. Therefore, magnetic fingerprints are often used to enhance the performance of WLAN systems, the latter gives an initial estimation of the correct region and the first refines the estimation based on local variations in the magnetic field [95]. Furthermore, moving objects containing ferromagnetic materials and electronic devices may also affect the magnetic field, which causes deviations from the fingerprint database and hence a lower localization accuracy [96].

1.6.2 Acoustic

The field of architectural acoustics has proven that the geometry of a room and the furniture strongly affect the room impulse response and hence the acoustic background spectrum (ABS) [97]. The intuition behind ABS-based positioning is that modern life is full of noises, e.g., buzzing lights, blowing air conditioners, and whirring computers. The combination of these persistent acoustic drivers and the room impulse responses result in unique acoustic fingerprints. Even similar looking rooms that sound indistinguishable to a human listener are likely to have different persistent acoustic characteristics [38], which can be used in a similar way as RSS measurements in the traditional fingerprinting technique (Section 1.4). Disadvantages of these systems are the performance deterioration due to acoustic noise that is not captured in the fingerprints, e.g., speaking occupants or environmental noise. Lately, acoustic ranging devices have been used in underwater target positioning to automate challenging tasks, e.g., pipeline inspection, ocean floor geodesy, or archaeological site examination [98, 99]. This is because acoustic signals are able to travel further distances, whereas RF signals rapidly attenuate and optical signals scatter, in underwater scenarios [100].

1.6.3 FM

Frequency modulation (FM) is the encoding of information in a carrier wave by varying the instantaneous frequency of the wave. The benefits of analogue FM audio signals for positioning purposes are the global availability (can be received both indoors and outdoors), readily available coordinates and power of the transmission antennas, low-cost, and low-power usage [39, 101]. Furthermore, FM signals are less susceptible to human pres-

ence and are more robust to temporal variations, i.e., vary less over time, due to the longer wavelength and hence lower frequency when compared to WiFi or BLE signals [102]. Similar to the traditional fingerprinting with RSS measurements from WiFi or BLE signals, the same technique can be applied to FM broadcast radio signals. Note that the positioning errors due to FM or WiFi signals are independent, which is interesting to obtain richer fingerprints or for sensor fusion techniques (discussed in Section 1.8).

1.7 Pedestrian dead reckoning

A completely other approach to location tracking is pedestrian dead reckoning (PDR), this technique predicts the current location based on inertial sensor measurements and the previous position [103]. Recent developments of microelectromechanical systems (MEMS) allow embedding multiple inertial sensors, e.g., accelerometers, magnetometers, compasses, and gyroscopes, in smartphones or in a compact IMU [104]. These inertial sensors can be used to detect steps, estimate the stride length, and the direction of motion.

The PDR technique works without a wireless signal and hence no infrastructure, connectivity, physical map information, pretrained databases, or measurement campaign is needed [105, 106]. A disadvantage of these systems is that they are typically prone to drift, i.e., the positioning error accumulates over time due to noisy measurements from low-cost inertial sensors and complicated human movements [107]. Furthermore, the start position and orientation of the IMU must be known in advance and need to be consistent with respect to the object that is being tracked. This can be solved by merging PDR with an external measurement source, e.g., WLAN or BLE measurements but then the advantages of not depending on any infrastructure disappear.

1.8 Sensor fusion

Sensor fusion or hybrid techniques combine two or more of the above mentioned technologies or signal parameters to estimate the location of an object or a user [108]. Smartphone-based location tracking systems often combine the measurements from the embedded sensors, e.g., accelerometer, gyroscope, and magnetometer, with the ability to scan WiFi access points [109] or additional floor plan information [110]. A multi-model system that fuses a wide range of sensor data from commonly available portable devices is extensively discussed in [111]. In outdoor environments, GPS signals can be combined with inertial sensor measurements to save energy by

lowering the duty cycle of the power-consuming GPS module [112]. Both Kalman filters and particle filters are well-suited for sensor fusion of inertial sensor measurements because of the link with the probabilistic transition model [113, 114].

1.9 Filtering and post-processing methods

Indoor and outdoor environments are complex and give rise to multipath, fading, and shadowing effects due to refraction, reflection and scattering from buildings, walls, and obstacles [115]. This results in random perturbations of the measured signals and hence location tracking systems that are often not sufficiently accurate. State-of-the-art location tracking algorithms try to improve the accuracy by using route constraints and advanced filters, e.g., Kalman filtering [116], particle filtering [117], and Hidden Markov Models [118].

1.9.1 Hidden Markov model

The hidden Markov model (HMM) is a technique to model a Markov process with unobservable states, i.e., the model contains an underlying stochastic process that is not directly observable (hidden states), but can be observed through another stochastic process (observations) [118]. In a location tracking system, these hidden states are the discrete locations in an area of interest, e.g., the reference locations in a fingerprint database, and the observations are the sequence of measurements [119]. The goal is to find the most likely explanation for the observation sequence and can be solved efficiently using the Viterbi algorithm [51], which is the topic of Chapter 2. Note that the applications of an HMM are not limited to measurements of an RF signal but can also be used to match user activities, measured with built-in smartphone sensors, to an indoor trajectory [120] or to map noisy GPS coordinates to an outdoor road network [121], which is the topic of Chapter 5.

1.9.2 Kalman filter

Kalman filtering (KF), also known as linear quadratic estimation (LQE), is an algorithm that recursively uses a set of mathematical equations to efficiently estimate an unknown state based on a series of noisy measurements observed over time [122]. In a location tracking system, this KF state is usually the unknown position and velocity of an object or person [123]. The KF results in the optimal solution if the posterior density at every time step is Gaussian but this highly restrictive assumption is hardly fulfilled in

real-life use cases. Furthermore, the KF can be seen as analogous to the HMM, with the key difference that the hidden state variables take values in a continuous space as opposed to the discrete state space that is used by the HMM. Two well-known generalizations of this filter for non-linear systems are the extended Kalman filter (EKF) [116, 124] and the unscented Kalman filter (UKF) [125, 126]. Disadvantages of the Kalman filter are the difficulty of getting a good estimate of the noise covariance matrices (to represent process and measurement noise), and that it is not possible to integrate distributed information like the walking distance, map information, or route constraints [127].

1.9.3 Particle filter

Particle filtering (PF) is a Sequential Monte Carlo (SMC) method, which can be applied to any state-space model and hence generalizes the traditional Kalman filtering method [127, 128]. The key idea behind PF is to represent the posterior distribution by a set of random samples (particles) with associated weights. In a location tracking system, these particles are positions on a map and the filtering consists of four steps [129].

- An initialization based on an initial probability density function (PDF).
- A prediction step that uses a probabilistic transition model to move each particle (predict new state)
- An update step that calculates the new weights based on the observations followed by a normalization of these weights
- A resampling step to generate an unweighted particle set to avoid degeneracy, i.e., particles with low weight are deleted and particles with high weight are duplicated [130].

Advantages of the PF are the ability to take into account map information, the number of particles and way of resampling can be adapted based on the processing possibilities, and motion models can be adapted based on the type of user to be tracked. Disadvantages of the PF are that they are non-deterministic (the same input can produce different output, which makes them harder to debug), computationally expensive (due to the huge amount of particles necessary for a consistently high performance), and there is no measure of confidence in the produced output.

1.10 Cooperative localization

In harsh environments where geographic positioning fails, communication between wireless nodes can be used to improve the accuracy of location information [74]. Usually a location tracking system locates all users or nodes independently without taking into account the relative location of each other. Cooperative localization is a paradigm in which wireless mobile or fixed sensor nodes help each other in a peer-to-peer manner to make measurements and then form a map of the network to determine their locations [131]. This approach has gained interest from the optimization, robotics, and wireless communications communities due to the increased performance in both accuracy and coverage when users or nodes in a wireless network cooperate [132–134].

1.11 User anonymity

A topic that currently attracts a lot of attention is user anonymity. Mobile network operators ensure anonymity between their mobile users by providing a temporary identifier (TMSI) instead of constantly using the longterm unique identifiers (IMSI). Lately, also anonymized location data has become a subject of concern [135, 136]. Countermeasures to tackle these exposed vulnerabilities are proposed in [137, 138]. In [139], simulations are used to calculate the number of devices necessary to locate non-participant individuals in urban environments. They prove that it is possible to track the movement of a significant portion of the population with a high-granularity over long periods of time when a small part of the population is part of a (malicious) sensor network. A dual privacy preserving scheme to protect the trajectory and query privacy of a user in continuous LBS, is presented in [140].

1.12 Main research contributions and outline

The main goal of this work is to improve the quality of indoor and outdoor location tracking systems and to validate their performance and robustness in realistic conditions. The first part of this work (Chapter 2–4) is about indoor location tracking with a focus on signal strength based systems and the second part (Chapter 5 and 6) is about GPS-based map matching with support for lane detection and outdoor location tracking based on cellular network data.

In Chapter 2, a real-time indoor location tracking system based on the Viterbi principle and semantic data is presented ([JT1], [JT9]). This sys-

tem serves as starting point for the compensation method of Chapter 3 to eliminate the effect of human body shadowing, e.g., to reduce the influence of a mobile tag's position with respect to the user being tracked ([JT2], [JT10], [JT11]). Chapter 4 presents an unsupervised learning technique to construct and optimize model-based radio maps or fingerprint databases for indoor location tracking systems, e.g., to make the radio map more accurate or to automatically cope with changes in an office layout ([JT3], [JT13]). Chapter 5 presents a fast, memory-efficient, and worldwide map matching algorithm that can optionally detect the driving lane based on accelerometer readings and open map data. A technique for outdoor location tracking of all mobile users residing on a cellular network, that does not depend on prior training data and does not require any cooperation on the mobile side or changes to the network side, is presented in Chapter 6. The technique was evaluated on a nationwide dataset collected and processed in real-time, in cooperation with a major network operator in Belgium ([JT4]). Finally, Chapter 7 concludes this book with a summary of the accomplished work and discusses some directions for future work.

1.13 Publications

1.13.1 Publications in international journals (A1)

Peer-reviewed publications in journals listed in the ISI Web of Science.

1.13.1.1 As first author

- [JT1] **Jens Trogh**, David Plets, Luc Martens, Wout Joseph. Advanced Real-Time Indoor Tracking Based on the Viterbi Algorithm and Semantic Data. *International Journal of Distributed Sensor Networks* 501 (2015): 271818.
- [JT2] **Jens Trogh**, David Plets, Arno Thielens, Luc Martens, Wout Joseph. Enhanced indoor location tracking through body shadowing compensation. *IEEE Sensors Journal*, 16(7):2105-2114, April 2016.
- [JT3] **Jens Trogh**, Wout Joseph, Luc Martens, David Plets. An Unsupervised Learning Technique to Optimize Radio Maps for Indoor Localization. *Sensors* 19.4 (2019): 752.
- [JT4] **Jens Trogh**, David Plets, Erik Surewaard, Mathias Spiessens, Mathias Versichele, Luc Martens and Wout Joseph. Outdoor Location Tracking of Mobile Devices in Cellular Networks. *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, p. 115, May 2019.

1.13.1.2 As co-author

- [JT5] Xu Gong, **Jens Trogh**, Quentin Braet, Emmeric Tanghe, Prashant Singh, David Plets, Jeroen Hoebeke, Dirk Deschrijver, Tom Dhaene, Luc Martens, Wout Joseph. Measurement-based wireless network planning, monitoring, and reconfiguration solution for robust radio communications in indoor factories. *IET Science, Measurement & Technology* (2016).
- [JT6] Said Benaissa, David Plets, Emmeric Tanghe, **Jens Trogh**, Luc Martens, Leen Vandaele, Leen Verloock, Frank Tuytens, Bart Sonck, Wout Joseph. Internet of animals: Characterisation of LoRa sub-GHz off-body wireless channel in dairy barns. *Electronics Letters* 53.18: 1281-1283 (2017).
- [JT7] Said Benaissa, Frank Tuytens, David Plets, Toon de Pessemier, **Jens Trogh**, Emmeric Tanghe, Luc Martens, Leen Vandaele, Annelies Van Nuffel, Wout Joseph, Bart Sonck. On the use of on-cow accelerometers for the classification of behaviours in dairy barns. *Research in veterinary science* (2017).
- [JT8] Nico Podevijn, **Jens Trogh**, Luc Martens, Pieter Suanet, Kim Hendrikse, David Plets, Wout Joseph. TDoA-Based Outdoor Positioning with Tracking Algorithm in a Public LoRa Network. *Special Issue on Advanced Signal Processing for Wireless Localization Systems in Wireless Communications and Mobile Computing* (2018).

1.13.2 Publications in international conferences (P1)

Peer-reviewed publications presented at international conferences or workshops.

1.13.2.1 As first author

- [JT9] **Jens Trogh**, David Plets, Luc Martens, Wout Joseph. Advanced Indoor Localisation Based on the Viterbi Algorithm and Semantic Data. *8th European Conference on Antennas and Propagation (EuCAP 2014), April 2015, Lisbon, Portugal*
- [JT10] **Jens Trogh**, David Plets, Arno Thielens, Luc Martens, Wout Joseph. Improved Tracking by Mitigating the Influence of the Human Body. *IEEE GLOBECOM 2015 Workshop on Localization for Indoors, Outdoors, and Emerging Networks (LION), December 2015, San Diego, United States*

- [JT11] **Jens Trogh**, David Plets, Arno Thielens, Luc Martens, Wout Joseph. A Comparison of Human Body Compensation Models for RSSI Based Localization and Tracking. *IEEE Second International Smart Cities Conference (ISC2)*, September 2016, Trento, Italy
- [JT12] **Jens Trogh**, David Plets, Luc Martens, Wout Joseph. Bluetooth Low Energy Based Location Tracking for Livestock Monitoring. *European Conference on Precision Livestock Farming (EC-PLF 2017)*, September 2017, Nantes, France
- [JT13] **Jens Trogh**, Luc Martens, Wout Joseph, David Plets. Radio Map Optimization Through Unsupervised Learning for Indoor Localization. *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, September 2018, Nantes, France

1.13.2.2 As co-author

- [JT14] Said Benaissa, Frank Tuytens, David Plets, Toon De Pessemier, **Jens Trogh**, Emmeric Tanghe, Luc Martens, Leen Vandaele, Annelies Van Nuffel, Wout Joseph, Bart Sonck. Behaviours recognition using neck-mounted accelerometers in dairy barns. *European Conference on Precision Livestock Farming (EC-PLF 2017)*, September 2017, Nantes, France
- [JT15] Said Benaissa, Frank Tuytens, David Plets, Toon De Pessemier, **Jens Trogh**, Emmeric Tanghe, Luc Martens, Leen Vandaele, Annelies Van Nuffel Van Nuffel, Wout Joseph and Bart Sonck. Behaviours classification using leg-mounted accelerometers in dairy barns. *European conference dedicated to the future use of ICT in the agri-food sector, bioresource and biomass sector, with World Congress on Computers in Agriculture (EFITA WCCA CONGRESS 2017)*, July 2017, Montpellier, France
- [JT16] David Plets, Alexander Sels, **Jens Trogh**, Kris Vanhecke, Luc Martens, Wout Joseph. An assessment of different optimization strategies for location tracking with an Android application on a smartphone. *11th European Conference on Antennas and Propagation (EuCAP 2017)*, March 2017, Paris, France
- [JT17] Nico Podevijn, **Jens Trogh**, Abdulkadir Karaagac, Jetmir Haxhibeqiri, Jeroen Hoebeke, Luc Martens, Pieter Suanet, Kim Hendrikse, David Plets, Wout Joseph. TDoA-based Outdoor Positioning in a Public LoRa Network. *12th European Conference on Antennas and Propagation (EuCAP 2018)*, April 2018, London, UK

-
- [JT18] David Plets, Nico Podevijn, **Jens Trogh**, Luc Martens, Wout Joseph. Experimental Performance Evaluation of Outdoor TDoA and RSS Positioning in a Public LoRa Network. *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, September 2018, Nantes, France
- [JT19] Nico Podevijn, David Plets, **Jens Trogh**, Abdulkadir Karaagac, Jetmir Haxhibeqiri, Jeroen Hoebeke, Luc Martens, Pieter Suanet, Wout Joseph. Performance Comparison of RSS Algorithms for Indoor Localisation in Large Open Environments. *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, September 2018, Nantes, France

References

- [1] Richard Becker, Ramón Cáceres, Karrie Hanson, Sibren Isaacman, Ji Meng Loh, Margaret Martonosi, James Rowland, Simon Urbanek, Alexander Varshavsky, and Chris Volinsky. *Human mobility characterization from cellular network data*. Communications of the ACM, 56(1):74–82, 2013.
- [2] Serdar Çolak, Lauren P Alexander, Bernardo G Alvim, Shomik R Mehndiratta, and Marta C González. *Analyzing cell phone location data for urban travel: current methods, limitations, and opportunities*. Transportation Research Record: Journal of the Transportation Research Board, (2526):126–135, 2015.
- [3] Mordor Intelligence. *Global Location-based Services Market - Segmented by Location (Indoor, Outdoor), Product (Hardware, Software, Services), Applications, Technology, End User, and Region - Growth, Trends, and Forecast (2018 - 2023)*, 2018. <https://www.researchandmarkets.com/research/8wn3zc>.
- [4] Market Research Future. *Location Based Service Market Report - Forecast till 2023*, 2019. <https://www.marketresearchfuture.com/reports/location-based-service-market-5439>.
- [5] Anahid Basiri, Elena Simona Lohan, Terry Moore, Adam Winstanley, Pekka Peltola, Chris Hill, Pouria Amirian, and Pedro Figueiredo e Silva. *Indoor location based services challenges, requirements and usability of current solutions*. Computer Science Review, 24:1–12, 2017.
- [6] F Viani, P Rocca, G Oliveri, Daniele Trincherro, and A Massa. *Localization, tracking, and imaging of targets in wireless sensor networks: An invited review*. Radio Science, 46(5), 2011.
- [7] Zaher Merhi, Mohamed Elgamel, and Magdy Bayoumi. *A lightweight collaborative fault tolerant target localization system for wireless sensor networks*. IEEE Transactions on Mobile Computing, 8(12):1690–1704, 2009.
- [8] Giovanni Zanca, Francesco Zorzi, Andrea Zanella, and Michele Zorzi. *Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks*. In Proceedings of the workshop on Real-world wireless sensor networks, pages 1–5. ACM, 2008.

- [9] Long Cheng, Chengdong Wu, Yunzhou Zhang, Hao Wu, Mengxin Li, and Carsten Maple. *A survey of localization in wireless sensor network*. International Journal of Distributed Sensor Networks, 2012, 2012.
- [10] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. *Challenges: device-free passive localization for wireless environments*. In Proceedings of the 13th annual ACM international conference on Mobile computing and networking, pages 222–229. ACM, 2007.
- [11] Moustafa Seifeldin, Ahmed Saeed, Ahmed E Kosba, Amr El-Keyi, and Moustafa Youssef. *Nuzzer: A large-scale device-free passive localization system for wireless environments*. Mobile Computing, IEEE Transactions on, 12(7):1321–1334, 2013.
- [12] Yang Zhao, Neal Patwari, Jeff M Phillips, and Suresh Venkatasubramanian. *Radio tomographic imaging and tracking of stationary and moving people via kernel distance*. In Proceedings of the 12th international conference on Information processing in sensor networks, pages 229–240. ACM, 2013.
- [13] Neal Patwari and Joey Wilson. *RF sensor networks for device-free localization: Measurements, models, and algorithms*. Proceedings of the IEEE, 98(11):1961–1973, 2010.
- [14] Joey Wilson and Neal Patwari. *See-through walls: Motion tracking using variance-based radio tomography networks*. Mobile Computing, IEEE Transactions on, 10(5):612–621, 2011.
- [15] Eiman Elnahrawy, John Austen-Francisco, and Richard P Martin. *Adding angle of arrival modality to basic rss location management techniques*. In Wireless Pervasive Computing, 2007. ISWPC’07. 2nd International Symposium on. IEEE, 2007.
- [16] Leila Gazzah, Leila Najjar, and Hichem Besbes. *Selective hybrid RSS/AOA weighting algorithm for NLOS intra cell localization*. In 2014 IEEE Wireless Communications and Networking Conference (WCNC), pages 2546–2551. IEEE, 2014.
- [17] Xinning Wei, Nico Palleit, and Tobias Weber. *AOD/AOA/TOA-based 3D positioning in NLOS multipath environments*. In Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on, pages 1289–1293. IEEE, 2011.

- [18] Davide Dardari, Andrea Conti, Ulric Ferner, Andrea Giorgetti, and Moe Z Win. *Ranging with ultrawide bandwidth signals in multipath environments*. Proceedings of the IEEE, 97(2):404–426, 2009.
- [19] Ismail Guvenc and Chia-Chin Chong. *A survey on TOA based wireless localization and NLOS mitigation techniques*. IEEE Communications Surveys & Tutorials, 11(3), 2009.
- [20] Yee Ming Chen, Chi-Li Tsai, and Ren-Wei Fang. *TDOA/FDOA Mobile Target Localization and Tracking with Adaptive Extended Kalman Filter*. In 2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), pages 202–206. IEEE, 2017.
- [21] Xin Li, Yimin Zhang, and Moeness G Amin. *Multifrequency-based range estimation of RFID tags*. In 2009 IEEE International Conference on RFID, pages 147–154. IEEE, 2009.
- [22] Yimin Zhang, Xin Li, and Moeness Amin. *Principles and techniques of RFID positioning*. RFID systems: Research trends and challenges, pages 389–415, 2010.
- [23] Ali H Sayed, Alireza Tarighat, and Nima Khajehnouri. *Network-based wireless location: challenges faced in developing techniques for accurate wireless location information*. IEEE signal processing magazine, 22(4):24–40, 2005.
- [24] Moustafa Youssef and Ashok Agrawala. *The Horus WLAN location determination system*. In Proceedings of the 3rd international conference on Mobile systems, applications, and services, pages 205–218. ACM, 2005.
- [25] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. *Indoor localization without the pain*. In Proceedings of the sixteenth annual international conference on Mobile computing and networking, pages 173–184. ACM, 2010.
- [26] P Misra and P Enge. *Special issue on global positioning system*. Proceedings of the IEEE, 87(1):3–15, 1999.
- [27] Fredrik Gustafsson and Fredrik Gunnarsson. *Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements*. IEEE Signal processing magazine, 22(4):41–53, 2005.

- [28] Paramvir Bahl and Venkata N Padmanabhan. *RADAR: An in-building RF-based user location and tracking system*. In INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 2, pages 775–784. IEEE, 2000.
- [29] Lionel M Ni, Dian Zhang, and Michael R Souryal. *RFID-based localization and tracking technologies*. IEEE Wireless Communications, 18(2), 2011.
- [30] Paul Meissner, Erik Leitingner, Markus Fröhle, and Klaus Witrisal. *Accurate and robust indoor localization systems using ultra-wideband signals*. arXiv preprint arXiv:1304.7928, 2013.
- [31] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A Al-Ammar, and Hend S Al-Khalifa. *Ultra wideband indoor positioning technologies: Analysis and recent advances*. Sensors, 16(5):707, 2016.
- [32] Richard I Hartley and Peter Sturm. *Triangulation*. Computer vision and image understanding, 68(2):146–157, 1997.
- [33] Vicente Osa, Joaquín Matamales, Jose F Monserrat, and Javier López. *Localization in wireless networks: the potential of triangulation techniques*. Wireless personal communications, pages 1–14, 2013.
- [34] Zheng Yang and Yunhao Liu. *Quality of trilateration: Confidence-based iterative localization*. IEEE Transactions on parallel and distributed systems, 21(5):631–640, 2009.
- [35] Enyang Xu, Zhi Ding, and Soura Dasgupta. *Source localization in wireless sensor networks from signal time-of-arrival measurements*. IEEE Transactions on Signal Processing, 59(6):2887–2897, 2011.
- [36] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman. *Indoor location sensing using geomagnetism*. In Proceedings of the 9th international conference on Mobile systems, applications, and services, pages 141–154. ACM, 2011.
- [37] Thomas Q Wang, Y Ahmet Sekercioglu, Adrian Neild, and Jean Armstrong. *Position accuracy of time-of-arrival based ranging using visible light with application in indoor localization systems*. Journal of Light-wave Technology, 31(20):3302–3308, 2013.

- [38] Stephen P Tarzia, Peter A Dinda, Robert P Dick, and Gokhan Memik. *Indoor localization without infrastructure using the acoustic background spectrum*. In Proceedings of the 9th international conference on Mobile systems, applications, and services, pages 155–168. ACM, 2011.
- [39] V. Moghtadaiee and A.G. Dempster. *Indoor Location Fingerprinting Using FM Radio Signals*. Broadcasting, IEEE Transactions on, 60(2):336–346, June 2014.
- [40] Rainer Mautz and Sebastian Tilch. *Survey of optical indoor positioning systems*. In 2011 international conference on indoor positioning and indoor navigation, pages 1–7. IEEE, 2011.
- [41] Mark Billinghurst, Adrian Clark, Gun Lee, et al. *A survey of augmented reality*. Foundations and Trends® in Human–Computer Interaction, 8(2-3):73–272, 2015.
- [42] Luca Mainetti, Luigi Patrono, and Ilaria Sergi. *A survey on indoor positioning systems*. In 2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pages 111–120. IEEE, 2014.
- [43] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. *Efficient & effective prioritized matching for large-scale image-based localization*. IEEE transactions on pattern analysis and machine intelligence, 39(9):1744–1756, 2016.
- [44] Zohair Abu-Shaban, Xiangyun Zhou, and Thushara D Abhayapala. *A novel TOA-based mobile localization technique under mixed LOS/NLOS conditions for cellular networks*. IEEE Transactions on Vehicular Technology, 65(11):8841–8853, 2016.
- [45] Cung Lian Sang, Michael Adams, Timm Hörmann, Marc Hesse, Mario Porrmann, and Ulrich Rückert. *An Analytical Study of Time of Flight Error Estimation in Two-Way Ranging Methods*. In 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–8. IEEE, 2018.
- [46] Gang Wang, Shu Cai, Youming Li, and Nirwan Ansari. *A bias-reduced nonlinear WLS method for TDOA/FDOA-based source localization*. IEEE Transactions on Vehicular Technology, 65(10):8603–8615, 2016.
- [47] Sheng Xu and Kutluyıl Doğançay. *Optimal sensor placement for 3-D angle-of-arrival target localization*. IEEE Transactions on Aerospace and Electronic Systems, 53(3):1196–1211, 2017.

- [48] 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.
- [49] Federico Thomas and Lluís Ros. *Revisiting trilateration for robot localization*. IEEE Transactions on robotics, 21(1):93–101, 2005.
- [50] Kareem El-Kafrawy, Moustafa Youssef, Amr El-Keyi, and Ayman Naguib. *Propagation modeling for accurate indoor WLAN RSS-based localization*. In 2010 IEEE 72nd Vehicular Technology Conference-Fall, pages 1–5. IEEE, 2010.
- [51] Jens Trogh, David Plets, Luc Martens, and Wout Joseph. *Advanced Real-Time Indoor Tracking Based on the Viterbi Algorithm and Semantic Data*. International Journal of Distributed Sensor Networks, 501:271818, 2015.
- [52] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. *CSI-based fingerprinting for indoor localization: A deep learning approach*. IEEE Transactions on Vehicular Technology, 66(1):763–776, 2017.
- [53] Torbjörn Wigren. *Adaptive enhanced cell-ID fingerprinting localization by clustering of precise position measurements*. IEEE Transactions on Vehicular Technology, 56(5):3199–3209, 2007.
- [54] Muhammad Waqas Khan, Naveed Salman, and Andrew H Kemp. *Enhanced hybrid positioning in wireless networks I: AoA-ToA*. In 2014 International conference on telecommunications and multimedia (TEMU), pages 86–91. IEEE, 2014.
- [55] Mike Chen, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian Smith, and Alex Varshavsky. *Practical metropolitan-scale positioning for gsm phones*. UbiComp 2006: Ubiquitous Computing, pages 225–242, 2006.
- [56] Avik Ray, Supratim Deb, and Pantelis Monogioudis. *Localization of LTE measurement records with missing information*. In Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on, pages 1–9. IEEE, 2016.
- [57] Mohamed Ibrahim and Moustafa Youssef. *CellSense: An accurate energy-efficient GSM positioning system*. IEEE Transactions on Vehicular Technology, 61(1):286–296, 2012.

- [58] Chavalit Koweerawong, Komwut Wipusitwarakun, and Kamol Kaemarungsi. *Indoor localization improvement via adaptive RSS fingerprinting database*. In The International Conference on Information Networking 2013 (ICOIN), pages 412–416. IEEE, 2013.
- [59] David Plets, Wout Joseph, Kris Vanhecke, Emmeric Tanghe, and Luc Martens. *Coverage prediction and optimization algorithms for indoor environments*. EURASIP Journal on Wireless Communications and Networking, 2012(1):1–23, 2012.
- [60] Vladimir Savic, Henk Wymeersch, and Erik G Larsson. *Target tracking in confined environments with uncertain sensor positions*. IEEE Transactions on Vehicular Technology, 65(2):870–882, 2016.
- [61] Ahmad Hatami and Kaveh Pahlavan. *Comparative statistical analysis of indoor positioning using empirical data and indoor radio channel models*. In Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE, volume 2, pages 1018–1022. IEEE, 2006.
- [62] Jens Trogh, Wout Joseph, Luc Martens, and David Plets. *An Un-supervised Learning Technique to Optimize Radio Maps for Indoor Localization*. Sensors, 19(4):752, 2019.
- [63] Robert Margolies, Richard Becker, Simon Byers, Supratim Deb, Ritwik Jana, Simon Urbanek, and Chris Volinsky. *Can you find me now? Evaluation of network-based localization in a 4G LTE network*. In INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, pages 1–9. IEEE, 2017.
- [64] Roy Want. *An introduction to RFID technology*. IEEE pervasive computing, (1):25–33, 2006.
- [65] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. *The active badge location system*. ACM Transactions on Information Systems (TOIS), 10(1):91–102, 1992.
- [66] Harald Vogt. *Multiple object identification with passive RFID tags*. In IEEE International Conference on Systems, Man and Cybernetics, volume 3, pages 6–pp. IEEE, 2002.
- [67] Reiner Jedermann, Luis Ruiz-Garcia, and Walter Lang. *Spatial temperature profiling by semi-passive RFID loggers for perishable food transportation*. Computers and Electronics in Agriculture, 65(2):145–154, 2009.

- [68] Rixin Xu, Liehuang Zhu, An Wang, Xiaojiang Du, Kim-Kwang Raymond Choo, Guoshuang Zhang, and Keke Gai. *Side-channel attack on a protected rfid card*. IEEE Access, 6:58395–58404, 2018.
- [69] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. *DeepFi: Deep learning for indoor fingerprinting using channel state information*. In 2015 IEEE wireless communications and networking conference (WCNC), pages 1666–1671. IEEE, 2015.
- [70] Edward Au. *The Latest Progress on IEEE 802.11 mc and IEEE 802.11 ai [Standards]*. IEEE Vehicular Technology Magazine, 11(3):19–21, 2016.
- [71] Ramsey Faragher and Robert Harle. *Location fingerprinting with bluetooth low energy beacons*. IEEE journal on Selected Areas in Communications, 33(11):2418–2428, 2015.
- [72] Joy Dutta, Firoj Gazi, Sarbani Roy, and Chandreyee Chowdhury. *AirSense: Opportunistic crowd-sensing based air quality monitoring system for smart city*. In 2016 IEEE SENSORS, pages 1–3. IEEE, 2016.
- [73] Heikki Karvonen, Konstantin Mikhaylov, Matti Hämäläinen, Jari Iinatti, and Carlos Pomalaza-Ráez. *Interference of wireless technologies on BLE based WBANs in hospital scenarios*. In 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pages 1–6. IEEE, 2017.
- [74] Henk Wymeersch, Jaime Lien, and Moe Z Win. *Cooperative localization in wireless networks*. Proceedings of the IEEE, 97(2):427–450, 2009.
- [75] Antonio Ramón Jiménez Ruiz and Fernando Seco Granja. *Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis*. IEEE Transactions on instrumentation and Measurement, 66(8):2106–2117, 2017.
- [76] Antonio Ramón Jiménez and Fernando Seco. *Comparing Decawave and Bespoon UWB location systems: Indoor/outdoor performance analysis*. In 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–8. IEEE, 2016.
- [77] Giulio Cossu, Marco Presi, Raffaele Corsini, Pallab Choudhury, Amir Masood Khalid, and Ernesto Ciaramella. *A visible light localization aided optical wireless system*. In 2011 IEEE GLOBECOM Workshops (GC Wkshps), pages 802–807. IEEE, 2011.

- [78] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. *Luxa-pose: Indoor positioning with mobile phones and visible light*. In Proceedings of the 20th annual international conference on Mobile computing and networking, pages 447–458. ACM, 2014.
- [79] W Zhang and M Kavehrad. *A 2-D indoor localization system based on visible light LED*. In 2012 IEEE photonics society summer topical meeting series, pages 80–81. IEEE, 2012.
- [80] Rashmi Bajaj, Samantha Lalinda Ranaweera, and Dharma P Agrawal. *GPS: location-tracking technology*. Computer, 35(4):92–94, 2002.
- [81] Omprakash Kaiwartya, Yue Cao, Jaime Lloret, Sushil Kumar, Nauman Aslam, Rupak Kharel, Abdul Hanan Abdullah, and Rajiv Ratn Shah. *Geometry-based localization for gps outage in vehicular cyber physical systems*. IEEE Transactions on Vehicular Technology, 67(5):3800–3812, 2018.
- [82] Santiago Alban, Dennis M Akos, Stephen M Rock, and Demoz Gebre-Egziabher. *Performance analysis and architectures for INS-aided GPS tracking loops*. In Proceedings of the Institute of Navigation National Technical Meeting, pages 611–622. Institute of Navigation, 2003.
- [83] Jan Wendel, Oliver Meister, Christian Schlaile, and Gert F Trommer. *An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter*. Aerospace Science and Technology, 10(6):527–533, 2006.
- [84] Xu Li and Qimin Xu. *A reliable fusion positioning strategy for land vehicles in GPS-denied environments based on low-cost sensors*. IEEE Transactions on Industrial Electronics, 64(4):3205–3215, 2017.
- [85] Jakub Borkowski and Jukka Lempiäinen. *Practical network-based techniques for mobile positioning in UMTS*. EURASIP Journal on Applied Signal Processing, 2006:149–149, 2006.
- [86] Martin Hellebrandt and Rudolf Mathar. *Location tracking of mobiles in cellular radio networks*. IEEE transactions on Vehicular Technology, 48(5):1558–1562, 1999.
- [87] Ahmed Iyanda Sulyman, AlMuthanna T Nassar, Mathew K Samimi, George R MacCartney, Theodore S Rappaport, and Abdulhameed Alsanie. *Radio propagation path loss models for 5G cellular networks in the 28 GHz and 38 GHz millimeter-wave bands*. IEEE Communications Magazine, 52(9):78–86, 2014.

- [88] S. Coleri Ergen, H.S. Tetikol, M. Kontik, R. Sevlian, R. Rajagopal, and P. Varaiya. *RSSI-Fingerprinting-Based Mobile Phone Localization With Route Constraints*. Vehicular Technology, IEEE Transactions on, 63(1):423–428, Jan 2014.
- [89] Stefania Bartoletti, Andrea Conti, Davide Dardari, and Andrea Giorgetti. *5G Localization and Context-Awareness*. 2018.
- [90] Henk Wymeersch, Gonzalo Seco-Granados, Giuseppe Destino, Davide Dardari, and Fredrik Tufvesson. *5G mmWave positioning for vehicular networks*. IEEE Wireless Communications, 24(6):80–86, 2017.
- [91] Wonil Roh, Ji-Yun Seol, Jeongho Park, Byunghwan Lee, Jaekon Lee, Yungsoo Kim, Jaeweon Cho, Kyungwhoon Cheun, and Farshid Aryanfar. *Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results*. IEEE communications magazine, 52(2):106–113, 2014.
- [92] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. *A survey on LPWA technology: LoRa and NB-IoT*. Ict Express, 3(1):14–21, 2017.
- [93] Nico Podevijn, David Plets, Jens Trogh, Luc Martens, Pieter Suanet, Kim Hendrikse, and Wout Joseph. *TDoA-based outdoor positioning with tracking algorithm in a public LoRa network*. Wireless Communications and Mobile Computing, 2018, 2018.
- [94] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. *Understanding the limits of LoRaWAN*. IEEE Communications Magazine, 55(9):34–40, 2017.
- [95] Yuanchao Shu, Cheng Bo, Guobin Shen, Chunshui Zhao, Liqun Li, and Feng Zhao. *Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing*. IEEE Journal on Selected Areas in Communications, 33(7):1443–1457, 2015.
- [96] Binghao Li, Thomas Gallagher, Andrew G Dempster, and Chris Rizos. *How feasible is the use of magnetic field alone for indoor positioning?* In 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–9. IEEE, 2012.
- [97] Ivan Dokmanić, Yue M Lu, and Martin Vetterli. *Can one hear the shape of a room: The 2-D polygonal case*. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 321–324. IEEE, 2011.

- [98] David Moreno-Salinas, Antonio Pascoal, and Joaquin Aranda. *Optimal sensor placement for acoustic underwater target positioning with range-only measurements*. IEEE Journal of Oceanic Engineering, 41(3):620–643, 2016.
- [99] Zhao Li, Stan E Dosso, and Dajun Sun. *Motion-compensated acoustic localization for underwater vehicles*. IEEE Journal of Oceanic Engineering, 41(4):840–851, 2016.
- [100] Melike Erol-Kantarci, Hussein T Mouftah, and Sema Oktug. *A survey of architectures and localization techniques for underwater acoustic sensor networks*. IEEE Communications Surveys & Tutorials, 13(3):487–502, 2011.
- [101] Sungro Yoon, Kyunghan Lee, YeoCheon Yun, and Injong Rhee. *ACMI: FM-based indoor localization via autonomous fingerprinting*. IEEE Transactions on Mobile Computing, 15(6):1318–1332, 2016.
- [102] Yin Chen, Dimitrios Lymberopoulos, Jie Liu, and Bodhi Priyantha. *FM-based indoor localization*. In Proceedings of the 10th international conference on Mobile systems, applications, and services, pages 169–182. ACM, 2012.
- [103] Moustafa Alzantot and Moustafa Youssef. *UPTIME: Ubiquitous pedestrian tracking using mobile phones*. In Wireless Communications and Networking Conference (WCNC), 2012 IEEE, pages 3204–3209. IEEE, 2012.
- [104] Antonio R Jimenez, Fernando Seco, Carlos Prieto, and Jorge Guevara. *A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU*. In Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on, pages 37–42. IEEE, 2009.
- [105] Ionut Constandache, Romit Roy Choudhury, and Injong Rhee. *Towards mobile phone localization without war-driving*. In Infocom, 2010 proceedings ieee, pages 1–9. IEEE, 2010.
- [106] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. *No need to war-drive: Unsupervised indoor localization*. In Proceedings of the 10th international conference on Mobile systems, applications, and services, pages 197–210. ACM, 2012.
- [107] Wonho Kang and Youngnam Han. *SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization*. IEEE Sensors journal, 15(5):2906–2916, 2015.

- [108] Slavisa Tomic, Marko Beko, and Rui Dinis. *Distributed RSS-AoA based localization with unknown transmit powers*. IEEE Wireless Communications Letters, 5(4):392–395, 2016.
- [109] William Wei-Liang Li, Ronald A Iltis, and Moe Z Win. *A smartphone localization algorithm using RSSI and inertial sensor measurement fusion*. In 2013 IEEE Global Communications Conference (GLOBE-COM), pages 3335–3340. IEEE, 2013.
- [110] Stefan Knauth and Athanasios Koukofikis. *Smartphone positioning in large environments by sensor data fusion, particle filter and FCWC*. In 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–5. IEEE, 2016.
- [111] Maarten Weyn. *Opportunistic Seamless Localization*. Lulu. com, 2011.
- [112] Honghui Qi and John B Moore. *Direct Kalman filtering approach for GPS/INS integration*. IEEE Transactions on Aerospace and Electronic Systems, 38(2):687–693, 2002.
- [113] Yuan Zhuang and Naser El-Sheimy. *Tightly-coupled integration of WiFi and MEMS sensors on handheld devices for indoor pedestrian navigation*. IEEE Sensors Journal, 16(1):224–234, 2016.
- [114] Hyungjin Kim, Bingbing Liu, Chi Yuan Goh, Serin Lee, and Hyun Myung. *Robust vehicle localization using entropy-weighted particle filter-based data fusion of vertical and road intensity information for a large scale urban area*. IEEE Robotics and Automation Letters, 2(3):1518–1524, 2017.
- [115] Homayoun Hashemi. *The indoor radio propagation channel*. Proceedings of the IEEE, 81(7):943–968, 1993.
- [116] Yu-Liang Hsu, Jeen-Shing Wang, and Che-Wei Chang. *A wearable inertial pedestrian navigation system with quaternion-based extended Kalman filter for pedestrian localization*. IEEE Sensors Journal, 17(10):3193–3206, 2017.
- [117] Zhenghuan Wang, Heng Liu, Shengxin Xu, Xiangyuan Bu, and Jianping An. *A diffraction measurement model and particle filter tracking method for RSS-based DFL*. IEEE Journal on Selected Areas in Communications, 33(11):2391–2403, 2015.
- [118] Jochen Seitz, Thorsten Vaupel, Steffen Meyer, Javier Gutiérrez Boronat, and Jörn Thielecke. *A hidden markov model for pedestrian*

- navigation*. In 2010 7th Workshop on Positioning, Navigation and Communication, pages 120–127. IEEE, 2010.
- [119] Carlo Morelli, Monica Nicoli, Vittorio Rampa, and Umberto Spagnolini. *Hidden Markov models for radio localization in mixed LOS/NLOS conditions*. IEEE Transactions on Signal Processing, 55(4):1525–1542, 2007.
 - [120] Baoding Zhou, Qingquan Li, Qingzhou Mao, Wei Tu, and Xing Zhang. *Activity sequence-based indoor pedestrian localization using smart-phones*. IEEE Transactions on Human-Machine Systems, 45(5):562–574, 2015.
 - [121] Paul Newson and John Krumm. *Hidden Markov map matching through noise and sparseness*. In Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems, pages 336–343. ACM, 2009.
 - [122] Greg Welch, Gary Bishop, et al. *An introduction to the Kalman filter*. 1995.
 - [123] İsmail Güvenc. *Enhancements to RSS based indoor tracking systems using Kalman filters*. PhD thesis, University of New Mexico, 2003.
 - [124] Thomas Moore and Daniel Stouch. *A generalized extended kalman filter implementation for the robot operating system*. In Intelligent Autonomous Systems 13, pages 335–348. Springer, 2016.
 - [125] Eric A Wan and Rudolph Van Der Merwe. *The unscented Kalman filter for nonlinear estimation*. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373), pages 153–158. Ieee, 2000.
 - [126] Siamak Yousefi, Xiao-Wen Chang, and Benoit Champagne. *Mobile localization in non-line-of-sight using constrained square-root unscented Kalman filter*. IEEE Transactions on Vehicular Technology, 64(5):2071–2083, 2015.
 - [127] Hui Wang, Henning Lenz, Andrei Szabo, Joachim Bamberger, and Uwe D Hanebeck. *WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors*. In 2007 4th workshop on positioning, navigation and communication, pages 1–7. IEEE, 2007.
 - [128] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*. IEEE Transactions on signal processing, 50(2):174–188, 2002.

- [129] Changjiang Yang, Ramani Duraiswami, and Larry Davis. *Fast multiple object tracking via a hierarchical particle filter*. In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, volume 1, pages 212–219. IEEE, 2005.
- [130] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman filter*. IEEE Aerospace and Electronic Systems Magazine, 19(7):37–38, 2004.
- [131] Neal Patwari, Joshua N Ash, Spyros Kyperountas, Alfred O Hero, Randolph L Moses, and Neiyer S Correal. *Locating the nodes: cooperative localization in wireless sensor networks*. IEEE Signal processing magazine, 22(4):54–69, 2005.
- [132] Alexander T Ihler, John W Fisher, Randolph L Moses, and Alan S Willsky. *Nonparametric belief propagation for self-localization of sensor networks*. IEEE Journal on Selected Areas in Communications, 23(4):809–819, 2005.
- [133] Andrea Conti, Matteo Guerra, Davide Dardari, Nicolo Decarli, and Moe Z Win. *Network experimentation for cooperative localization*. IEEE Journal on Selected Areas in Communications, 30(2):467–475, 2012.
- [134] Jing Dong, Erik Nelson, Vadim Indelman, Nathan Michael, and Frank Dellaert. *Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach*. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 5807–5814. IEEE, 2015.
- [135] Hui Zang and Jean Bolot. *Anonymization of location data does not work: A large-scale measurement study*. In Proceedings of the 17th annual international conference on Mobile computing and networking, pages 145–156. ACM, 2011.
- [136] Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. *New privacy issues in mobile telephony: fix and verification*. In Proceedings of the 2012 ACM conference on Computer and communications security, pages 205–216. ACM, 2012.
- [137] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, and Mark Ryan. *Privacy through Pseudonymity in Mobile Telephony Systems*. In NDSS, 2014.

- [138] Altaf Shaik, Ravishankar Borgaonkar, N Asokan, Valtteri Niemi, and Jean-Pierre Seifert. *Practical attacks against privacy and availability in 4G/LTE mobile communication systems*. arXiv preprint arXiv:1510.07563, 2015.
- [139] Nathaniel Husted and Steven Myers. *Mobile location tracking in metro areas: malnets and others*. In Proceedings of the 17th ACM conference on Computer and communications security, pages 85–96. ACM, 2010.
- [140] Shaobo Zhang, Guojun Wang, Md Zakirul Alam Bhuiyan, and Qin Liu. *A dual privacy preserving scheme in continuous location-based services*. IEEE Internet of Things Journal, 5(5):4191–4200, 2018.

Part I

Indoor localization

Chapter 2

Indoor location tracking based on signal strength data

2.1 Introduction

Indoor environments are complex and give rise to multipath and shadowing effects due to refraction, reflection, and scattering from walls and obstacles [1]. This chapter presents a real-time indoor location tracking system based on the Viterbi principle and semantic data, to cope with these influences. The Viterbi principle is used in combination with semantic data to limit possible transitions between positions and hence improve the location tracking accuracy. The semantic data comprises the environment of the object that is being tracked and a motion model. The focus of this work is on accurately tracking a person through a building-wide environment in real-time while using the existing wireless sensor network (WSN) or wireless local area network (WLAN) infrastructure to avoid the need for dedicated positioning hardware.

The remainder of this chapter is structured as follows, Section 2.2 describes related work and Section 2.3 presents the path loss models that are used to construct a radio map for the fingerprinting technique. This approach is sometimes called model-based or predicted fingerprinting since the fingerprints in the radio map are not based on real measurements. This avoids an expensive and time-consuming measurement campaign, which makes the positioning system very fast to deploy. The proposed location tracking algorithm and its optimizations are discussed in Section 2.4. The simulations and experimental validation in a building-wide testbed are de-

scribed in Sections 2.6 and 2.7. Furthermore, a sensitivity analysis was conducted in Section 2.8 to estimate the influence of node density, grid size, memory usage, and semantic data on the performance. Finally, in Section 2.9, conclusions are provided.

2.2 Related work

A positioning method based on path planning that uses a weighted maximum-likelihood estimation (MLE) for the location prediction while the path planning model constrains the movement trajectory of the mobile target, is presented in [2]. The paths on which a target can move are vastly restricted and the approach is solely verified by simulations. The average positioning error depends on the simulated transmission range and average connectivity of the anchor nodes, and lies between 1.5 m and 3 m. In [3], a positioning algorithm based on maximum a posteriori probability (MAP) and RSS ranging is proposed. The goal is to locate the access points (sensor nodes), whereas in this work the location of the fixed access points are known in advance and the goal is to track a mobile user. A performance evaluation was carried out based on simulations but no practical experiments were performed. In [4], a real-time particle filter for 2D and 3D hybrid indoor positioning is presented. Floor plan restrictions and a particle smoother are used to correct previous positions. The obtained accuracy is 2 m with the particle filter and 1.4 m with the smoother. The difference with this work is that besides WLAN-based position measurements, it also depends on a hand-held inertial sensor unit and a barometer. Also, the learning phase depends on training data for the positioning with WLAN, which implies a time-consuming measurement campaign. In [5], a positioning system is proposed, based on frequency modulation (FM) broadcast as a signal of opportunity (SoOP). It combines both deterministic and probabilistic techniques and the obtained accuracy is 2.5 m when 150, manually recorded, reference points are used. In [6], a grid based filter and Viterbi algorithm are used as the central processor for data fusion to estimate the location. Motion dynamics information (MDI), which are similar to inertial measurement unit (IMU) data, measured with smartphone sensors, are used to calculate the state transition probabilities. In [7], a WiFi positioning method to locate mobile terminals is presented. A hybrid model based on the RADAR [8] model and Friis-based calibrated model is suggested [9]. The RADAR model is improved by taking topological elements into account, for which the neighborhood of each point needs to be given in advance. Our approach does not rely on MDI or IMU data but instead solely relies on a floor plan to automatically take into account the environment, e.g., walls, doors, and obstacles. Furthermore, extensive

experiments were conducted to validate the proposed approach instead of solely relying on simulations.

2.3 RSS fingerprinting

The starting point of the proposed location tracking algorithm is an RSS fingerprinting technique [10]. This technique differs from traditional ranging and multilateration techniques because the location is not determined based on estimated distances between transmitter and receiver. Instead, it consists of two phases: an offline training phase and an online positioning phase.

During the offline training phase a radio map of the area of interest is constructed. This radio map contains the signal strength values at every possible grid point for all fixed APs or sensor nodes, e.g., ZigBee [11]. The radio map and signal strength values are also known as fingerprint database and reference fingerprints. The grid points on the floor plan represent the positions where an object or person that is being tracked, can be located. The density of these grid points is determined by the resolution or grid size, this is the distance between two neighboring grid points. The reference fingerprints can be calculated with a theoretical model, ray-tracer, or obtained through a measurement campaign. During the online phase, the user's location is estimated by looking for the closest match in signal space, i.e., by comparing the measured signal strengths to the reference fingerprints. This closest match is where a certain metric, e.g., Euclidean or Manhattan distance, is the lowest.

An advanced indoor path loss model was used to construct the fingerprint database, avoiding an expensive and time-consuming measurement campaign (WHIPP tool [12]). This approach results in a slightly reduced accuracy but allows an immediate deployment. The only prerequisite to generate the fingerprint database for a certain building is to draw its floor plan with the right materials. Most common materials are already available in the tool: brick, drywall, wood, glass, and metal, both in thin and thick format, with material parameters from literature. Due to the changing nature of furniture and other non-static objects, these are not included. The tool uses an advanced heuristic path loss model that was constructed based on path loss samples collected in an office building and was verified in three other types of buildings: a retirement home, a congress center, and an arts center. Without any additional data or tuning, validation measurements showed an excellent correspondence with estimations. Three contributions are taken into account to calculate the total path loss (from which the received signal strength can be deducted): the sum of the distance loss along the path, the total wall loss along the path, and the interaction loss along

the path.

$$PL_{ref} = \underbrace{PL_0 + 10\gamma \log_{10}\left(\frac{d}{d_0}\right)}_{\text{distance loss}} + \underbrace{\sum_i L_{W_i}}_{\text{cumulated wall loss}} + \underbrace{\sum_j L_{B_j}}_{\text{interaction loss}} + X_\sigma \quad [dB] \quad (2.1)$$

PL_{ref} [dB] is the total path loss calculated with the WHIPP tool, PL_0 [dB] is the path loss at a reference distance d_0 [m], γ [-] is the path loss exponent and d [m] is the distance along the path between transmitter and receiver. The first two terms represent the path loss due to the traveled distance (*distance loss*), the third term (*cumulated wall loss*) is the sum of all wall losses L_{W_i} when a signal propagates through a wall W_i and the fourth term (*interaction loss*) takes into account the cumulated losses L_{B_j} caused by all propagation direction changes B_j of the propagation path from transmitter to receiver. X_σ [dB] is a log-normally distributed variable with zero mean and standard deviation σ , corresponding to the large-scale shadow fading.

2.4 Location tracking algorithm

2.4.1 Viterbi principle

The proposed location tracking system exploits the Viterbi principle, a technique named after Andrew Viterbi, who proposed it in 1967 as a decoding algorithm for convolutional codes over noisy digital communication links [13]. This dynamic programming algorithm is used in the context of hidden Markov models (HMM) to determine the most likely sequence of hidden states, called the Viterbi path, resulting in the sequence of observed events. To apply this technique on a location tracking algorithm, the proposed approach interprets the states as real locations on a floor plan. Then, this principle comes down to determining the most likely sequence of positions instead of only the most likely current position. In the remainder of this work the term *path* is used for a sequence of positions. During the online phase, all possible paths are reconstructed and stored in memory with an associated cost. This cost is defined as the sum of mean square errors (MSE) between measurements and reference fingerprints. At each time step, all costs are calculated and used as decision metric to determine the most likely path.

$$MSE_{p,t} = \frac{1}{N_{AP}} \sum_{i=1}^{N_{AP}} (RSS_{meas}^{i,t} - RSS_{ref}^{i,p,t})^2 \quad (2.2)$$

$$cost_{p,T} = \sum_{t=1}^T MSE_{p,t} \quad (2.3)$$

$MSE_{p,t}$ is the mean square error of path p at time step t , N_{AP} is the number of access points (AP) that measure the RSS from the user, $RSS_{meas}^{i,t}$ is the RSS measurement from access point i at time step t and $RSS_{ref}^{i,p,t}$ is the reference RSS value from access point i for the grid point along path p at time step t , calculated with the WHIPP tool. $cost_{p,T}$ is the associated cost of the p th path stored in memory at time step T and T is the number of time steps that went by since the beginning. The last position of the path with the lowest associated cost is taken as most likely current location.

The proposed algorithm is not recursive in the strictest sense, i.e., the function to determine the location is not applied within its own definition but the algorithm is also not restarted every time a measurement is received. The paths and their associated costs from a previous iteration serve as input to the current iteration, along with the new measurements. After every location update, i.e., after processing the latest signal strength measurements, a fixed number of paths are kept in memory, e.g., 1000. Historical data, i.e., measurements from previous time steps, are represented by the paths that are kept in memory. In other words, the current state is summarized in the last position of each path in memory and future estimations do not depend on positions preceding the current state (Markov property). Note that the memory usage depends on this fixed number of paths and can be compared to the number of particles in a particle filter. The required processing power depends on the maximum speed, location update rate, grid size, and environment as all physically possible paths are considered for the next location update.

2.4.2 Map information and motion model

To apply the Viterbi principle in a useful manner and improve the location tracking accuracy, the number of allowed transitions between two grid points is restricted by using semantic data, i.e., the environment of the object that is being tracked and a motion model. In Figure 2.1 an example is given to illustrate this principle.

The environment is modeled by the floor plan that was used in Section 2.3 to generate the fingerprint database. The grid is based on a Cartesian coordinate system and grid points are generated for all areas enclosed by a set of walls and doors. These doors are used to indicate positions where a user can leave a room, assuring that no walls are crossed. All grid points between which a transition is possible, given the walls and doors of the

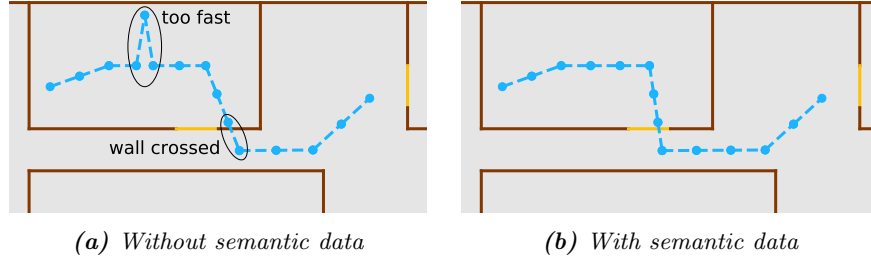


Figure 2.1: Reconstructed trajectory with and without semantic data taken into account.

environment, are calculated in advance and stored in a hash table. This alleviates the need for geometric intersection during the online phase, making the proposed technique computationally much more efficient. Furthermore, the motion model sets a maximum speed limit, ensuring that no unrealistically large distances are traveled within a given time frame. Overall, this leads to realistic and physically possible paths.

The pseudo-code of the location tracking system is shown in Algorithm 1.

Note that the proposed system is also useful for sensor fusion, e.g., the detected steps from an accelerometer, changes in orientation from a gyroscope, or additional magnetic fingerprints can be combined in a weighted function to calculate the cost of each path. In summary, the proposed system is based on an RSS fingerprinting technique that uses an advanced indoor path loss model to construct the radio map and uses the Viterbi principle and semantic data to reconstruct a realistic trajectory. To the best of the author’s knowledge this is the first location tracking algorithm that uses this combination of techniques and optimizations. It is easily deployed, works in real-time and is accurate for positioning in a building-wide environment.

2.4.3 Robust start position

Because the most likely sequence of positions is determined and the allowed transitions between two positions are restricted, the proposed location tracking algorithm is sensitive to a wrong starting position. One could start off in the wrong room, which implies a certain recovery time before predictions can be accurate again, because walls cannot be crossed. To counteract this, additional possible starting positions are added as soon as the tracking begins. These additional starting positions are located on circles around the best initial prediction and there are eight such positions per circle (Figure 2.2). This method takes two arguments: *levels* and *inter distance*. *Levels* cor-

Algorithm 1: Indoor location tracking system.

Data: RSS measurements collected by fixed APs
Result: most likely trajectory given all measurements

```

1  $MP \leftarrow 1000$  // maximum paths in memory
2  $t_{prev} \leftarrow$  first timestamp with measurements
   // RSP: robust start position (Section 2.4.3)
3  $pathsInMem \leftarrow$  list initialized with grid points from RSP method
4 while measurements do
5    $t \leftarrow$  current timestamp
6    $\Delta t \leftarrow t - t_{prev}$ 
7    $pathsTemp \leftarrow$  empty list
8   for  $path \in pathsInMem$  do
9      $cost \leftarrow$  cost of  $path$ 
10     $PGP \leftarrow$  endpoint of  $path$  (parent grid point)
11     $RGP \leftarrow$  reachable grid points (given semantic data and  $\Delta t$ )
       // CP: candidate position
12    for  $CP \in RGP$  do
13       $MSE_{p,t} \leftarrow$  mean square error between measurements
        and reference values of  $CP$  // Equation 2.2
14       $path_{new} \leftarrow path + CP$ 
15       $cost_{new} \leftarrow cost + MSE_{p,t}$ 
16      add  $(path_{new}, cost_{new})$  to  $pathsTemp$ 
17    $pathsInMem \leftarrow$  retain  $MP$  paths from  $pathsTemp$  based on
     lowest  $cost$ 
18    $Viterbi-RT \leftarrow$  add endpoint of  $path$  with lowest  $cost$ 
19    $t_{prev} \leftarrow t$ 
20  $Viterbi \leftarrow path$  with lowest  $cost$  in  $pathsInMem$ 

```

respond to the number of circles that are added and *inter distance* is the distance between these circles. In all experiments *levels* was set to 5 and *inter distance* to 1 m because increasing the number of circles or reducing the distance between them did not further improve the results.

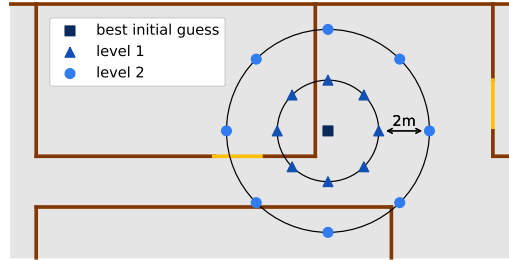


Figure 2.2: Example of additional starting positions with 2 levels and an inter distance of 2 m for visibility.

The technique allows the algorithm to easily correct itself by switching to another path when new measurements suggest being located inside a different room. Note that as the user starts walking and more measurements become available, the best paths will move towards the correct location and coincide if they are on the same grid point at the same time instance. Consequently, the proposed system is robust against a wrong starting position and does not need information about this position in advance, like, e.g., pedestrian dead reckoning. This location tracking algorithm, also referred to as route mapping filter, serves as basis for the proposed systems in the next chapters, i.e., to compensate human body shadowing (Chapter 3), to optimize radio maps based on unsupervised learning (Chapter 4), and map matching based on raw GPS data (Chapter 5).

2.5 Experimental configuration

The measurements are conducted in a generic wireless testbed for sensor experiments (w-iLab.t [14]), located on the third floor of a modern office building in Ghent. It consists of several computer classes, offices, and meeting rooms (Figure 2.3).

The core of the building is made of concrete walls (light gray), the inner structure is movable and made of layered drywall (yellow), the doors are made of wood (brown) and the outside of the building consists of windows with metal blinds (dark gray) and measures 90 m by 17 m (Figure 2.4). The fixed infrastructure is a testbed consisting of 57 sensor nodes and an



Figure 2.3: Mobile nodes and indoor environment.

equal amount of WiFi nodes (access points) that were installed at a height of 2.5 m (blue dots in Figure 2.4).

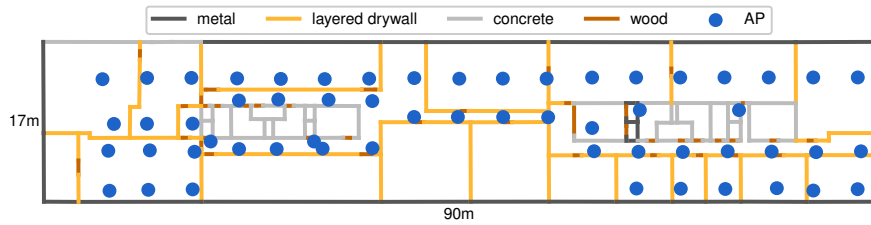


Figure 2.4: Floor plan with indication of the access points and the material of all walls and doors.

Two mobile nodes were used: a TMote Sky node [15], that uses ZigBee, and a WiFi node, both fed by an external battery of 19 Volt (Figure 2.3a). Both mobile nodes have a transmission rate of 10 packets/s and are operating in the 2.4 GHz frequency band. They have a bandwidth of 2 MHz and 20 MHz and both have an external antenna with a gain of 5 dBi. The RSS values, measured by the fixed APs, are used to estimate the location but the fingerprint database consists of path loss values, calculated with the WHIPP tool (Section 2.3). A quick calibration is needed to determine the shift between both, therefore a broadcast of 30 seconds was performed at four different locations. The average difference was used as shift between the RSS and the path loss values (assumed to be fixed). This was done once for ZigBee and once for WiFi.

Nine test trajectories are used to evaluate the proposed location tracking algorithm. The simulations and experimental validation use the same trajectories to allow an easy comparison between both (Section 2.8.1). These nine test trajectories had an average length of 87 m and were conducted in different areas of the building, by a person who walked at constant speed. The mobile nodes were hand-carried at a height of approximately 1.5 m and

the average walking speed was 1.10 m/s (about 4 km/h), which is a normal velocity for an indoor environment. Figure 2.5 shows three such trajectories in dashed orange, dotted green, and solid red. The other six trajectories are similar but pass through different rooms.

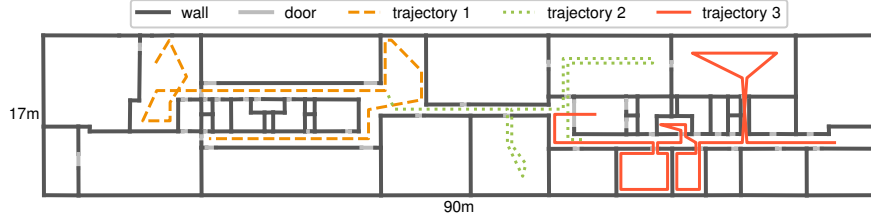


Figure 2.5: Floor plan with indication of the walls, doors, and three test trajectories shown in dashed orange, dotted green, and solid red.

2.5.1 Algorithm settings

The settings of the proposed location tracking algorithm are summarized in Table 2.1. The maximum allowed speed v_{max} is fixed at 2 m/s (7.2 km/h) and the *location update rate* is set to 1 Hz. Remark that the transmission rate of the mobile node was set to 10 Hz and the average RSS value (per AP) of the packets received within a second is used as input to the location tracking algorithm. The parameter MP in Table 2.1 represents the maximum number of paths that are retained in memory at each time step. For example, when set to 1000, all paths currently in memory are updated (in all possible directions depending on the measurements and semantic data) and the 1000 best paths, i.e., with the lowest associated cost, are retained. Initially, there are 41 paths, i.e., begin positions after the first location update, because the number of levels in the method for a robust start position is set to 5 (Figure 2.2). The impact of this parameter on the calculation time is studied in Section 2.8.3. The *grid size* determines the resolution of possible positions on the floor plan where a person can be located, i.e., candidate locations, this is set to 0.5 m unless stated otherwise. The influence of this parameter is investigated in Section 2.8.2. The *node density* determines how many of the 57 fixed nodes are used in an experiment (blue dots in Figure 2.4). As the current goal is achieving a high accuracy, this parameter is set to the maximum value. In Section 2.8.1, the impact of the node density is studied, this is important because in a realistic environment this node density will be much lower. On average, 25 out of 57 fixed nodes received the packets from the mobile node. At each time step, the 10 strongest access point measurements (if available) are used to estimate the

new location because increasing this number did not further improve the performance but only adds to the calculation time.

Setting	Value
v_{max}	2 m/s (7.2 km/h)
Sample rate	1 sample/s
MP	1000 paths
Grid size	0.5 m
Node density	57 nodes

Table 2.1: Settings of the proposed indoor location tracking algorithm. MP : maximum number of paths retained in memory with every location update.)

2.5.2 Evaluation metrics

The proposed algorithm has two outputs: *Viterbi-RT* and *Viterbi*, as mentioned in Algorithm 1. The former works in real-time (RT) because it only uses the available information at the present time (current and past measurements) to estimate the current location. The latter allows estimated locations from the past to be corrected by future measurements. In other words, the *Viterbi* result is the most likely path, given all the measurements. This is interesting for applications where a small delay is allowed or real-time is not required, e.g., modeling or analyzing pedestrian and traffic flows.

A *basic* positioning algorithm, Kalman filter [16], and particle filter with smoother [4] are used as references to estimate the performance improvement with the proposed location tracking technique. The basic algorithm uses only the RSS fingerprinting technique from Section 2.2 and does not make use of the Viterbi principle or semantic data from Section 2.4. The Kalman filter and particle filter are included in the simulations to evaluate the influence of an increasing noise level on the location accuracy (Section 2.6.1). The measurement and process noise of the implemented Kalman filter are based on the maximum allowed speed (v_{max}) and level of added noise (Section 1.9.2). Furthermore, the particle filter uses a speed distribution based on v_{max} , the number of particles is equal to the number of paths (MP), and map information is taken into account by resampling the particles that went through a wall in the update phase (Section 1.9.3). The particle filter has two outputs as well: PF and PFS , the first is the real-time output of the particle filter, i.e., weighted average of the particles, and the second is a smoothed trajectory based on the Forward Filtering-Backward Smoothing (FFBS) method [17].

To ensure a fair comparison, all positioning techniques make use of the same fingerprint database (constructed with the advanced indoor path loss model). In Section 2.8.4 the influence of this model is investigated by comparing the results when a free-space path loss model is used to construct the fingerprint database.

The mean (μ), standard deviation (σ), median (50^{th}), and 95^{th} percentile value of the error are chosen as evaluation metrics for the above algorithms. The accuracy, i.e., location error, is defined as the Euclidean distance (in meters) between the predicted and true position.

$$accuracy = \sqrt{(x_{est} - x_{act})^2 + (y_{est} - y_{act})^2} \quad [m] \quad (2.4)$$

The estimated and actual position are located at coordinates (x_{est}, y_{est}) and (x_{act}, y_{act}) . To determine the exact location of these actual positions a constant walking speed is assumed, i.e., the true positions are equally spread along a trajectory.

2.6 Simulations

To simulate real measurements, Gaussian noise with zero mean and a configurable standard deviation is added to the reference fingerprints (RSS_{ref} in Equation 2.2) and are used as input for the location tracking algorithms [18]. An example reconstruction of such a simulated trajectory with the proposed and basic algorithm is shown in Figure 2.6.

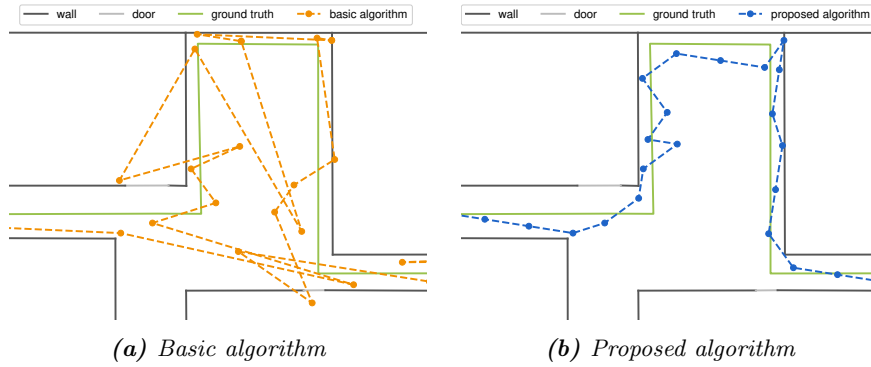


Figure 2.6: Reconstruction of a test trajectory by the basic (a) and the proposed location tracking algorithm (b).

The ground truth test trajectory is indicated in solid green and the reconstructed trajectories are visualized as orange and blue dashed lines, for the *basic* and *proposed* algorithm. The standard deviation of the added

Gaussian noise was set to 5 dB (denoted as *noise level*) and both algorithms have processed the same input. As expected, the reconstructed path with the *proposed* algorithm is realistic and physically possible, whereas in the *basic* algorithm some walls are crossed and impossibly large traveled distances are present.

2.6.1 Influence of noise level

This section evaluates the influence of an increasing noise level on the location accuracy. The impact on the performance of both algorithms is studied for four types of node density: sparse, normal, dense, and very dense. These four node densities use 5, 10, 20, and 57 fixed nodes, respectively, for a surface of 90 m by 17 m or 1530 m² (Figure 2.7). The average inter access point distance (IAPD) for these four node densities are 22.2 m, 12.1 m, 6.4 m, and 3.6 m, respectively.

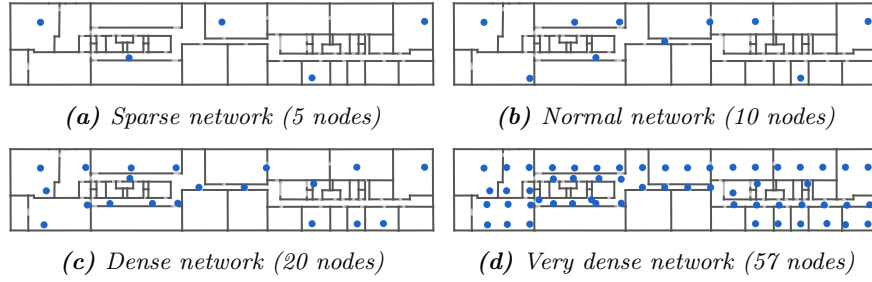


Figure 2.7: Location of the fixed nodes for a sparse, normal, dense, and very dense network.

Figure 2.8 shows a plot of the mean accuracy μ as a function of the added noise level from 0 dB up to 20 dB in steps of 2 dB. Note that in the simulations a transmission rate of 1 Hz is simulated, as such there is no averaging over 10 packets as in the experimental validation (Section 2.5.1). Each simulation was repeated 100 times and the accuracies are averaged over the results of the nine test trajectories.

Figure 2.8 shows that the proposed location tracking technique (*Viterbi*) always outperforms the basic algorithm, Kalman filter, and particle filter in mean accuracy. Moreover, *Viterbi-RT* achieves a similar accuracy as the best performing reference algorithm, i.e., the particle filter with smoother, although the latter does not work in real-time. The absolute difference in mean accuracy between the positioning techniques decreases as node density increases, e.g., for a noise level of 10 dB the differences between *Viterbi* and *Basic* are 6.60 m, 4.62 m, 2.72 m, and 1.08 m for the four

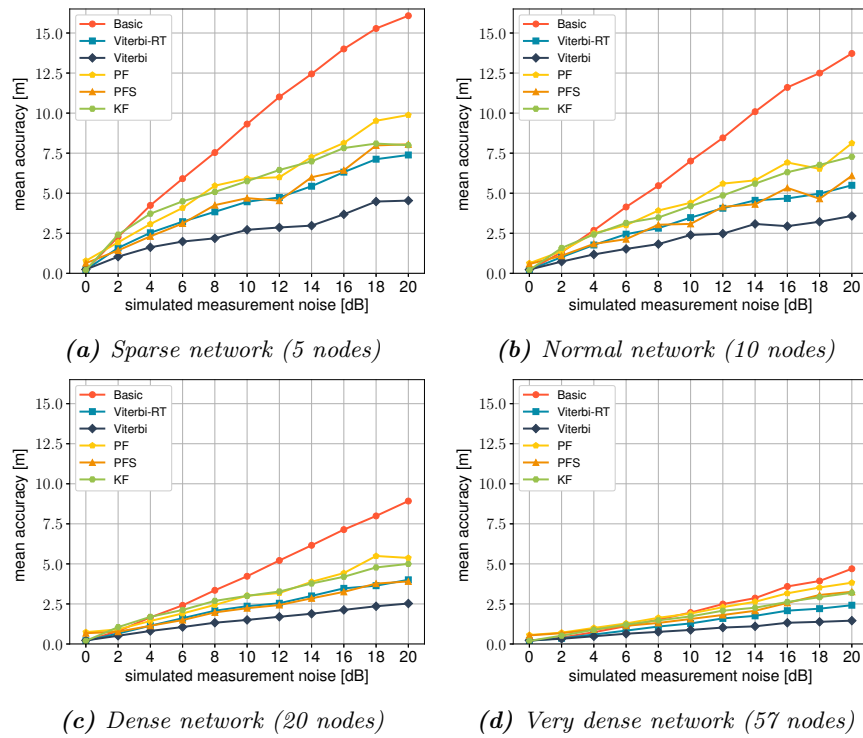


Figure 2.8: Influence of noise for various node densities (simulation). RT: real-time, KF: Kalman filter, PF: particle filter, and PFS: particle filter with smoother.

node densities, respectively. Note that most buildings are not equipped with a dense wireless network. The relative improvements in accuracy of the proposed location tracking algorithm compared to the reference and real-time positioning techniques for four node densities and a noise level of 10 dB are summarized in Table 2.2.

Node density	Algorithm	Accuracy [m]				Viterbi improvement [%]			
		μ	σ	50 th	95 th	μ	σ	50 th	95 th
Sparse	Basic	9.32	8.69	6.80	26.96	70.9	77.6	67.7	74.9
	Kalman filter	5.75	3.20	5.35	12.10	52.8	39.1	58.9	44.0
	Particle filter	5.92	4.00	5.16	13.81	54.1	51.3	57.4	50.9
	Particle smoother	4.69	3.10	4.00	10.94	42.1	37.2	45.1	38.1
	Viterbi _{RT}	4.47	3.19	3.72	10.95	39.2	38.9	40.9	38.1
	Viterbi	2.72	1.95	2.20	6.78	/	/	/	/
Normal	Basic	7.01	6.57	5.22	19.58	65.9	70.6	65.6	67.1
	Kalman filter	4.20	2.50	3.81	9.02	43.0	22.9	52.9	28.5
	Particle filter	4.40	3.05	3.78	10.68	45.7	36.8	52.4	39.7
	Particle smoother	3.08	1.91	2.70	6.96	22.4	-1.0	33.5	7.4
	Viterbi _{RT}	3.48	2.46	3.00	8.32	31.2	21.5	40.1	22.5
	Viterbi	2.39	1.93	1.80	6.45	/	/	/	/
Dense	Basic	4.23	3.52	3.37	11.25	64.5	63.8	68.9	60.2
	Kalman filter	3.00	1.64	2.81	6.03	50.0	22.3	62.7	25.8
	Particle filter	3.01	2.14	2.56	7.46	50.2	40.2	59.1	40.0
	Particle smoother	2.23	1.41	2.01	5.06	32.8	9.4	48.0	11.5
	Viterbi _{RT}	2.36	1.73	1.90	5.84	36.3	26.4	44.8	23.3
	Viterbi	1.50	1.28	1.05	4.48	/	/	/	/
Very dense	Basic	1.96	1.72	1.44	5.44	55.3	61.4	50.2	56.9
	Kalman filter	1.72	1.04	1.58	3.59	49.1	36.2	54.6	34.7
	Particle filter	1.90	1.42	1.55	4.78	53.8	53.2	53.8	51.0
	Particle smoother	1.56	0.98	1.37	3.54	43.9	32.5	47.4	33.8
	Viterbi _{RT}	1.28	1.03	0.94	3.51	31.7	35.8	23.8	33.2
	Viterbi	0.88	0.66	0.72	2.34	/	/	/	/

Table 2.2: Improvement of the proposed location tracking algorithm (Viterbi) compared to the reference and real-time (RT) positioning techniques for four node densities and a noise level of 10 dB.

The proposed location tracking algorithm always improves the performance in mean (μ), standard deviation (σ), median (50th), and 95th percentile value of the accuracy (except for the standard deviation of the particle smoother for the normal node density). The largest relative improvements compared to the reference algorithms are achieved for the sparse network with only 5 nodes. The mean accuracy improves by 70.9%, 52.8%, 54.1%, 42.1%, and 39.2% for the basic algorithm, Kalman filter, particle filter, particle smoother, and Viterbi-RT, respectively. The standard deviation, median, and 95th percentile value show similar improvements in accuracy. In the very dense network, the relative improvements are still significant but drop to 55.3%, 49.1%, 53.8%, 43.9%, and 31.7% for the basic

algorithm, Kalman filter, particle filter, particle smoother, and *Viterbi-RT*, respectively.

Note that, the standard deviation of the measurement noise is assumed to be equal for all measurements and every fixed access point receives all packets sent by the mobile node, independently of the traveled distance, in practice, these two assumptions will not hold. However, all considered positioning techniques have to process the same input, so the comparison remains valid.

2.7 Experimental validation

In this section, the performance evaluation is based on the results obtained with the measurements conducted by a person who hand-carried the mobile ZigBee or WiFi node.

2.7.1 ZigBee

The results with the ZigBee node are averaged over all nine test trajectories and are summarized in Table 2.3.

Algorithm	μ [m]	σ [m]	50 th [m]	95 th [m]
Basic	3.06	4.04	2.30	7.34
Kalman filter	3.01	3.58	2.28	6.85
Particle filter	2.75	2.35	2.21	5.63
Particle smoother	2.64	1.96	2.20	5.47
Viterbi-RT	2.54	1.70	2.14	5.87
Viterbi	2.26	1.40	1.99	5.00

Table 2.3: Average performance with the ZigBee node. *RT*: real-time.

Table 2.3 shows that the median accuracy and standard deviation are below 2 m with *Viterbi*. Overall, the proposed algorithms always outperform the basic algorithm, Kalman filter, particle filter, and particle smoother, especially the standard deviation and 95th percentile values are significantly reduced. The relative improvement in mean accuracy of *Viterbi* is 26.1%, 24.9%, 17.8%, 14.4%, and 11.0% compared to the basic algorithm, Kalman filter, particle filter, particle smoother, and *Viterbi-RT*, respectively.

2.7.2 WiFi

Besides the ZigBee node, three test trajectories were investigated with the WiFi node. Because of the higher maximum transmit power, more fixed

nodes received the packets from this mobile node (on average 40 out of 57 fixed nodes). This time, the 20 strongest access point measurements were used to estimate the location, for the same reason as with the ZigBee node (Section 2.5.1). The other settings were the same as in Table 2.1 and the results are summarized in Table 2.4.

Algorithm	μ [m]	σ [m]	50 th [m]	95 th [m]
Basic	3.61	2.66	3.09	7.55
Kalman filter	3.46	2.56	2.93	6.92
Particle filter	3.17	1.81	2.83	6.38
Particle smoother	2.99	1.67	2.65	6.17
Viterbi-RT	3.02	1.93	2.65	6.21
Viterbi	2.86	1.61	2.59	6.05

Table 2.4: Average performance with the WiFi node. RT: real-time.

Again, the proposed *Viterbi* algorithms perform better than the reference algorithms but with slightly lower improvements than the ZigBee results: 20.8%, 17.3%, 9.8%, 4.3%, and 5.3% compared to the basic algorithm, Kalman filter, particle filter, particle smoother, and *Viterbi-RT*, respectively. Less accurate results were obtained with the WiFi node because compared to the used ZigBee channel, there was more interference on the used WiFi channel, which resulted in a larger variance in the RSS measurements. The channels used were 26 for ZigBee (2480 MHz with a bandwidth of 2 MHz), which is usually relatively unaffected by WiFi, and channel 1 for WiFi (2412 MHz with a bandwidth of 20 MHz), which is also used for regular wireless traffic.

2.8 Sensitivity analysis

In this section a sensitivity analysis is conducted, based on the measurements from the experimental validation, to investigate the influence of node density, grid size, path loss model, semantic data, maximum number of paths updated and retained with each location update, and the amount of previous positions taken into account. This is important to estimate the influence of the algorithm's parameters on the performance (location accuracy, execution time, and memory usage). Unless stated otherwise, the settings from Table 2.1 are used and the results are averaged over all nine trajectories, conducted with the ZigBee node.

2.8.1 Node density

The majority of testbeds used for validating a new positioning system have a very high node or access point density but in a typical environment this node density is much lower. The recommended node density for a wireless network infrastructure will depend on several factors, e.g., the number of users, required bandwidth, type of building, etc., whereas the most important factor for a positioning system will be the required accuracy. Cisco recommends a node density between 0.0015 nodes/m² and 0.0043 nodes/m² for their Cisco 1000 series lightweight access point [19]. The test environment has a surface area of 1530 m², which results in 7 APs for a node density of 0.0043 APs/m². The used testbed is equipped with 57 nodes, which leaves us many possibilities to investigate the performance from sparse to very dense networks. Four scenarios are considered: 5, 10, 20, and 57 fixed nodes (as in Section 2.6.1). The mean accuracy (μ) and standard deviation (σ) are summarized in Table 2.5.

Scenario	#nodes	nodes/m ²	IAPD [m]	Basic		Viterbi-RT		Viterbi	
				μ [m]	σ [m]	μ [m]	σ [m]	μ [m]	σ [m]
Sparse	5	0.0033	22.2	8.30	11.89	6.04	9.10	5.11	7.81
Normal	10	0.0065	12.1	6.58	11.41	3.76	2.58	3.26	2.24
Dense	20	0.013	6.4	4.78	8.34	3.20	2.10	2.70	1.72
Very dense	57	0.037	3.6	3.06	4.04	2.54	1.70	2.26	1.40

Table 2.5: Performance for different node densities. IAPD: inter access point distance

It is clear that the proposed algorithm shows acceptable results in a network with normal node density. More specifically, *Viterbi-RT* and *Viterbi* perform 2.82 m and 3.32 m better compared to the basic algorithm in terms of mean accuracy, corresponding to an improvement of 42.9% and 50.5%. This is due to the many poor predictions of the basic algorithm, which worsen the mean accuracy a lot. Comparing the 95th percentile values with normal node density give similar insights: 15.00 m, 8.20 m, and 7.94 m for the basic algorithm, *Viterbi-RT*, and *Viterbi*, respectively. Note that a difference between the experimental validation and the simulations is that the latter does not take into account the influence of the human body itself. This so-called human body shadowing causes additional losses, which can further worsen the location accuracy and is the topic of Chapter 3.

2.8.2 Grid size

The grid size determines the resolution of the possible positions on the floor plan where a person can be located. A lower grid size has the benefit of

a finer resolution at the cost of a higher execution time or needed computational power because the search space is bigger. In this section, the influence of this parameter on the performance of the proposed algorithm is investigated for four different grid sizes (Table 2.6).

Grid size [cm]	Viterbi-RT				Viterbi			
	μ [m]	σ [m]	50 th [m]	95 th [m]	μ [m]	σ [m]	50 th [m]	95 th [m]
20	2.60	1.74	2.22	6.24	2.34	1.47	2.02	5.38
50	2.54	1.70	2.14	5.87	2.26	1.40	1.99	5.00
100	2.64	1.81	2.15	6.28	2.22	1.42	1.96	5.09
200	3.01	2.12	2.52	7.28	2.82	1.89	2.40	6.64

Table 2.6: Performance for different grid resolutions.

Grid sizes of 0.2 m, 0.5 m, and 1 m yield similar performance: a mean accuracy around 2 m, a standard deviation of 1.5 m, a median accuracy just below 2 m, and a 95th percentile value around 5 m. This is because as soon as the grid size is small compared to the achieved median accuracy, the performance will not further improve by reducing the grid size. Thus, a grid size of 1 m is recommended when time or computational power is limited because smaller grid sizes will increase the required memory and the number of calculations of both the offline as online phase and hence the execution time.

2.8.3 Execution time

The average time needed to calculate one location update determines the latency and ability for real-time usage and depends on the available computational power but also on parameters related to the proposed location tracking system. Notably, the grid size and the number of paths stored in memory will have an influence on the execution time. Figure 2.9 shows the execution time as a function of the number of paths in memory, for four different grid sizes (logarithmic scale). The solid lines are the execution times with the proposed algorithm and the dotted lines with the basic algorithm. An upper limit of 1000 paths was used because an increase in number of stored paths did not result in further improvements in accuracy; for larger grid sizes even less paths are needed. The experiments are run on a desktop computer with an Intel Core i7 3.40 GHz processor, 8.00 GB DDR3-SDRAM, and 64 bit operating system.

Figure 2.9 shows that the time needed to calculate one location update increases exponentially with the number of paths that are stored and updated at every time step (MP in Table 2.1 and Algorithm 1) and quadratically with a smaller grid size because halving the grid size results in four

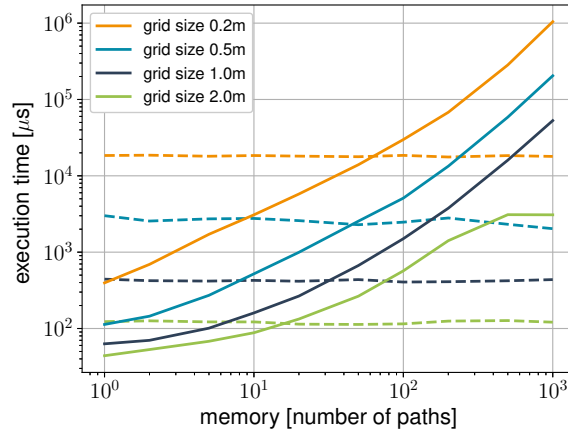


Figure 2.9: Average execution time to calculate one location update for different grid sizes (solid line: proposed algorithm, dotted line: basic algorithm).

times as many grid points (on a two-dimensional floor plan). The basic algorithm does not depend on the number of paths and hence has a nearly constant execution time. For a low number of paths, the proposed system executes faster than the basic algorithm because to determine the most likely position, the latter examines every grid point on the floor plan and the proposed algorithm only considers those centered around a previous position (due to the motion model). For a grid size of 2 m it can be seen that the execution time remained equal for 500 to 1000 paths because along the nine trajectories, the maximum possible paths retained in memory never exceeded 500. Note that this is determined by the start position, location of walls and doors, and the maximum speed along the reconstructed paths in memory. The maximum delay is about 1 s when using 1000 paths and a grid size of 20 cm. This is a possible bottleneck if the proposed system is used in combination with a VLP or UWB approach because a grid size below 20 cm would be required to take advantage of the accurate ranging measurements. Note that a particle filter operates in a continuous space and hence its performance is only dependent on the number of particles and number of obstacles for the geometric intersection if floor plan information is taken into account. For real-time location tracking applications based on signal strength measurements it is better to use a grid size of 1 m and 100 paths, which results in a maximum delay of 1.5 ms to calculate a location update with almost no loss in positioning accuracy. This leaves margin to run the proposed location tracking algorithm in real-time on devices with less computational power or memory.

2.8.4 Path loss model

Until now the advanced indoor path loss model (WHIPP [12]) was used to construct the fingerprint database (Section 2.3). In Table 2.7 the performance with a free-space path loss model is compared to the one with WHIPP. The free-space path loss is calculated as follows:

$$FSPL = 20 \log_{10}(d) + 20 \log_{10}(f) - 27.55 \quad [dB] \quad (2.5)$$

$FSPL$ [dB] is the total path loss calculated with the free-space model, d [m] is the distance between transmitter and receiver, and f [MHz] is the signal frequency. To ensure a fair comparison with the WHIPP model, the free-space model was calibrated with the same measurements to calculate the fixed RSS shift (Section 2.5).

Algorithm	Path loss model	μ [m]	σ [m]	50 th [m]	95 th [m]
Basic	free-space	3.35	3.43	2.68	7.98
Basic	WHIPP	3.06	4.04	2.30	7.34
Viterbi	free-space	2.62	1.60	2.36	5.82
Viterbi	WHIPP	2.26	1.40	1.99	5.00

Table 2.7: Influence of used path loss model.

Except for the standard deviation with the basic algorithm, the results with the WHIPP model slightly outperform the ones with a free-space path loss model but the main improvement originates from the Viterbi technique. The mean, standard deviation, median, and 95th percentile accuracy improve by 32.5%, 59.1%, 25.7%, and 37.3%, respectively, when using the proposed algorithm and advanced indoor path loss model (*Viterbi+WHIPP*) compared to *Basic+free-space*.

2.8.5 Semantic data

This section discusses the added value of the semantic data (environmental data and motion model). The location tracking accuracy for four combinations of semantic data used in the proposed algorithm are summarized in Table 2.8.

When no semantic data is taken into account, the Viterbi principle has no added value because any transition between two positions is possible, which gives the same result as with the basic algorithm. Using the environmental data, e.g., no wall crossing and leaving a room through the doors, gives a small improvement and the motion model, i.e., a limitation on the assumed maximum speed, results in the largest improvement. Using both,

Semantic data	μ [m]	σ [m]	50 th [m]	95 th [m]
None (Basic)	3.06	4.04	2.30	7.34
Environmental data	2.71	2.59	2.17	6.62
Motion model	2.27	1.42	2.03	5.05
Both	2.26	1.40	1.99	5.00

Table 2.8: Influence of semantic data.

still yields the best results but the additional value is limited because there are only a few room changes over the course of an entire trajectory. Most of the time, a user is walking in a room or in a hall way. The values of Table 2.8 are of course specific for the considered configuration.

2.8.6 Future measurement data

This section investigates the effect of future measurement data, i.e., location updates, on the positioning accuracy. Note that the proposed algorithm allows future measurements to correct previously estimated positions by keeping all paths in memory and selecting the most likely trajectory when the location tracking ends. The amount of future measurement data is represented by the elapsed time after the current location update. Figure 2.10 shows a plot of the four metrics as a function of this parameter for a normal node density, i.e., 10 nodes or 0.0065 nodes/m².

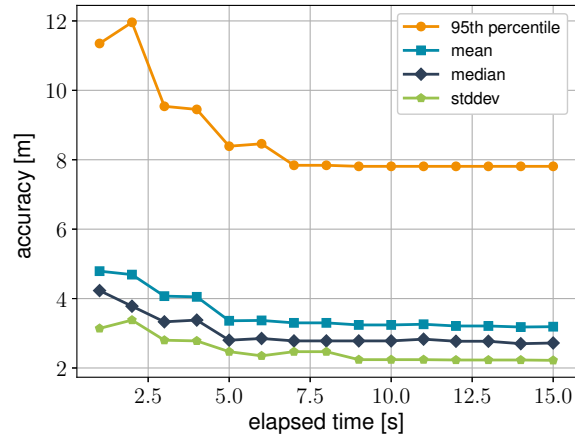


Figure 2.10: Influence of the amount of future measurement data on the four metrics (in a normal node density).

No further improvement in any of the four metrics is noticeable when

nine or more seconds of future measurement data is taken into account. This means that a delay of nine seconds suffices to obtain the highest possible accuracy with the nine test trajectories of this experimental validation. Another advantage is that the required memory can be reduced if a person is tracked for a long time, e.g., by periodically calculating the most likely trajectory, storing the results and keeping only the paths and costs of the last nine seconds for the next calculation. Intuitively, one would indeed expect the bigger the time difference between two location updates, the smaller the influence both will have on each other.

2.9 Conclusion

In this chapter, a real-time indoor location tracking system based on the Viterbi principle and semantic data is presented. The system is evaluated by both simulations and an extensive experimental validation in a real office environment. The simulations confirmed that the proposed location tracking system was more robust against measurement noise, especially for networks with smaller node densities, e.g., a simulated noise level of 10 dB resulted in improvements of 70.9%, 52.8%, 54.1%, and 42.1% in mean accuracy compared to a fingerprinting technique, Kalman filter, particle filter, and particle smoother, respectively. In the experimental validation, an average median accuracy below 2 m was obtained over nine test trajectories with a total length of 783 m, in an office building that has 57 nodes and measures 90 m by 17 m (covering over 1500 m²). It is shown that the node density has a major impact on the accuracy, although acceptable results were obtained for normal node densities, i.e., a mean accuracy of 3.26 m with the proposed location tracking algorithm compared to 6.58 m with the basic algorithm. Furthermore, it is shown that semantic data is necessary to exploit the Viterbi principle, especially the motion model has a major influence. Compared to a free-space path loss model, the usage of an advanced indoor path loss model to construct the fingerprint database improved the median accuracy of the proposed Viterbi algorithm by 15.68%. The grid size has a huge impact on the execution time, required computational power, and memory usage, which is important to work in real-time on low-cost portable devices. The amount of future measurement data used to correct previously estimated locations was found to improve the positioning accuracy up to nine future location updates. This means that a delay of nine seconds suffices to obtain the highest possible accuracy in the experimental validation for the considered configuration. In Chapter 3, a positioning system is proposed to cope with the signal deterioration caused by the user being tracked itself, so-called human body shadowing.

References

- [1] Homayoun Hashemi. *The indoor radio propagation channel*. Proceedings of the IEEE, 81(7):943–968, 1993.
- [2] Peng Gao, Weiren Shi, Wei Zhou, Hongbing Li, and Xiaogang Wang. *A location predicting method for indoor mobile target localization in wireless sensor networks*. International Journal of Distributed Sensor Networks, 2013, 2013.
- [3] Kezhong Lu, Xiaohua Xiang, Dian Zhang, Rui Mao, and Yuhong Feng. *Localization algorithm based on maximum a posteriori in wireless sensor networks*. International Journal of Distributed Sensor Networks, 8(1):260302, 2011.
- [4] Henri Nurminen, Anssi Ristimäki, Simo Ali-Loytty, and Robert Piché. *Particle filter and smoother for indoor localization*. In Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on, pages 1–10. IEEE, 2013.
- [5] V. Moghtadaiee and A.G. Dempster. *Indoor Location Fingerprinting Using FM Radio Signals*. Broadcasting, IEEE Transactions on, 60(2):336–346, June 2014.
- [6] Jingbin Liu, Ruizhi Chen, Ling Pei, Robert Guinness, and Heidi Kuusniemi. *A hybrid smartphone indoor positioning solution for mobile lbs*. Sensors, 12(12):17208–17233, 2012.
- [7] Frederic Lassabe, Philippe Canalda, Pascal Chatonnay, F Spies, and D Charlet. *Refining WiFi indoor positioning renders pertinent deploying location-based multimedia guide*. In Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on, volume 2, pages 126–132. IEEE, 2006.
- [8] Paramvir Bahl and Venkata N Padmanabhan. *RADAR: An in-building RF-based user location and tracking system*. In INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 2, pages 775–784. IEEE, 2000.
- [9] Frederic Lassabe, Philippe Canalda, Pascal Chatonnay, Francois Spies, and Oumaya Baala. *A Friis-based calibrated model for WiFi terminals positioning*. In Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, pages 382–387. IEEE, 2005.

- [10] Ville Honkavirta, Tommi Perala, Simo Ali-Loytty, and Robert Piché. *A comparative survey of WLAN location fingerprinting methods*. In Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on, pages 243–251. IEEE, 2009.
- [11] *ZigBee*. <https://www.zigbee.org/zigbee-for-developers/about-us/>. Accessed: 2015-06-16.
- [12] David Plets, Wout Joseph, Kris Vanhecke, Emmeric Tanghe, and Luc Martens. *Coverage prediction and optimization algorithms for indoor environments*. EURASIP Journal on Wireless Communications and Networking, 2012(1):1–23, 2012.
- [13] Andrew J Viterbi. *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*. Information Theory, IEEE Transactions on, 13(2):260–269, 1967.
- [14] Stefan Bouckaert, Wim Vandenberghe, Bart Jooris, Ingrid Moerman, and Piet Demeester. *The w-iLab. t testbed*. In Testbeds and Research Infrastructures. Development of Networks and Communities, pages 145–154. Springer, 2011.
- [15] *TMote Sky Datasheet*. <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>. Accessed: 2014-11-20.
- [16] İsmail Güvenc. *Enhancements to RSS based indoor tracking systems using Kalman filters*. PhD thesis, University of New Mexico, 2003.
- [17] Arnaud Doucet and Adam M Johansen. *A tutorial on particle filtering and smoothing: Fifteen years later*. Handbook of nonlinear filtering, 12(656-704):3, 2009.
- [18] Paramvir Bahl, Venkata N Padmanabhan, and Anand Balachandran. *Enhancements to the RADAR user location and tracking system*. Microsoft Research, 2(MSR-TR-2000-12):775–784, 2000.
- [19] Cisco Systems. *Deployment Guide: Cisco Aironet 1000 Series Lightweight Access Points*.

Chapter 3

Compensation of human body shadowing

3.1 Introduction

Many location tracking systems already try to cope with performance deterioration caused by multipath fading and diffraction as discussed in Chapter 2. However, an important factor that has already been noted in literature but is often still neglected is the influence caused by the user being tracked itself, so-called human body shadowing [1]. The presence of such a user will influence the radio-frequency (RF) signal paths between a body-worn tag and the receiving nodes, and can block the line-of-sight (LoS) depending on the user's orientation and location. This causes additional propagation losses that are currently not accounted for and will generally decrease the accuracy of signal strength based localization techniques. Furthermore, the performance of positioning systems is often verified by stepwise moving a node placed on a tripod, hereby explicitly removing the human from the equation, while practical human tracking applications always imply the presence of a user's body.

This chapter presents an RF-based location tracking system that improves its performance by compensating for the shadowing caused by the human body of the user being tracked. Two novel approaches to mitigate the human body shadowing are investigated and combined in one location tracking system. The first approach combines the measured signal strengths from multiple mobile nodes, placed on different parts of a human body. Combining their measured signal strengths allows reducing the variation caused by the user's body and hence limits its influence. The second approach takes into account the user's orientation towards the fixed infras-

structure and the body-worn tag's relative position. Next, a human body loss model is used to explicitly compensate for the user's influence. In this method, the orientation of a user is determined without making use of the classic compass or gyroscope approaches, but with a novel orientation estimator built on top of the tracking algorithm of Chapter 2. Both approaches can be independently combined and reduce the influence caused by body shadowing, hereby improving the tracking accuracy.

The remainder of this chapter is structured as follows, Section 3.2 describes related work, Section 3.3 presents an experimental motivation for the proposed approaches, and an overview of the system is given in Section 3.4. Section 3.5 revisits the equations of the location tracking algorithm introduced in Chapter 2 and Section 3.6 elaborates on these equations to embed the human body compensation models and orientation estimator into the algorithm. Section 3.7 discusses the configuration and results of the experimental validation. Finally, in Section 3.8, conclusions are provided.

3.2 Related work

Many state-of-the-art approaches try to cope with the complexity of indoor environments by making use of advanced processing techniques, e.g., Kalman filters [2, 3], particle filters [4–6], and machine learning techniques [7], but most approaches neglect the influence of the human body itself. In [8], a body shadowing mitigation method is used on top of an RSS-based Monte Carlo positioning technique, achieving meter scale accuracies for a wrist-worn personnel tracking tag. The shadowing caused by a user's body is mitigated by using LoS and non-line-of-sight (NLoS) channel models. A disadvantage of this approach is that it depends on a manual differentiation of the LoS conditions and separate measurements that need to be conducted for each LoS condition. In [9], video cameras are used to detect the human orientation and an empirical compensation model is used to compensate for body electronic interference. In [10], multiple sensors are placed on a user and measured power level values are used for estimating the position and orientation of a user in a single room. They present a theoretical procedure to evaluate the maximum attainable performance based on ray-tracing to compute a fine grid of RSS values and a maximum-likelihood approach for localization. In [11], a fingerprinting system based on neural networks is used for indoor positioning with Bluetooth devices. They use a compass module to provide information about the user's orientation, which improves the selection of the most adequate neural network to use. The achieved results are highly accurate, but a lot of training data is needed, since for every user orientation a neural network has to be trained. In [12, 13], the

losses caused by a human body are used as an advantage, i.e., a user is asked to rotate in place, simulating the behavior of a directional antenna. This directional analysis technique is used to localize an outdoor access point in [12] and to estimate the location of the user in [13].

In this work, the user's direction is obtained with an orientation estimator based on previous and current location predictions. Next, the estimated orientation is used to automatically mitigate the shadowing caused by the human body. This approach requires no manual differentiation [8], video cameras [9], or orientation tracking sensors like accelerometers, gyroscopes, and compasses [11]. The compensation model is based on three-dimensional electromagnetic simulations with a human phantom and hence no need for extensive measurements [8, 11]. Note that the simulations need to be performed only once per position of the mobile tag. Furthermore, all possible orientations are taken into account, which results in a continuous three-dimensional model to compensate for body shadowing. The performance evaluation is done at 2.4 GHz using ZigBee nodes [14], on a building-wide testbed, not limited to a single room or theoretical framework [10].

3.3 Characterization of human body influence

In this section, the influence of human body shadowing on the exchanged signals between a body-worn mobile device and a fixed node, is investigated and characterized. This is relevant for all signal strength based positioning and location tracking algorithms, which purpose is to track humans. In this work, the proposed body shadowing compensation methods are verified with an RSS fingerprinting based tracking algorithm [15]. Other signal strength based positioning methods can also benefit from this approach, e.g., the aforementioned RSS-based algorithms that use a Kalman filter [2] or particle filters [4, 6]. The proposed methods in this work are aimed at signal strength based systems but human body shadowing can also influence the performance of ToA-based positioning methods. The presence of a human body can, e.g., block the line of sight path or cause the creation of additional paths with multipath, biases in the estimated delays, and performance degradation as a result. Different compensation techniques are needed to cope with the shadowing in ToA-based systems but this is beyond the scope of this work. In general, eliminating the influence of human shadowing results in better and more robust (ranging) measurements, which lead to better positioning accuracies.

As mentioned before, human body shadowing occurs when the RF signal path between a body-worn tag and a receiving node is completely or partially blocked by a person. The exchanged RF signals are altered due

to dielectric losses in the human body, which manifests itself as a drop in signal strength. Additional propagation losses of around 10 to 30 dB are reported in literature [12]. To verify this effect, an experiment is conducted: a (male) user is asked to turn 360° clockwise around his axis, while wearing a mobile tag on his chest and back (Figures 3.4a and 3.4b). The user turns 45° every 15 seconds, taking 2 minutes for a full rotation. Forty fixed nodes received the packets sent by the mobile tags and measured the RSS values. Note that the testbed consists of 48 fixed nodes in total, meaning that 8 nodes did not receive packets due to signal attenuation (more details on this testbed are provided in Section 3.7.1 and Figure 3.2). Figure 3.1 shows the measured RSS values, averaged over a window of 15 s, as a function of the user's orientation for a nearby (12 m) and far away (66 m) node, indicated in green and red. The location of the user is indicated with a yellow star in Figure 3.2. The experiment starts with the user facing the left side of the building, i.e., towards the nearby and far away node (orientation 0°).

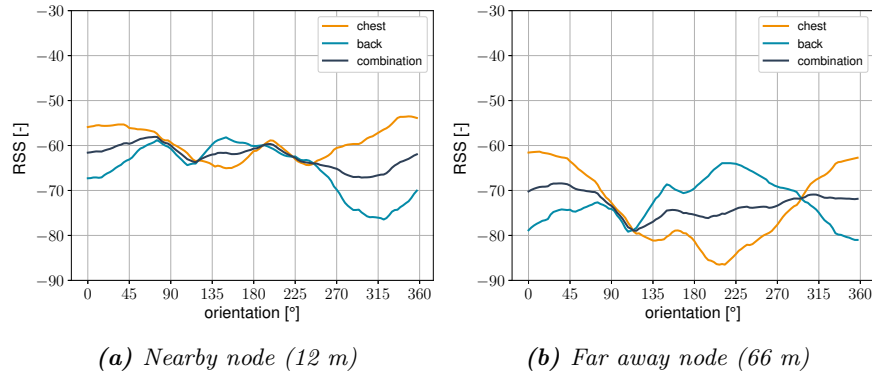


Figure 3.1: The effect of a full rotation and human body shadowing in an indoor environment.

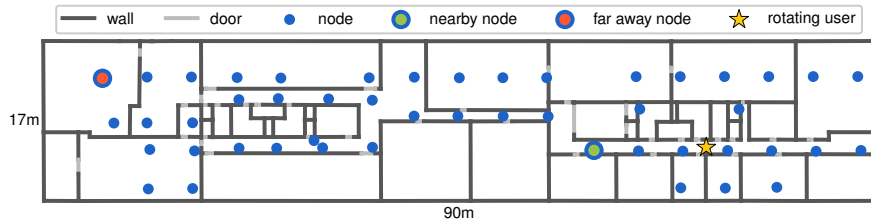


Figure 3.2: Floor plan of testbed with indication of walls, doors, access points, near access point, far access point, and location of the rotating user.

From Figure 3.1, it is clear that the orientation of a user has a significant

influence on the measured RSS values (on top of the variance caused by multipath). A high RSS value measured by the chest tag corresponds always to a low RSS value measured by the back tag (and the other way around). This is because the human body is located between both body-worn tags and can block or attenuate the strongest path between the nodes and mobile tags. The shape of the RSS plots are not 180°shifted copies of each other because of differences in the individual tags and the human anatomy front and back side. The maximum observed differences between the chest and back tag, measured by the forty fixed nodes, range from 1.8 dB to 25.2 dB with an average value of 14.8 dB. The highest differences are observed in LoS situations, whereas the lowest differences are observed in situations where the RF signal suffers from severe multipath. As a result, the average RSS value of both tags shows less variation: the standard deviation, averaged over all forty nodes, is 4.9 dB, 4.3 dB, and 2.5 dB for the chest, back, and combination of both tags, respectively. In other words, the orientation of a user will have less influence on the measured signal strengths and hence location tracking accuracy when multiple mobile tags are combined. The next section introduces the proposed location tracking system that mitigates the human body shadowing effect with a compensation and combination model, which can work independently.

3.4 System overview

A complete overview of the location tracking system with compensation for human body shadowing is shown as a flow graph in Figure 3.3.

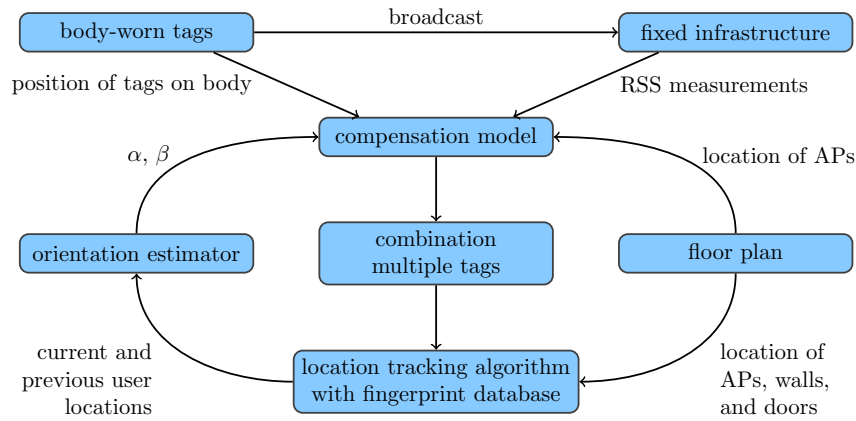


Figure 3.3: Flow graph of the location tracking system with human body compensation.

The starting point is the location tracking algorithm of Chapter 2, located on the bottom of the flow graph. This algorithm uses the preprocessed RSS measurements together with a fingerprint database and a floor plan with the location of all walls, doors, and APs, as input to calculate the user's location (Section 3.5). Next, the outputted locations are used to estimate the user's orientation (Section 3.6.2.1), which serves as an input for the compensation model (Section 3.6.2.2). To determine the influence caused by the user's body, this compensation model also uses the position of the mobile tags, the RSS measurements, and the location of the access points as input. Note that the RSS measurements originate from the fixed infrastructure as the body-worn mobile tags broadcast packets that are received by the fixed APs. Finally, the compensated RSS measurements of all tags are combined (Section 3.6.1) and passed on as input to the location tracking algorithm, and the process restarts.

3.5 Location tracking algorithm

This section reformulates the equations of the location tracking algorithm of Chapter 2, which serves as starting point for this work [15]. The algorithm is based on the Viterbi principle and uses the well-known RSS fingerprinting technique [16], a motion model, and off-the-shelf devices. In the standard RSS fingerprinting technique, without the Viterbi principle or semantic data, the current location is estimated by looking for the closest match in signal space, i.e., by comparing the measured signal strengths to the reference values in a fingerprint database. The closest match is based on the mean squared error (MSE), which results in the following cost function and least squares solution:

$$cost^{fp,j} = \frac{1}{N_{AP}} \sum_{i=1}^{N_{AP}} (RSS_{meas}^i - RSS_{ref}^{i,j})^2 \quad (3.1)$$

$cost^{fp,j}$ is the cost function for a grid point j in the fingerprint database, N_{AP} is the number of access points that received packets broadcasted by the body-worn tag(s), RSS_{meas}^i is the measured RSS value from access point i , and $RSS_{ref}^{i,j}$ is the reference RSS value for access point i and a grid point j in the fingerprint database. This results in following equation for the least square solution:

$$loc_{est} = \underset{j \in GP}{\operatorname{argmin}} cost^{fp,j} \quad (3.2)$$

loc_{est} is the most likely current location, which is the grid point j that

has the lowest associated cost ($cost^{fp,j}$) over all grid points GP in the fingerprint database.

The more advanced location tracking algorithm uses the environment of the user that is being tracked and a motion model as constraints to determine the most likely sequence of positions instead of only the most likely current position. These two constraints ensure that no walls are crossed and that no unrealistically large distances are traveled within a given time frame. The following cost function is used to determine the most likely path (after applying both constraints):

$$cost_{p,T}^{lta} = \sum_{t=1}^T \sum_{i=1}^{N_{AP}} (RSS_{meas}^{i,t} - RSS_{ref}^{i,p,t})^2 \quad (3.3)$$

$cost_{p,T}^{lta}$ is the associated cost of the location tracking algorithm (lta) for path p after T time steps. N_{AP} is the number of access points that received packets broadcasted by the body-worn tag(s). $RSS_{meas}^{i,t}$ is the RSS measurement at time step t from access point i and $RSS_{ref}^{i,p,t}$ is the reference RSS value from access point i for the position along path p at time step t . Note that the reference RSS values for a certain position are static, the subscript t is only used to indicate the position along path p . The last position of the path with the lowest associated cost is taken as most likely current location. The calculations of paths and costs are not restarted every time a new measurement is received but the paths and costs from a previous iteration serve as input for the current iteration along with the new measurements (Algorithm 1).

The reference RSS values are stored in a fingerprint database and are derived from the heuristic path loss model introduced in Section 2.3. These fingerprints are generated for a grid size of 50 cm, which determines the density of possible positions. In Section 2.8.2 it was proven that a higher density did not further improve the results and that a grid size of 50 cm is suited to work in real time. The advantage of a theoretical model is that it avoids an expensive and time-consuming measurement campaign but allows for an immediate deployment at the expense of a slightly reduced accuracy.

3.6 Compensation techniques

3.6.1 Combining multiple tags

As indicated in Section 3.3, the combination of measurements from multiple mobile tags, worn on different parts of the human body, reduces the influence of body shadowing. This results in a closer match between the preprocessed measurements and the reference RSS values from the fingerprinting

database, which suggests a possible improvement in tracking accuracy. Determining the optimal number and positions for the mobile tags depends on the required accuracy and wearing comfort in particular. Three positions are considered to improve the performance while maintaining practical tag positions and are shown in Figure 3.4: the central area of the chest and back (to have diversity in the forward and backward direction) and the right wrist (can be worn like a watch or integrated in one). Note that a user's wrists are slightly moving due to natural walking behavior and attenuation changes over time but the packets that are received within a second are averaged before they are passed as input to the tracking algorithm and compensation models. Interference between the different signals can be limited by the used modulation technique, e.g., direct-sequence spread spectrum (DSSS) [17].

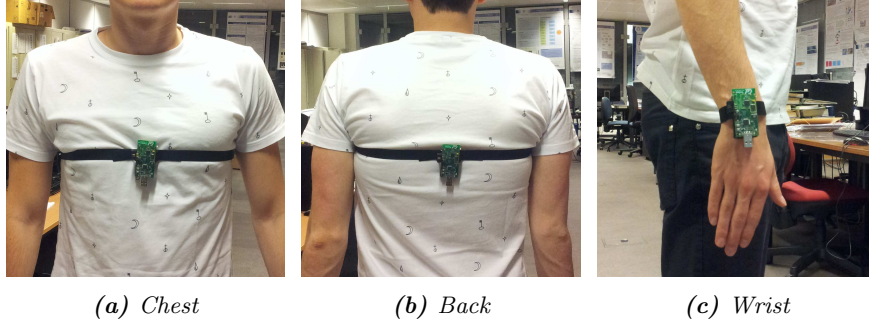


Figure 3.4: Body-worn tags.

These body-worn tags broadcast packets and the measured RSS values are combined in following cost function (based on Equation 3.4):

$$cost_{p,T}^{cmt} = \sum_{t=1}^T \sum_{b=1}^B \sum_{i=1}^{N_{AP}} (RSS_{meas}^{i,b,t} - RSS_{ref}^{i,p,t})^2 \quad (3.4)$$

$cost_{p,T}^{cmt}$ is the associated cost of the location tracking algorithm that combines multiple tags (*cmt*), for path p after T time steps. B is the amount of body-worn tags that are in use and b is a number referring to the body-worn tag's position, i.e., 1 = chest, 2 = back, and 3 = wrist. N_{AP} is the number of access points that received packets broadcasted by these body-worn tags. $RSS_{meas}^{i,b,t}$ is the RSS measurement from access point i of a packet broadcasted by the body-worn tag b at time step t . $RSS_{ref}^{i,p,t}$ is defined in Equation (3.3).

3.6.2 Compensating user orientation

The second approach compensates explicitly for the influence caused by a user with a certain position and tag orientation towards the infrastructure APs. It consists of two parts: an orientation estimator and a compensation model.

3.6.2.1 Orientation estimator

People tend to walk in the same direction for at least a few seconds due to the nature of most building structures and typical human behavior, which can be exploited to estimate a user's orientation. More specifically, the angle between previous and current location predictions is used as an estimation for the next orientation. Such an estimator is built on top of an existing tracking system that provides the current and previous positions as input. The next orientation is estimated as:

$$O_{t+1}^{est} = \frac{180}{\pi} \arctan \left(\frac{P_{t,y} - P_{t,y}^{am}}{P_{t,x} - P_{t,x}^{am}} \right) \quad [^\circ] \quad (3.5)$$

$$P_t^{am} = \frac{1}{K} \sum_{k=1}^K P_{t-k} \quad (3.6)$$

$O_{t+1}^{est} [^\circ]$ is the estimated orientation for the next time step ($t+1$), P_t^{am} is the arithmetic mean of K previous predicted positions P_t at time step t and the x and y subscript indicate the x and y coordinates. The two-argument arctangent function (*atan2*) is used to obtain the appropriate quadrant of the computed angle. The performance of this orientation estimator depends on the accuracy of the tracking system itself and on the movement of a user, i.e., frequency of turns taken and standing still moments. The goal is to investigate whether human body shadowing can be mitigated with this approach, instead of using the more precise measurements of an accelerometer, compass, or gyroscope, which require additional hardware and increase the cost of the location tracking system. For now, it is assumed that the user is walking forward. Including more previous location predictions (parameter K) improves the robustness and performance of the orientation estimator when few turns are present. However, this decreases rapidly when a trajectory with more abrupt changes is followed because a higher K implies a lower responsiveness. The reason for the initial improvement is that the predicted locations are not completely accurate, e.g., variations around an actually followed straight line, and taking into account multiple positions reduces the effect of prediction inaccuracies (averages out the error). This improvement stops when the user takes a turn during the last K positions

and the orientation error starts to deteriorate when even more previous positions are taken into account. Two trajectories are outlined on a floor plan to evaluate the orientation estimator with a simulation. The first trajectory has a length of 100 m and contains twenty-three 90° turns and two 180° turns, which results in an average of one turn every 4 m (blue trajectory in Figure 3.5). The second trajectory has fewer turns, longer straight segments, a total length of 56 m, and contains eight 90° turns, which results in an average of one turn every 7 m (red trajectory in Figure 3.5).

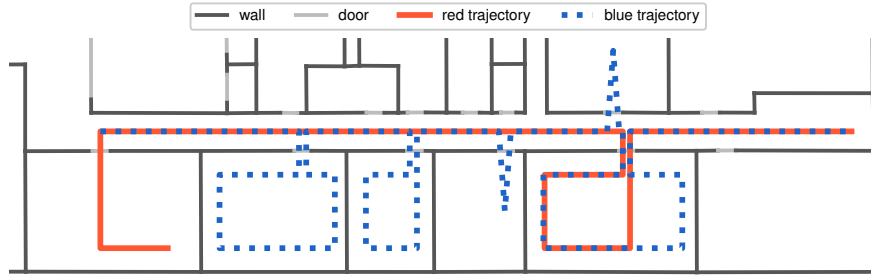


Figure 3.5: Red and blue trajectory for testing the orientation estimator (simulation).

The RSS values corresponding to the positions along these trajectories are selected from the fingerprint database (Section 3.5) and are used as input for the tracking algorithm (the walking speed is set to 1 m/s). Gaussian noise with a standard deviation of 1 dB, 3 dB, and 6 dB is added to these RSS values to simulate realistic conditions. Finally, the positions predicted by the tracking algorithm are used to estimate the orientation with Equations (3.5) and (3.6). The user's orientation ranges from 0° to 360° , meaning that an orientation error of 180° is the worst possible result. The orientation error is defined as:

$$\epsilon_{or} = \begin{cases} |O^{est} - O^{exact}| & |O^{est} - O^{exact}| \leq 180^\circ \\ 360^\circ - |O^{est} - O^{exact}| & |O^{est} - O^{exact}| > 180^\circ \end{cases} \quad [^\circ] \quad (3.7)$$

ϵ_{or} is the orientation error, O^{est} is the estimated orientation (Equation 3.5), and O^{exact} is the exact orientation, derived from the trajectories that are known in advance and visualized in Figure 3.5. Figure 3.6 shows a plot of the median orientation error as a function of the number of previous locations taken into account (parameter K from Equation 3.6). Note that the location update rate in the simulations is set at 1 Hz. The simulations are repeated five times for averaging purposes.

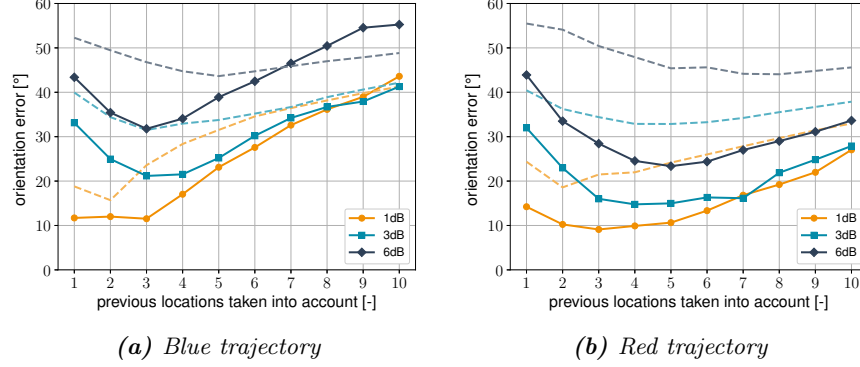


Figure 3.6: Performance of orientation estimator for the two trajectories and three noise levels. Solid line: median error, dashed line: standard deviation.

As expected, the orientation error initially decreases when more previous positions are taken into account (averages out inaccuracies) and then increases again for higher K values (due to turns taken during the last K positions). For the trajectory with many turns (blue), the optimal K value is 3 with median orientation errors (and standard deviations) of 12° (24°), 21° (32°), and 32° (47°), depending on the amount of added noise: 1 dB, 3 dB, and 6 dB, respectively. For the trajectory with longer straight segments (red), the optimal value for K is 5 with median orientation errors (and standard deviations) of 11° (24°), 15° (33°), and 23° (45°), depending on the amount of added noise: 1 dB, 3 dB, and 6 dB, respectively. Note that, the individual orientation errors are typically largest just after a turn because of the transition period K (Equation 3.6).

In the remainder of this work, the default K value is set to 4, which can deal with more abrupt trajectory changes and is still sufficiently accurate to compensate for the user's orientation (Section 3.7.4). Alternatively, this K value could be made adaptive based on the estimated trajectory up to the current location update, e.g., the value can be increased if the algorithm detects that a user is walking straight in the hallway or decreased if the user enters a room or approaches the end of a straight segment. The largest advantage of this approach is the ease of use and cost-effectiveness as there is no need for additional hardware to obtain the user's orientation. Disadvantages are the inability to detect a user rotating in place and the rather harsh orientation estimations but in Section 3.7.4 it is shown that this approach suffices to reduce the impact of human body shadowing.

Once the orientation of a user is known, the azimuth angle α and the elevation angle β between the body-worn tag(s) and the fixed infrastructure

nodes is calculated (Figure 3.7). It is assumed that the tag's position on the body and a floor plan with the locations of the fixed access points are known in advance and that the user is walking forward.

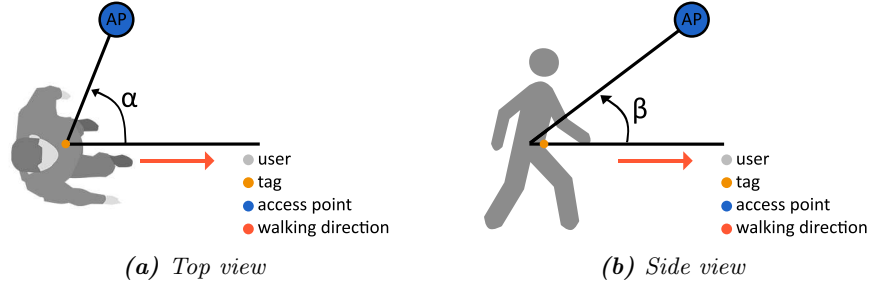


Figure 3.7: Angles between body-worn tag and AP.

3.6.2.2 Compensation models

Next, a compensation model is needed to estimate the influence due to the presence of a user. The mobile tags are calibrated based on the same measurements from Section 3.3: the average value between the measurements and the RSS fingerprinting database during the full rotation is taken as offset. This offset is inherent to a mobile tag and their transmit power, and is calculated once. This means that the measured RSS values will be an overestimation when tag and receiving node face each other directly and be an underestimation when the human body is completely blocking the signal. Two compensation models are used: a basic over / underestimation model and a simulation-based three-dimensional model:

- The first model labels the measurements as an over- or underestimation based on the azimuth angle α between tag and receiving node:

$$C_{o/u}(\alpha) = \begin{cases} +\frac{\text{compensation}}{2} & -\frac{\pi}{2} < \alpha \leq \frac{\pi}{2} \\ -\frac{\text{compensation}}{2} & \frac{\pi}{2} < \alpha \leq \frac{3\pi}{2} \end{cases} \quad [dB] \quad (3.8)$$

Figure 3.8a shows a plot of the $C_{o/u}$ (over/underestimation) compensation model. The default value of the *compensation* parameter from Equation 3.8 is set to 6 dB, which is found to be a good compromise between compensating for worst-case body shadowing and compensating for when almost no shadowing is present. Note that the maximum observed differences between the chest and back tag, measured by the forty fixed nodes, varied from 1.8 dB to 25.2 dB in the experiment of Section 3.3.

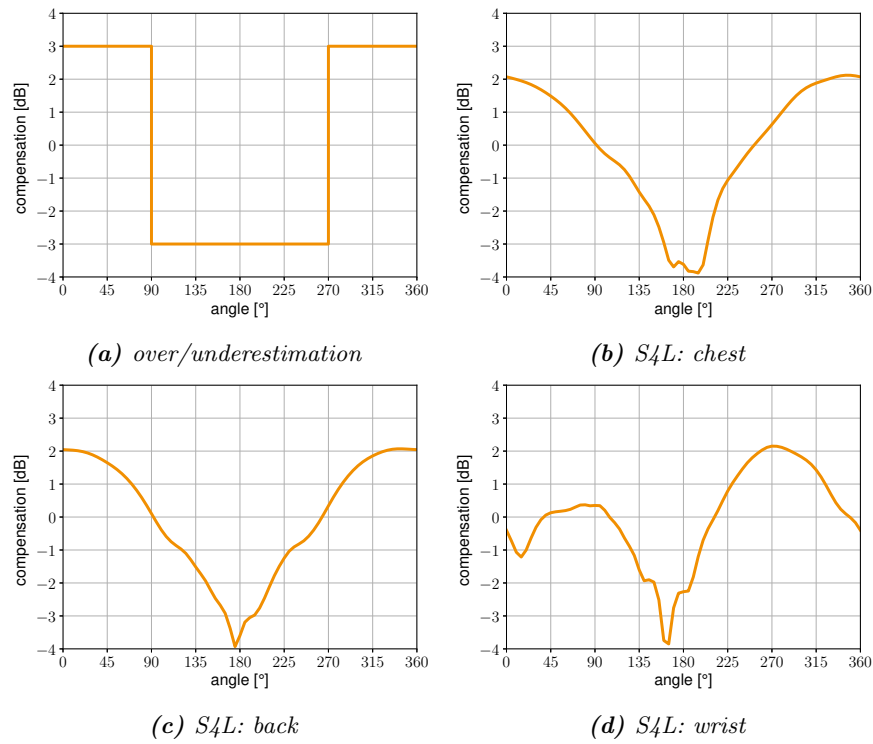


Figure 3.8: Graphical representation of the two compensation models.

- The second model is based on simulations carried out in Sim4Life (S4L), a three-dimensional full wave simulation environment based on the finite-difference time-domain (FDTD) method [18]. The simulations are executed at 2.45 GHz using the Virtual Family Male (VFM) [19]. The VFM is a heterogeneous phantom with a BMI of 22.3 kg/m². An accurate model of the integrated onboard antenna of the mobile tag is built in the simulation platform [20] (Figure 3.9a). This Planar Inverted-F antenna (PIFA) is made out of metal and mounted on a 1 mm thick, dielectric substrate with a relative permittivity of 4 and measures 65 mm by 31 mm [14]. The antenna is optimized to resonate at 2.45 GHz (a plot of the reflection coefficient S11 [dB] is shown in Figure 3.9b).

The mobile tag is placed on three practical positions of the VFM: chest, back, and right wrist. The simulated directivity pattern reflects the influence of the human body (Figure 3.9c). The values of the directivity pattern are shifted in such a way that the average value is zero and the range is normalized to the same *compensation* value of Equation 3.8 to be compatible with the mobile tag calibration. In this way a three-dimensional compensation model, that reflects the influence caused by the human body, is obtained. The second model is expressed as $C_{S4L,b}(\alpha, \beta)$, α and β are the azimuth and elevation angle between tag and receiving node, and b indicates the body-worn tag's position.

Including the compensation for a user's orientation, results in following cost function for the tracking algorithm (based on Equation 3.4):

$$cost_{p,T}^{cmt+c} = \sum_{t=1}^T \sum_{b=1}^B \sum_{i=1}^{N_{AP}} \left((RSS_{meas}^{i,b,t} - C_{j,b}(\alpha, \beta)) - RSS_{ref}^{i,p,t} \right)^2 \quad (3.9)$$

$cost_{p,T}^{cmt+c}$ is the associated cost of the location tracking algorithm that combines multiple tags and explicitly compensates ($cmt + c$) for the additional losses caused by human body shadowing, for path p after T time steps. N_{AP} is the number of access points and B is the amount of body-worn tags that are in use. $RSS_{meas}^{i,b,t}$ and $RSS_{ref}^{i,p,t}$ are already defined in Equations 3.3 and 3.4. $C_{j,b}(\alpha, \beta)$ is the j th compensation model, for mobile tag b and uses the angles α and β as input arguments. The subscript j is either *o/u* (over / underestimation) or *S4L*.

Note that the *o/u* compensation model makes no use of the elevation angle β and uses the same model for all mobile tags b . Figure 3.8 shows a

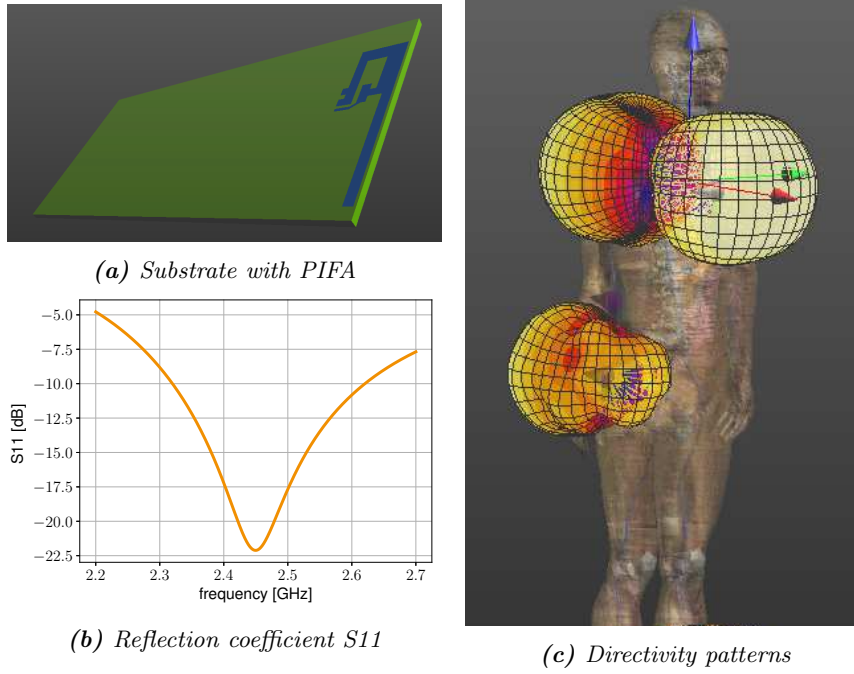


Figure 3.9: Simulations in S_4L .

graphical representation of the two compensation models. To obtain a two-dimensional plot, an elevation angle β of 20° is chosen for the visualization of the S_4L compensation model.

It is clear from Figure 3.8 that the S_4L compensation models do depend on the tag position but there are many similarities, especially between the tags placed on the central area of the chest and back. This is as expected because of the symmetry between both tag positions and hence RF signals will propagate through the same areas of the VFM. The wrist-worn tag compensation model deviates from the others, due to the human posture where there is some open space between arm and body.

3.7 Experimental validation

3.7.1 Configuration

The experiments are conducted in a wireless testbed, located on the third floor of an office building in Ghent, measuring 17 m by 90 m, and covering over 1500 m² (Figure 3.2). It consists of several computer labs, offices, and meeting rooms. The core of the building is made of concrete walls, the movable inner structure is made of layered drywall and the doors are made of wood. The wireless network consists of 48 fixed access points that are installed at a height of 2.5 m (blue dots in Figure 3.2). TelosB motes from Crossbow are used as the body-worn tags [14]. These are equipped with an embedded PIFA antenna and Chipcon CC2420 radio operating at 2.4 GHz (IEEE 802.15.4/ZigBee compliant [21]). There are 31 transmission power levels between -25 dBm and 0 dBm (set to 0 dBm in all experiments). The mobile tags broadcast 10 packets per second that are received by the fixed infrastructure nodes. Every second a location update is generated and the average RSS value of the packets received within this second per access point are used as input for the tracking algorithm. The test trajectory (indicated in red in Figure 3.10) has a total length of 140 m, passes through three meeting rooms, a computer lab, and the hallway (note that there are no access points installed in the second meeting room).

The ground truth, i.e., the correct locations for comparison, are provided by fragmenting the test trajectory based on the number of location updates. During the experiments the user walked as continuously as possible with an average speed of 1.2 m/s (4.32 km/h).

3.7.2 Results

This section investigates the impact of using multiple mobile tags and body shadowing compensation on the tracking accuracy. The presented results

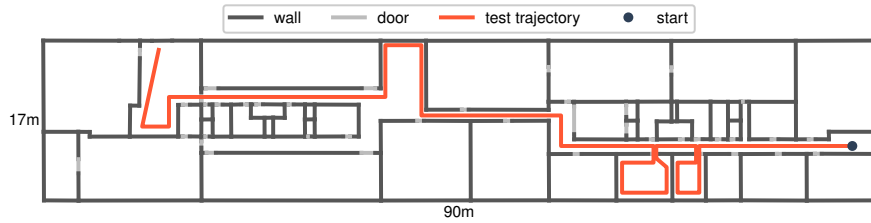


Figure 3.10: Floor plan of testbed with indication of test trajectory and start point (red dot).

are obtained from experiments performed on the testbed described in Section 3.7.1. The mean (μ), standard deviation (σ), median (50th), and 95th percentile value of the tracking accuracy are used as evaluation metrics. This accuracy is defined as the Euclidean distance between the estimated and actual location (Equation 2.4):

3.7.3 Impact of multiple mobile tags

To evaluate the performance when using multiple tags, the test trajectory is repeated 5 times by a human wearing three tags. The tags are placed on the central area of the chest, back, and right wrist (Figure 3.4). The two positioning algorithms from Section 3.5 are used for testing. The standard RSS fingerprinting technique is referred to as *basic* and the location tracking algorithm from Section 3.5 is referred to as *advanced*. The measured RSS values from the mobile tags are combined with Equation 3.4 (cost function from Section 3.6.1). Figure 3.11 shows the mean tracking accuracy as a function of the number of used mobile tags for the basic and advanced positioning algorithm. These accuracies are averaged over all possible combinations of the mobile tags, i.e., the results with one tag are the average of the three tags separately, the results with two tags are the average of the three possible combinations of two tags (*chest+back*, *chest+wrist*, and *back+wrist*), and for the results with three tags there is only one combination possible (*chest+back+wrist*). The standard deviation in mean accuracy of the individual tags compared to each other is 19 cm and for the three combinations of two tags this is 8 cm.

From Figure 3.11 it is clear that the mean tracking accuracy improves as more mobile tags are used, for both positioning algorithms. The relative improvement in mean accuracy is 19.1% (from 3.76 m to 3.04 m) when using two instead of one mobile tag, 14.8% (from 3.04 m to 2.59 m) when using three instead of two mobile tags, and 31.1% (from 3.76 m to 2.59 m) when using three instead of one mobile tag (calculated for the advanced location

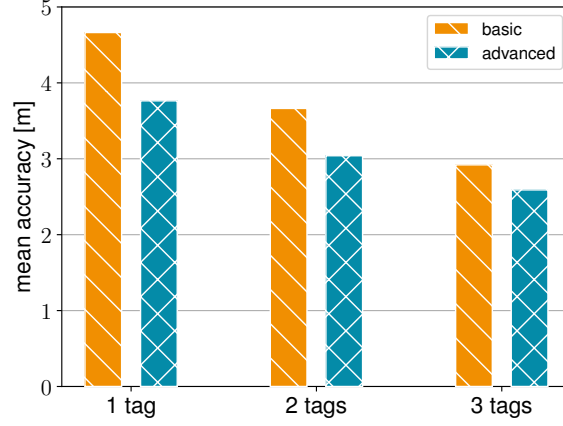


Figure 3.11: Mean accuracy as a function of the number of used mobile tags for the basic and advanced positioning algorithm.

tracking algorithm). Table 3.1 summarizes all evaluation metrics of the positioning accuracy and the improvement when using multiple tags, i.e., two vs. one, three vs. two, and three vs. one tag, as well as the improvement between the basic and advanced algorithm when one, two, and three mobile tags are used.

# tags	Basic [m]				Advanced [m]				Improvement [%]			
	μ	σ	50th	95th	μ	σ	50th	95th	μ	σ	50th	95th
1	4.66	3.21	4.17	10.68	3.76	2.28	3.48	8.50	19.3	29.1	16.6	20.5
2	3.66	2.41	3.28	7.89	3.04	1.86	2.77	6.71	17.0	22.8	15.4	15.0
3	2.92	2.03	2.63	6.41	2.59	1.63	2.30	5.66	11.3	19.7	12.5	11.7
2 vs. 1 [%]	21.5	24.9	21.3	26.1	19.1	18.4	20.4	21.1				
3 vs. 2 [%]	20.2	15.8	19.8	18.8	14.8	12.4	17.0	15.6				
3 vs. 1 [%]	37.3	36.8	36.9	40.0	31.1	28.5	33.9	33.4				

Table 3.1: Positioning accuracy as a function of the used mobile tags for the basic and advanced algorithm, and the improvement when using multiple tags (on the bottom) as well as the improvement between both algorithms (on the right).

Taking into account multiple mobile tags improves the standard deviation, median, and 95th percentile value in a similar way as the mean accuracy. Furthermore, the advanced algorithm always outperforms the basic one, as was already shown in Chapter 2. More specifically, the mean accuracy improves by 19.3%, 17.0%, and 11.3% when using one, two, and three mobile tags, respectively. The improvements in standard deviation are slightly higher: 29.1%, 22.8%, and 19.7% when using one, two, and three mobile tags, respectively. The median and 95th percentile value show sim-

ilar improvements as the mean positioning accuracy. Note that, although the results with the basic algorithm are less accurate, the improvement is higher when using multiple tags (values on the bottom of Table 3.1).

3.7.4 Impact of body shadowing compensation

The orientation estimator and three-dimensional compensation model ($S4L$) are verified with the same measurement data from Section 3.7.3. The *advanced* location tracking algorithm is used to provide the previous positions that are needed to estimate the user's orientation (Section 3.6.2.1). Two scenarios are considered: no compensation and compensation with the estimated orientation. The first scenario is the same as the best result from Section 3.7.3 (*advanced* in Figure 3.11). The second scenario compensates for the influence of a user's body with the three-dimensional $S4L$ compensation model from Section 3.6.2.2 and uses the orientation estimator to determine the user's orientation. Table 3.1 summarizes all evaluation metrics of the positioning accuracy for the advanced algorithm with and without compensation and the improvement between both when one, two, and three mobile tags are used.

# tags	No compensation [m]				Compensation [m]				Improvement [%]			
	μ	σ	50th	95th	μ	σ	50th	95th	μ	σ	50th	95th
1	3.76	2.28	3.48	8.50	3.29	2.10	2.99	7.56	12.5	7.9	14.1	11.1
2	3.04	1.86	2.77	6.71	2.66	1.80	2.28	6.52	12.5	3.2	17.7	2.8
3	2.59	1.63	2.30	5.66	2.33	1.54	1.91	5.23	10.0	5.5	17.0	7.6

Table 3.2: Positioning accuracy as a function of the used mobile tags with and without compensation, and the improvement between both.

Including the user's orientation to compensate for human body shadowing always improves the tracking accuracy (on top of the improvement thanks to using multiple mobile tags). Usage of the orientation estimator and compensation model result in additional improvements of 12.5% (from 3.76 m to 3.29 m), 12.5% (from 3.04 m to 2.66 m), and 10.0% (from 2.59 m to 2.33 m) in mean accuracy when using one, two, and three mobile tags, respectively. The total improvement in mean accuracy when using three mobile tags and compensating for body shadowing, compared to only one mobile tag and no compensation, is 38.0% (from 3.76 m to 2.33 m). The median accuracy is now below 2 m when using all three tags and the body shadowing compensation. The improvements in median accuracy are slightly better compared to the ones in mean accuracy: 14.1% (from 3.48 m to 2.99 m), 17.7% (from 2.77 m to 2.28 m), and 17.0% (from 2.30 m to 1.91 m) when using one, two, and three mobile tags, respectively. The

improvements in standard deviation and 95th percentile value are slightly lower compared to the improvements in mean and median accuracy.

The median *orientation* error for this trajectory, averaged over the 5 runs, is 39° , 35° , and 28° , for one, two, and three tags, respectively. These results are comparable to the simulations of Section 3.6.2.1. To estimate the effect of this orientation error on the positioning accuracy a third scenario is considered. In this scenario the exact orientation is passed to the positioning algorithm as an additional input. This is possible because the followed trajectory is known in advance and it is assumed a user walks forward, hence the exact orientation can be determined. Figure 3.12 shows the mean accuracy for the three scenarios with one, two, and three mobile tags.

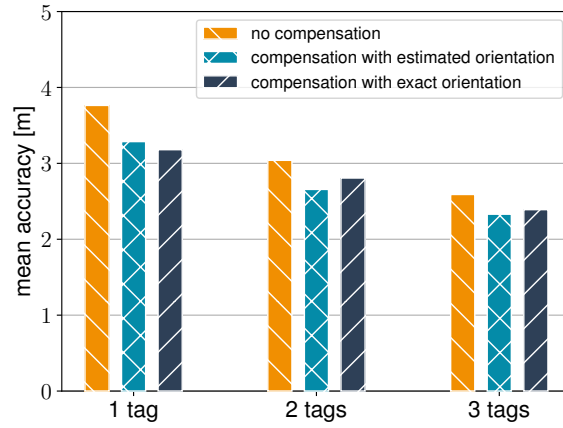


Figure 3.12: Impact of body shadowing compensation on the tracking accuracy.

As was already clear from Table 3.1 the location tracking accuracy always benefits from the compensation for human body shadowing. Furthermore, using the exact orientation has no added value compared to using the developed orientation estimator. Normally, this exact orientation is provided by a compass or gyroscope, but here this is accounted for by giving the exact orientation as an additional input. Therefore, using a compass or gyroscope has no significant added value for body shadowing compensation. In other positioning schemes they can be used as an additional feature to estimate the position itself, for example in dead reckoning their usage will be beneficial [22].

3.7.5 Comparison of compensation models

The compensation with exact orientation from Section 3.7.4 is used to evaluate the compensation models. In this way it is assured that the orientation

estimator (and indirectly the tracking algorithm itself) has no impact on the comparison.

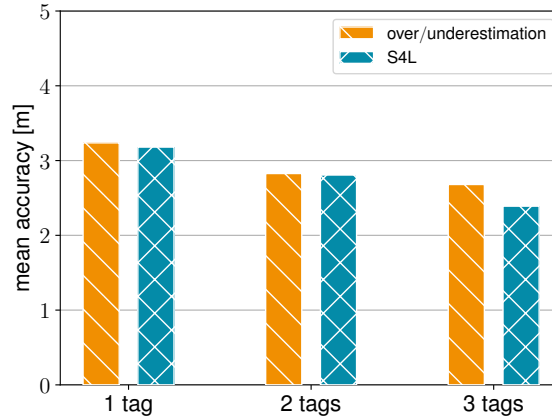


Figure 3.13: Comparison of different compensation models on the tracking accuracy.

Figure 3.13 shows the comparison between the *over/underestimation* and *S4L* compensation models from Section 3.6.2.2. The three-dimensional model (*S4L*) still has the most accurate results for all tag combinations, but the benefit is limited. Only for three tags it performs clearly better: an improvement of 10.8% in mean accuracy (and 5.9% in standard deviation). Although the simulations to determine the directivity pattern have to be performed only once (per position and mobile tag), they are rather time-consuming. It will depend on the application if it is worth performing these to obtain more detailed models.

3.8 Conclusion

In this chapter, novel techniques to reduce the effects of human body shadowing on a tracking algorithm's performance, are presented. Two methods are found to be effective: the first one combines multiple mobile tags, placed on different parts of the human body, and the second one compensates explicitly for the body shadowing caused by the user that is being tracked. Both methods can be independently combined, resulting in the most accurate performance. The first approach exploits the measured signal strengths from multiple mobile nodes, placed on different parts of a human body. Combining their measured signal strengths allows reducing the variation caused by the user's body and hence limits its influence. The second

method uses a compensation model based on three-dimensional electromagnetic simulations with a human phantom and relies on the orientation of a user's tag towards the infrastructure nodes. The user's direction is provided by an orientation estimator developed on top of a location tracking algorithm. This orientation estimator does not rely on dedicated hardware like a compass or gyroscope but uses previously estimated positions as input and achieves similar performance. Using three instead of one body-worn tag results in a mean accuracy of 2.59 m (versus an original accuracy of 3.76 m, corresponding to an improvement of 31.1%), in an office building that has 48 nodes and measures 90 m by 17 m (covering over 1500 m²). Compensating for the user's orientation, further improves this result to a mean accuracy of 2.33 m (a total and additional improvement of 38.0% and 10.0%). In Chapter 4, an unsupervised learning technique is proposed to construct, maintain, and optimize model-based radio maps for indoor positioning systems.

References

- [1] Paramvir Bahl, Venkata N Padmanabhan, and Anand Balachandran. *Enhancements to the RADAR user location and tracking system*. Microsoft Research, 2(MSR-TR-2000-12):775–784, 2000.
- [2] Anthea Wain Sy Au, Chen Feng, Shahrokh Valaee, Sophia Reyes, Sameh Sorour, Samuel N Markowitz, Deborah Gold, Kevaughn Gordon, and Moshe Eizenman. *Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device*. Mobile Computing, IEEE Transactions on, 12(10):2050–2062, 2013.
- [3] Paul Meissner, Erik Leitingner, Markus Fröhle, and Klaus Witrisal. *Accurate and robust indoor localization systems using ultra-wideband signals*. arXiv preprint arXiv:1304.7928, 2013.
- [4] Pavel Davidson, Jussi Collin, and Jarmo Takala. *Application of particle filters for indoor positioning using floor plans*. In Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010, pages 1–4. IEEE, 2010.
- [5] Begümhan Turgut and Richard P Martin. *Restarting particle filters: an approach to improve the performance of dynamic indoor localization*. In Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, pages 1–7. IEEE, 2009.
- [6] Henri Nurminen, Anssi Ristimäki, Simo Ali-Loytty, and Robert Piché. *Particle filter and smoother for indoor localization*. In Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on, pages 1–10. IEEE, 2013.
- [7] Henk Wymeersch, Stefano Marano, Wesley M Gifford, and Moe Z Win. *A machine learning approach to ranging error mitigation for UWB localization*. Communications, IEEE Transactions on, 60(6):1719–1728, 2012.
- [8] William PL Cully, Simon L Cotton, William G Scanlon, and JB McQuiston. *Body shadowing mitigation using differentiated los/nlos channel models for rssi-based monte carlo personnel localization*. In Wireless Communications and Networking Conference (WCNC), 2012 IEEE, pages 694–698. IEEE, 2012.
- [9] Xinfeng Li, Jin Teng, Qiang Zhai, Junda Zhu, Dong Xuan, Yuan F Zheng, and Wei Zhao. *EV-Human: Human localization via visual estimation of body electronic interference*. In INFOCOM, 2013 Proceedings IEEE, pages 500–504. IEEE, 2013.

- [10] Paolo Barsocchi, Francesco Furfari, Paolo Nepa, and Francesco Potortì. *RSSI localisation with sensors placed on the user*. In Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on, pages 1–6. IEEE, 2010.
- [11] Marco Altini, Davide Brunelli, Elisabetta Farella, and Luca Benini. *Bluetooth indoor localization with multiple neural networks*. In Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on, pages 295–300. IEEE, 2010.
- [12] Zengbin Zhang, Xia Zhou, Weile Zhang, Yuanyang Zhang, Gang Wang, Ben Y Zhao, and Haitao Zheng. *I am the antenna: accurate outdoor AP location using smartphones*. In Proceedings of the 17th annual international conference on Mobile computing and networking, pages 109–120. ACM, 2011.
- [13] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. *SpinLoc: Spin once to know your location*. In Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications, page 12. ACM, 2012.
- [14] *Crossbow TelosB Datasheet*. http://www.willow.co.uk/TelosB_Datasheet.pdf. Accessed: 2015-06-16.
- [15] Jens Trough, David Plets, Luc Martens, and Wout Joseph. *Advanced Real-Time Indoor Tracking Based on the Viterbi Algorithm and Semantic Data*. International Journal of Distributed Sensor Networks, 501:271818, 2015.
- [16] Yutian Wen, Xiaohua Tian, Xinbing Wang, and Songwu Lu. *Fundamental limits of RSS fingerprinting based indoor localization*. In 2015 IEEE Conference on Computer Communications (INFOCOM), pages 2479–2487. IEEE, 2015.
- [17] Lee Freitag, Milica Stojanovic, Sandipa Singh, and Mark Johnson. *Analysis of channel effects on direct-sequence and frequency-hopped spread-spectrum acoustic communication*. IEEE journal of oceanic engineering, 26(4):586–593, 2001.
- [18] Zurich MedTech. *Sim4Life (S4L)*. <https://zmt.swiss/sim4life/>. Accessed: 2015-06-16.
- [19] Andreas Christ, Wolfgang Kainz, Eckhart G Hahn, Katharina Honegger, Marcel Zefferer, Esra Neufeld, Wolfgang Rascher, Rolf Janka, Werner Bautz, Ji Chen, et al. *The Virtual Family-development of*

- surface-based anatomical models of two adults and two children for dosimetric simulations*. Physics in Medicine & Biology, 55(2):N23, 2009.
- [20] *2.4 GHz Inverted F Antenna*. <http://www.ti.com/lit/an/swru120b/swru120b.pdf>. Accessed: 2015-06-16.
- [21] Andreas F Molisch, Kannan Balakrishnan, Dajana Cassioli, Chia-Chin Chong, Shahriar Emami, Andrew Fort, Johan Karedal, Juergen Kunisch, Hans Schantz, Ulrich Schuster, et al. *IEEE 802.15. 4a channel model-final report*. IEEE P802, 15(04):0662, 2004.
- [22] Moustafa Alzantot and Moustafa Youssef. *UPTIME: Ubiquitous pedestrian tracking using mobile phones*. In Wireless Communications and Networking Conference (WCNC), 2012 IEEE, pages 3204–3209. IEEE, 2012.

Chapter 4

Radio map optimization through unsupervised learning

4.1 Introduction

Signal strength-based fingerprinting techniques estimate the position of an unknown user or object by finding the closest match in a radio map (online phase) [1, 2]. The radio map, which is also known as a fingerprint database, is a signal space that links RSS values to positions in a building. A major burden of signal strength-based fingerprinting for indoor positioning is the generation and maintenance of these radio maps.

Generally, the fingerprint database is constructed in an offline phase by making use of a radio channel simulator or an elaborate measurement campaign, also known as drive-testing or wardriving [3]. The first approach is simulation-based and hence much faster but will generally lead to less accurate location estimations. The second approach consists of manually performing RSS measurements at known locations and needs to be redone each time the wireless network or even the office layout undergoes changes.

This chapter presents a method to automatically construct, maintain, and optimize radio maps, without the need for inertial sensor units, calibration, or extensive measurements. The method is based on unsupervised learning, i.e., random walks for which the ground truth locations are unknown serve as input for the optimization, along with a floor plan, path loss model, and a location tracking algorithm. The objective is to learn the differences between reference values and real measurements on a per room basis to improve the quality of a radio map and hence enhance the

accuracy of an application that depends on this fingerprint database, such as a location or tracking system. No measurement campaign, site survey or calibration, which are labor-intensive and time-consuming, or inertial sensor measurements, which are often not available and consume additional power, are needed for this approach. Since a path loss model and location tracking algorithm are used as starting point for the learning phase and therefore domain knowledge is used, this approach is sometimes called semi-supervised learning. The remainder of this chapter uses the term unsupervised learning.

The remainder of this chapter is structured as follows, Section 4.2 describes related work and Section 4.3 presents an overview of the proposed system followed by a description of the experimental configuration, the initial radio-maps, and the self-calibration technique. Section 4.4 gives a motivation for the proposed approach, and discusses the radio map update method and the integration in the location tracking algorithm of Chapter 2. The simulation and experimental validation in a building-wide testbed are described in Sections 4.5 and 4.6. Finally, in Section 4.7, conclusions are provided.

4.2 Related work

Given the widespread use of WiFi access points or BLE beacons for indoor positioning purposes, it is paramount to find methods to obtain reliable fingerprinting maps with a minimal effort. In the past, other techniques for indoor positioning without the need for pre-deployment efforts, such as site surveys, measurement campaigns, or device calibrations, were proposed [4–8]. A related topic is simultaneous localization and mapping (SLAM), which constructs or updates a map of an unknown environment while keeping track of the agent’s location within it [9]. This map refers to the actual physical environment, whereas the proposed technique in this work focuses on optimizing the radio map or signal space of a building. The EZ algorithm is a configuration-free indoor positioning scheme that uses a genetic algorithm and occasionally available GPS locks, e.g., at the entrance or near a window, to localize mobile devices [4]. Another technique that bypasses wardriving is UnLoc [5], which uses dead-reckoning, urban sensing, and WiFi-based partitioning. A dead-reckoning scheme is used to track a user’s smartphone between so-called internal landmarks of a building, such as a distinct pattern on a smartphone’s accelerometer or an unusual magnetic fluctuation in a specific spot.

In [6], WiFi and inertial sensor information are combined with constraints imposed by a map of the indoor space of interest and augmented

particle filtering is used to estimate the position concurrently with other variables such as the stride length. A robust crowdsourcing-based indoor localization system to automatically construct a radio map is presented in [10]. The trajectories of the crowdsourced data are reconstructed based on activity detection, pedestrian dead-reckoning, and a semantic graph of the floor plan. A geomagnetism-aided indoor radio-map construction method based on crowdsourcing is presented in [11]. This method utilizes magnetism sequence similarity and a clustering algorithm to form the pathway graph of a floor plan, without needing an exact floor layout, but with the assumption of straight corridors. A factor graph optimization method is combined with the inertial-based readings of a smartphone for pedestrian dead-reckoning to generate a WiFi radio map in [12]. The factor graph optimization method is used to re-estimate the trajectory by adding constraints originated from collected WiFi fingerprints and landmark positions. A joint indoor positioning and radio map construction scheme is presented in [7]. This scheme transfers a source data set to a limited number of calibration fingerprints using manifold alignment. A crowdsourcing-based scheme to construct a probabilistic radio map based on parametric fitting is presented in [8]. This technique describes location signatures by transforming RSS into signal envelopes but relies on an additional positioning mechanism and a very large amount of RSS samples. A location tracking system that iteratively updates the channel parameters to generate the fingerprints is presented in [13, 14]. The possible trajectories for a user are based on a Voronoi diagram and particle filtering.

The approach in this work does not rely on any manual calibrations, pre-defined trajectories, known initial positions [14], measurement campaigns [3], GPS fixes [4], landmark positions [12], or inertial sensor units [6], such as accelerometers, gyroscopes, or magnetic compasses [11], to perform pedestrian dead-reckoning [5, 10, 15]. Only unlabeled training data, i.e., random walks in a building, and a floor plan are needed to construct, maintain, and optimize radio maps for indoor localization, e.g., to make model-based databases more accurate or to automatically cope with changes in the environment, office layout, or network infrastructure.

4.3 Methodology

Our approach consists of an initial radio map based on a theoretical path loss model (Section 4.3.2) and a self-calibration technique to match a user's device with this radio map (Section 4.3.3). The unsupervised learning technique to optimize the radio map uses a location tracking algorithm to reconstruct the most likely trajectory of unlabeled training data by including

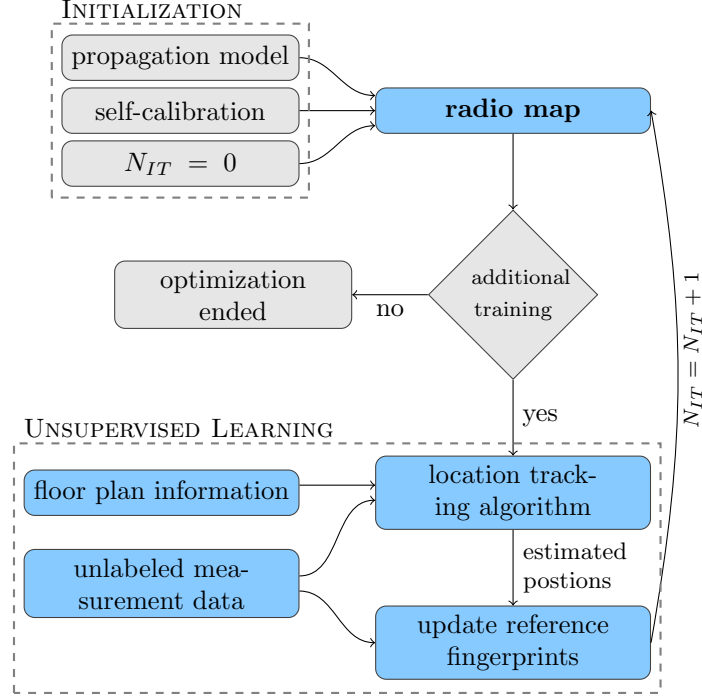


Figure 4.1: Flow graph of the proposed radio map construction and optimization technique. N_{IT} is the current training iteration.

floor plan information and the current radio map (Section 4.4.2). Next, the estimated positions from the reconstructed trajectories are used to update the reference fingerprints from the radio map (Section 4.4.3). The optimization ends when the maximum number of training iterations is reached or when the learned values remain the same between two training iterations. Figure 4.1 shows a flow graph of the proposed technique to construct and optimize a radio map.

The proposed unsupervised learning technique could run at a central location if the network infrastructure collects all RSS measurements or locally at a user's device if the latter collects all RSS measurements by scanning the access points in the area. The next sections describe each part in more detail, including a motivation for the chosen approach based on experiments conducted in an office building. Furthermore, three path loss models and three access point configurations are considered in the simulations and experimental validation for the proposed technique.

4.3.1 Experimental configuration

The experiments are conducted in a wireless testbed, located on the ninth floor of an office building in Ghent, Belgium, covering over 1100 m² (41 m by 27 m) and is visualized in Figure 4.2. The inner structure of the building is made of thick concrete walls (light gray) and the meeting rooms, offices, and kitchen have plaster walls (yellow), wooden doors (brown), and some glass walls (light blue).

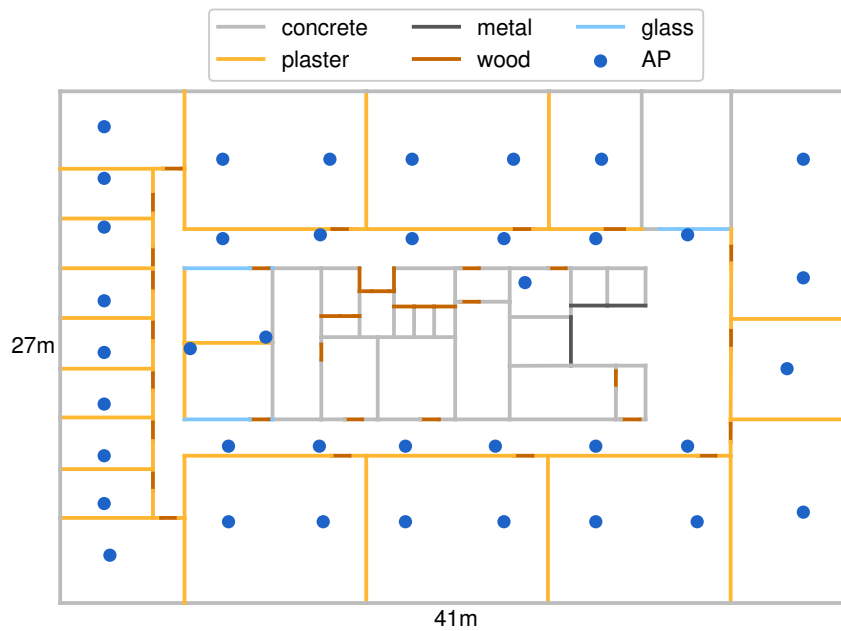


Figure 4.2: Floor plan of the office building with indication of walls, doors, and access points.

The wireless network consists of 35 fixed access points (sensor nodes) that are installed at a height of 3 m and are indicated with a blue dot. These sensor nodes are based on the Zolertia RE-mote platform, which is based on the Texas Instruments CC2538 ARM Cortex-M3 system on chip, with an on-board 2.4 GHz IEEE 802.15.4 RF interface. This interface runs at up to 32 MHz with 512 KB of programmable flash and 32 KB of RAM, bundled with a Texas Instruments CC1200 868/915 MHz RF transceiver to allow dual band operation [16]. The battery-powered mobile node is based on the same platform and is mounted on a tripod with a height of 1.5 m to collect static validation data (Figure 4.3).

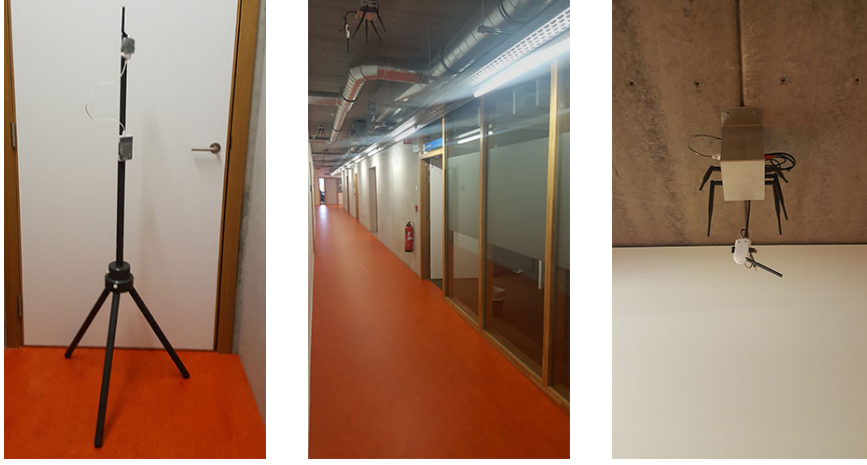


Figure 4.3: Tripod with battery-powered mobile node and ceiling-mounted fixed access points in corridor.

4.3.2 Radio map

The initial fingerprint database is a simulated radio map, that can be based on any propagation model. Note that to simulate path losses, the access point locations need to be known in advance. This is independent of the proposed unsupervised learning technique because the latter solely needs an initial radio map as starting point, which can be both measurement or simulation-based. Three different path loss models are considered as initial radio map to be optimized by the self-learning technique: free-space [17], IEEE 802.11 TGn [18], and WHIPP model [19]. Note that the first and third model have already been introduced in Chapter 2 (Section 2.3 and Section 2.8.4), but are repeated for completeness.

- Free-space model: the free-space path loss (FSPL) is the attenuation of radio energy between a sender and receiver antenna in idealized conditions, i.e., the antenna polarizations are perfectly matched, the environment is unobstructed free-space and the antennas are in each others far field. The FSPL is calculated as follows:

$$FSPL = 20 \log_{10}(d) + 20 \log_{10}(f) - 27.55 \quad [dB] \quad (4.1)$$

$FSPL$ [dB] denotes the free-space path loss, d [m] is the distance between the sender and receiver antenna and f [MHz] is the operating frequency, if this is set to 2400 MHz, then the model reduces to:

$$FSPL = 40.05 + 20 \log_{10}(d) \quad [dB] \quad (4.2)$$

- IEEE 802.11 TGn: the IEEE 802.11 TGn model is a two-slope path loss model, which is suitable for path-loss predictions in office environments. The TGn is calculated as follows:

$$TGNL = \begin{cases} PL_0 + 10n_1 \log_{10}(d) & d \leq d_{br} \\ PL_0 + 10n_2 \log_{10}(d) - 32 & d > d_{br} \end{cases} \quad [dB] \quad (4.3)$$

$TGNL$ [dB] denotes the path loss predicted by the TGn model, PL_0 [dB] is the reference path loss and is equal to 40.05 dB, n_1 and n_2 are the path loss exponents for the first and second part of the two-slope model and are equal to 2 and 5.2, and d_{br} [m] is the breakpoint distance and is equal to 10 m. For $d \leq d_{br}$, the TGn model and free-space model result in the same path loss.

- WHIPP model: the WHIPP model is a theoretical model for indoor environments that includes wall and interaction losses. This model does not use a ray tracing algorithm, but is based on a heuristic algorithm where the dominant path is searched, i.e., the path along which the path loss is the lowest (as discussed in Chapter 2, Section 2.3). Here, the path loss values are modeled as:

$$WL = \underbrace{PL_0 + 10\gamma \log_{10}\left(\frac{d}{d_0}\right)}_{\text{distance loss}} + \underbrace{\sum_i L_{W_i}}_{\text{cumulated wall loss}} + \underbrace{\sum_j L_{B_j}}_{\text{interaction loss}} + X_\sigma \quad [dB] \quad (4.4)$$

WL [dB] denotes the path loss predicted by the WHIPP path loss model, PL_0 [dB] is the path loss at a reference distance d_0 [m], γ [-] is the path loss exponent, d [m] is the distance along the path between transmitter and receiver. These two terms represent the path loss due to the traveled distance. The cumulated wall loss represents the sum of all wall losses L_{W_i} when a signal propagates through a wall W_i . The interaction loss represents the cumulated losses L_{B_j} caused by all propagation direction changes B_j along the path between sender and receiver, and X_σ [dB] is a log-normally distributed variable with zero mean and standard deviation σ , corresponding to the large-scale shadow fading.

4.3.3 Self-calibration

Off-the-shelf devices are usually not capable of measuring a path loss or true received RF power but instead report a received signal strength (RSS) value, which gives an indication of the received power. To be compatible with one of the theoretical path loss values, the RSS values are converted to a path loss value (or vice versa):

$$pathloss = -RSS + RSS_{bias} \quad [dB] \quad (4.5)$$

The RSS value is preceded by a minus sign because a higher path loss corresponds to a lower RSS value and the other way around. The RSS_{bias} is a fixed offset that is calculated once and depends on the access points and the user's device, e.g., transmitting power and antenna configuration, both of which are often unknown. Therefore, a self-calibration method [20] is used to obtain a good mapping between the measured RSS values and the reference path loss values from the radio map, also called fingerprints. This method relates the histogram of the reference radio map to the RSS histogram of a user's device, which requires no user intervention or ground truth location data, and results in the following bias:

$$RSS_{bias} = Md(F_{ref}^{-1}(y) - F_{meas}^{-1}(y)), \quad y \in \{0.1, 0.2, \dots, 0.9\} \quad (4.6)$$

RSS_{bias} is the estimated bias between the measurements of the user device and the fingerprint database, and is equal for all grid points and access points. The latter is only valid if all access points have the same configuration, e.g., transmit power, otherwise the self-calibration could be done for each access point individually. $Md(.)$ indicates the median value, F_{ref} is the cumulative distribution function (CDF) of the model-based reference fingerprints and F_{meas} is the empirical CDF of the RSS measurements from the user's device, multiplied by -1 to be compatible with the path loss values (Equation 4.5). The assumption behind this approach is that the empirical CDF of raw measurement values, collected during a random walk, resembles the respective empirical CDF of the mean measurement values collected with the same device at several uniformly distributed known positions. For the unlabeled training data, the value of RSS_{bias} stabilized after 14 s of measurement data (the sending rate was fixed at 5 Hz). Note that after the self-calibration the free-space and TGn path loss model can be seen as an intercept-fitted one-slope and two-slope path loss model with fixed path loss exponents. If always the same access points and mobile devices would be used, this automatic self-calibration would only have to be done once.

4.4 Unsupervised learning

This section starts with a motivation for the proposed learning technique. Next, the location tracking algorithm and update step for the reference fingerprints in the radio map are discussed (purple blocks in the flow graph of Figure 4.1).

4.4.1 Motivation

An experiment in an indoor environment showed that measurements of neighboring locations are similar and errors, i.e., measured deviations, from the initial model-based radio map tend to be correlated per room and access point. A mobile node, placed at 200 different locations, precisely measured with a laser meter, broadcasts packets at 5 Hz for 1 min while 35 fixed access points were listening. The 200 locations, ordered in a grid with a spacing of 2 m, are indicated in Figure 4.4 and referred to as *grid points* in the remainder of this section.

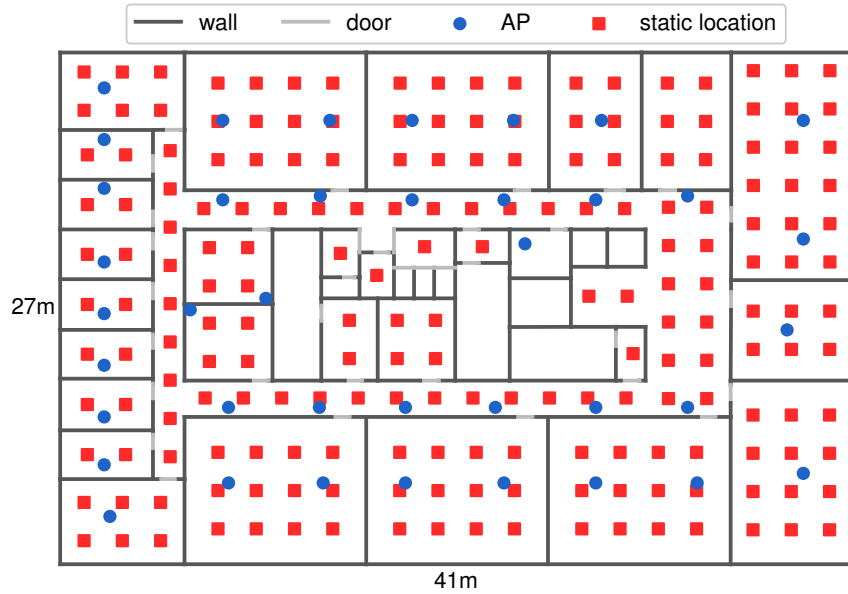


Figure 4.4: Floor plan of the office building with indication of walls, doors, access points, and static validation locations where a mobile node broadcasted packets at 5 Hz.

Next, the median self-calibrated values of the received measurements are compared to the corresponding reference values in the model-based radio map and are grouped per room and access point. These differences are

visualized for one access point in Figure 4.5, for each of the propagation models from Section 4.3.2.

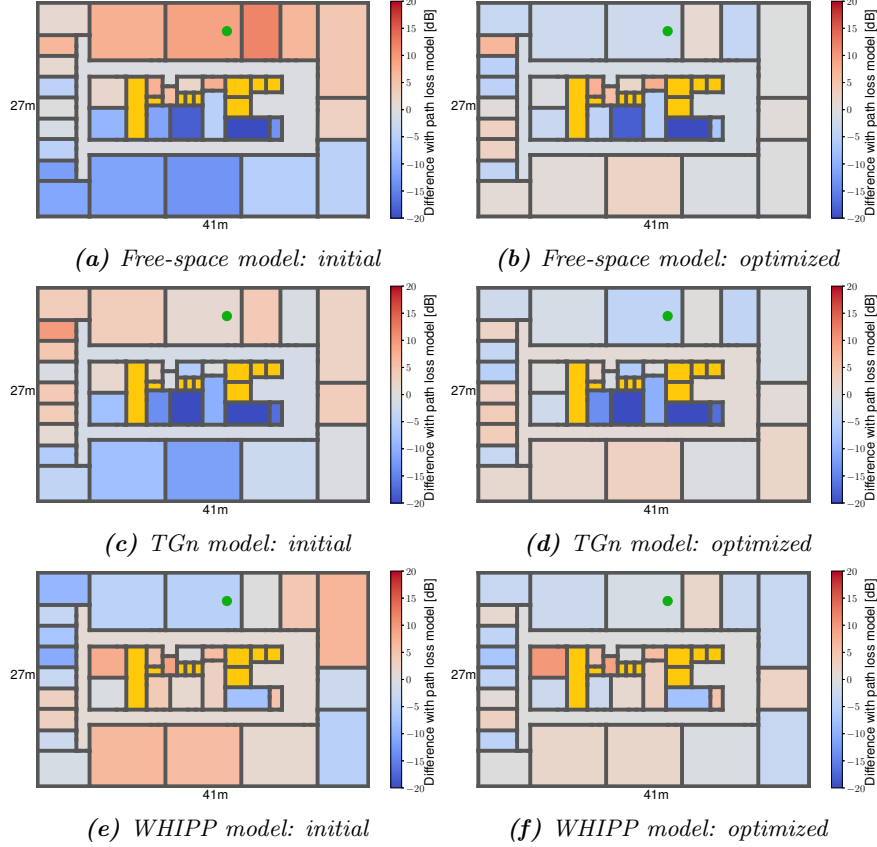


Figure 4.5: Average difference between model-based radio map and measurements, grouped per room, for one access point (green dot), before (a–c) and after the unsupervised learning (d–f). Unvisited rooms are indicated in yellow.

The access point is indicated with a green dot and the color in each room indicates the average difference from the path loss model (after the conversion from RSS to path loss). Rooms that are not visited are indicated in yellow. A blue color means that the theoretical model predicts a lower path loss value than what was actually measured, i.e., the received signal is weaker than what the theoretical path loss model predicts. This is clearly visible for the free-space and TGn model in the rooms on the bottom left (Figures 4.5a and 4.5e). The access point is located at the top middle and the inner structure of the building is made of thick concrete walls, which weaken the radio signals significantly but the free-space and TGn model

do not explicitly take into account the presence of these walls. A red color means that the theoretical model predicts a higher path loss value than what was actually measured, i.e., the received signal is stronger than what the theoretical path loss model estimates. This is the case for the WHIPP model, the same rooms on the bottom left have a reddish color because the model predicts a slightly higher path loss value than what was actually measured (Figure 4.5d).

Note that because of the self-calibration from Section 4.3.3, measurements can be stronger (lower path loss) than what the theoretical models predict because the calibration shifts all reference values with one fixed value to minimize the average differences between the measurements and the theoretical predicted values. Without this calibration phase, all deviations from the free-space path loss model would have a bluish color because the free-space model is more or less a lower limit for the measured path loss. The other access points show similar behavior but will have different values because this depends on the building’s layout and the relative location between room and access point. Comparison with the optimized radio map is discussed in Section 4.6 (Figures 4.5c, 4.5b, and 4.5f).

Additional sources of deviation from the theoretical predicted path loss values are temporal fading and human body shadowing [21]. Temporal fading is the variability of received power over time at a static location in the propagation environment. The influence of temporal fading is diminished by taking the median value over 300 values (1 min broadcast at 5 Hz) but will have an influence if only one sample is available, e.g., when a user is tracked while walking through a building. Human body shadowing is caused by the presence of a user, who can block the line-of-sight (LoS) between a body-worn tag and a receiving node, and causes additional propagation losses [22]. The amount of additional path loss depends on both the orientation of a person and the relative placement of the mobile tag on the body (Chapter 3). Methods to compensate for this loss include techniques that generate orientation-independent fingerprints by measuring RSS values for multiple orientations [23] or by modeling the signal attenuation caused by the human body [24]. Alternatively, the accuracy of a tracking algorithm can be improved by eliminating the shadowing caused by the human body as discussed in Chapter 3.

In this work, the influence of human body shadowing is eliminated by using a tripod instead of body-worn or hand-held device. This ensures that the performance of the proposed unsupervised learning technique is evaluated independently of human body shadowing. Note that this is only possible to collect the static validation data, the dynamic training data will be collected by a user with a hand-held device (Section 4.6). The deviations

between reference fingerprints and real measurements are classified in three categories:

- **overall deviation:** the overall deviation represents the variation for the whole building and is used as an indication of radio map quality. A value of zero would mean that the measured path losses are exactly equal to the theoretically predicted values at all locations, for all access points.

$$RSS_{diff}^{i,j} = RSS_{meas,sc}^{i,j} - RSS_{ref,pl}^{i,j} \quad (4.7)$$

$$\mu_{diff}^i = \frac{1}{N_{GP}} \sum_j^{N_{GP}} RSS_{diff}^{i,j} \quad (4.8)$$

$$dev_{overall} = \sqrt{\frac{1}{N_{AP} \cdot N_{GP}} \sum_i^{N_{AP}} \sum_j^{N_{GP}} (RSS_{diff}^{i,j} - \mu_{diff}^i)^2} \quad (4.9)$$

$RSS_{diff}^{i,j}$ is the difference between the self-calibrated (*sc*) measurement $RSS_{meas,sc}^{i,j}$ and the reference value $RSS_{ref,pl}^{i,j}$ for access point i , grid point j , and path loss model pl . The average difference for access point i is denoted by μ_{diff}^i , the total number of grid points by N_{GP} , the overall deviation by $dev_{overall}$, and the total number of access points by N_{AP} .

- **room deviation:** the room deviation models the difference between the radio map and the measurements, averaged over a whole room.

$$DIFF_{room}^{i,k} = \frac{1}{N_{GP}^k} \sum_j^{N_{GP}^k} RSS_{diff}^{i,j} \quad (4.10)$$

$$\mu_{room}^i = \frac{1}{N_{rooms}} \sum_k^{N_{rooms}} DIFF_{room}^{i,k} \quad (4.11)$$

$$dev_{room} = \sqrt{\frac{1}{N_{AP} \cdot N_{rooms}} \sum_i^{N_{AP}} \sum_k^{N_{rooms}} (DIFF_{room}^{i,k} - \mu_{room}^i)^2} \quad (4.12)$$

$DIFF_{room}^{i,k}$ is the average difference for access point i and room k , these values are visualized for one access point in Figure 4.5. N_{GP}^k are

the grid points within room k , $RSS_{diff}^{i,j}$ is defined in Equation (4.7), μ_{room}^i is the average room difference for access point i , N_{rooms} are the number of rooms in the building, dev_{room} is the room deviation, and N_{AP} the number of access points.

- **local deviation:** the local deviation represents the variation within a room on top of the room deviation, i.e., the differences between measured path loss values and the theoretical path loss values from the radio map are similar within a room but not exactly the same for all locations in this room.

$$DIFF_{local}^{i,j} = RSS_{meas,sc}^{i,j} - RSS_{ref,pl}^{i,j} - DIFF_{room}^{i,k_j} \quad (4.13)$$

$$\mu_{local}^i = \frac{1}{N_{GP}} \sum_j^{N_{GP}} DIFF_{local}^{i,j} \quad (4.14)$$

$$dev_{local} = \sqrt{\frac{1}{N_{AP} \cdot N_{GP}} \sum_i^{N_{AP}} \sum_j^{N_{GP}} (DIFF_{local}^{i,j} - \mu_{local}^i)^2} \quad (4.15)$$

$DIFF_{local}^{i,j}$ represents the local difference for access point i and grid point j , $DIFF_{room}^{i,k_j}$ is the room difference for access point i and room k_j , i.e., the room of grid point j (Equation 4.10), μ_{local}^i is the average local difference for access point i , N_{GP} is the number of grid points, dev_{local} is the local deviation, and N_{AP} is the number of access points.

Table 4.1 summarizes the statistics for the differences between the measurements and theoretical path loss models after self-calibration. Note that the measurements from all access points during the 1 minute broadcast on 200 different static locations are used.

Path loss model	RSS_{diff} [dB]			Deviation [dB]		
	min	max	avg	$dev_{overall}$	dev_{room}	dev_{local}
Free-space	-30.3	19.7	-0.4	10.9	9.7	3.5
IEEE 802.11 TGn	-37.5	19.1	-0.7	9.6	8.8	3.5
WHIPP	-22.0	24.0	0.8	7.6	5.7	3.7

Table 4.1: Experimentally measured differences and deviations compared to the theoretical path loss models after self-calibration.

The differences compared to the theoretical path loss values vary from -30.3 dB to $+19.7$ dB for the free-space model, from -37.5 dB to $+19.1$ dB for the IEEE 802.11 TGn model and from -22.0 dB to $+24.0$ dB for the WHIPP path loss model. The average differences are around 0 due to the self-calibration phase (-0.4 dB, -0.7 dB and 0.8 dB, respectively). The minimum and maximum difference could solely be caused by outliers, a better indication of the radio map quality is the overall deviation $dev_{overall}$ (Equation 4.9). This value is 10.9 dB for the free-space path loss model, 9.6 dB for the IEEE 802.11 TGn model, and 7.6 dB for the WHIPP path loss model, which is to be expected given the increased complexity. The room deviations dev_{room} (Equation 4.12) are 9.7 dB, 8.8 dB, and 5.7 dB, for the three path loss models, respectively. The local deviations dev_{local} (Equation 4.15) are similar for all three path loss models (3.5 dB, 3.5 dB and 3.7 dB, respectively). The dev_{local} is significantly lower than dev_{room} and $dev_{overall}$. This indicates that the RSS_{diff} from Equation 4.7 are correlated per room and hence knowledge of $DIFF_{room}^{i,k}$ can improve the radio map quality (visualized in Figure 4.5).

Under the assumption that a user's trajectory can be roughly reconstructed, i.e., with room-level accuracy, the $DIFF_{room}^{i,k}$ values (Equation 4.10) can be learned, resulting in a radio map or fingerprint database that matches the actual measurements more closely. Consequently, this optimized fingerprint database can increase the positioning accuracy of the trajectories or static locations of other users or objects. Note that a lower $dev_{overall}$ indicates that the theoretically predicted values are a closer match to the real measurements, which makes it easier to learn the $DIFF_{room}^{i,k}$ values because mapping unlabeled training data to the correct room will be more likely.

4.4.2 Location tracking algorithm

In this work, the trajectories of unlabeled training data are first reconstructed with location tracking algorithm of Chapter 2 and subsequently passed to the learning algorithm to optimize the radio map. By processing all available data at once, previously estimated locations can be corrected by future measurements (similar to backward belief propagation). This location tracking algorithm makes it possible to optimize the radio map because the estimated positions, along the reconstructed trajectories, are generally assigned to the correct room. Consequently, the discrepancies between reference fingerprints and real measurements can be learned, which improves the radio map quality and positioning accuracy. This is less likely with stateless positioning techniques, where consecutive estimated positions can fluctuate between different rooms because of measurement noise and outliers. Alternatively, a Kalman filter [25] or Particle filter [26] could be used

to reconstruct the trajectories but the location tracking algorithm resulted in a better accuracy as discussed in Section 2.6.1 (Chapter 2).

4.4.3 Radio map update step

After all training data is processed by the location tracking algorithm, the measurements are linked to a grid point and a room based on the estimated positions. The differences between self-calibrated RSS measurements and the corresponding reference fingerprints are grouped and averaged per estimated room and access point. Next, the reference fingerprints in the radio map are updated and the training iteration number N_{it} is increased by one (feedback loop in the flow graph of Figure 4.1). Note that reference fingerprints of unvisited rooms (or rooms where no training data is mapped) are unaffected by this update step. Furthermore, if training data is mapped to the wrong room, the radio map gets a partial erroneous update.

$$RSS_{ref,pl}^{i,j} = RSS_{ref,pl}^{i,j} + DIF_{room}^{i,k_j} \quad (4.16)$$

$RSS_{ref,pl}^{i,j}$ is the reference value of access point i , grid point j , and path loss model pl , that is updated in this training iteration. DIF_{room}^{i,k_j} represents the average difference between a set of self-calibrated measurements from access point i and room k_j , i.e., the room of grid point j . This is similar to Equation (4.10) but the input locations are now based on the training data and location tracking algorithm instead of the 200 static validation points. It is recommended to update the $RSS_{ref,pl}^{i,j}$ values after all training data has been processed at once. This gives the location tracking algorithm a chance to correct previously estimated positions by taking into account future measurements and to reduce the effect of outliers.

The update process is applied iteratively on the same unlabeled data until the learned values stagnate or when the maximum number of training iterations is reached. The reason for this is that the estimated trajectories from the unlabeled data tend to become more accurate in the next training iteration because it uses the current optimized radio map, which in turn results in a better update of the reference fingerprints. Also, if new unlabeled data becomes available, the optimization can start again to update the current radio map.

4.5 Simulations

A simulation with three different access point configurations is carried out to test the proposed unsupervised learning technique.

4.5.1 Settings

The simulation environment is the same as in Section 4.3.1, i.e., an office building in Ghent, Belgium, covering over 1100 m². The access point configurations are subsets of the access points from Figure 4.4: a dense scenario with 35 access points (Figure 4.6a), a normal scenario with 15 access points (Figure 4.6b), and a sparse scenario with 9 access points (Figure 4.6c). The WHIPP path loss model serves as basis to simulate real measurements because this model resembles a real-world scenario more closely [19], as shown in Section 4.4.1.

The first simulation takes into account the room differences $DIFF_{room}^{i,k}$ (Equation 4.10) and the local differences $DIFF_{local}^{i,j}$ (Equation 4.13). Both are modeled by Gaussian noise with zero mean and are added to the reference fingerprints, i.e., the theoretical path loss values from the WHIPP model. The standard deviation is varied from 0 dB to 16 dB in steps of 2 dB, which gives a total of 81 combinations to be simulated. The test environment consists of 40 rooms and 35 access points, resulting in 1400 $DIFF_{room}^{i,k}$ values for the dense scenario. The $DIFF_{local}^{i,j}$ values are generated for each grid and access point. A grid size of 50 cm results in 4386 grid points and 153510 $DIFF_{local}^{i,j}$ values for the dense scenario.

The second simulation considers two potential sources of additional noise on the measurements: temporal fading and human body shadowing. Both will have a major influence on the performance of the proposed unsupervised learning technique because it affects the accuracy of the reconstructed training data with the location tracking algorithm, as well as the learned values in the radio map update step. Both can be added together and are simulated by a single Gaussian noise with zero mean and the standard deviation is varied from 0 dB to 20 dB in steps of 2 dB. Note that the maximum observed influence of the human body varied from 1.8 dB to 25.2 dB with an average value of 14.8 dB in the experiment of Chapter 3 (Section 3.3). These noise values are generated each time the users passes at a location, whereas the $DIFF_{local}^{i,j}$ and $DIFF_{room}^{i,k}$ are fixed for each location and room in the building.

The objective is to learn the $DIFF_{room}^{i,k}$ values based on unlabeled training data to improve the quality of a radio map and hence enhance the accuracy of an application that relies on this fingerprint database, such as a location or tracking system. All other values act as additional noise on the measurements that make it harder to learn the $DIFF_{room}^{i,k}$ values (which will also be the case for a real-world scenario). Also, not all $DIFF_{room}^{i,k}$ values are equally important, e.g., a correctly learned value for a large conference room will have a greater impact on the accuracy than a wrongly learned value of a small storage room. Since more random training data

will pass through a larger room, it is also more likely that these values will be learned correctly.

One could argue that it is better to learn the difference for each grid point and access point individually instead of a global value per room and access point. However, simulations showed that this is unfeasible because a few learned values in a room can attract all future measurements of training data in that room, which will worsen the radio map and positioning accuracy. For example, suppose training data passes at a few grid points in a room and the reference fingerprints of these grid points are updated while all other fingerprints in that room remain unchanged. Next, if new training data becomes available in that room, these will be mapped to the few updated grid points if these are closer in signal space than the actual locations, i.e., when the difference with respect to their reference fingerprints is smaller than for the actual locations. This causes the optimization technique to assign all updates to only a few grid points per room.

The training data consists of a random walk simulating a user that walks freely in the building during 1 h with a random and variable walking speed between 4 km/h and 8 km/h. Every second, measurements are simulated by adding the generated noise values for that location and access point to the reference fingerprints. Depending on the sparse, normal, or dense scenario this number of reference fingerprints will vary between 9, 15, and 35, respectively. The quality of the radio map can be evaluated by comparing the learned values to the initially generated $DIFF_{room}^{i,k}$ values. Another indication of radio map quality is the accuracy with a static location algorithm, e.g., by taking the closest match in the optimized radio map as location estimation. This closest match is based on weighted least squares, i.e., strong signals have a greater weight than the weaker signals.

$$loc_{est} = \underset{j \in GP}{\operatorname{argmin}} \sqrt{\sum_i^{N_{AP}} RSS_{meas,sc}^{i,j} \cdot \left(RSS_{meas,sc}^{i,j} - RSS_{ref,pl}^{i,j} \right)^2} \quad (4.17)$$

loc_{est} is the estimated location, i.e., the grid point with the closest match in signal space, GP is the set of all grid points in the building, N_{AP} is the set of all access points i that have measurements for this location update, $RSS_{meas,sc}^{i,j}$ is the self-calibrated measurement and $RSS_{ref,pl}^{i,j}$ is the reference value for access point i , grid point j , and path loss model pl . Note that Equation 4.17 assumes positive RSS values, if the range of RSS measurements is negative, e.g., -90dB up to -40dB, the weighted effect is obtained by dividing by the absolute value of these negative RSS measurements. The maximum number of training iterations during the unsupervised learning is set to 10, i.e., the same unlabeled training data is used 10 times to learn the

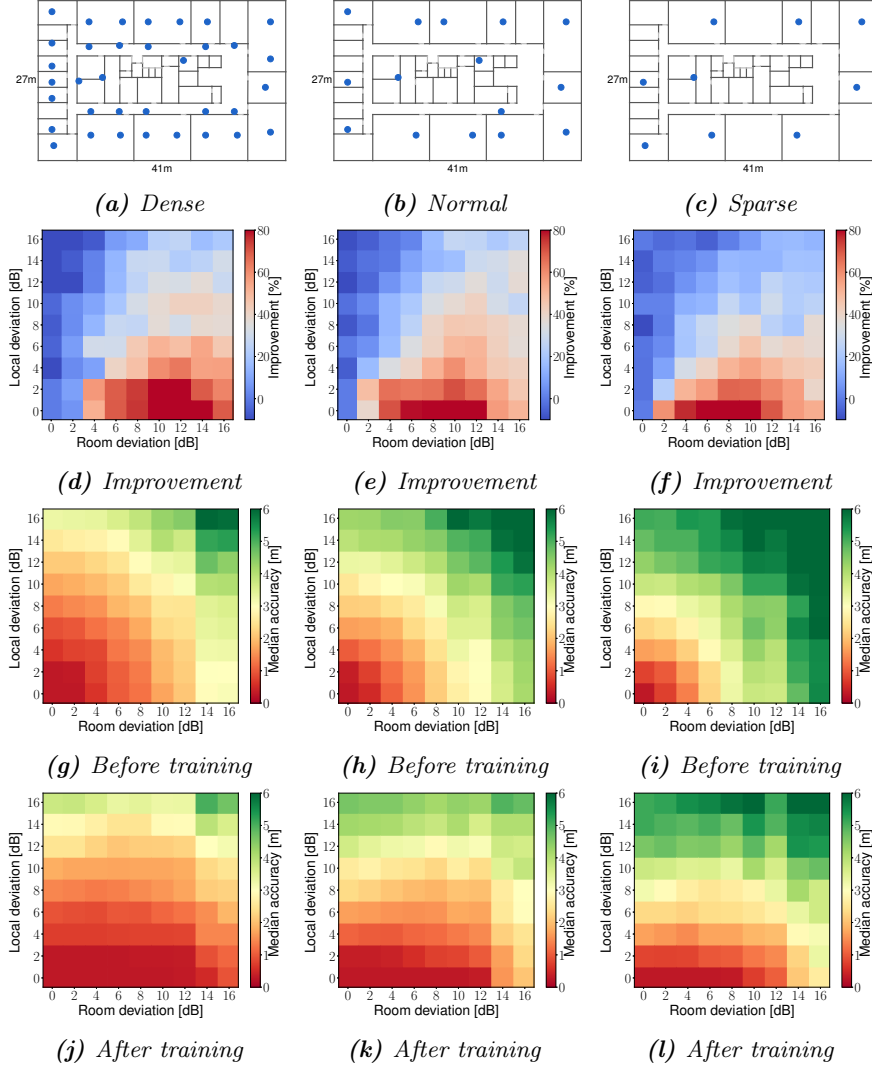


Figure 4.6: Location of the access points for the simulated scenarios with 35, 15, and 9 fixed access points: dense (a), normal (b), and sparse (c) configuration. The relative improvement in median accuracy after unsupervised learning with 1 h of unlabeled training data, evaluated on 1000 uniformly spread locations for the three scenario's (d–f). The median accuracy before (g–i) and after learning (j–l) for the three scenario's. The room deviation (x-axis) and local deviation (y-axis) vary from 0 dB to 16 dB in steps of 2 dB.

$DIFF_{room}^{i,k}$ values and update the radio map. The default value is set to 10 because the quality of the radio map and hence location accuracy always stopped improving before the seventh training iteration in the simulations (Section 4.5.2) and experimental validation (Section 4.6). Furthermore, the optimization is stopped early to speed up the process when the learned values remain the same between two training iterations (*additional training* block in the flow graph of Figure 4.1). Afterwards, the positioning accuracy is calculated based on 1000 uniformly spread locations with the original and optimized radio map to quantify the improvement.

4.5.2 Results

4.5.2.1 Influence of room and local deviation

The relative improvement in median accuracy as well as the median accuracy before and after the last training iteration are visualized per access point configuration in Figure 4.6. The x -axis and y -axis represent the room and local deviations, and the color indicates the relative improvement and median accuracy for the 81 scenarios. Each scenario, i.e., a colored square in Figure 4.6, is trained with 1 h of unlabeled training data, for a maximum of 10 training iterations, and is validated on 1000 uniformly spread locations. The relative improvements are similar for all three access point scenarios and the absolute accuracies are better for the dense access point scenario, as is expected.

The highest improvement is 89.7%, the median accuracy goes from 3.3 m to 0.3 m in the sparse access point scenario with a room and local deviation of 8 dB and 0 dB. In this scenario, the absolute values of the $DIFF_{room}^{i,k}$ to be learned are on average 6.1 dB. The average absolute difference between the true and learned values, after each training iteration, are: 1.5 dB, 1 dB, 0.7 dB, 0.6 dB, and stay at 0.5 dB from the fifth training iteration onwards. A local deviation of 0 dB means that all $DIFF_{local}^{i,j}$ values are zero, which makes it easier to learn the correct $DIFF_{room}^{i,k}$ values but is not a realistic scenario as shown in Section 4.4.1. The proposed unsupervised learning technique always improves the location accuracy, except for low values of the room deviation in combination with high values of the local deviation. This is to be expected as the $DIFF_{local}^{i,j}$ values are high, meaning there is very little correlation between the locations within a room, which act as an additional source of noise, but the $DIFF_{room}^{i,k}$ values are near zero, meaning the path loss model is already a good fit, which is also not a realistic scenario. In other words, the unsupervised learning technique works if the errors are correlated per room, which is the case according to the experiment of Section 4.4.1 (Figure 4.5). The nearest values to the experimentally

derived room and local deviation from Section 4.4.1 for the WHIPP path loss model would lead to an improvement of 40.7%, 40.1%, and 40.3% in median accuracy, for the three access point scenario's, respectively. Note that these simulations do not take into account the influence of additional noise, e.g., temporal fading and human body shadowing.

4.5.2.2 Influence of additional noise

A more realistic scenario is to include the additional noise caused by, e.g., temporal fading or human body shadowing. Figure 4.7a shows the relative improvement in mean, 50th, standard deviation, and 75th percentile of the accuracy for increasing noise levels. Figure 4.7b shows the absolute 50th and 75th percentile values of the accuracy before and after training. The unsupervised learning phase is the same as in the previous section: each noise level is trained with 1 h of unlabeled training data, for a maximum of 10 training iterations, and is validated on 1000 uniformly spread locations. The access point configuration is set to normal (15 access points) and the room and local deviation are set to 5.7 dB and 3.7 dB, which are the experimentally derived and hence realistic values from Section 4.4.1 for the WHIPP path loss model. The improvement in mean, median, standard deviation, and 75th percentile value of the accuracy show a very similar trend. The median location accuracy improvement starts at 43.8% (from 1.85 m to 1.04 m) without additional noise (0 dB) and decreases roughly linearly. The accuracy of the reconstructed training data needs to be at least accurate on room level, otherwise the radio map's reference values are affected by measurements from a neighboring room, which will worsen the accuracy in the next iteration. The proposed unsupervised learning technique can improve the radio map and positioning accuracy up to an additional noise of 16 dB, from then onwards the improvements are negative because the location tracking algorithm is no longer able to assign the measurements to the correct room, which results in a deterioration of the radio map.

4.6 Experimental validation

The test data for the experimental validation are the 200 static locations, uniformly spread over a whole floor in an office building (Section 4.4.1, Figure 4.4). The goal is to improve the quality of a model-based radio map and the location accuracy. The training data consists of a random walk of 15 min (or 900 location estimates) along the corridor, kitchen, offices, and meeting rooms, and is indicated with blue lines in Figure 4.8.

Note that this is a rough indication of the trajectory and that during

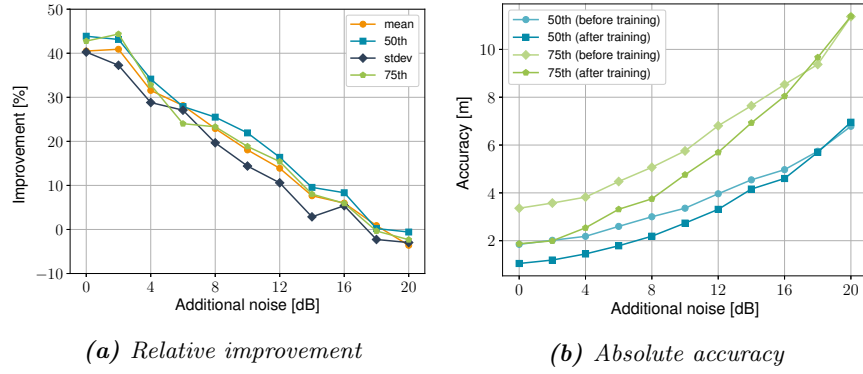


Figure 4.7: Accuracy before and after unsupervised learning for a varying level of additional noise with the normal access point configuration.

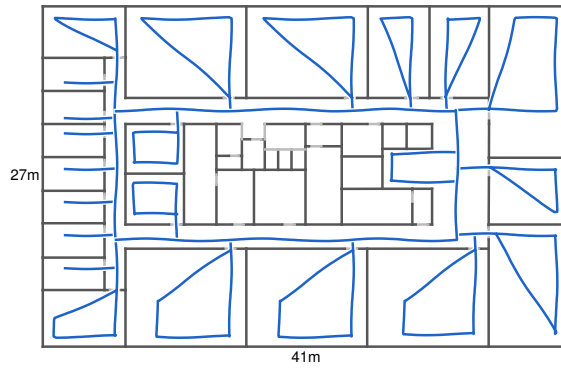


Figure 4.8: Floor plan with rough indication of the training data trajectory.

the random walk the mobile node is hand-held instead of mounted on a tripod, which causes additional deviation, as was previously mentioned. The exact positions are unknown and not needed for the learning phase, hence unsupervised, but are indicated to give an idea of the covered area. Obviously, only the rooms where the random walks pass by, can be learned. Therefore, most areas were covered except for the server room, elevators, bathrooms, stairwells, and storage rooms in the center. Every second a location update is generated, the average RSS values of the packets received within this second are used as input for the location tracking algorithm. Next, the estimated positions serve as input to the radio map update step.

Figure 4.5 shows the $DIFF_{room}^{i,k}$ values before and after the radio map optimization for one access point in the dense scenario. The colors after optimization (right hand side figures) are more grayish compared to the initial situation, which indicates that $DIFF_{room}^{i,k}$ values are closer to zero and thus, that the proposed technique learns the correct values and improves the radio map. For this scenario, the average absolute difference between the experimentally derived and learned $DIFF_{room}^{i,k}$ values decreases from 7.4 dB to 4.0 dB for the free-space model, from 5.7 dB to 3.7 dB for the TGN model, and from 4.7 dB to 2.9 dB for the WHIPP model. Note that in rooms where no training data passes, no values can be learned, i.e., their color remains the same. The accuracies and relative improvement, before and after training, are summarized for all access point configurations and path loss models in Table 4.2. Figure 4.9 shows the cumulative distribution function (CDF) of the positioning accuracy before and after training, for the three access point configurations and the WHIPP path loss model. These accuracies are based on the 200 static locations estimated with the initial and optimized radio map, and the weighted least squares algorithm (Equation 4.17). The location tracking algorithm is only used to reconstruct the unlabeled training data.

The initial accuracies before training are similar for the three path loss models, the largest difference in mean accuracy is 0.69 m (5.04 m and 4.35 m between the free-space and TGN model for the sparse access point scenario). The standard deviation of the accuracy is always best with the WHIPP model because the free-space and TGN model show larger positioning outliers on locations where there is a lot of additional path loss. The latter is caused by the concrete walls, which are better modeled in the radio map of the WHIPP model. The scenario with the dense access point configuration and WHIPP path loss model has the highest improvement; the median accuracy improves from 2.90 m to 2.07 m (28.6%). For this scenario, the median accuracy starts at 2.90 m and is consecutively: 2.40 m, 2.33 m, 2.22 m, 2.15 m, and finally 2.07 m after the fifth training iteration. The learned

#APs	PL Model	Accuracy [m]			
		μ	σ	50th	75th
9	Free-space	5.04 \rightarrow 5.11 (−1.3%)	3.93 \rightarrow 3.98 (−1.3%)	4.12 \rightarrow 3.93 (4.8%)	6.50 \rightarrow 6.35 (2.2%)
	TGn	4.35 \rightarrow 4.23 (2.9%)	3.73 \rightarrow 3.72 (0.3%)	3.14 \rightarrow 3.07 (2.1%)	5.72 \rightarrow 5.13 (10.4%)
	WHIPP	4.66 \rightarrow 3.77 (19.0%)	3.24 \rightarrow 2.49 (23.3%)	3.94 \rightarrow 3.03 (23.3%)	5.97 \rightarrow 4.83 (19.1%)
15	Free-space	4.28 \rightarrow 3.97 (7.4%)	3.43 \rightarrow 2.88 (16.1%)	3.49 \rightarrow 3.40 (2.4%)	5.03 \rightarrow 4.99 (0.8%)
	TGn	4.22 \rightarrow 3.96 (6.0%)	3.48 \rightarrow 3.18 (8.7%)	3.42 \rightarrow 3.31 (3.2%)	5.42 \rightarrow 4.71 (13.0%)
	WHIPP	4.33 \rightarrow 3.50 (19.1%)	2.98 \rightarrow 2.38 (20.0%)	3.50 \rightarrow 3.02 (13.7%)	6.03 \rightarrow 4.44 (26.4%)
35	Free-space	3.13 \rightarrow 3.22 (−2.9%)	3.09 \rightarrow 2.62 (15.5%)	2.40 \rightarrow 2.30 (4.1%)	3.61 \rightarrow 3.98 (−10.4%)
	TGn	3.65 \rightarrow 2.92 (20.1%)	3.18 \rightarrow 2.10 (33.9%)	2.75 \rightarrow 2.43 (11.7%)	4.51 \rightarrow 3.65 (19.1%)
	WHIPP	3.23 \rightarrow 2.66 (17.6%)	2.14 \rightarrow 1.74 (18.7%)	2.90 \rightarrow 2.07 (28.6%)	4.31 \rightarrow 3.52 (18.4%)

Table 4.2: Accuracy of experimental validation test set per access point configuration (sparse, normal, and dense) and path loss model. The first and second value are the accuracy before and after training and the third value is the relative improvement.

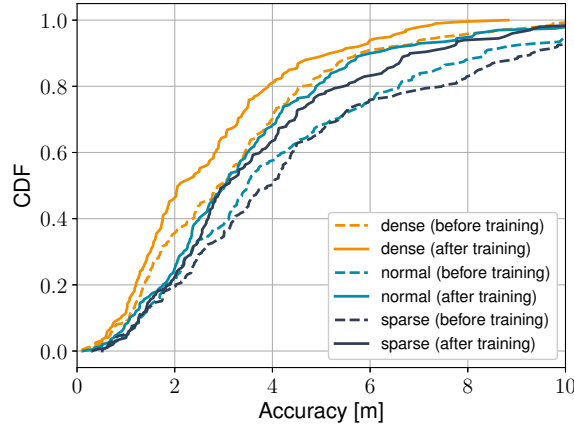


Figure 4.9: Cumulative distribution function of the positioning accuracy before (dashed line) and after training (solid line), for the three access point configurations and the WHIPP path loss model.

$DIFF_{room}^{i,k}$ values and hence the median location accuracy, remained the same in the sixth training iteration, which ended the optimization process. The largest relative improvement always occurs in the first training iteration and the learned values (and hence the location accuracy) stagnate after a maximum of six iterations, for all scenarios. There is only one substantial negative improvement, the 75th percentile accuracy for the free-space model in the dense access point configuration, decreases from 3.61 m to 3.98 m or a degradation of 10.4%. This occurs when the training data's estimated trajectory deviates too much from the ground truth locations, which causes the fingerprint database to learn RSS values measured in another room.

The highest accuracy, for all metrics and access point configurations, is achieved by the optimized WHIPP model, despite that in some scenarios the initial accuracy with the free-space or TGn model was better before training. The initial accuracy for those models are adequate due to the large number of line-of-sight connections with the dense access point configuration, resulting in stronger signals, which have a higher weight in the static location algorithm (Section 4.5). Furthermore, the improvement in their learning phase is limited because larger outliers occur with the free-space model and TGn model, e.g., if training measurements get assigned to a wrong room then this room attracts similar measurements in the next training iteration. Averaged over all access point configurations, the WHIPP model shows an improvement in mean, standard deviation, median, and 75th percentile accuracy of 18.6%, 20.7%, 21.9%, and 21.3%, respectively. This is similar to the simulation with the experimentally derived room and local deviation, and an additional noise level of 8 dB (Figure 4.7). The IEEE 802.11 TGn model is only slightly worse than the WHIPP model, which makes it a valuable alternative if the access point locations are known but further information about the building's layout is limited. The free-space model can result in adequate results as long as the access point configuration is not sparse but since the implementation effort is the same as with the TGn model, the latter is preferred.

4.7 Conclusions

This chapter presented an unsupervised learning technique to construct and optimize model-based radio maps or fingerprint databases for indoor positioning systems, e.g., to make the radio map more accurate or to automatically cope with changes in an office layout. The proposed technique does not rely on time-consuming measurement campaigns, device calibrations, or additional inertial measurement units, that are power consuming. Instead, it uses an initial radio map based on a theoretical path loss model, unlabeled

training data, a self-calibration method, and a location tracking algorithm. The premise of this approach is that the differences between real measurements and reference values, derived from a model-based radio map, tend to be correlated per room and access point. Three theoretical path loss models are considered: the free-space model, the IEEE 802.11 TGn model, and a model that takes into account wall and interactions losses (WHIPP). It was shown by measurements and simulations that the discrepancies between reference fingerprints and real measurements could be learned in various scenarios, based on the random walks by a typical person. This results in reference fingerprints that match the real measurements more closely and hence will lead to better radio maps and location accuracies. An experimental validation on a testbed in a large office building, measuring 41 m by 27 m (covering over 1100 m²) and that has 35 nodes, confirmed the simulations. The highest relative improvement is 28.6%, the median accuracy with the WHIPP path loss model improved from 2.90 m to 2.07 m after unsupervised learning with only 15 min of unlabeled training data. Furthermore, it is shown that the IEEE 802.11 TGn model is a valuable alternative if the information about the building's layout is limited. The next part of this dissertation is about map matching with support for lane detection based on GPS and accelerometer data (Chapter 5) and outdoor location tracking based on cellular network data (Chapter 6).

References

- [1] Paramvir Bahl and Venkata N Padmanabhan. *RADAR: An in-building RF-based user location and tracking system*. In INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 2, pages 775–784. IEEE, 2000.
- [2] Paramvir Bahl, Venkata N Padmanabhan, and Anand Balachandran. *Enhancements to the RADAR user location and tracking system*. Microsoft Research, 2(MSR-TR-2000-12):775–784, 2000.
- [3] Ionut Constandache, Shravan Gaonkar, Matt Sayler, Romit Roy Choudhury, and Landon Cox. *Enloc: Energy-efficient localization for mobile phones*. In INFOCOM 2009, IEEE, pages 2716–2720. IEEE, 2009.
- [4] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. *Indoor localization without the pain*. In Proceedings of the sixteenth annual international conference on Mobile computing and networking, pages 173–184. ACM, 2010.
- [5] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. *No need to war-drive: Unsupervised indoor localization*. In Proceedings of the 10th international conference on Mobile systems, applications, and services, pages 197–210. ACM, 2012.
- [6] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. *Zee: Zero-effort crowdsourcing for indoor localization*. In Proceedings of the 18th annual international conference on Mobile computing and networking, pages 293–304. ACM, 2012.
- [7] Sameh Sorour, Yves Lostanlen, Shahrokh Valaee, and Khaqan Majeed. *Joint indoor localization and radio map construction with limited deployment load*. IEEE Transactions on Mobile Computing, 14(5):1031–1043, 2015.
- [8] Qideng Jiang, Yongtao Ma, Kaihua Liu, and Zhi Dou. *A probabilistic radio map construction scheme for crowdsourcing-based fingerprinting localization*. IEEE Sensors Journal, 16(10):3764–3774, 2016.
- [9] Hugh Durrant-Whyte and Tim Bailey. *Simultaneous localization and mapping: part I*. IEEE robotics & automation magazine, 13(2):99–110, 2006.

- [10] Baoding Zhou, Qingquan Li, Qingzhou Mao, and Wei Tu. *A robust crowdsourcing-based indoor localization system*. *Sensors*, 17(4):864, 2017.
- [11] Wen Li, Dongyan Wei, Qifeng Lai, Xianghong Li, and Hong Yuan. *Geomagnetism-Aided Indoor Wi-Fi Radio-Map Construction via Smartphone Crowdsourcing*. *Sensors*, 18(5):1462, 2018.
- [12] Jian Tan, Xiangtao Fan, Shenghua Wang, and Yingchao Ren. *Optimization-Based Wi-Fi Radio Map Construction for Indoor Positioning Using Only Smart Phones*. *Sensors*, 18(9):3095, 2018.
- [13] Widyawan Widyawan. *Paten: A Location and Tracking System*. 2010.
- [14] Martin Klepal, Dirk Pesch, et al. *Location and tracking system*, December 22 2015. US Patent 9,217,788.
- [15] Antonio R Jimenez, Fernando Seco, Carlos Prieto, and Jorge Guevara. *A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU*. In *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, pages 37–42. IEEE, 2009.
- [16] *Zolertia Online Resources and documentation*, 2018 (accessed December 10, 2018). <https://github.com/Zolertia/Resources/wiki>.
- [17] *Free-space path loss*. https://en.wikipedia.org/wiki/Free-space_path_loss. Accessed: 2018-10-08).
- [18] V Erceg, L Schumacher, P Kyritsi, et al. *IEEE 802.11 document 03/940r4 (TGn Channel Models)*. Garden Grove, California, 2004.
- [19] David Plets, Wout Joseph, Kris Vanhecke, Emmeric Tanghe, and Luc Martens. *Coverage prediction and optimization algorithms for indoor environments*. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):1–23, 2012.
- [20] Christos Laoudias, Robert Piché, and Christos G Panayiotou. *Device self-calibration in location systems using signal strength histograms*. *Journal of Location Based Services*, 7(3):165–181, 2013.
- [21] Jens Trough, David Plets, Arno Thielens, Luc Martens, and Wout Joseph. *Enhanced indoor location tracking through body shadowing compensation*. *IEEE Sensors Journal*, 16(7):2105–2114, 2016.
- [22] Simon Schmitt, Stephan Adler, and Marcel Kyas. *The effects of human body shadowing in RF-based indoor localization*. In *Indoor Positioning*

- and Indoor Navigation (IPIN), 2014 International Conference on, pages 307–313. IEEE, 2014.
- [23] Ville Honkavirta, Tommi Perala, Simo Ali-Loytty, and Robert Piché. *A comparative survey of WLAN location fingerprinting methods*. In Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on, pages 243–251. IEEE, 2009.
- [24] Ngewi Fet, Marcus Handte, and Pedro José Marrón. *A model for WLAN signal attenuation of the human body*. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, pages 499–508. ACM, 2013.
- [25] Zhenghua Chen, Han Zou, Hao Jiang, Qingchang Zhu, Yeng Soh, and Lihua Xie. *Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization*. Sensors, 15(1):715–732, 2015.
- [26] Frédéric Evennou, François Marx, and Emil Novakov. *Map-aided indoor mobile positioning system using particle filter*. In Wireless Communications and Networking Conference, 2005 IEEE, volume 4, pages 2490–2494. IEEE, 2005.

Part II

Outdoor localization

Chapter 5

Map matching and lane detection

5.1 Introduction

Map matching is the problem of how to link recorded geographic coordinates to a road network of the real world, which is usually represented by a geographic information system (GIS). A typical example of map matching is to determine the trajectory of a moving object based on GPS data due to the nearly ubiquitous availability of the GPS signal [1]. The moving object can be a vehicle, cyclist, or runner, and the input GPS signals can be augmented with the data of an inertial measurement unit (IMU) to improve the accuracy or to lower the GPS update frequency and hence reduce the power consumption [2]. Other applications of map matching are, e.g., satellite navigation, freight tracking, activity recognition, road usage patterns, intelligent transportation systems (ITS), or urban traffic modeling. An example of the latter is to build statistical models of traffic delays that can be used to give suggestions to avoid traffic jams by finding the time-optimal driving route.

Map matching algorithms can be divided in real-time (online) and non-time critical (offline) algorithms. Real-time systems map the location to a road network during the recording process, whereas offline techniques are used to map geographic coordinates to a road network after all data has been recorded, which generally results in a better accuracy at the cost of a delay.

The most basic approach is to map each geographic input coordinate to the nearest point on the road network, but due to measurement noise this can easily result to a wrong reconstruction [3]. This basic approach

is also known as point-to-point map matching. The noise in GPS data is usually caused by the urban canyon effect, tunnels, or terrestrial features that affect the GPS signal, e.g., hills, forests, or valleys. Figure 5.1 shows an example of the typical problem with a basic nearest neighbor algorithm that maps the GPS point to the closest grid point on a road network, resulting in an unrealistic and physically impossible trajectory. Note that due to the urban canyon effect, the error between the raw GPS points and ground truth trajectory is up to 53 m in this real-world example.



Figure 5.1: Map matching problem: nearest grid point

Another problem is shown in Figure 5.2, the reconstruction is physically possible but the ground truth trajectory is more likely from a typical driver's point of view. However, the reconstructed trajectory can be the most likely in a map matching algorithm, e.g., if the objective is to minimize the Euclidean distance between the geographic input coordinates and the mapped trajectory. Merely minimizing a distance-based metric can result in unnecessary loops, U-turns, and overall weird driving behavior.

The map matched geographic coordinates can be enriched with driving lane information if additional data is available, e.g., monocular or stereo vision [4, 5], LIDAR (light detection and ranging) [6], or data from an IMU [7]. This is useful for accurate road surface monitoring or to obtain lane-specific statistical models about driving behavior. Note that the IMU data also



Figure 5.2: Map matching problem: minimizing a distance-based metric

allows to estimate human drivers' behavior, e.g., speeding, swerving, hard braking, or maneuvers, which is useful for, e.g., insurance companies, mobility operators, or to prevent potential car accidents [8, 9].

The remainder of this chapter is structured as follows, Section 5.2 describes related work of map matching and lane detection techniques. Section 5.3 discusses the grid, road segments, and proposed map matching algorithm and Section 5.4 introduces the proposed lane detection technique. Section 5.5 describes the evaluation trajectories and performance metrics, and discusses the results of the simulations and experimental validation. Finally, in Section 5.6, conclusions are provided.

5.2 Related work

5.2.1 Map matching

Curve-to-curve map matching is a geographic approach that uses a sequence of input coordinates to form a smoothed curve that is matched to a road segment with a similar geometry [10, 11]. Disadvantages are their sensitivity to measurement noise and the sampling rate, e.g., if only once every minute a GPS sample is available to limit the power consumption in devices with

limited battery capacity, then the curve based on the input coordinates can highly deviate from the real trajectory unless a user is driving on a straight road segment.

A possible solution are map matching algorithms based on the Hidden Markov Model (HMM) [1, 12, 13]. These systems can model the road infrastructure and take into account measurement noise and many different path hypotheses simultaneously. A prerequisite for HMMs is to model the observation and transition probabilities, i.e., to select candidate roads for the GPS observations. An HMM-based map matching solution that uses a route choice model estimated from real-world drive data to evaluate the paths generated by the HMM is presented in [13]. The route choice model is based on the concept of drivers' route and is used to avoid unreasonable paths generated by HMMs for highly noisy geographic data. The path-related parameters for this model are estimated based on a 1000 drives.

A conditional random field (CRF) is a type of undirected graphical model that is used to encode known relationships between observations, e.g., to segment and label sequential data [14]. In [15], a CRF is used to map match GPS trajectories at a low sample rate based on floating car data (FCD) with GPS trajectories of 70 taxis from one day. The dataset is split in training and test data (ratio 7:3) to select features, fine-tune hyper parameters, and evaluate the approach. A disadvantage of these data-driven techniques is that they depend on the quantity and quality of available historical data to improve the map mapping accuracy and hence will fail if parts of the input trajectory are not present in the dataset. A multi-track technique that simultaneously matches a collection of trajectories to a map is presented in [16]. This multi-track map matching is based on the observation that human drivers show a high degree of temporal and spatial regularity [17]. They exploit the regular structure in a large set of GPS traces by enforcing a set of partially overlapping trajectories to coincide their mapped paths in the intersection regions.

The basic assumption underlying HMM-based map matching algorithms is that the true mobility is Markovian [18]. However, in [19], it is argued that mobility is non-Markovian based on the Chapman-Kolmogorov equation [20] and a dataset with thousands of real taxicab rides spanning several weeks. The results show that there is a strong connection between the shortest path and the real trajectory traversed by the moving objects. Their proposed technique relies exclusively on shortest path computations and could achieve an improvement of 20% in accuracy over HMM-based approaches but this is largely due to the nature of the dataset, i.e., the moving objects (taxicabs) have intent to reach a specific destination in a timely manner, which is inherently suited for shortest path algorithms.

Furthermore, most map matching algorithms are evaluated by calculating the overlap between the ground truth path and the estimated path. This means that even if both paths completely overlap, the exact time at which a vehicle is at each road segment can be unknown or differ due to traffic delays, different road speed limits, unpredictabilities, or randomness in the driver behavior. This is disadvantageous for applications that need the exact location on the road network at each time instance, e.g., autonomous driving, precise registration of telematics data [21], pothole identification [22], or lane detection [7].

5.2.2 Lane detection

The problem of lane detection is usually tackled with vision-based techniques. These vision-based lane detection algorithms are categorized into feature-based and model-based. A feature-based algorithm uses low-level features, e.g., the solid or dashed painted lines on public roads, and image segmentation [23], deep learning [23, 24], or sensor fusion [25] to detect the lanes. A model-based approach uses a few parameters to represent the lanes, e.g., straight lines or parabolic curves, these parameters can be estimated by a Hough transformation [26, 27] or a likelihood function [28, 29]. A lane-detection method that extracts the lane marks based on color information in traffic scenes with moving vehicles is presented in [30]. Typical difficulties of vision-based lane detection algorithms are the large diversity in color and width of lane markings, image clarity, e.g., due to nearby vehicles, headlight glare, low sun angles, strong reflections, or faded lane markings, and the illumination problem at night or in bad weather, e.g., due to haze, fog, or rain [31, 32].

A possible solution is to use inertial sensor data to facilitate lane detection in all conditions. A probabilistic lane estimation algorithm that uses an unsupervised crowdsourcing approach to learn the position and lane-span distribution of the different lane-level anchors based on accelerometer, gyroscope, and magnetometer, is presented in [33]. The lane anchors are based on empirical assumptions, e.g., vehicle stops occur on the right most lane, U-turns occur on the left lane, and potholes typically span only one lane. A GNSS/INS integration system, i.e., global navigation satellite system / inertial navigation system, that utilizes ray tracing and a 3D building map to rectify ranging errors caused by multipath or non line-of-sight (NLoS) scenarios is presented in [34]. Experiments in an urban canyon demonstrated half-lane width errors 60% of the time. A lane-level positioning system based on crowdsourced location estimates of roadway landmarks and vehicular sensors is presented in [35]. The position estimates of other cars at these landmarks are combined with odometry and bearing information

from the vehicular sensors in a sequential Monte Carlo (SMC) method.

In this chapter, a method is proposed that combines the Markovian behavior and the shortest path aspect while taking into account the road speed limits and driving behavior. The proposed technique is a fast, memory-efficient, and worldwide map matching algorithm based on geographic coordinates and open map data, with support for lane detection that self-adapts to different driving behaviors. Usually map matching is aimed solely at car rides [15] but this algorithm is compatible with walks, bikes, and car rides, and is evaluated on real GPS data for a varying levels of measurement noise and temporal sparsity, based on precision, recall, and location accuracy on a per point basis instead of merely the overlap between the ground truth and estimated path. The total validation dataset is 9.4 hours, covers 301 km, consists of 33k GPS points, and 402k accelerometer samples. Furthermore, neither the map matching technique nor the lane detection algorithm depend on large training sets [15, 16] or crowdsourced measurement campaigns [33, 35]. The proposed techniques are offline algorithms, i.e., the geographic coordinates are mapped to a road network and the driving lanes are assigned after all data has been recorded. In Chapter 6 the map matching algorithm will be extended with an online version to process cellular network data in real-time.

5.3 Map matching

The goal of the proposed map matching algorithm is to output a continuous trajectory, i.e., connected road segments, based on timestamped geographic data as input.

5.3.1 Grid and road segments

The road segments are based on publicly available OpenStreetMap data, consisting of straight line segments enriched with metadata about the type of road, e.g., sidewalk, bike path, or highway, information about one-way traffic, relative layering, street name, number of driving lanes, and maximum allowed speed. Note that a default value is used when the maximum speed for a line segment is unknown or invalid, e.g., 120 km/h for highways, 90 km/h for primary roads, 70 km/h for secondary roads, and 50 km/h for all other road types. To take into account cars that are speeding and to avoid that location estimations are lagging behind, the allowed speed limit (for the reconstructed trajectory) can be increased by, e.g., 20% for each road segment. The straight line segments are further divided into equal pieces, i.e., road segments, based on the grid size. Note that line segments

smaller than the grid size also occur, e.g., at roundabouts, and are automatically included without further separation. The begin and end points of all road segments constitute the grid. For Belgium, this results in 13.4 million road segments and 12.2 million grid points for a grid size of 10 m. Figure 5.3 shows a sample of these road segments and grid points.

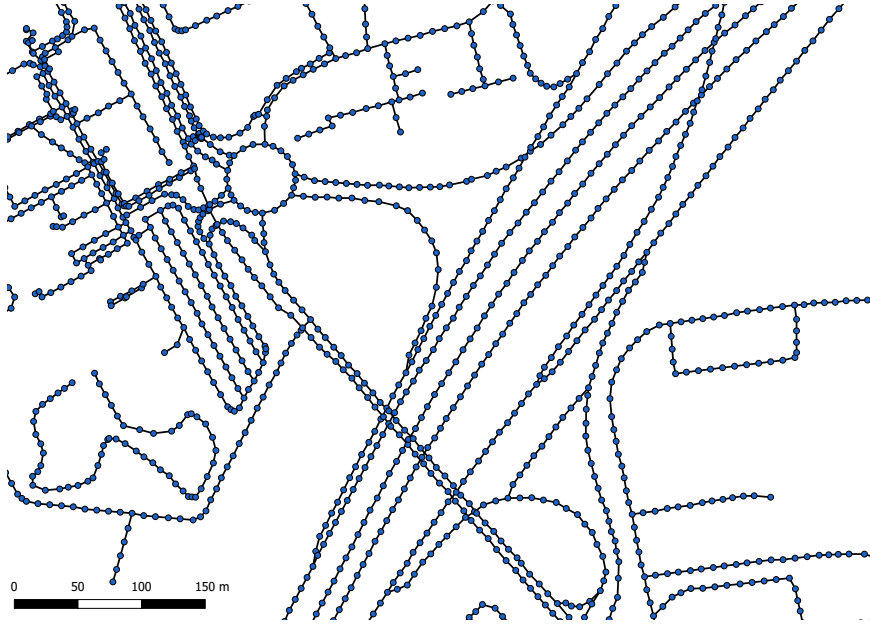


Figure 5.3: Grid and road segments based on OpenStreetMap data and a grid size of 10 m.

Note that, the outputted map matched locations are not limited to the grid points but can lie anywhere on the road segments because of the interpolation phase (discussed in Section 5.3.2). The grid and road segments are organized in square area blocks of 10 km^2 and calculated once based on the Winkel tripel projection [36], which is also the standard projection for world maps made by the National Geographic Society [37, 38]. The OpenStreetMap data for the whole world is 40.6 GB (June, 2019) [39]. The map matching starts with a scan of the geographic input data to load only the necessary area blocks and reduce the required memory. Furthermore, the proposed map matching algorithm is online available through a web service [40].

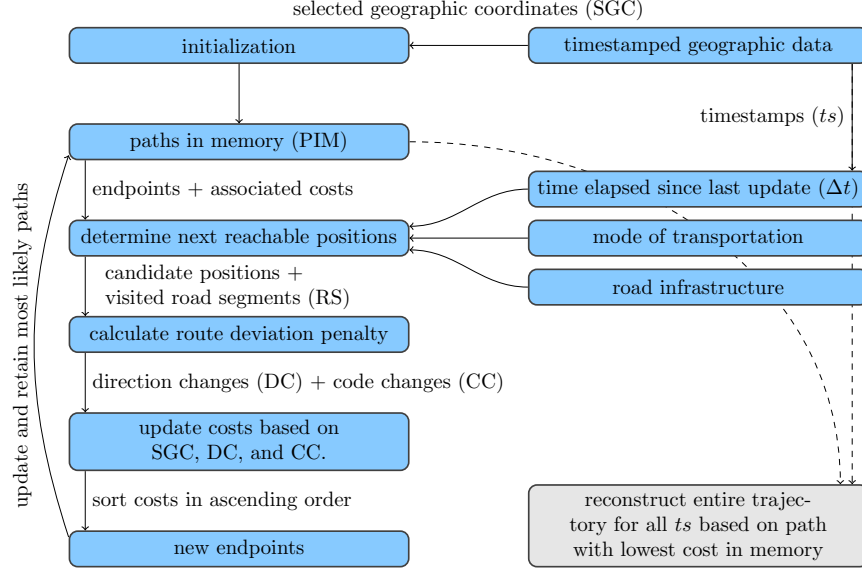


Figure 5.4: Flow graph of the map matching algorithm. The dashed lines are only executed at the end if all timestamped geographic data is processed. SGC: selected geographic coordinates, PIM: paths in memory, RS: road segments, DC: direction changes, and CC: code changes.

5.3.2 Algorithm

5.3.2.1 General

The proposed map matching algorithm is based on the Viterbi path, a technique related to hidden Markov models [41, 42]. The principle is roughly the same as in Chapter 2 but there are differences, e.g., the algorithm's input are raw locations, e.g., unmatched GPS data, instead of RSS values, the time between two location updates is usually much larger (up to a couple of minutes instead of every second), the semantic data is the road network instead of a floor plan, and the maximum speed depends on the mode of transportation and is variable per road segment. By processing all available data at once, previous estimated locations can be corrected by future measurements (similar to backward belief propagation). Naturally, this is only possible if the intended application tolerates a certain delay. Figure 5.4 shows a flow graph of the map matching algorithm, which ensures realistic and physically possible paths.

5.3.2.2 Geographic data selection

The proposed map matching algorithm starts with a selection of the geographic input coordinates (*initialization* in Figure 5.4) based on a minimum update distance and standing still detection, to avoid the map matching problem of Figure 5.2 by having a shortest path effect between the selected points combined with a direction and code change penalty.

The minimum update distance is the required traveled distance, based on the raw input data, before the next geographic coordinates are added to the selection. This is to establish a smoothing effect in the map matching, e.g., in the extreme case of only the begin and end point of a trajectory, the output is the shortest path between these two points.

The direction change penalty is the absolute sum of direction changes along the visited road segments between the estimations of two selected geographic coordinates, multiplied by a weight (β). The code change penalty is the total amount of road code changes along these visited road segments, multiplied by a weight (γ). Note that the road code indicates the type of road, e.g., highway, primary, secondary, or tertiary roads, residential area, sidewalks, or bicycle path. This direction and code change penalty are introduced to encourage the reconstructed trajectory to go straight on its current road segment and to diminish the weird route effect caused by noisy geographic data. For example, the unnecessary detour in Figure 5.2 caused by solely minimizing a distance-based metric can be avoided by adding a direction and code change penalty.

The default minimum update distance is 50 m, and the default weights for the distance-based penalty (α), direction change penalty (β), and code change penalty (γ) are 1 per meter, 10 per turn of 90°, and 10 per code change, respectively, which results in a unitless cost. The explanation behind these default values and the effect of these four parameters on the map matching accuracy are discussed in Section 5.5.3.3. Note that, the default minimum update distance has no effect if the sample rate of the geographic coordinates is low and the tracked object is on the move, e.g., a car driving at a rate of 120 km/h and a sample interval of 5 s already results in 167 m, and hence all geographic coordinates will be selected, i.e., are used as input in the map matching algorithm (*SGC* in Figure 5.4).

The interpolation between two selected geographic coordinates results in significant location errors if the tracked person or object is barely moving for a certain period between these two coordinates. Therefore, a standing still technique detects the start and end of intervals where there is no movement, e.g., a car waiting at a red traffic light. The begin and end of these intervals are included in the selection to ensure a correct interpolation.

5.3.2.3 Mapping and interpolation

The pseudo-code of the map matching method is shown in Algorithm 2 and the variables and steps are discussed in the text below.

After a selection of geographic coordinates is made based on the minimum update distance and standing still detection, the mode of transportation (*MoT*) is estimated based on the 95th percentile value of the maximum speed along the trajectory (to avoid picking an outlier). It is labeled as walking if it is below 10 km/h, as cycling if it is between 10 km/h and 40 km/h, and otherwise as driving a motorized vehicle. Note that if it is labeled as on foot or by bike, the highway road segments are discarded from the grid and if it is labeled as by car the sidewalks and bicycle road segments are discarded from the grid. Note also that a runner with a 95th percentile value of 12 km/h would be labeled as a biker but this has no influence on the performance; on the contrary, if he or she would be labeled as *walking*, the reconstructed path would lag behind the real trajectory. The map matching algorithm is initialized with the first geographic coordinate (GC) as current position (GC_0), e.g., a GPS data point. Then, a predefined number of other locations (*MP*) are selected around this position and their cost is initialized to zero, e.g., the 1000 closest grid points to the current position. This ensures that the map matching algorithm can recover from initially noisy GPS data, e.g., 1000 grid points and a grid size of 10 m resulted in covered surfaces between 18 and 75 hectares in the experimental validation of Section 5.5 (the exact area depends on the density of the road network). The initialization forms the starting point of all possible paths that are kept in the memory of the location tracking algorithm (*pathsInMem*).

Next, for the subsequent geographic coordinate (*GC*), all reachable positions (*RGP*) starting from the path's current last grid point (*PGP*: parent grid point) are determined for all *paths* in memory by making use of the surrounding road network, the time elapsed since last location update (Δt), the mode of transportation (*MoT*), and OpenStreetMap metadata, i.e., maximum speed, type of road, and one-way information. These reachable positions, which are also grid points, are the candidate positions for the next location update. This is similar to the reachable grid points of Chapter 2 (Algorithm 1) but transitions between grid points are now limited by an outdoor road infrastructure instead of the walls and doors of an indoor environment. Each candidate position (*CP*) retains a link to the parent grid point (*PGP*), a list with visited road segments *RS*, and a cost that represents this new branch along the road network. This new path (branch) and updated cost are added to the temporary list (*pathsTemp*) as a tuple ($(path_{new}, cost_{new})$). The updated cost is a weighted sum of a distance (*ED*), direction change (*DC*), and code change (*CC*) penalty. The

Algorithm 2: Map matching technique.

Data: timestamped geographic data (TGD)
Result: map matched trajectory (MMT)

- 1 $SGC \leftarrow$ selected geographic coordinates based on update distance and standing still detection
- 2 $MoT \leftarrow$ (estimated) mode of transportation
- 3 $GC_0 \leftarrow$ first geographic coordinates
- 4 $t_{prev} \leftarrow$ first timestamp
- 5 $MP \leftarrow 1000$ // maximum paths in memory
- 6 $pathsInMem \leftarrow$ list with MP grid points closest to GC_0 initialized with cost 0
- // iterate over all geographic coordinates in SGC
- 7 **for** $GC \in SGC$ **do**
- 8 $t \leftarrow$ current timestamp
- 9 $\Delta t \leftarrow t - t_{prev}$
- 10 $pathsTemp \leftarrow$ empty list
- 11 **for** $path \in pathsInMem$ **do**
- 12 $cost \leftarrow$ current cost of $path$
- 13 $PGP \leftarrow$ current endpoint of $path$ (parent grid point)
- 14 $RGP \leftarrow$ reachable grid points along roads with MoT within time span Δt starting from PGP
- // calculate new path cost for each candidate position (CP) based on distance, direction, and code change penalty
- 15 **for** $CP \in RGP$ **do**
- 16 $RS \leftarrow$ road segments between PGP and CP
- 17 $ED \leftarrow eucl_dist(CP, GC)$ // Euclidean distance
- 18 $DC \leftarrow$ penalty due to direction changes along RS
- 19 $CC \leftarrow$ penalty due to code changes along RS
- 20 $path_{new} \leftarrow path + RS + CP$
- 21 $cost_{new} \leftarrow cost + \alpha \cdot ED + \beta \cdot DC + \gamma \cdot CC$
- 22 add $(path_{new}, cost_{new})$ to $pathsTemp$
- 23 $pathsInMem \leftarrow$ retain MP paths from $pathsTemp$ based on lowest cost
- 24 $t_{prev} \leftarrow t$
- 25 $MMT \leftarrow$ reconstruct trajectory along $path$ with lowest cost in $pathsInMem$ for all timestamps t in TGD based on interpolation

distance penalty is the Euclidean distance between CP and GC , and the DC and CC penalty is calculated based on the directions and codes of the visited road segments RS , i.e., a physically possible path between CP and PGP .

Lastly, the MP paths with lowest cost are retained to serve as input for the next iteration. After all timestamped geographic data is processed, the entire trajectory of the path with lowest cost in memory is reconstructed for all timestamps in the geographic input data. The interpolation between the selected geographic coordinates is based on the visited road segments and a fixed or variable speed depending on the mode of transportation, i.e., for walks and bike rides this is fixed and for car rides this is the maximum road speed. The latter increases the accuracy if the road speed limit changes between two selected geographic coordinates, e.g., if once every minute a GPS sample is available and the car drives 30 s on a 90 km/h road and 30 s on a 30 km/h road, using regular interpolation would map too many locations to the first road. Note that this does not affect the precision, recall, or F_1 score (Section 5.5.2) but solely the prediction accuracy in meter (Section 5.5.3.1).

5.4 Lane detection

The mapped GPS coordinates can be enriched with information about the current driving lane, which is useful to map potholes to their exact location or to derive driving behavior statistics per lane on a highway. The number of driving lanes per road segment are included in the open map data and are added as metadata to the road segment (Section 5.3.1). Figure 5.5 shows a detail of the number of available lanes per road segment for an area surrounding a driveway exit.

5.4.1 Lane changes

The lane change detection algorithm is based on pattern recognition with accelerometer data. Note that, the sensor coordinate system of the accelerometer must be transformed to the car coordinate system if they are not aligned, e.g., based on inertial measurements or knowledge of the sensor placement. In the experimental validation (Section 5.5), the sensor placement was known and the coordinate systems of the accelerometer and car are equal (Figure 5.6), i.e., the x-axis is aligned with the driving direction (longitudinal axis), the y-axis is aimed towards the left lane (lateral axis), and the z-axis is aligned along the direction of gravity (vertical axis).

The accelerometer values are low-pass filtered to reduce the influence



Figure 5.5: Available driving lanes per road segment based on open map data.

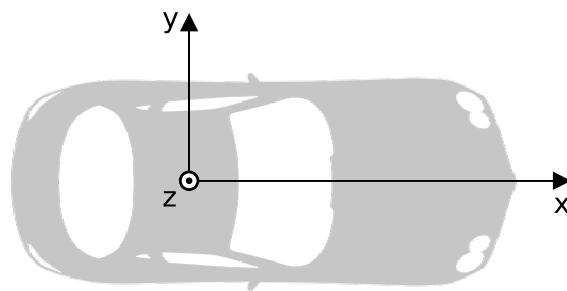


Figure 5.6: Alignment of sensor and car coordinate systems.

of random noise with a one second time window, which was found to be a good compromise between suppressing noise and distinguishable patterns.

$$acc_t^{lp,\{x,y,z\}} = \frac{1}{W} \sum_{i=-\frac{W}{2}}^{\frac{W}{2}} acc_{t+i}^{\{x,y,z\}} \quad (5.1)$$

$acc_t^{\{x,y,z\}}$ and $acc_t^{lp,\{x,y,z\}}$ are the raw and low-pass filtered accelerometer values for the x-, y-, and z-axis at time t , and W is the window size, e.g., for a sample rate of 100 Hz and a one second time window W is equal to 100. Figure 5.7 shows a sample of the raw and low-pass filtered sensor values for the lateral axis with indication of left and right lane changes for two trajectories with a different car and driver (Section 5.5.1).

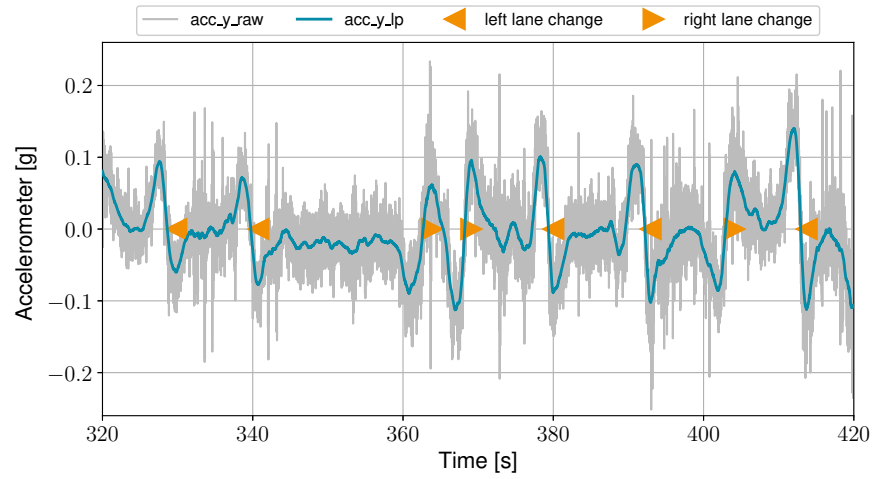
It is clear from this real example that left and right lane changes show unique and similar patterns on the lateral acceleration and that the low-pass filtering is necessary to distinguish these patterns. A lane change to the left starts with a positive peak followed by a slightly lower negative peak along the lateral axis, for a right lane change this is the other way around. Figure 5.7a shows a sample of 100 s on the highway with five lane changes to the left and three to the right. The three peaks around the 210 s time mark in Figure 5.7b are due to driving on a roundabout and taking the first exit. The small negative and positive peaks before the 210 s time mark in Figure 5.7b are caused by speed ramps and road bumps on a tertiary road with one lane. The peaks are detected with following formulas:

$$t_{peak_{pos}} = \left\{ t \mid acc_t^{lp,y} > acc_{t+i}^{lp,y}, |acc_t^{lp,y}| > \delta_{acc}, |i| \leq \frac{N}{2}, i \neq 0 \right\} \quad (5.2)$$

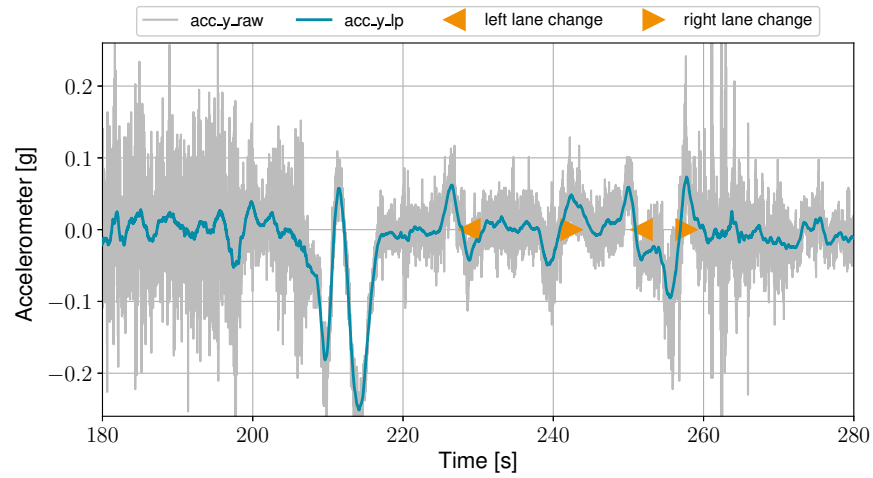
$$t_{peak_{neg}} = \left\{ t \mid acc_t^{lp,y} < acc_{t+i}^{lp,y}, |acc_t^{lp,y}| > \delta_{acc}, |i| \leq \frac{N}{2}, i \neq 0 \right\} \quad (5.3)$$

$t_{peak_{pos}}$ and $t_{peak_{neg}}$ are the timestamps of all positive and negative peaks, $acc_t^{lp,y}$ is the low-pass filtered lateral accelerometer value at time t (Equation 5.1), δ_{acc} is the noise floor for peak detection and is set at 0.02 g, and N is the total number of considered accelerometer samples and is set at a value that matches an interval of 2 s, e.g., 200 if the sample rate is 100 Hz. In the remainder of this section, $t_{peak_{pos}}$ and $t_{peak_{neg}}$ are referred to as (positive and negative) peaks.

Figure 5.8 shows a sample of the driving lane at each road segment for both trajectories (the parts on the highway are traveled in both ways).

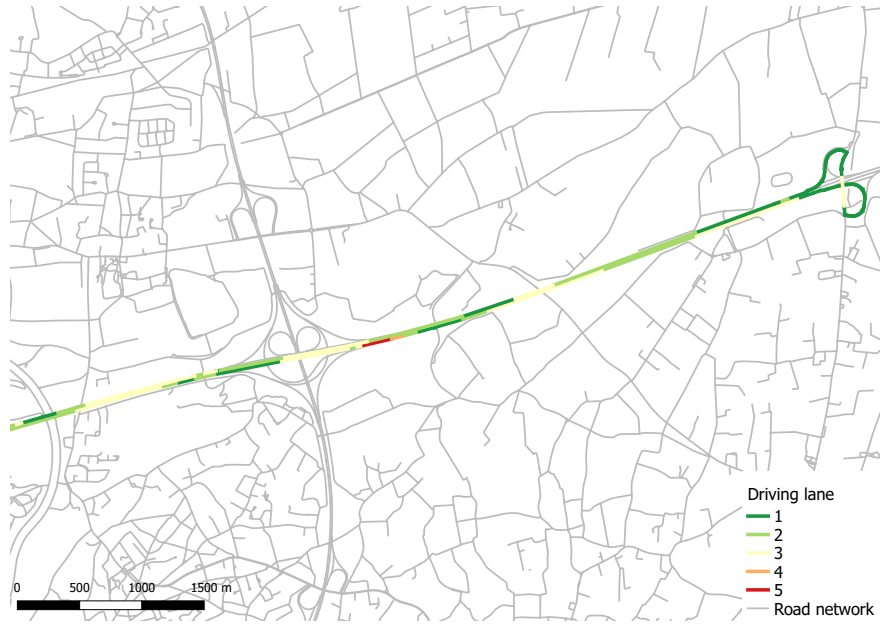


(a) Trajectory 1

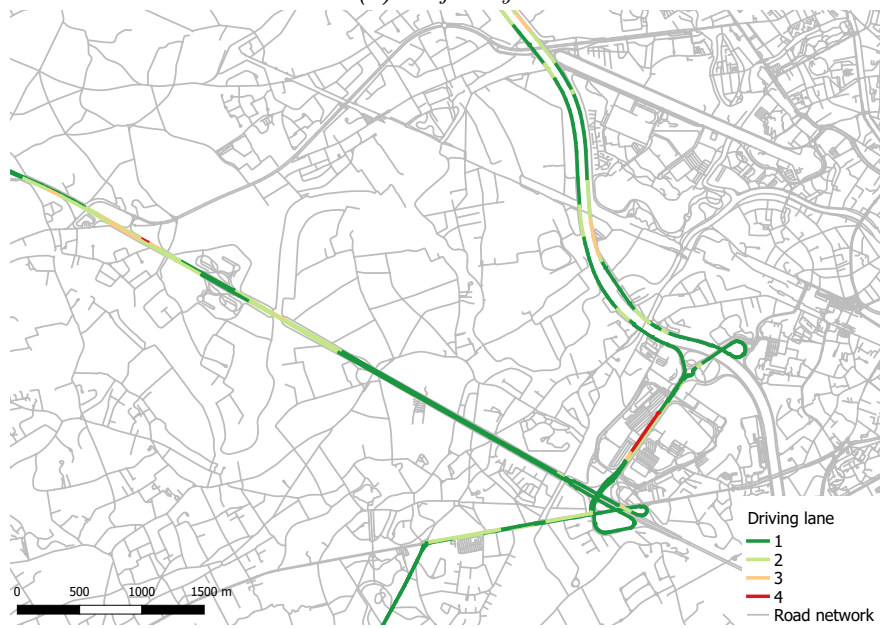


(b) Trajectory 2

Figure 5.7: Sample of raw and low-pass filtered sensor values for the lateral axis of the accelerometer with indication of left and right lane changes for both trajectories.



(a) Trajectory 1



(b) Trajectory 2

Figure 5.8: Sample of the driving lane at each road segment for both trajectories.

Table 5.1 summarizes the mean, standard deviation, 75th percentile value, minimum, and maximum values of the positive peaks (peak_{pos}), negative peaks (peak_{neg}), peak-to-peak values, and time between two peaks associated with a lane change (Δt), for both trajectories.

Trajectory	LC	Type	μ	σ	75th	Min	Max
1	86	peak_{pos} [g]	0.10	0.03	0.12	0.03	0.20
		peak_{neg} [g]	0.10	0.03	0.12	0.02	0.15
		peak-to-peak [g]	0.20	0.04	0.21	0.11	0.35
		Δt [s]	2.14	0.43	2.37	1.27	3.30
2	67	peak_{pos}	0.06	0.02	0.07	0.02	0.10
		peak_{neg}	0.07	0.02	0.08	0.03	0.12
		peak-to-peak	0.12	0.03	0.15	0.08	0.22
		Δt [s]	2.93	0.87	3.21	1.48	4.87

Table 5.1: Lane change peak statistics based on lateral accelerometer data. LC: amount of lane change.

Note that the absolute values of the negative peaks are used to simplify the comparison. Obviously, the driving style has an influence on the lateral acceleration patterns. Higher peak-to-peak values indicate fiercer lane changes, which can also be deducted from the shorter lane change times Δt for the first trajectory. The lane changes in the second trajectory are milder and slower, i.e., an average peak-to-peak and Δt value of 0.12 g and 2.93 s vs. 0.20 g and 2.14 s for the first trajectory, making them harder to detect because the patterns are less distinct (Section 5.5.4.1). To cope with different driving styles, a lane change detection algorithm based on a variable and bounded threshold, is proposed.

First, the peaks are filtered based on the map matched data, i.e., all peaks that are detected while only one lane is available are discarded, as these peaks cannot be caused by a lane change but are due to, e.g., bad road conditions or driving maneuvers. These peaks are filtered again based on a lower (peak_{min}) and upper bound (peak_{max}) threshold, i.e., all peaks outside these boundaries are discarded as well. The remaining peaks are visited chronologically and two peaks (a positive and a negative peak for a left lane change and vice versa for a right lane change), are paired and marked as a left or right lane change if two conditions are met: their peak-to-peak value is above a threshold ($p2p_{\text{min}}$) and the time between both peaks (Δt) is below t_{max} .

The t_{max} value is fixed at 5 s, which is larger than the maximum duration of any lane change in the experimental validation over a total of 153 lane changes (LC in Table 5.1). The other threshold values (peak_{min} , peak_{max} ,

and $p2p_{min}$) are estimated per trajectory based on the low-pass filtered lateral accelerometer values on the road segments with at least two lanes. Only straight road segments where the car was driving at least 50 km/h are taken into account to remove the influence of left or right turns and traffic jams. Note that, these selected road segments are used to estimate the threshold values but afterwards all lateral accelerometer values on road segments with at least two lanes, are processed. The negative peaks that have at a neighboring positive peak within t_{max} are taken into account (and vice versa for the positive peaks). The minimum and maximum absolute value of these selected peaks are used as $peak_{min}$ and $peak_{max}$, and the minimum peak-to-peak value of these negative and positive peaks is taken as $p2p_{min}$.

5.4.2 Driving lane allocation

After the detection of lane changes based on the lateral accelerometer data, the driving lane can easily be updated by adding one to the current driving lane if it is a lane change to the left and subtracting one if it is to the right. Lane one is used for the slow lane and the far left lane, i.e., fast lane, depends on the number of available lanes and could be anywhere from 2 to 8. Note that in left-driving countries the fast lane is the far right lane. Therefore, in the remainder of this section a right-driving country is assumed to avoid confusion. Only the initial driving lane is unknown and although it seems plausible to start at lane one, this is not always the case because, e.g., after a highway ramp a driver can go immediately to the second or third lane without being noticed by the lane change detection algorithm. Another source of errors are the addition of lanes on the right, e.g., when two highways merge, the current driving lane changes while the car was going straight.

The first problem is solved by accounting for a penalty for each possible starting lane when a driver goes from a road segment with one lane to a road segment with multiple lanes, e.g., when passing a highway ramp or when going from a small local road to a secondary road with two lanes. The penalties are calculated based on all future lane changes until the next road segment with only one lane. Each time an impossible lane change occurs, i.e., a left lane change when driving on the outside lane or a right lane change when driving on the first lane, this penalty is increased by one. The starting lane with the lowest penalty is taken as most likely initial lane.

The second problem is solved by monitoring the number of available lanes when the road segment changes, e.g., if this number increases and a road is merged from the right side, the current driving lane is increased by one otherwise it is unaffected. Note that, for left-driving countries this is

the other way around.

5.5 Experimental validation

5.5.1 Trajectories

The experimental validation encompasses twelve trajectories on foot, by bike, and by car. Table 5.2 summarizes the total distance, duration, average speed, and number of GPS data points for all trajectories per mode of transportation.

MoT	#routes	Distance [km]	Duration [min]	Speed [km/h]	#GPS points
Walk	3	25	268	6	15700
Bike	3	25	73	20	4255
Car	6	251	222	68	13165
Total	12	301	563	32	33120

Table 5.2: GPS dataset details per mode of transportation (MoT).

The trajectories are recorded at a sample rate of 1 Hz with a GPS logging application on a smartphone. The smartphone was put in the dashboard holder for the car rides, which were done by four different drivers, and carried in the pocket for the trajectories on foot and by bike. The total dataset is 9.4 hours, covers 301 km, consists of 33120 geographic coordinates (GPS points). The trajectories pass through rural and urban areas, on primary and secondary roads, sidewalks, bicycle paths, and highways. Note that the environment has a strong influence on the performance of a map matching algorithm, e.g., in rural areas the road network is usually sparser, which reduces the chance to select a wrong road segment and there are less tall buildings that can cause additional noise on the GPS signals.

Accelerometer data is available for two car rides on the highway with a lot of forced lane changes to have a sufficient amount of data to validate the lane detection, i.e., around four and two lane changes per minute (Analog Devices ADXL345 [43]). Note that these are the two trajectories from Section 5.4.1 (Figures 5.7 and 5.8). The driving lane ground truth is manually annotated based on video recordings from a dashcam. A script was written to facilitate this process, i.e., a video can be watched at an adjustable speed and is annotated by pushing the left or right button when the car makes a left or right lane change. The driving lane details, i.e., the number of lane changes, the average lane changes per minute (where lane changes are possible), total duration, time spent in each driving lane, and accelerometer samples, are summarized in Table 5.3 for the two car rides.

Trajectory	LC	LCPM	Duration [min]	Time per lane [min]					#samples
				1-av	1	2	3	4	
1	86	4.0	24.6	3.3	5.9	8.1	6.7	0.6	143229
2	67	2.1	43.4	11.3	18.5	10.3	3.0	0.2	258606

Table 5.3: Driving lane details per trajectory based on dashcam video recordings. *LC*: amount of lane changes, *LCPM*: average number of lane changes per minute, *1-av*: only one lane available, and number of accelerometer samples.

5.5.2 Performance metrics

The quality of the map matching and lane detection algorithm are each validated with two performance metrics. The map matching algorithm is evaluated with the F_1 score (also F-score or F-measure) and the average error (Euclidean distance) between the ground truth and estimated location. The F_1 score is the harmonic average of the precision and recall, where an F_1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. Broadly speaking, the precision, also called positive predictive value, is the fraction of relevant instances among the retrieved instances and the recall, also known as sensitivity, is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Precision can be seen as a measure of exactness or quality, whereas recall is a measure of completeness or quantity.

$$\text{precision} = \frac{tp}{tp + fp} \quad (5.4)$$

$$\text{recall} = \frac{tp}{tp + fn} \quad (5.5)$$

tp are the true positives or hits, fp are the false positives or false alarm (Type I error: asserting something that is absent), fn are the false negatives or misses (Type II error: failing to assert what is present), and tn are the true negatives or correct rejections but these are abundant to calculate the precision or recall.

In the context of map matching, this results in following formulas:

$$F_1^{mm} = 2 \cdot \frac{\text{precision}_{mm} \cdot \text{recall}_{mm}}{\text{precision}_{mm} + \text{recall}_{mm}} \quad (5.6)$$

$$\text{precision}_{mm} = \frac{L_{corr}}{L_{mm}} \quad (5.7)$$

$$\text{recall}_{mm} = \frac{L_{corr}}{L_{gt}} \quad (5.8)$$

The mm in F_1^{mm} , $precision_{mm}$, and $recall_{mm}$ refers to *map matching* so that it can be distinguished from the recall and precision definition for the lane detection. L_{gt} is the length of the ground truth trajectory, L_{mm} is the length of the map matched trajectory, and L_{corr} is the length of the (correct) overlapping segments between the map matched and ground truth trajectory. Most map matching algorithms use this metric to evaluate the approach because it suffices for a broad range of applications, e.g., road usage patterns or urban traffic modeling.

The ground truth trajectories, i.e., continuous sequence of road segments and individual locations on these segments, are constructed based on the shortest path between manually indicated points where the route must pass at certain timestamps. Note that the trajectories were known beforehand and indication points are added as long as the constructed paths were not completely correct. Figure 5.9 shows the raw GPS input data with a sample rate of 1 Hz and the constructed ground truth segments and locations.

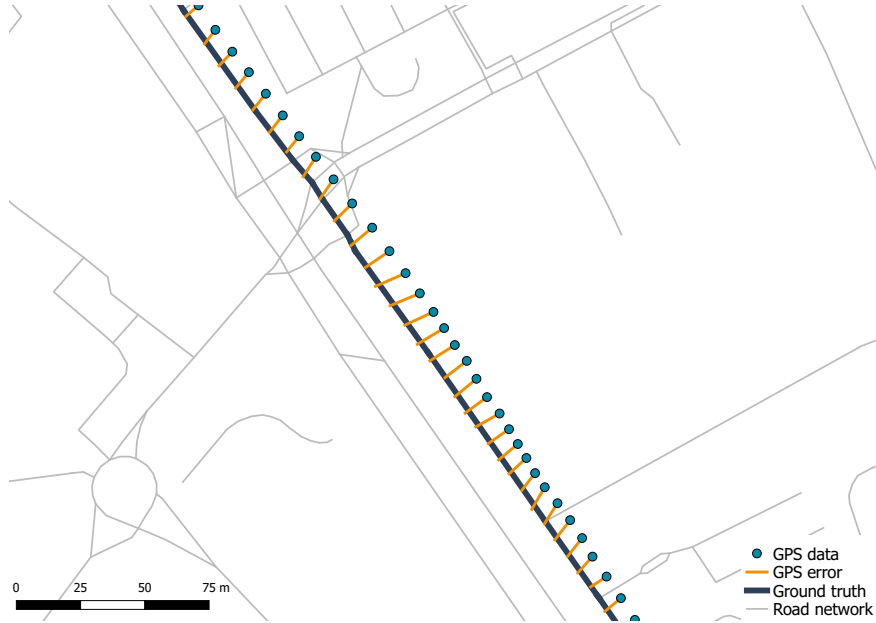


Figure 5.9: Raw GPS input data with ground truth construction.

The individual timestamped locations are used to calculate the average error of the map matched trajectories. This metric is generally ignored but is important for applications that map events or other sensor measurements to an exact location.

The performance of the lane detection is evaluated with the F_1 score

based on the precision and recall in the classification context.

$$F_1^{ld} = 2 \cdot \frac{\text{precision}_{ld} \cdot \text{recall}_{ld}}{\text{precision}_{ld} + \text{recall}_{ld}} \quad (5.9)$$

$$\text{precision}_{ld} = \frac{tp_{ld}}{tp_{ld} + fp_{ld}} \quad (5.10)$$

$$\text{recall}_{ld} = \frac{tp_{ld}}{tp_{ld} + fn_{ld}} \quad (5.11)$$

The *ld* subscript refers to *lane detection* so that it can be distinguished from the F_1 score, recall and precision definition for the map matching. tp_{ld} are the true positives, i.e., a left or right lane change is correctly identified. fp_{ld} are the false positives, e.g., a driver stays in his lane but the algorithm detects a lane change due to a bump or maneuver. fn_{ld} are the false negatives, i.e., failing to identify a left or right lane change.

The second performance metric for the lane detection is the amount of time the correct lane is estimated based on the detected lane changes. Incorrectly predicted driving lanes are further divided into 1-off and 2-off when the predicted lane is one or two lanes off, i.e., the absolute difference between the ground truth and detected lane (three lanes off did not occur in our experimental validation).

5.5.3 Map matching accuracy

The accuracy of the map matching is evaluated as a function of the sample interval, GPS noise, and three algorithm parameters: update distance, direction change, and code change penalty.

5.5.3.1 GPS sample interval

Lowering the GPS sample rate saves battery power and reduces the communication cost by limiting the bandwidth usage but increases the computational cost per location update as a larger area needs to be considered. Note that, since GPS devices need a lock on the available satellites it is not possible to turn the device completely off between samples because the time to first fix (TTFF) with a cold or warm start is too high [44]. However, a rapid acquisition of satellite signals is enabled in standby (hot) mode because the receiver already has valid time, position, almanac (approximate information on all the other satellites), and ephemeris data (detailed orbital information). The influence of the GPS sample interval is simulated by downsampling the input data, i.e., discarding GPS samples to acquire the intended sample rate, and each simulation was repeated ten times for

averaging. Figure 5.10 shows the F_1^{mm} score and the median error between the estimated and ground truth locations, averaged over all simulations and trajectories per mode of transportation, as a function of the sample interval.

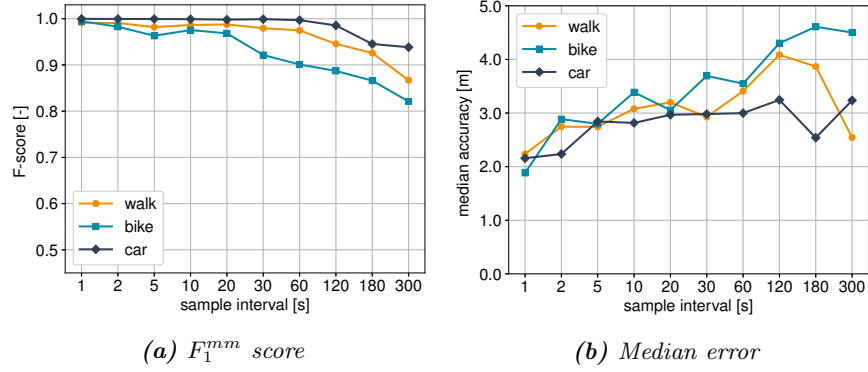


Figure 5.10: F_1^{mm} score and median error as a function of the sample interval per mode of transportation.

As expected, decreasing the sample rate results in lower F_1^{mm} scores and a larger median error but the performance remains accurate, e.g., if only once every 5 min a GPS sample is available the F_1^{mm} scores and median errors are still 86.7 % and 2.54 m, 82.1 % and 4.5 m, and 93.8 % and 3.24 m, for the walks, bike rides, and car routes, respectively. Increasing the sample rate to once every minute, which can still be considered as a power saving mode, improves the average F_1^{mm} score to 97.5%, 90.1%, and 99.7%, for the trajectories on foot, by bike, and by car, respectively. Although the median errors are similar for the three considered modes of transportation, the F_1^{mm} score is higher for the car rides because these have more restrictions on the road segments than a walk and bike ride, e.g., the reconstruction of a person on foot has access to all type of roads (except highways) in both ways because one-way streets are usually not applicable to pedestrians and a separate sidewalk is not always included in the open street map data. A typical error for the walks and bike rides is to select a parallel road that is very close to the correct road, which has a small impact on the location error but decreases the F_1^{mm} score because the trajectories do not overlap. Note that the median error is based on the (downsampled) geographic input coordinates, e.g., once every 5 minute, which does not reveal much about the exact location accuracy of the points along the reconstructed trajectory (based on the timestamps). Figure 5.11 shows the same plot but with interpolation, e.g., for a sample interval of 5 min and interpolation at 1 Hz, the algorithm outputs all road segments and 300 estimated locations with

every update, i.e., once every second instead of only one estimated location and the road segments to get there.

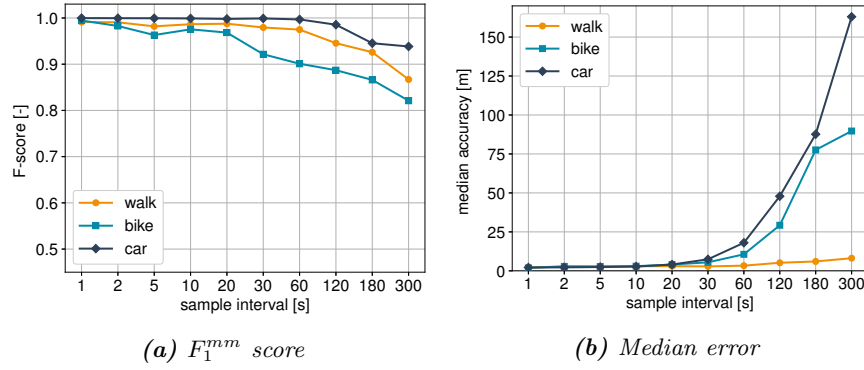


Figure 5.11: F_1^{mm} score and median error as a function of the sample interval per mode of transportation with interpolation at 1 Hz.

Naturally, the F_1^{mm} score is exactly equal and the median error is larger, e.g., with a sample interval of 1 min, the median errors of all estimated locations during this minute are 3.33 m, 10.59 m, and 18.01 m, for the trajectories on foot, by bike, and by car, respectively. Note that a car driving at 120 km/h travels a distance of 2 km during a one minute interval. The location error along the interpolated points starts to increase rather fast for the bike and car rides if the sample rate is further decreased because the traveling speed can be variable due to traffic lights, road speed limits, or oncoming traffic. This is detrimental for the interpolation, e.g., with a sample interval of 5 min there is no way of knowing if a car is standing still for a minute or just driving slow. The reason for the lower location error of the trajectories on foot is because a walking speed is more constant than a biking or driving speed due to accelerations and slowing down.

5.5.3.2 GPS noise

The noise in GPS measurements can be modeled as zero-mean Gaussian [45]. To assess the influence of noise on the proposed map matching algorithm, geographic input data is simulated by taking the ground truth locations from Section 5.5.1 and adding Gaussian noise with a varying standard deviation. In the remainder of this section, the standard deviation of the added Gaussian noise is referred to as *noise level*. Figure 5.12 shows the F_1^{mm} score and the median error between the estimated and ground truth locations, as a function of the added noise, for two sample intervals T (1 s and 1 min).

Obviously, increasing the noise level results in lower F_1^{mm} scores and

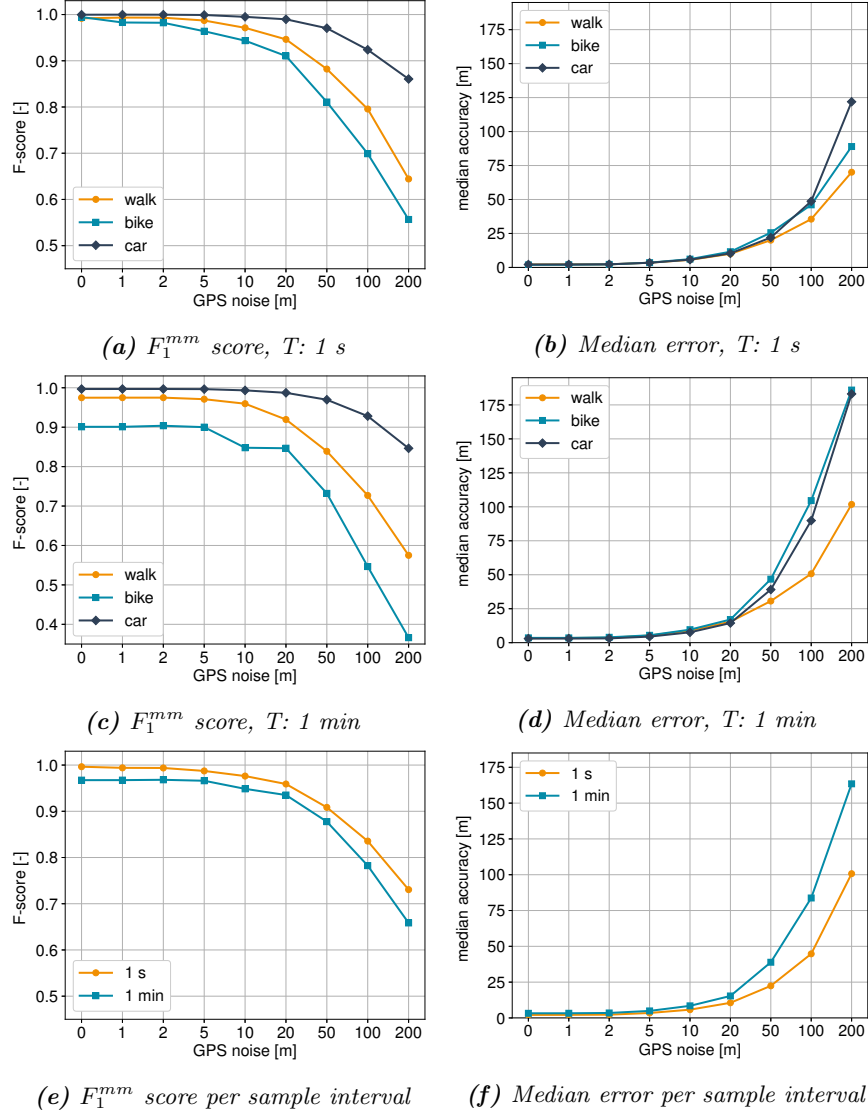


Figure 5.12: F_1^{mm} score and median error as a function of the GPS noise per mode of transportation for a sample interval T of 1 s (a–b) and 1 min (c–d). Overall F_1^{mm} score and median error as a function of the GPS noise per sample interval T (e–f)

a larger median error. The F_1^{mm} scores and median errors, for a sample interval of 1 s and noise levels of 2 m and 20 m, start at 99.3% and 2.26 m, 98.2% and 2.22 m, and 100.0% and 2.33 m, and drop to 94.6% and 10.0 m, 91.0% and 11.57 m, and 99.0% and 10.34 m, for the walks, bike rides, and car routes, respectively.

Increasing the sample interval from once every second to once every minute degrades the performance but has a limited effect on the median error for low noise levels (Figure 5.12f). The F_1^{mm} score is shifted by an average value of 3.5% for all noise levels. The average F_1^{mm} scores and median errors for a noise level of 10 m start at 95.9% and 10.56 m for a sample interval of 1 s, and drop to 93.5% and 15.34 m for a sample interval of 1 min, although the amount of available data is reduced by a factor of 60.

5.5.3.3 Sensitivity analysis

This section discusses the influence of four parameters of the proposed map matching technique (Algorithm 2): update distance, weight of the distance metric (α), weight of the direction change penalty (β), and weight of the code change penalty (γ). The update distance is used to establish a smoothing effect in the map matching by making a selection of geographic input coordinates (Section 5.3.2). The α parameter is used to assign more weight to the geographic input coordinates and the β and γ parameters are used to discourage deviations from the current road or driving direction. Figure 5.13 shows the F_1^{mm} score and the median error between the estimated and ground truth locations, as a function of these four parameters.

An update distance of 0 m, i.e., every geographic coordinate is used as input, results in an average F_1^{mm} score and the median error of 95.1% and 3.41 m. Increasing the update distance to 50 m results in an average F_1^{mm} score and the median error of 99.5% and 2.07 m. A distance weight of 0 gives poor results because the geographic data is not taken into account. Best performance is obtained with a distance weight of 1 and increasing this weight slowly deteriorates the F_1^{mm} score (Figure 5.13c). Including the direction change penalty slightly improves the accuracy for weights up to 20 per 90°, i.e., the F_1^{mm} score improves from 96.9% to 98.7% (Figure 5.13e). Including the code change penalty slightly improves the accuracy as well for weights up to 20, i.e., the F_1^{mm} score improves from 96.9% to 99.3% (Figure 5.13g). The best global accuracy, averaged over all 12 trajectories on foot, by bike, or by car, is obtained with an update distance of 50 m, α of 1 per meter, β of 10 per 90°, and γ of 10 per code change. Note that, γ is set to zero to evaluate the influence of the direction change penalty and β is set to zero to evaluate the influence of the code change penalty.

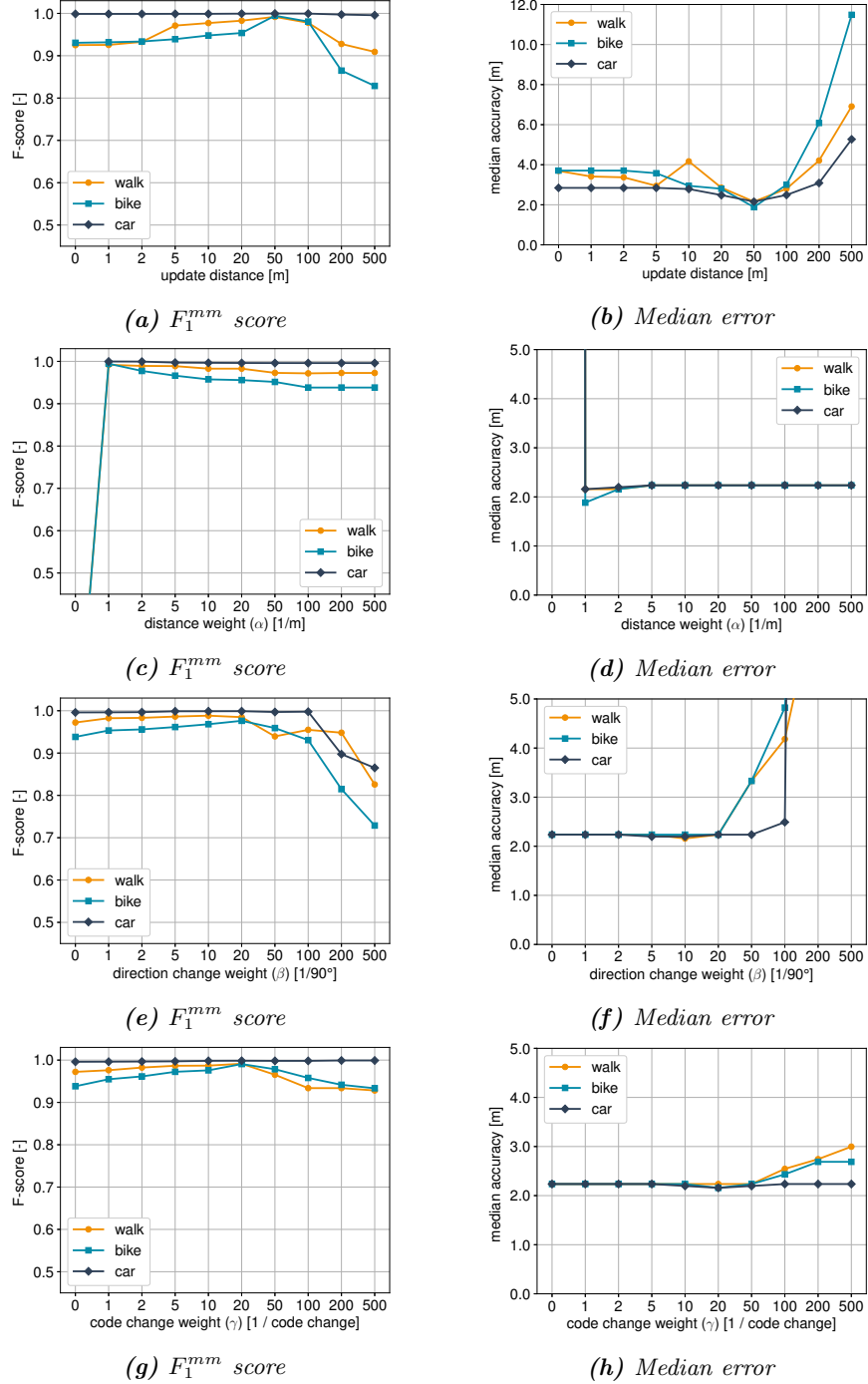


Figure 5.13: F_1^{mm} score and median error as a function of the update distance (a–b), distance weight α (c–d), direction change weight β (e–f), and code change weight γ (g–h).

Furthermore, the considered parameter values were on a logarithmic scale to avoid overfitting on the dataset of this experimental validation.

5.5.4 Lane detection accuracy

The lane detection is evaluated by the accuracy of the lane change detection algorithm, the driving lane estimation, and the influence of the sample rate of the accelerometer.

5.5.4.1 Lane changes

The precision, recall, and F_1^{ld} score of the lane change detection algorithm is summarized in Table 5.4 for both trajectories.

Trajectory	Precision [%]	Recall [%]	F_1^{ld} score [%]	tp [-]	fp [-]	fn [-]
1	100.0	96.5	98.2	83	0	3
2	85.0	67.2	75.1	45	8	22

Table 5.4: Precision, recall, and F_1^{ld} score for the lane change detection. tp: true positives, fp: false positives, fn: false negatives.

For the first trajectory, the proposed algorithm correctly labels 83 lane changes, detects no false positives (precision 100.0%) and fails to detect only three lane changes (recall 96.5%). This results in an F_1^{ld} score of 98.2%. For the second trajectory, the proposed algorithm correctly labels 45 lane changes, detects 8 false positives (precision 85.0%) and fails to detect 22 lane changes (recall 67.2%). This results in an F_1^{ld} score of 75.1%. The difference in performance is due to different driving styles and making the detection harder by shifting lanes while turning, carrying out small maneuvers, or changing lanes in a slow manner.

5.5.4.2 Driving lane

The amount of time the lane detection algorithm estimates the correct and wrong lane are summarized in Table 5.5 for both trajectories. These amounts of time are expressed as a percentage with respect to the total duration of a trajectory.

The driving lane is estimated correctly 99.2% and 86.8% of the time for the first and second trajectory, which is a logical consequence of the lane change detection performance. Note that there was only one lane available during 14.8% and 21.6% of the time, which automatically results in the correct lane because the lane change detection and driving lane allocation are disabled when the map matching algorithm estimates a road segment

Trajectory	Correct lane [%]			Wrong lane [%]		
	1-av	match	total	1-off	2-off	total
1	14.8	84.4	99.2	0.7	0.1	0.8
2	21.6	65.2	86.8	12.3	0.9	13.2

Table 5.5: Accuracy of the lane detection algorithm as a percentage of the complete trajectory. 1-av indicates that there is only one lane available, and 1-off or 2-off that the estimated lane is one or two lanes next to the correct lane.

with only one lane. Furthermore, a wrongly estimated lane is nearly always only one lane off, two lanes off occurs only in 0.1% and 0.9% of the time, and three lanes off does not occur. This is better than the IMU-based lane detection technique that is proposed in [33], where an average accuracy of 80% is achieved.

5.5.4.3 Accelerometer sample rate

The sample rate of the accelerometer is simulated by discarding samples in the input data. Similarly to the GPS sample interval (Section 5.5.3.1), lowering the sample rate of the accelerometer saves battery power and reduces the communication cost by limiting the bandwidth usage, e.g., if the data is collected or processed centrally. Unlike with the GPS data, the computational cost decreases because less samples need to be examined during the peak detection. However, this can affect the accuracy of the lane change detection algorithm and hence driving lane estimation. Figure 5.14 shows the precision, recall, and F_1^{ld} score as a function of the accelerometer sample rate averaged over both trajectories.

Lowering the sample rate results in a negligible performance loss up to 5 Hz, i.e., a reduction by a factor of 20. Increasing the sample interval to 5 s results in a recall near zero because all lane changes happen within 5 s. The lane changes that are still detected are because the lateral accelerometer values at the selected time instances are still above or below the thresholds values from Section 5.4.1.

5.6 Conclusion

In this chapter, a fast, memory-efficient, and worldwide map matching algorithm with support for trajectories on foot, by bike, and motorized vehicles, is presented. The input for the map matching algorithm are raw geographic coordinates and enriched open map data. The proposed algorithm combines the Markovian behavior and the shortest path aspect while taking

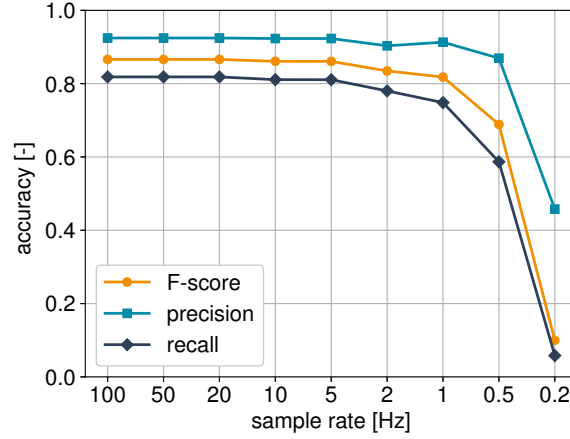


Figure 5.14: Precision, recall, and F_1^{ld} score as a function of the accelerometer sample rate over both trajectories.

into account the type and direction of all road segments, information about one-way traffic, maximum allowed speed per road segment, and driving behavior. Furthermore, a lane detection algorithm based on accelerometer readings and traffic lane information from the open map data, that self-adapts to different driving behaviors, is added on top of the map matching algorithm. An experimental validation consisting of 12 trajectories on foot, by bike, and by car, showed the efficiency and accuracy of the proposed algorithms. The total dataset is 9.4 hours, covers 301 km, consists of 33k GPS points, and 402k accelerometer samples. The average F1-scores and median errors of the map matching algorithm, if all GPS samples are used, were 99.1 % and 2.24 m, 99.5 % and 1.88 m, and 100.0% and 2.16 m, for the walks, bike rides, and car routes, respectively. The performance remained adequate if the input data was downsampled to only one sample every 5 min, i.e., the average F1-scores were 86.7 %, 82.1 %, and 93.8 %, for the walks, bike rides, and car routes, respectively. Two trajectories with accelerometer data were used to evaluate the lane detection algorithm with F1-scores of 98.2% and 75.1% for the lane change detection, which resulted in the correctly estimated lane 99.2% and 86.8% of the time. In Chapter 6, an outdoor location tracking algorithm based on cellular network data that uses a modified version of the map matching algorithm with support for real-time positioning, is proposed.

References

- [1] Britta Hummel. *Map matching for vehicle guidance*. In Dynamic and Mobile GIS, pages 196–207. CRC Press, 2006.
- [2] Hone-Jay Chu, Guang-Je Tsai, Kai-Wei Chiang, and Thanh-Trung Duong. *GPS/MEMS INS data fusion and map matching in urban areas*. Sensors, 13(9):11280–11288, 2013.
- [3] David Bernstein, Alain Kornhauser, et al. *An introduction to map matching for personal navigation assistants*. 1996.
- [4] David C Andrade, Felipe Bueno, Felipe R Franco, Rodrigo Adamshuk Silva, João Henrique Z Neme, Erick Margraf, William T Omoto, Felipe A Farinelli, Angelo M Tusset, Sergio Okida, et al. *A Novel Strategy for Road Lane Detection and Tracking Based on a Vehicle’s Forward Monocular Camera*. IEEE Transactions on Intelligent Transportation Systems, (99):1–11, 2018.
- [5] Wenjie Song, Yi Yang, Mengyin Fu, Yujun Li, and Meiling Wang. *Lane detection and classification for forward collision warning system based on stereo vision*. IEEE Sensors Journal, 18(12):5151–5163, 2018.
- [6] Luca Caltagirone, Samuel Scheidegger, Lennart Svensson, and Matthias Wahde. *Fast LIDAR-based road detection using fully convolutional neural networks*. In 2017 IEEE intelligent vehicles symposium (iv), pages 1019–1024. IEEE, 2017.
- [7] Mohamed Maher Atia, Allaa R Hilal, Clive Stellings, Eric Hartwell, Jason Toonstra, William B Miners, and Otman A Basir. *A low-cost lane-determination system using GNSS/IMU fusion and HMM-based multistage map matching*. IEEE Transactions on Intelligent Transportation Systems, 18(11):3027–3037, 2017.
- [8] Jiadi Yu, Zhongyang Chen, Yanmin Zhu, Yingying Jennifer Chen, Linghe Kong, and Minglu Li. *Fine-grained abnormal driving behaviors detection and identification with smartphones*. IEEE transactions on mobile computing, 16(8):2198–2212, 2016.
- [9] Haluk Eren, Semiha Makinist, Erhan Akin, and Alper Yilmaz. *Estimating driving behavior by a smartphone*. In 2012 IEEE Intelligent Vehicles Symposium, pages 234–239. IEEE, 2012.
- [10] Christopher E White, David Bernstein, and Alain L Kornhauser. *Some map matching algorithms for personal navigation assistants*. Transportation research part c: emerging technologies, 8(1-6):91–108, 2000.

- [11] Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. *On map-matching vehicle tracking data*. In Proceedings of the 31st international conference on Very large data bases, pages 853–864. VLDB Endowment, 2005.
- [12] Paul Newson and John Krumm. *Hidden Markov map matching through noise and sparseness*. In Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems, pages 336–343. ACM, 2009.
- [13] George R Jagadeesh and Thambipillai Srikanthan. *Online map-matching of noisy and sparse location data with hidden Markov and route choice models*. IEEE Transactions on Intelligent Transportation Systems, 18(9):2423–2434, 2017.
- [14] John Lafferty, Andrew McCallum, and Fernando CN Pereira. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. 2001.
- [15] Jian Yang and Liqiu Meng. *Feature selection in conditional random fields for map matching of GPS trajectories*. In Progress in Location-Based Services 2014, pages 121–135. Springer, 2015.
- [16] Yang Li, Qixing Huang, Michael Kerber, Lin Zhang, and Leonidas Guibas. *Large-scale joint map matching of GPS traces*. In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 214–223. ACM, 2013.
- [17] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. *Understanding individual human mobility patterns*. nature, 453(7196):779, 2008.
- [18] Carlos Alberto V Campos, Daniel C Otero, and Luis Felipe M de Moraes. *Realistic individual mobility markovian models for mobile ad hoc networks*. In 2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No. 04TH8733), volume 4, pages 1980–1985. IEEE, 2004.
- [19] Mudhakar Srivatsa, Raghu Ganti, Jingjing Wang, and Vinay Kolar. *Map matching: Facts and myths*. In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 484–487. ACM, 2013.
- [20] Ralf Metzler. *Generalized Chapman-Kolmogorov equation: A unifying approach to the description of anomalous transport in external fields*. Physical Review E, 62(5):6233, 2000.

- [21] Johan Wahlström, Isaac Skog, and Peter Händel. *Smartphone-based vehicle telematics: A ten-year anniversary*. IEEE Transactions on Intelligent Transportation Systems, 18(10):2802–2825, 2017.
- [22] Artis Mednis, Girts Strazdins, Reinholds Zviedris, Georgijs Kanonirs, and Leo Selavo. *Real time pothole detection using android smartphones with accelerometers*. In 2011 International conference on distributed computing in sensor systems and workshops (DCOSS), pages 1–6. IEEE, 2011.
- [23] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. *Towards end-to-end lane detection: an instance segmentation approach*. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 286–291. IEEE, 2018.
- [24] Jun Li, Xue Mei, Danil Prokhorov, and Dacheng Tao. *Deep neural network for structural prediction and lane detection in traffic scene*. IEEE transactions on neural networks and learning systems, 28(3):690–703, 2016.
- [25] Qingquan Li, Long Chen, Ming Li, Shih-Lung Shaw, and Andreas Nüchter. *A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios*. IEEE Transactions on Vehicular Technology, 63(2):540–555, 2013.
- [26] Kamarul Ghazali, Rui Xiao, and Jie Ma. *Road lane detection using H-maxima and improved hough transform*. In 2012 Fourth International Conference on Computational Intelligence, Modelling and Simulation, pages 205–208. IEEE, 2012.
- [27] Xianwen Wei, Zhaojin Zhang, Zongjun Chai, and Wei Feng. *Research on Lane Detection and Tracking Algorithm Based on Improved Hough Transform*. In 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), pages 275–279. IEEE, 2018.
- [28] Wei Liu, Hongliang Zhang, Bobo Duan, Huai Yuan, and Hong Zhao. *Vision-based real-time lane marking detection and tracking*. In 2008 11th International IEEE Conference on Intelligent Transportation Systems, pages 49–54. IEEE, 2008.
- [29] Marcos Nieto, Andoni Cortés, Oihana Otaegui, Jon Arróspide, and Luis Salgado. *Real-time lane tracking using Rao-Blackwellized particle filter*. Journal of Real-Time Image Processing, 11(1):179–191, 2016.

- [30] Hsu-Yung Cheng, Bor-Shenn Jeng, Pei-Ting Tseng, and Kuo-Chin Fan. *Lane detection with moving vehicles in the traffic scenes*. IEEE Transactions on intelligent transportation systems, 7(4):571–582, 2006.
- [31] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. *Recent progress in road and lane detection: a survey*. Machine vision and applications, 25(3):727–745, 2014.
- [32] Jongin Son, Hunjae Yoo, Sanghoon Kim, and Kwanghoon Sohn. *Real-time illumination invariant lane detection for lane departure warning system*. Expert Systems with Applications, 42(4):1816–1824, 2015.
- [33] Heba Aly, Anas Basalamah, and Moustafa Youssef. *Lanequest: An accurate and energy-efficient lane detection system*. In 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom), pages 163–171. IEEE, 2015.
- [34] Yanlei Gu, Li-Ta Hsu, and Shunsuke Kamijo. *GNSS/onboard inertial sensor integration with the aid of 3-D building map for lane-level vehicle self-localization in urban canyon*. IEEE Transactions on Vehicular Technology, 65(6):4274–4287, 2015.
- [35] Yurong Jiang, Hang Qiu, Matthew McCartney, Gaurav Sukhatme, Marco Gruteser, Fan Bai, Donald Grimm, and Ramesh Govindan. *Carloc: Precise positioning of automobiles*. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, pages 253–265. ACM, 2015.
- [36] *Winkel Tripel Projections*. <http://www.winkel.org/other/Winkel%20Tripel%20Projections.htm>. Accessed: 2018-09-01.
- [37] John P Snyder. *Flattening the earth: two thousand years of map projections*. University of Chicago Press, 1997.
- [38] Bojan Vrsavcic, Bernhard Jenny, Denis White, and Daniel R Strebe. *User preferences for world map projections*. Cartography and Geographic Information Science, 42(5):398–409, 2015.
- [39] *OpenStreetMap Data Extracts*. <http://download.geofabrik.de/>. Accessed: 2019-06-01.
- [40] *Map Matching Tool*. <https://www.waves.intec.ugent.be/exposure-tool/map-matching-tool>. Accessed: 2018-10-22.

- [41] Jens Trogh, David Plets, Luc Martens, and Wout Joseph. *Advanced Real-Time Indoor Tracking Based on the Viterbi Algorithm and Semantic Data*. International Journal of Distributed Sensor Networks, 501:271818, 2015.
- [42] Jens Trogh, David Plets, Arno Thielens, Luc Martens, and Wout Joseph. *Enhanced indoor location tracking through body shadowing compensation*. IEEE Sensors Journal, 16(7):2105–2114, 2016.
- [43] Analog Devices ADXL345. <https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>.
- [44] Goran M Djuknic and Robert E Richton. *Geolocation and assisted GPS*. Computer, 34(2):123–125, 2001.
- [45] Frank Van Diggelen. *SYSTEM DESIGN & TEST-GNSS Accuracy-Lies, Damn Lies, and Statistics-This update to a seminal article first published here in 1998 explains how statistical methods can create many different*. GPS world, 18(1):26–33, 2007.

Chapter 6

Outdoor location tracking based on cellular data

6.1 Introduction

In this chapter a technique and large-scale experimental validation for anonymous outdoor location tracking of all mobile users residing on a cellular network, is presented. Network-based positioning algorithms locate a mobile user based on measured radio signals from base stations in its vicinity. The growing amount of available cellular data has led to many location-based services (LBS) for outdoor environments. These LBS represent an additional stream of revenue for mobile network operators through targeted advertising or improving the customers experience [1]. However, the available cellular data is also useful for governments, e.g., to analyze traffic patterns or to estimate population movements during disasters or outbreaks. The latter requires timely and accurate location data that large-scale surveys cannot provide, whereas mobile network operators manage data that can potentially be used to provide location data in real-time [2].

The proposed positioning algorithm: AMT (antenna, map, and timing information based tracking) is able to accurately locate all mobile users in a cellular network without any required modifications at the mobile side (client) or network side (server). The latter is useful for applications where there is typically no cooperation at the mobile side, e.g., traffic monitoring, population movement estimation, or criminal activity detection and makes it immediately applicable for mobile network operators. This location tracking algorithm exploits enriched open map data [3], a mode of transportation estimator, and advanced route filtering based on the map matching algorithm of Chapter 5, on top of the cellular network topology

and measurements to track the movement and locations of mobile devices. Furthermore, it does not depend on additional or custom software, forced messages, dedicated infrastructures, direct communication between mobile users, or prior training data via, e.g., offline fingerprinting, drive-testing, or crowd-sourced measurement campaigns.

Most tracking algorithms for cellular networks are validated in optimal or controlled environments on a small dataset or are merely validated by simulations. In this work, nationwide validation data consisting of millions of parallel location estimations from over a million users are collected and processed in real-time, in cooperation with a major network operator in Belgium. The experimental validation includes trajectories on foot, by bike, and by car, in urban and rural environments while a person was actively using his or her smartphone, but also in standby mode. In this mode, all applications that use the mobile network are blocked, e.g., email and messaging services, and as such, standby mode represents a worst-case scenario in terms of number of location updates. The latter shows measurement gaps of up to 6 min while a user was on the move, i.e., time periods where no measurement data is available, which mainly occur in rural areas. Current existing location tracking algorithms for cellular networks are not able to cope with large measurement gaps but instead are deployed in optimal or controlled environments with a high base station density, regularly available measurement updates, large training sets, or are merely validated by simulations with a fixed location update rate.

The remainder of this chapter is structured as follows, Section 6.2 describes related work and Section 6.3 outlines the mobile network and grid configuration, type of measurements, and trajectories for the experimental validation. Section 6.5 discusses the proposed location tracking algorithm in detail and Section 6.6 presents the results. Finally, in Section 6.7, conclusions are provided.

6.2 Related work

Network-based location tracking poses several problems due to multipath and non line-of-sight (NLoS) conditions, small-scale and large-scale fading, low signal-to-noise ratios, and interference by other mobile users [4]. These affect the radio signal parameters used as input data to location tracking algorithms. To process the noisy signal parameters and improve the accuracy, location tracking algorithms use additional intelligence and information. NLoS mitigation techniques use more robust estimators or simply discard the NLoS component [5]. Map-based algorithms use information about the environment to limit possible locations and transitions between two location

updates, this can be done in combination with Kalman filters [6], particle filters [7], hidden Markov models (HMM) [8, 9], data fusion [10], or a least squares estimator [11].

In [9], a Received Signal Strength Indication (RSSI) fingerprinting technique with GSM signals and route constraints is used to locate a moving vehicle. The approach uses road segments as the states for an HMM, that is trained based on the statistics of the average driver's behavior on the road and the probabilistic distribution of the RSSI vectors observed in each road segment, consequently it is not applicable for all mobile users in the mobile network. Besides that it is aimed at outdoor localization, the difference with this work is that measurement data is needed to train an HMM. Furthermore, possible locations for the mobile user are limited to places where a car can pass, meaning no indoor, pedestrian, or off-road locations can be estimated. A database correlation technique over RSSI data that is based on advanced map- and mobility-based filtering is presented in [12]. The algorithm is validated in a field environment with trips by car, a location update rate forced to 2 Hz, and an electromagnetic field simulator.

A cooperative positioning technique for cellular systems using RF pattern matching is presented in [13]. It is shown in simulations that leveraging the Device-to-Device (D2D) communications protocol can improve positioning performance if insufficient base stations are visible to a user entity. A crowd-sourced measurement campaign to develop radio-frequency (RF) coverage maps and a similarity based location algorithm is presented in [14]. A proprietary application, installed on the smartphone of a sample set of users in the network, periodically reports the RF channel measurement along with the GPS tag to a central server, which are then processed into the RF coverage map. This resulted in accuracies up to 50 m and 300 m, depending on the cell's coverage range. A semi- and unsupervised learning technique that minimizes the effort to label signal strength measurements for the network-side cellular positioning problem is presented in [15]. This technique uses Gaussian mixture models to model the signal strength vectors and an expectation maximization approach to learn the distributions. Accuracies up to 30 m are reported as long as enough training data is available and the base station density is high. A machine learning technique for indoor-outdoor classification and particle filter with HMM for cellular positioning is presented in [7]. The trajectory of a moving user was synthesized and reconstructed based on a data training set of around 129000 drive test data points and a fixed location update interval of 10 s, which led to accuracies up to 20 m in urban environments. Note that the latter accuracies are only achieved with large (crowd-sourced) training sets, synthesized data, and high location update rates, which the approach in this work does

not require. Furthermore, the proposed location tracking algorithm in this work is confirmed to execute in real-time for more than a million users in parallel and outperforms state-of-the-art particle filters [7].

6.3 Methodology

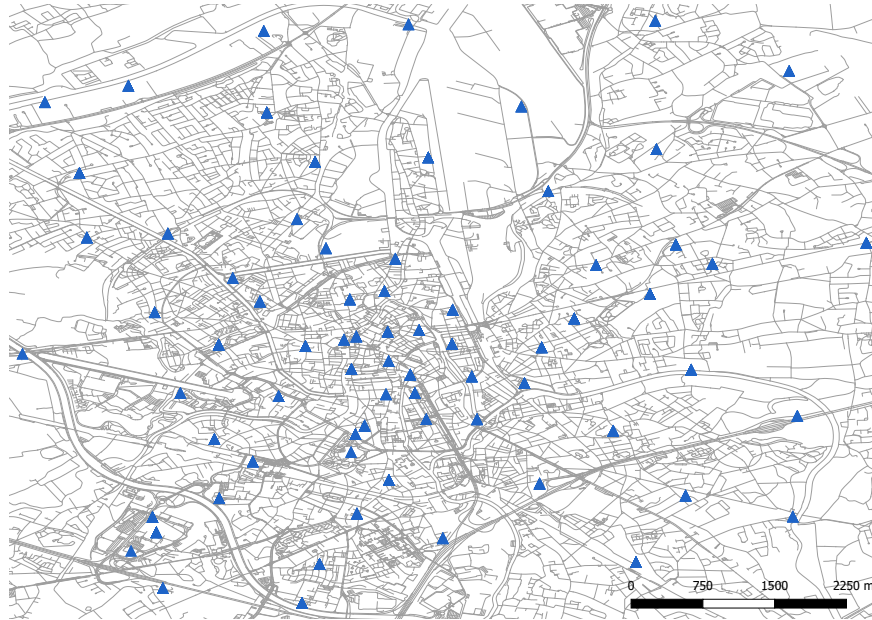
6.3.1 Cellular network

The cellular network or mobile network, that is used in the experimental validation, consists of more than 2500 NodeBs (September 2017), distributed over Belgium's territory (30528 km²). In a 3G network, the base stations are referred to as NodeBs. Figure 6.1 shows the NodeB locations in a representative urban and rural environment on the same scale (i.e., Ghent and Melsele). It is clear that the environment will have a major influence on the positioning accuracy, because of the difference in NodeB density on the one hand and in urban planning on the other: a sparser road network can limit plausible locations, and the type and height of buildings can affect the signal parameters used as input to location tracking algorithms (e.g., apartments vs. stand alone houses vs. office buildings). There are more than 50 NodeBs in an area of approximately 45 km² for the experiments in an urban environment whereas for the rural environment there are roughly 10 NodeBs in an area of the same size. The comparison and influence on the performance are discussed in Section 6.6.

A NodeB has multiple antennas with unique cell-IDs, oriented towards different directions (Figure 6.2). Antenna configurations with one up to six distinct orientations occur in the mobile network, that is used in the experimental validation, the most common ones are with three (92%), one (4%) and two (3%) different antenna directions. Usually, a mobile user will connect to the NodeB antenna that is directed towards him. Likewise, a user for which measurements are available from antennas with different orientations but from the same NodeB, has a large chance to be located between both zones. The aforementioned observations provide information that is exploited in the proposed location tracking algorithm (Section 6.5).

6.3.2 Grid

The grid represents a collection of points in the area of interest where a mobile user can be located. In a regular Cartesian grid, all elements are unit squares. It is a simplistic approach where all areas are equally important and take the same resources in both database size and processing time. Alternatively, a map-based grid can be used to limit possible points along the major (motorway, freeway, primary, secondary and tertiary) and



(a) Urban environment



(b) Rural environment

Figure 6.1: NodeB density in urban and rural environments (NodeBs are indicated by blue triangles).

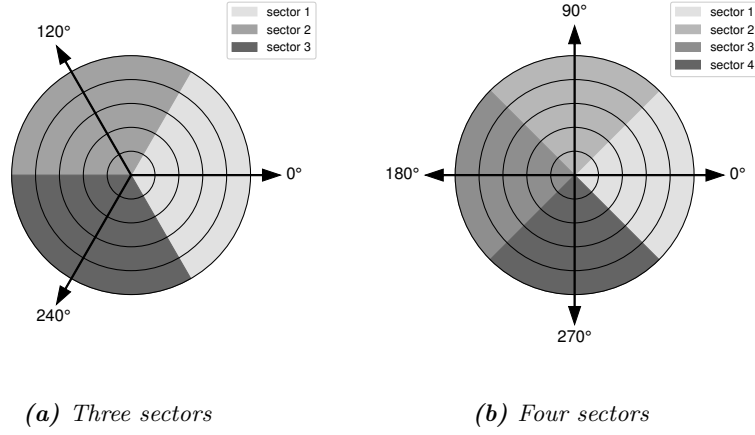


Figure 6.2: Antenna directions and configuration.

minor (local and residential) roads from the area of interest. The grid size determines the number and density of these points. Our grid is based on OpenStreetMap data, consisting of straight line segments enriched with metadata about the type of road, information about one-way traffic, relative layering, street name, and maximum allowed speed. Every start point and endpoint of a straight road segment is automatically included in the grid and road segments are further divided into pieces equal to the grid size. The dots in Figure 6.3 represent such a grid with grid size 50 m. For Belgium, this results in 3.2 million grid points for the map-based technique instead of 12.2 million for a Cartesian grid.

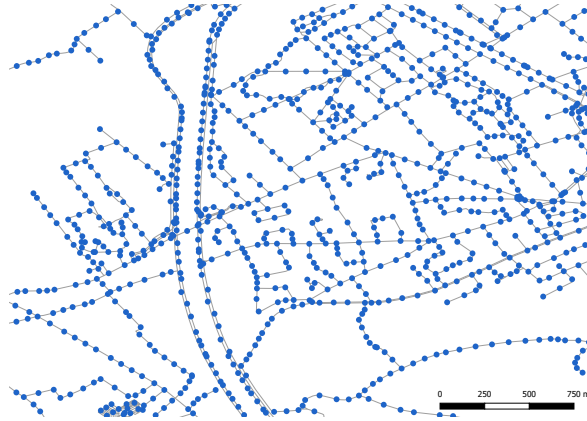


Figure 6.3: Grid based on OpenStreetMap data with grid size 50 m.

6.3.3 Experimental data

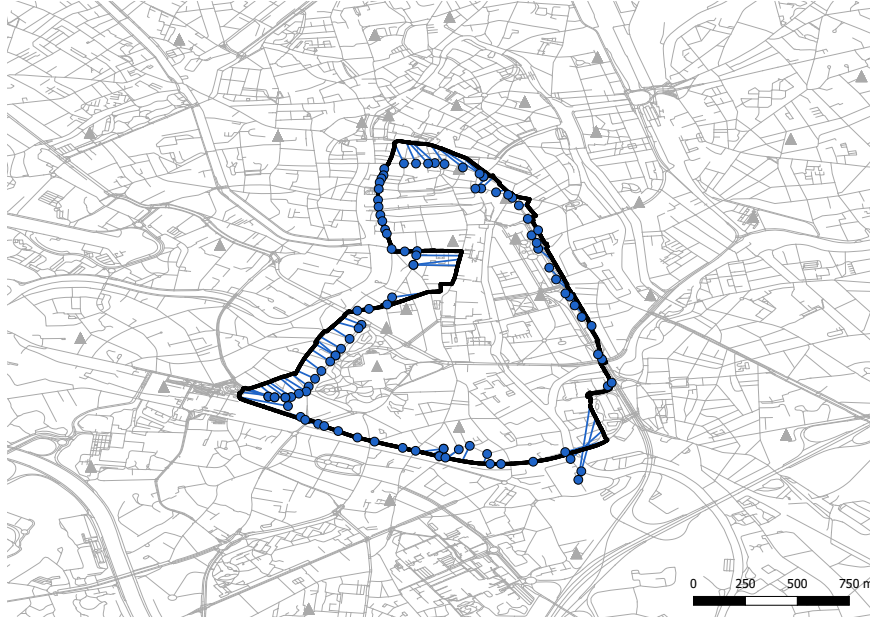
Experimental data is collected in cooperation with a major network operator in Belgium. The experiments are conducted in and around the city center of Ghent and in a smaller town near Antwerp (Melsele), to represent urban and rural environments, and on the highway between both cities. The mobile network collects 3G data for more than a million mobile subscribers but to quantify the location errors (accuracy), the real position or ground truth needs to be known, for which permission and cooperation of a mobile user is needed. The experimental validation encompasses trajectories on foot, by bike, and by car. A smartphone with a GPS logging application is carried in all scenarios by a mobile user. It was put in the dashboard holder for the car rides and carried in the pocket for the trajectories on foot and by bike. The smartphone was forced on 3G to make the experiments independent of having 4G coverage and to ensure a fair comparison between urban and rural environments. This is because in the cooperation with the network operator only access to the 3G measurements was granted and a user would disappear from the collected data if he or she switched to 4G. Figures 6.4 and 6.5 show the GPS trajectories as black lines, the sample rate of the GPS logging application was set to 1 location per second. The NodeB locations are indicated with gray triangles. The GPS trajectories are post-processed with the map matching algorithm of Chapter 5 to increase the accuracy, this is especially useful in urban areas near tall buildings (urban canyoning). Section 6.5 describes the location tracking algorithm and Section 6.6 discusses the performance and accuracy for all trajectories in detail. The total distance, duration, and average speed for all trajectories are summarized in Table 6.1.

Scenario	Distance [km]	Duration [min]	Average Speed [km/h]
Walk (urban)	8	84	6
Bicycle tour (urban)	8	25	19
Car ride (urban)	39	25	47
Walk (rural)	8	101	5
Bicycle tour (rural)	8	22	22
Car ride (rural)	19	28	41
Car ride (highway)	48	36	80

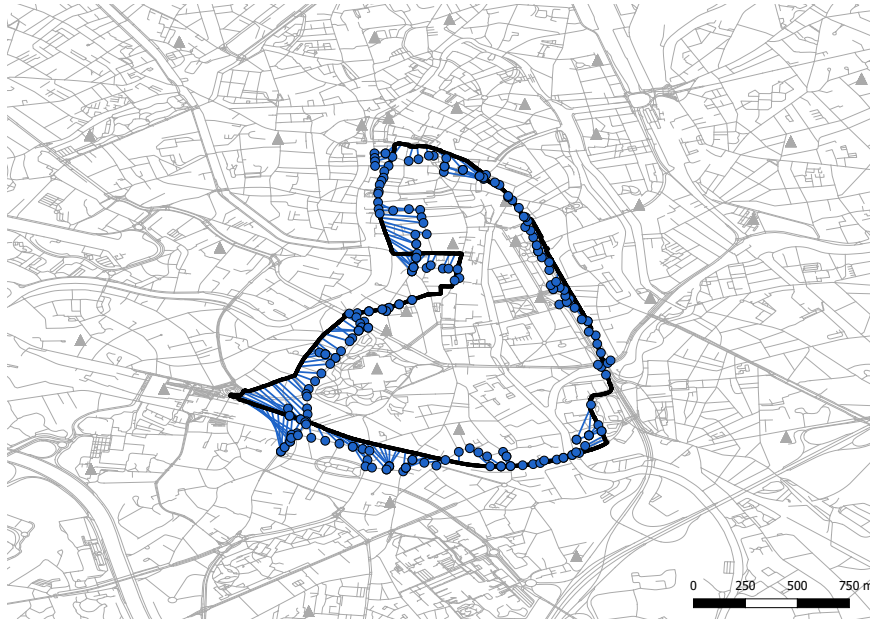
Table 6.1: Trajectory details.

6.3.4 Measurement format

3G measurements are made by the mobile network, i.e., by the radio network controller that controls the NodeBs. The input data for the proposed loca-

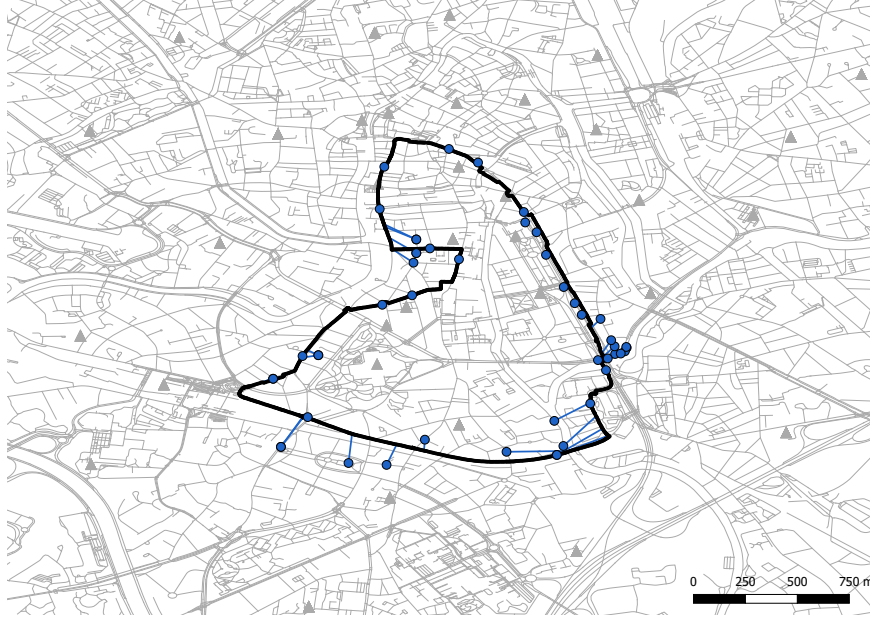


(a) Trajectory on foot (urban + standby)

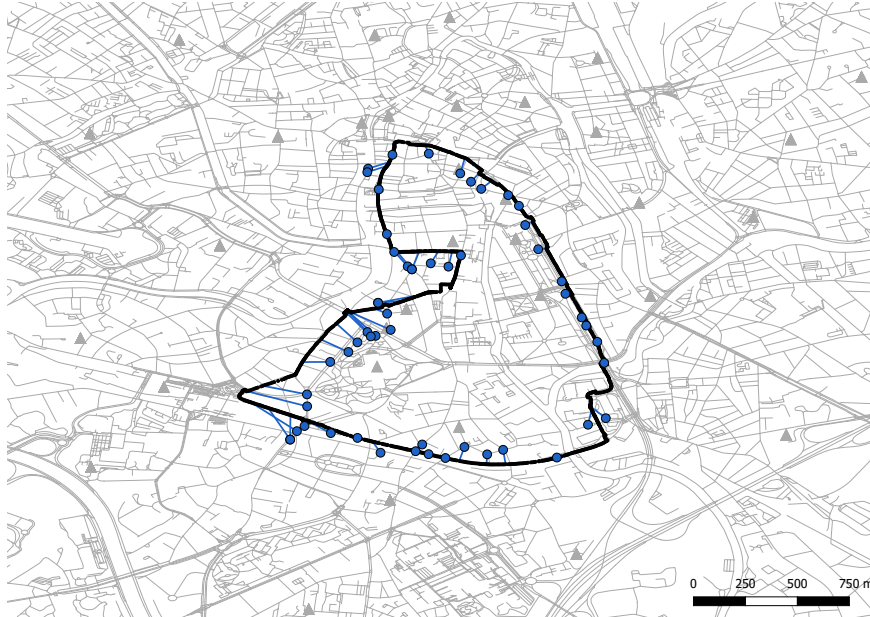


(b) Trajectory on foot (urban + streaming)

Figure 6.4: GPS trajectories on foot in the city center of Ghent (black lines), estimated positions (blue dots), error between estimation and ground truth (blue lines), and NodeBs (gray triangles).

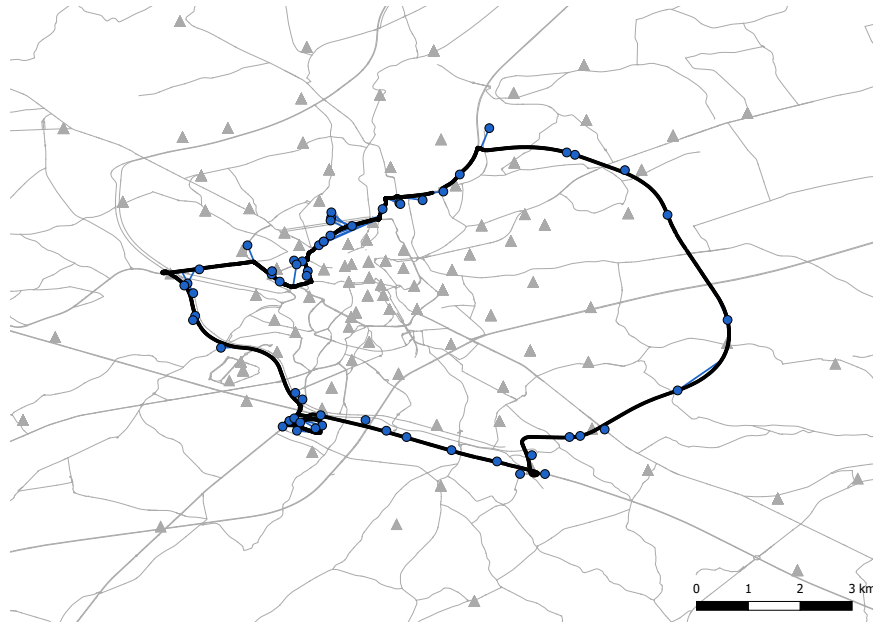


(a) Trajectory by bike (urban + standby)

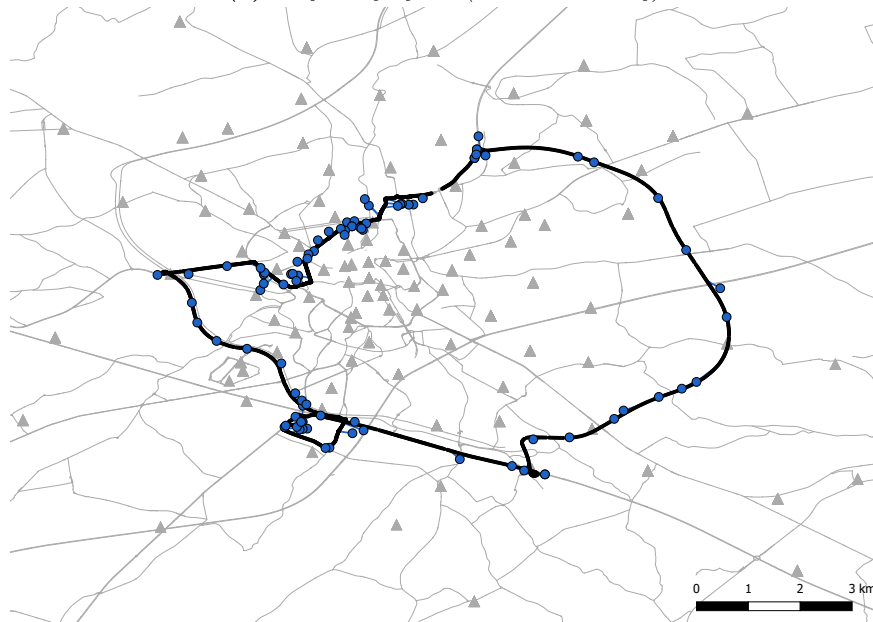


(b) Trajectory by bike (urban + streaming)

Figure 6.5: GPS trajectories by bike in the city center of Ghent (black lines), estimated positions (blue dots), error between estimation and ground truth (blue lines), and NodeBs (gray triangles).

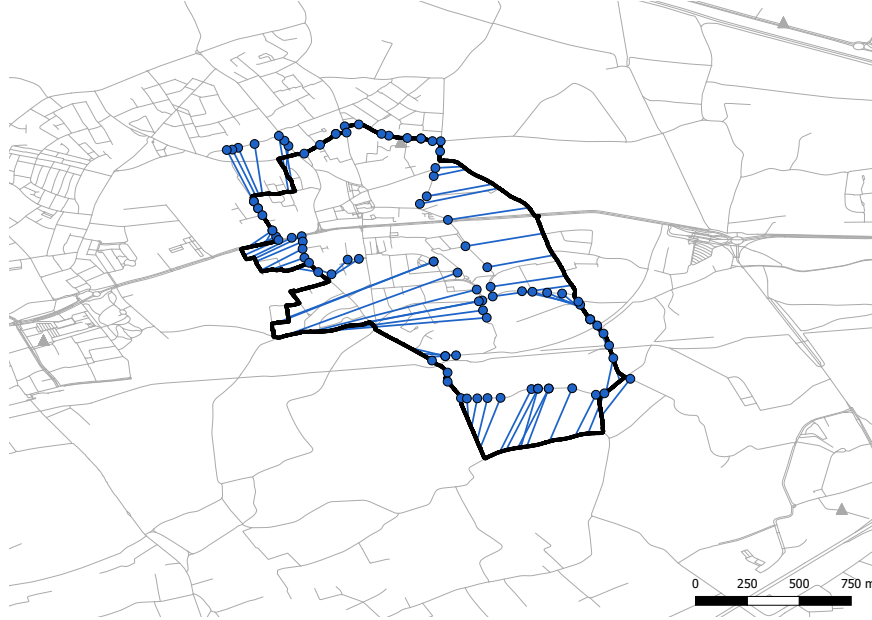


(a) Trajectory by car (urban + standby)

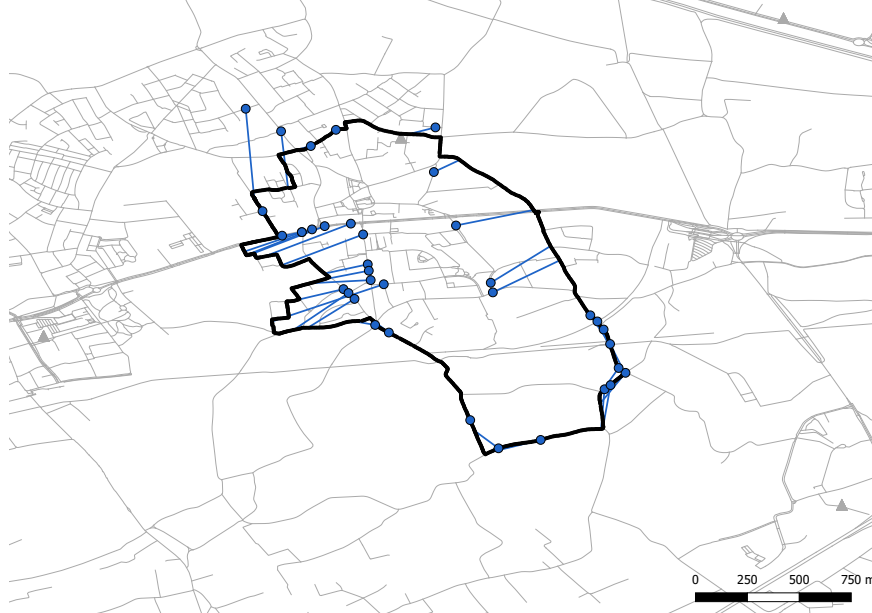


(b) Trajectory by car (urban + streaming)

Figure 6.6: GPS trajectories by car around the city center of Ghent (black lines), estimated positions (blue dots), error between estimation and ground truth (blue lines), and NodeBs (gray triangles).

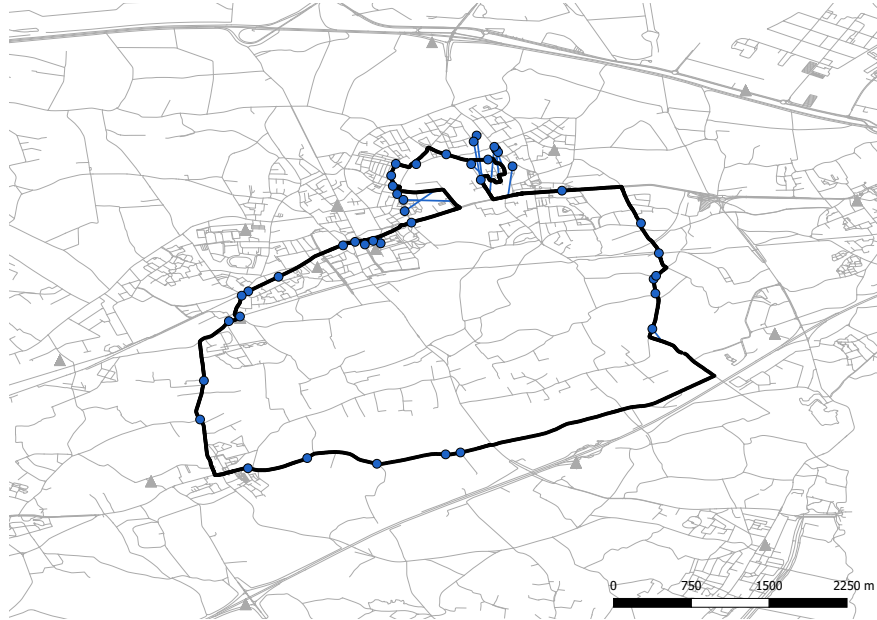


(a) Trajectory on foot (rural + standby)

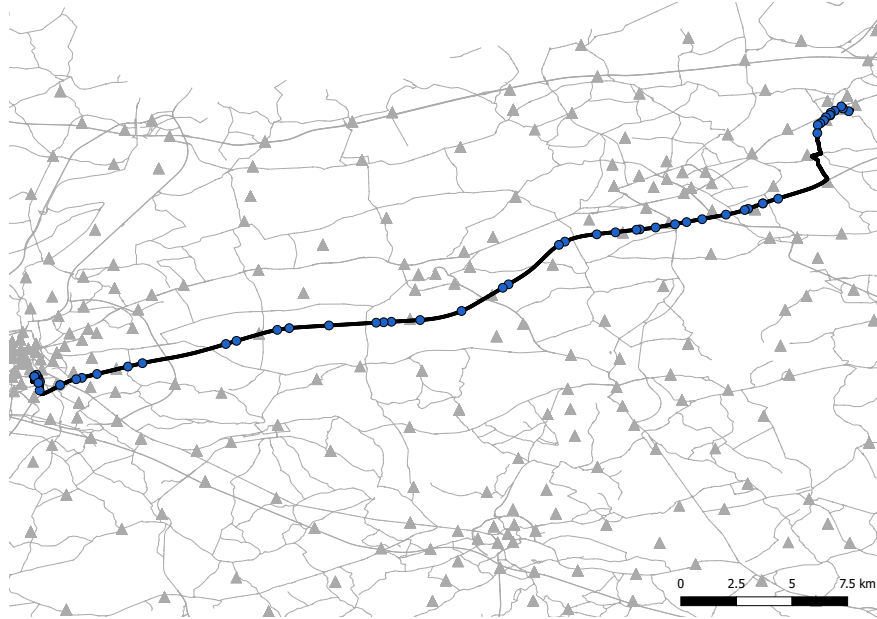


(b) Trajectory by bike (rural + standby)

Figure 6.7: GPS trajectories on foot and by bike in a rural area (black lines), estimated positions (blue dots), error between estimation and ground truth (blue lines), and NodeBs (gray triangles).



(a) Trajectory by car (rural + standby)



(b) Trajectory by car (highway + standby)

Figure 6.8: GPS trajectories by car in a rural area (black lines), estimated positions (blue dots), error between estimation and ground truth (blue lines), and NodeBs (gray triangles).

tion tracking algorithm are timing information and received signal strength values from a set of NodeBs. Both are reported on regular time periods but independently from each other. The timing information comes in the form of a *propagation delay* and is reported only by the serving base station, i.e., the NodeB the mobile device is currently connected to. The signal strength values originate from the *measurement reports* and are reported for all NodeBs that a mobile device currently sees (i.e., from which it receives a broadcast message). Timing information to these other NodeBs would require network changes and increases the load in the mobile network and is consequently not used in the approach in this work.

6.3.5 Propagation delay

The propagation delay parameter can be used to estimate the distance between a mobile device and its serving cell. This delay is used by the radio network controller to make communication possible. It checks and adjusts this delay to allow transmission and reception synchronization. The propagation delay has a time granularity of 780 ns, which corresponds to 234 m [16]. A value of 1 means the mobile user is located in the interval between 234 m and 468 m from the NodeB. Figure 6.9 shows a plot of the real distance (between mobile user and NodeBs) as a function of the observed propagation delay parameter, during a walk of 8 km in the city center of Ghent, Belgium (Figure 6.4b). The theoretical distances for the propagation delays are indicated as a green band. For this test, a radio application was installed on the mobile device and was permanently streaming audio to ensure regular network updates and measurement data. The walk took 84 min during which 234 propagation delay measurements with 49 different cell-IDs from 15 NodeBs were recorded (one physical NodeB can have multiple cell-IDs depending on the number of supported frequencies and different orientations of its antennas). The maximum measured propagation delay during this walk in the city center of Ghent was 6, which corresponds to 1521 m. In rural areas propagation delays up to 22 (≈ 5 km) were recorded with the same mobile device, which is to be expected due to the sparser base station density. The measured propagation delays fall in the correct interval in 69% of the observations. They are one, two and three units apart in 27%, 3%, and 0.4% of the cases, respectively. The mean and standard deviation of the absolute differences between the real and calculated distance are 94 m and 82 m. These values are to be expected with a distance granularity of 234 m (i.e., the calculated distances, based on the 3G propagation delays, are in steps of 234 m). Note that the proposed technique can also be applied to 4G and 5G measurements, which have a higher base station density and more accurate timing information, and therefore, will yield a better location

precision, e.g., 4G has a time granularity of 260 ns, corresponding to 78 m or three times as accurate as 3G.

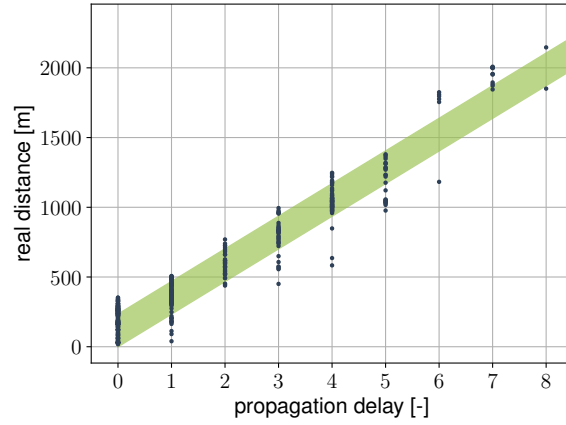


Figure 6.9: *Propagation delay granularity and accuracy.*

6.3.6 Measurement report

Measurement reports contain information about channel quality and are reported by a user entity (mobile device) to a NodeB. They assist the network in making handover and power control decisions. The received signal code power (RSCP) denotes the power measured by a mobile user on a particular physical communication channel, also known as common pilot channel. It continuously broadcasts the scrambling code from the NodeBs and carries no other information. These broadcast messages are transmitted with a constant transmit power and gain but can differ per NodeB (information available in network topology). The measurement reports contain measured signal strength values from all NodeBs the mobile user currently sees. As such, the RSCP values can be converted to a path loss value:

$$PL = P_{TX} + G_{TX} - RSCP \quad (6.1)$$

PL [dB] denotes the total path loss, P_{TX} [dB] and G_{TX} [dB] are the transmit power and gain of a NodeB, and $RSCP$ is the received signal strength code power measured by a mobile device.

Figure 6.10 shows these path loss values on the y-axis and associated distances between mobile user and NodeBs on the x-axis (the measurement reports are collected in the same experiment as the propagation delays from

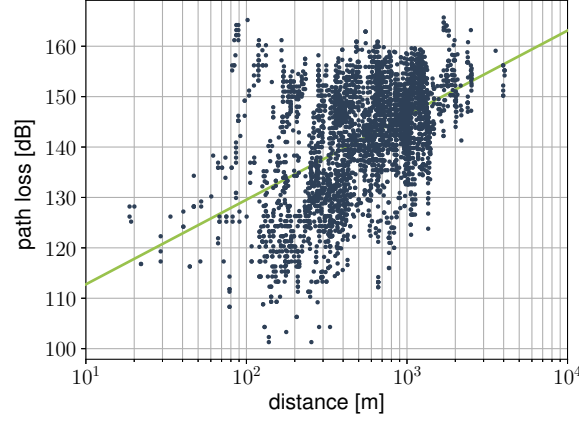


Figure 6.10: Measured path loss as a function of distance between mobile user and NodeBs (blue dots). A fitted path loss model is plotted as a red line.

Section 6.3.5). During the experiment, 578 measurement reports were collected with 4106 RSCP values to 136 different cell-IDs from 32 NodeBs. The fitted one-slope path loss model (red line) has following form:

$$PL = PL_0 + 10\gamma \log_{10} \left(\frac{d}{d_0} \right) + X_\sigma \quad (6.2)$$

PL [dB] denotes the total path loss, PL_0 [dB] is the path loss at a reference distance d_0 [m], γ [-] is the path loss exponent, d [m] is the distance along the path between transmitter and receiver and X_σ [dB] is a log-normally distributed variable with zero mean and standard deviation σ , corresponding to the large-scale shadow fading. The measurement data from this experiment yields a PL_0 of 112.8 dB at a reference distance of 10 m with a γ of 1.68, resulting in an R-squared of 23% and a standard deviation of 9.8 dB. Low R-squared values indicate that the data is not close to the fitted line, which results in bad estimations. Also, deviations in measured path loss will result in larger errors at greater distances to the NodeBs, e.g., for a deviation of 5 dB: a value of 140 dB (420 m) instead of 135 dB (211 m) results in a distance error of 209 m and a value of 150 dB (1655 m) instead of 145 dB (834 m) results in a distance error of 881 m. These larger errors occur rather often, 26% of the measurements have a user-to-NodeB distance that is greater than 1 km. As such, the mean and median absolute distance errors for all 4106 measured values are 1143 m and 473 m, respectively. These distance errors are much larger compared to those derived from the propagation delay, suggesting that many received path loss measurements contain no additional information and can worsen

the accuracy when used together with the timing information as input to a location tracking algorithm. Note that these path loss measurements can be useful in combination with fingerprint maps based on test-driving or crowd-sourced measurement campaigns but these are labor intensive or require modifications on the client side [7, 17, 18]. Also, these crowd-sourced measurement campaigns will be heavily influenced by, e.g., passing cars, new buildings, or other infrastructure changes.

6.4 Data availability

The problem with cellular network data is the limited amount of available data, which determines the number of possible updates. Mobile devices can support a range of different wireless technologies, e.g., infrared, Bluetooth, WiFi, GPS, Universal Mobile Telecommunications System (UMTS) in 3G networks and Long-Term Evolution (LTE) in 4G systems but not all data are available to the network operator and this also depends on the usage of a mobile user.

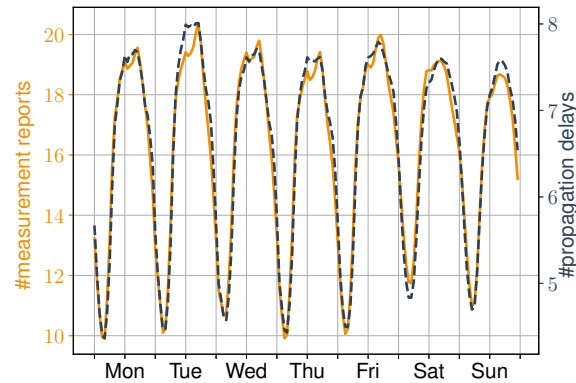


Figure 6.11: Average number of measurement reports (solid orange line) and propagation delays (dotted blue line) per user, per hour for more than a million distinct active users during one week in Belgium.

Figure 6.11 shows the average number of measurement reports and propagation delays, per user, per hour, during one week, measured on a 3G mobile network in Belgium for more than a million distinct active users. It is immediately clear that every day exhibits a similar pattern for both the measurement reports and propagation delays with the difference that there are about twice as many measurement reports. The least and most active hours are 3 a.m. and 6 p.m. respectively (x-axis ticks are set every

12 hours and the labels are set at 12 p.m.). Saturdays and Sundays show a flatter and lower curve than weekdays because more people are staying at home, which translates into fewer measurements per user on the mobile network during the day. On Friday and Saturday between 11 p.m. and 5 a.m., there is an average increase of 20% in number of measurements for a similar amount of people compared to weekdays, indicating that there is more movement or usage of mobile devices (whether or not outdoors). There are more than a million distinct active users during the whole week but the maximum number of active users in one hour is only 700k. This is because not all users send updates to their mobile network when he or she is not moving, has WiFi coverage, or is on a different mobile network (2G or 4G). Current time-series or map-based tracking algorithms assume regular measurement updates to filter outliers and improve the accuracy [6, 12]. This assumption does not hold for many mobile users, making the aforementioned algorithms not generally applicable. The proposed location tracking algorithm can cope with this and consists of multiple phases, depending on the amount of available measurements. Also, it is successfully validated, in cooperation with a major network operator in Belgium, to work in real-time on more than a million subscribers with an Apache Spark implementation to support fast cluster-computing. The used cluster consists of 9 nodes with a total memory of 1.58 TB and 408 physical cores.

6.5 Tracking algorithms

The performance of the proposed location tracking algorithm will be compared with two reference algorithms: cell-ID (Section 6.5.1) and centroid-based (Section 6.5.2). The new tracking algorithm is presented in Section 6.5.3.

6.5.1 Cell-ID

The first reference algorithm is the most simplistic, where a mobile user is mapped to the NodeB it is currently connected to (also known as serving NodeB or serving cell-ID). This approach is easy to implement, has a low cost and short response time but usually has the lowest accuracy [19].

6.5.2 Centroid

The second reference algorithm takes all different NodeBs from the measurement reports into account and calculates the centroid. In case there is only one NodeB with measurements, this approach results in the same location as the cell-ID technique. Alternatively, a weighted centroid algorithm can

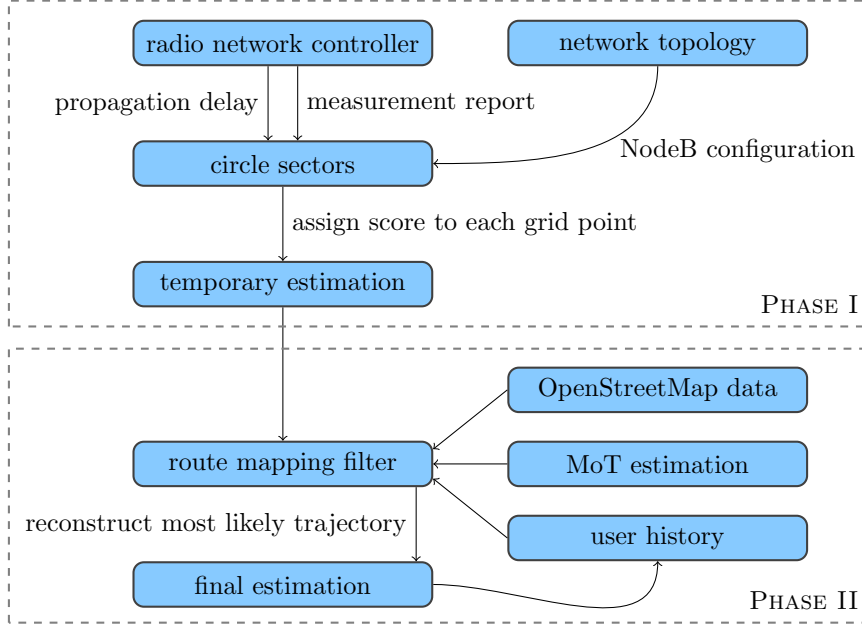


Figure 6.12: Flow graph of the proposed location tracking algorithm: phase I and phase II.

be used, where NodeBs get a weight assigned based on their measurement frequency or received signal strength information [20].

6.5.3 AMT: antenna, map, and timing information based tracking

Figure 6.12 shows a flow graph of the proposed location tracking algorithm, which uses the orientation of NodeB antennas, map, and timing information as input (AMT).

Phase I processes the data measured by the radio network controllers and calculates the temporary estimations (TEs). Phase II further refines these estimated locations with a route mapping filter that uses OpenStreetMap (meta)data, previously estimated positions (user history), and an estimated mode of transportation as input.

6.5.3.1 Phase I: temporary estimation

The pseudo-code to calculate the temporary estimation of a mobile user, residing on the cellular network, is shown in Algorithm 3 and the variables

and steps are discussed with an example in the text below.

Algorithm 3: Calculating the temporary estimation.

Data: measurements reported by radio network controller

Result: temporary estimation (TE)

```

1  $cell_{sc} \leftarrow$  cell-id of the serving cell
2  $loc_{sc} \leftarrow$  location of  $cell_{sc}$ 
3  $\alpha_{sc} \leftarrow$  antenna orientation of  $cell_{sc}$ 
4  $\beta_{sc} \leftarrow$  opening angle of  $cell_{sc}$ 
5  $pd \leftarrow$  reported propagation delay
6  $CA_{sc} \leftarrow$  area bounded by circular arcs based on  $loc_{sc}$ ,  $\alpha_{sc}$ ,  $\beta_{sc}$ , and
    $pd$ 
7  $circleSectors \leftarrow$  empty list
8  $JoinedMeasReport \leftarrow$  measurement reports where  $\alpha$  en  $\beta$  are
   joined if same cell location // explained in text
9 for  $MR \in JoinedMeasReport$  do
10    $cell_{nb} \leftarrow$  cell-id of NodeB that reported  $MR$ 
11    $loc_{nb} \leftarrow$  location of  $cell_{nb}$ 
12    $\alpha_{nb} \leftarrow$  antenna orientation of  $cell_{nb}$ 
13    $\beta_{nb} \leftarrow$  opening angle of  $cell_{nb}$ 
14    $CS_{nb} \leftarrow$  circle sector based on  $loc_{nb}$ ,  $\alpha_{nb}$ , and  $\beta_{nb}$ 
15   add grid points within  $CS_{nb}$  to  $circleSectors$ 
16  $GPMO \leftarrow$  grid points within  $CA_{sc}$  with maximum occurrences in
    $circleSectors$ 
17  $TE \leftarrow$  median location of  $GPMO$ 

```

Consider the example in Figure 6.13: a mobile user is located in the center (yellow square), its serving NodeB ($cell_{sc}$) is indicated with a green star (loc_{sc}) and there are three other NodeBs ($cell_{nb}$) for which there are signal strength measurements (loc_{nb} indicated with red triangles). The antenna orientations (α_{sc} and α_{nb}) of cell-IDs with measurements are indicated with a red line. The other NodeBs in this area (without measurements in this real example) are shown as gray triangles and the grid points are shown as regular dots on top of the road network. The radio network controller reports a propagation delay of 4 from the serving NodeB, which triggers a new location update. This propagation delay corresponds to 1053 m, which limits the possible locations to an area (CA_{sc}) bounded by two circular arcs with an opening angle (β_{sc}) of 120° and radii of 936 m and 1170 m (indicated in transparent green). The distance between both arcs is based on the time granularity of 3G (780 ns corresponds to 234 m). The measurement reports are linked to a propagation delay based on a window of 5 s since they are not reported at the exact same time instances, i.e., all

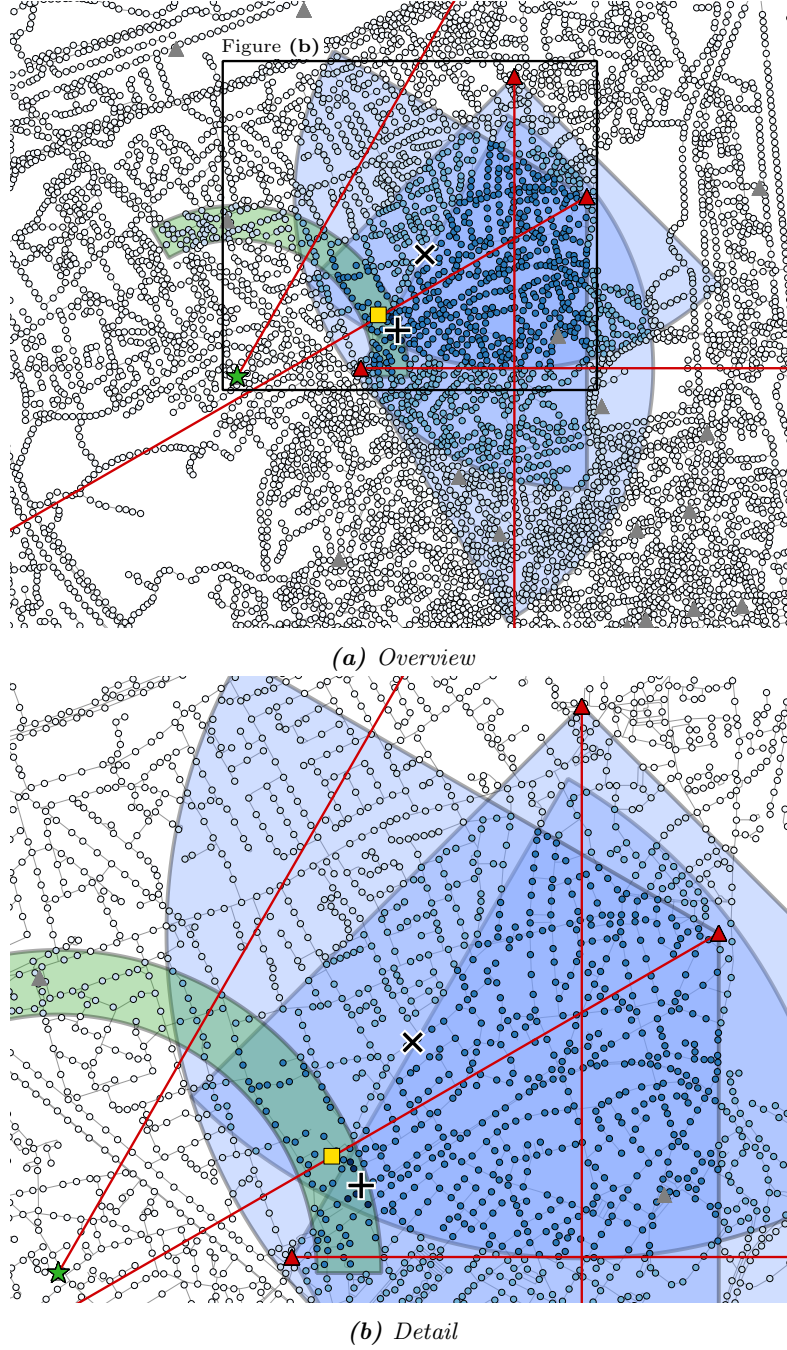


Figure 6.13: Working principle of proposed location tracking algorithm: phase I. Mobile user (yellow square), NodeBs (gray triangles), serving NodeB (green star), NodeBs with measurements (red triangles), antenna orientation (red lines), black cross sign \times (centroid estimation), black plus sign $+$ (AMT estimation), CA_{sc} (green arc), and CS_{nb} (blue circle sectors).

measurements within the interval 2.5 s before and after the timestamp of the propagation delay are taken into account. A window of 5 s resulted in an average of 3.3 measurements that were linked to a propagation delay in the experimental validation of Section 6.6.

Because the calculated distances based on the reported signal strengths from the measurement reports are not reliable (Section 6.3.6), only the orientation (α_{nb}) and opening angle (β_{nb}) of the antennas corresponding to these measurements are used. These are retrieved by looking up the reported cell-ID in the network topology, resulting in three additional circle sectors (CS_{nb}), indicated in transparent blue. The opening angle of the sectors depends on the number of antennas and different orientations the NodeBs have, and is equally divided between all orientations. The most common case of three distinct and equally spread antenna orientations corresponds to an opening angle of 120° (similar to the different gray zones in Figure 6.2a). If there are multiple measurements to one NodeB and the reported cell-IDs correspond to antennas with different orientation, then both measurements are merged (*JoinedMeasReport*) and a new circle sector is used instead, i.e., the smallest area between both orientations. For example, if there are measurements received on the antennas with directions 0° and 90° then the new circle sector would be the first quadrant (0° to 90°) instead of the area from -45° to 135° (Figure 6.2b). Because users that are located just outside a circle sector could be picked up by the antenna, as is visible in Figure 6.13 for the antenna on the bottom center, i.e., the yellow square is not contained in the circle sector although the user is visible for this antenna. A margin of 10° is added to the left and right side of a sector to solve this issue (Figure 6.13 shows the original sectors without the additional margins).

The coloring of the grid points corresponds to the number of NodeBs (cell-IDs) that are visible from this grid point (it is visible if a grid point falls within the sector areas defined above). In this case, there are only 6 locations that satisfy all measurements, i.e., inside the propagation delay area and in all three circle sectors (green and blue areas). The median location of this set (*GPMO*) is the temporary estimation, indicated with a black plus sign (+) in Figure 6.13.

If there is no overlap between the propagation delay area and the circle sectors (green and blue areas) then the median location of the propagation delay area is used as temporary estimation. The latter occurs in only 2% of all location updates during the experimental validation (Section 6.6). Using this approach results, for the depicted example, in an error of 132 m, whereas the cell-ID approach would map the mobile user to the serving NodeB (indicated with a green star), resulting in an error of 1103 m, and

the centroid approach results in an error of 490 m (indicated with a black cross \times).

6.5.3.2 Phase II: route mapping filter

These temporary estimations can be improved with a route mapping filter if there are location updates available from a recent past (user history). For example, a user on foot will travel far less than a user by bike or by car, given a certain time interval. Furthermore, the most likely trajectory over a certain time period can be reconstructed by making use of OpenStreetMap (meta)data: road infrastructure (ways), maximum speed limits, one-way street information, type of road, e.g., sidewalk, bike path, or highway, and the user's history. To take into account cars that are speeding and to avoid that location estimations are lagging behind, the allowed speed limit (for the reconstructed trajectory) can be increased by, e.g., 10% for each road segment. This route mapping filter is based on the map matching algorithm of Chapter 5 and is used to process the temporary estimation, which can be seen as sparse GPS data points with more noise. The pseudo-code of the adapted route mapping filter is shown in Algorithm 4 (the modifications compared to Algorithm 2 are indicated in blue).

There are five main differences with Algorithm 2:

- The input are still geographic coordinates but are now temporary estimations instead of raw GPS data.
- A location update is triggered every time the mobile network reports measurements instead of based on a selection of geographic input coordinates (Section 5.3.2).
- The mode of transportation (MoT) is not assumed to be fixed per trajectory but can change in real-time, i.e., every time the serving cell changes the MoT is re-estimated (Section 6.5.4).
- There is a differentiation between real-time (RT) and non-time-critical (NTC) in the route mapping filter's output (similar to Section 2.5.2). For time-critical applications, the path that currently has the lowest cost, is used as real-time location estimation at each time instance (*AMT-RT*). In this case, previous estimated locations will not be corrected by future measurements, only the user's current history is taken into account. The non-time-critical location estimations (*AMT-NTC*), i.e., for applications that tolerate a certain delay, are calculated when all measurements are processed or after a fixed delay, allowing future measurement to correct previous estimated locations (Section 6.6.3).

Algorithm 4: Route mapping filter.

Data: temporary estimations (TE)
Result: most likely trajectory mapped on roads: AMT-RT (real-time) and AMT-NTC (non-time-critical)

- 1 $TE_0 \leftarrow$ first temporary estimation
- 2 $t_{prev} \leftarrow$ first timestamp with measurements
- 3 $MP \leftarrow 1000$ // maximum paths in memory
- 4 $pathsInMem \leftarrow$ list with MP grid points closest to TE_0 initialized with cost 0
- 5 **while** new measurements **do**
- 6 $t \leftarrow$ current timestamp
- 7 $\Delta t \leftarrow t - t_{prev}$
- 8 $TE \leftarrow$ current temporary estimation
- 9 $MoT \leftarrow$ (re-)estimated mode of transportation
- 10 $pathsTemp \leftarrow$ empty list
- 11 **for** $path \in pathsInMem$ **do**
- 12 $cost \leftarrow$ current cost of $path$
- 13 $PGP \leftarrow$ current endpoint of $path$ (parent grid point)
- 14 $RGP \leftarrow$ reachable grid points along roads with MoT within time span Δt starting from PGP
- // calculate new path cost for each candidate position (CP) based on distance, direction, and code change penalty
- 15 **for** $CP \in RGP$ **do**
- 16 $RS \leftarrow$ road segments between PGP and CP
- 17 $ED \leftarrow eucl_dist(CP, GCP)$ // Euclidean distance
- 18 $DC \leftarrow$ penalty due to direction changes along RS
- 19 $CC \leftarrow$ penalty due to code changes along RS
- 20 $path_{new} \leftarrow path + RS + CP$
- 21 $cost_{new} \leftarrow cost + \alpha \cdot ED + \beta \cdot DC + \gamma \cdot CC$
- 22 add $(path_{new}, cost_{new})$ to $pathsTemp$
- 23 $pathsInMem \leftarrow$ retain MP paths from $pathsTemp$ based on lowest cost
- // current most likely position at time t
- 24 $AMT-RT_t \leftarrow$ endpoint of $path$ with lowest cost
- 25 $t_{prev} \leftarrow t$
- 26 $AMT-NTC \leftarrow$ reconstruct trajectory along $path$ with lowest cost in $pathsInMem$

- The grid size is set to 50 m instead of 10 m because the location error of cellular network positioning is typically a couple orders of magnitude larger than GPS-based tracking [21]. Furthermore, this increases the processing speed and enlarges the covered area in the initialization phase to recover from faulty first estimations, i.e., 1000 grid points and a grid size of 50 m resulted in covered surfaces between 0.5 km² and 2.4 km² in the experimental validation of Section 6.6 (the exact area depends on the density of the road network).

Figure 6.14 shows a detail of the locations before and after the route mapping filter for the trajectory on foot in Ghent (Figure 6.4b). The temporary estimations are indicated with green crosses and the final estimated trajectory with blue dots. In Figures 6.4–6.8 only the final estimated trajectory (blue dots) and ground truth (black line) are shown for clarity.

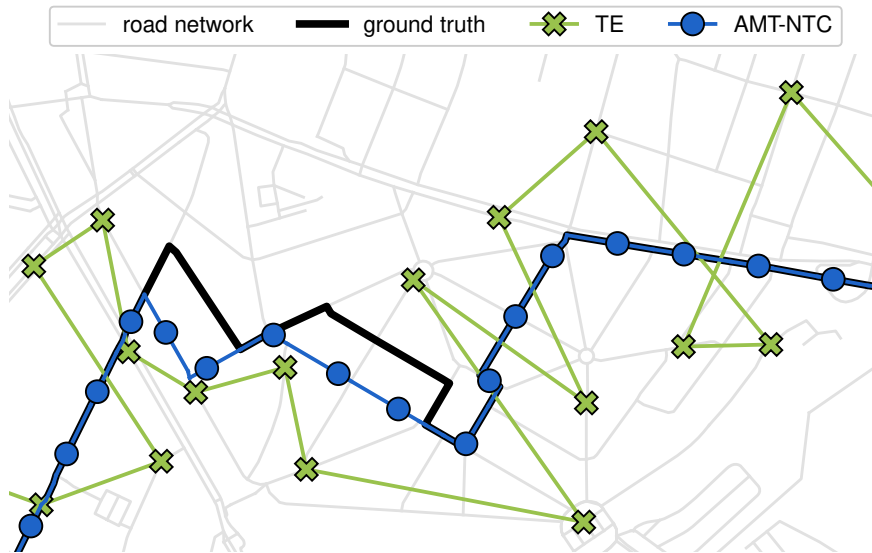


Figure 6.14: Detail of the estimated locations before and after the route mapping filter: temporary estimations (green crosses) and final estimated trajectory (blue dots). The road network and GPS trajectory are indicated in gray and black lines.

6.5.4 Mode of transportation estimator

The maximum allowed speed used by the route mapping filter can be refined if the mode of transportation is correctly estimated, e.g., pedestrians or cyclists will usually not move faster than 6 km/h or 30 km/h. In Chapter 5, the mode of transportation could easily be derived from the GPS

input data, which is accurate enough to give an indication of the maximum speed, but this is not the case for cellular network positioning. In this chapter, the mode of transportation is estimated based on the rate and distance between serving cell handover zones, i.e., when a new NodeB becomes the serving cell. When a handover takes place, the middle between both NodeBs (estimated handover location) is saved together with the timestamp the handover took place. The average speed between all estimated handover locations that took place during a certain moving window, is used to label the mode of transportation. A moving window of 10 min (5 min before and after the location update) could be used for the non-time-critical route mapping filter but this is not possible for real-time applications (as no future measurements are available). For this reason only the last 5 min (counting backwards from the location update that is being calculated) are considered to estimate the average speed. It is labeled as walking if it is below 10 km/h, as cycling if it is between 10 km/h and 25 km/h, and otherwise as driving a motorized vehicle. In the latter case, the route mapping filter will continue to use the maximum allowed road speed for each segment. Although the location updates (TEs) are more frequent and accurate than the estimated handover locations, they show more fluctuations, which results in an overestimation of the average speed (Figure 6.14). For example, during the walk in the city center of Ghent (Figure 6.4b), there are 232 location updates whereas there are only 48 handovers, which result in an average estimated speed of 25 km/h based on the location updates and 7 km/h based on the estimated handover locations with a moving window of 5 min.

6.5.5 Particle Filter

An existing location tracking algorithm [7] based on a particle filter and map information was implemented to validate the proposed route mapping filter (Section 1.9.3). They used regression on drive test data to estimate the probability distribution of an observation. Since drive test data is generally not available for a nationwide mobile network, the likelihood function for the particles is modified to work with the temporary estimations as input (similar to the proposed route mapping filter, ensuring a fair comparison). This particle filter is configured with 2000 particles and the mean μ and variance σ^2 of the initial speed distribution are based on the mode of transportation and the maximum allowed speed of the road segments under consideration. Likewise, at each time step with measurements, the proposed route mapping filter retains the 1000 paths with the lowest associated costs in memory (MP in Algorithm 4).

6.6 Experimental validation

The experimental validation is done in urban and rural environments, near Ghent and Antwerp, with trajectories on foot, by bike, and by car, while a user's smartphone was used in standby and streaming mode, in the months May and September 2017.

6.6.1 General

Figures 6.4 and 6.5 show the estimated positions with the proposed location tracking algorithm as blue dots. The errors between the GPS ground truth and estimated positions are indicated with a blue line. The ground truth is defined as the GPS position, which is closest in time to the timestamp from when the network received measurements that initiated the location update. The GPS logging application takes 1 sample per second and is mapped to the road network (which includes footpaths, paths for cycling, and service roads), ensuring a sufficient time synchronization and accuracy between the estimated positions and their ground truth.

Table 6.2 summarizes the mean, standard deviation, median, and 95th percentile value of the accuracy for all scenarios (walking, cycling, and driving in urban and rural environments with a user's smartphone in standby and streaming mode).

The two basic algorithms are referred to as *Cell-ID* (Section 6.5.1) and *Centroid* (Section 6.5.2). The first phase of the proposed location tracking algorithm (without the route mapping filter) is referred to as *TE* (temporary estimation). The location tracking algorithm with route mapping filter, road speed limits, and mode of transportation estimation is referred to as *AMT*, named after the used inputs: antenna orientation, map, and timing information (phase II). To differentiate between the estimated locations that are available in real-time and those that are corrected by future measurements, *AMT-RT* (real-time) and *AMT-NTC* (non-time-critical) are used. The results based on the particle filter are included in Table 6.2 and referred to as *PF*. The latitude and longitude coordinates from all NodeBs in the mobile network, data from the GPS logging application, and OpenStreetMap data are projected to the Belgian Lambert 72 coordinate system. Consequently, the grid points and estimated locations are in the same plane coordinate reference system. This enables the use of the Euclidean distance between the estimated and actual position to define the accuracy. The total number of location updates and the average time and distance between two consecutive location updates are also included in Table 6.2. Figure 6.15 shows the median accuracy per scenario with the *TE*, *PF*, *AMT-RT*, and *AMT-NTC* technique. The *Cell-ID* and *Centroid* approach are omitted to

Description		Algorithm	μ [m]	σ [m]	50 th [m]	95 th [m]	#updates [-]	Update time [s]	Update distance [m]
Scenario	1	Cell-ID	410	287	342	1006	96	51	76
Environment	urban	Centroid	333	226	276	767			
MoT	foot	TE	270	338	150	744			
Length	8 km	AMT-RT	150	95	125	351			
Speed	5 km/h	AMT-NTC	126	76	119	262			
Smartphone	standby	PF	165	112	139	391	234	21	32
Scenario	2	Cell-ID	453	324	390	1042			
Environment	urban	Centroid	349	240	292	791			
MoT	foot	TE	205	208	151	517			
Length	8 km	AMT-RT	141	82	130	325			
Speed	5 km/h	AMT-NTC	128	82	115	303			
Smartphone	streaming	PF	158	110	137	367	48	32	150
Scenario	3	Cell-ID	586	346	540	1127			
Environment	urban	Centroid	426	270	407	923			
MoT	bike	TE	246	223	172	611			
Length	8 km	AMT-RT	189	139	158	452			
Speed	18 km/h	AMT-NTC	132	83	119	296	55	26	127
Smartphone	standby	PF	193	122	160	486			
Scenario	4	Cell-ID	380	238	305	880			
Environment	urban	Centroid	277	169	226	644			
MoT	bike	TE	187	254	150	384			
Length	8 km	AMT-RT	136	78	131	317	58	46	600
Speed	18 km/h	AMT-NTC	122	80	112	301			
Smartphone	streaming	PF	147	103	131	331			
Scenario	5	Cell-ID	982	587	1030	2013			
Environment	urban	Centroid	808	629	661	2013	91	33	411
MoT	car	TE	441	486	290	1589			
Length	39 km	AMT-RT	370	485	243	1311			
Speed	47 km/h	AMT-NTC	306	291	220	1012			
Smartphone	standby	PF	471	425	372	1322			
Scenario	6	Cell-ID	955	630	901	1989	92	66	81
Environment	urban	Centroid	780	549	645	1966			
MoT	car	TE	382	398	257	1093			
Length	39 km	AMT-RT	336	431	241	844			
Speed	47 km/h	AMT-NTC	217	141	200	467			
Smartphone	streaming	PF	427	439	273	1181	37	36	196
Scenario	7	Cell-ID	1937	1388	1764	4096			
Environment	rural	Centroid	1269	983	1056	3389			
MoT	foot	TE	559	577	433	1469			
Length	8 km	AMT-RT	336	268	276	955			
Speed	5 km/h	AMT-NTC	294	222	275	821	43	39	380
Smartphone	standby	PF	385	313	344	1176			
Scenario	8	Cell-ID	2393	1310	2578	4096			
Environment	rural	Centroid	1175	659	1100	2443			
MoT	bike	TE	522	326	430	1309			
Length	8 km	AMT-RT	305	166	311	709	59	35	775
Speed	22 km/h	AMT-NTC	268	127	243	491			
Smartphone	standby	PF	391	208	389	778			
Scenario	9	Cell-ID	1297	1131	972	3630			
Environment	rural	Centroid	746	490	670	1803	81	38	283
MoT	car	TE	488	594	308	1385			
Length	19 km	AMT-RT	280	224	188	733			
Speed	41 km/h	AMT-NTC	188	186	129	589			
Smartphone	standby	PF	401	357	280	904			
Scenario	10	Cell-ID	1059	702	1021	3096	81	38	283
Environment	highway	Centroid	833	537	790	1851			
MoT	car	TE	352	454	231	1224			
Length	48 km	AMT-RT	235	232	167	826			
Speed	80 km/h	AMT-NTC	138	92	122	277			
Smartphone	standby	PF	395	341	283	1253			
Average over all scenarios		Cell-ID	1045	694	984	2298	81	38	283
		Centroid	700	475	612	1659			
		TE	365	386	257	1032			
		AMT-RT	248	220	197	682			
		AMT-NTC	192	138	165	482			
		PF	313	253	251	819			

Table 6.2: Accuracy, number of positioning updates, and average time and distance between two consecutive location updates, per scenario and algorithm. TE: temporary estimation (phase I), AMT-RT: real-time route mapping filter (phase II), AMT-NTC: non-time-critical route mapping filter (phase II), PF: particle filter.

enhance clarity.

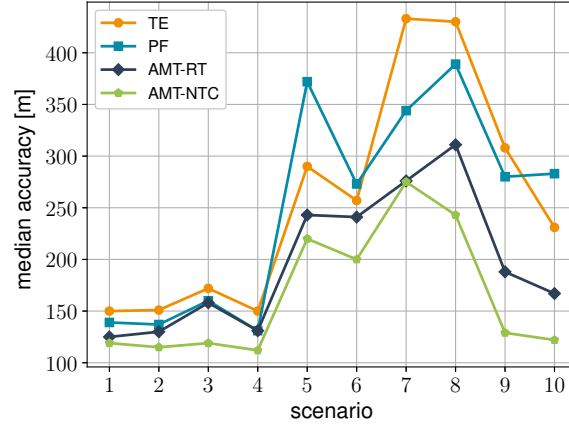


Figure 6.15: Median accuracy per scenario with the TE, PF, AMT-RT, and AMT-NTC technique.

6.6.2 Comparison with other algorithms

It is immediately clear that the proposed location tracking algorithms outperform the classic *Cell-ID* and *Centroid* approach in all ten scenarios. The particle filter [7] performs slightly worse than the proposed route mapping filter (real-time and non-time-critical version) in scenarios 1–4 (trajectories in urban areas on foot and by bike) and is significantly outperformed in scenarios 5–10. The main reason for this is that the time between two location updates is variable and can be rather large (it ranges from 5 s to 6 min). In the update step of the particle filter, a new state is sampled for all particles, based on the previous state, current time, and a new random sample, and are then mapped on the road network. This can cause large deviations if the user’s real speed or direction changes in this time period, which can happen multiple times during a sizeable measurement gap. The trajectories done by car and the ones in rural areas are most affected by this, which can even cause the particle filter to perform worse than *TE* (scenarios 5, 6, and 10).

In phase II of the *AMT* approach, i.e., route mapping filter (Section 6.5.3.2), all possible locations that can be reached along the road network in this time period are considered as candidate positions for the next location update (given the paths in memory, estimated mode of transportation, maximum speed limits, type of roads, and one-way street information). The median

TE accuracy varies between 150 m and 433 m, and has an average improvement, over all scenarios, of 68% and 55% compared to the *Cell-ID* and *Centroid* approach, respectively. The median *PF* accuracy varies between 131 m and 389 m, and has an average improvement, over all scenarios, of 69%, 56%, and 2% compared to the *Cell-ID*, *Centroid*, and *TE* approach respectively. The median *AMT-RT* accuracy varies between 125 m and 311 m, and has an average improvement, over all scenarios, of 74%, 64%, 20%, and 18% compared to the *Cell-ID*, *Centroid*, *TE*, and *PF* approach, respectively. The median *AMT-NTC* accuracy varies between 112 m and 275 m, and has an average improvement, over all scenarios, of 78%, 69%, 33%, 31%, and 16% compared to the *Cell-ID*, *Centroid*, *TE*, *PF*, and *AMT-RT* approach, respectively. The mean accuracies, standard deviations, and 95th percentile values show similar improvements.

The largest relative improvements compared to the reference algorithms are achieved with the trajectory on the highway (scenario 10). The median accuracy improves by 88% (from 1021 m to 122 m) compared to the *Cell-ID* approach, by 85% (from 790 m to 122 m) compared to the *Centroid* approach, and by 57% (from 283 m to 122 m) compared to the *PF* approach.

The most accurate results reported in the state-of-the-art processing techniques from Section 6.2 are better than the results in this work (accuracies up to 20 m [7], 30 m [15], and 50 m [14]) but these are achieved with simulated data, large training sets, which are not feasible for nationwide deployments, optimal environments, crowd-sourced measurement campaigns, and forced location update rates. However, applying the processing technique from [7] (particle filter with map information) on the validation data of Section 6.6 resulted in worse accuracies but gives a realistic idea of the achievable performance without crowd-sourcing or modifications to the network or mobile side (*PF* in Table 6.2).

6.6.3 Non-time-critical vs. real-time

The non-time-critical version of the route mapping filter (*AMT-NTC*), that takes into account all measurements at once, can also work with a smaller delay (instead of at the end of a trajectory). Previously predicted locations can be corrected by multiple future measurements but the impact tends to decrease as more time has passed between the previous update and those future measurements. For the experimental validation this time period is 8 min; taking into account additional future measurements does not further improve the overall accuracy. Even with only 2 min of future measurement data, the mean and median overall accuracy are already 200 m and 174 m (compared to 192 m and 165 m if all future measurements are taken into account). This means that if a time delay of 2 min is allowed for the

intended application, the overall mean accuracy can already be improved by 19% compared to the real-time algorithm (*AMT-RT*).

6.6.4 Impact of environment

The highest accuracies are achieved for the scenarios in an urban environment with trajectories on foot or by bike (scenarios 1–4). For example, the trajectory by bike in the city center of Ghent with a smartphone in streaming mode (scenario 4) has a mean, standard deviation, median, and 95th percentile value of 122 m, 80 m, 112 m, and 301 m, respectively. These better accuracies are mainly due to the higher base station density, which is typical in urban environments. This ensures that the serving base stations have smaller separations and hence this limits the possible grid points because of the lower propagation delays, i.e., the green sector in Figure 6.13 will cover a smaller area. When driving a car, the absolute accuracy in urban environments is worse than in rural scenarios. For example, the improvements between two trajectories by car in an urban (scenario 5) and rural environment (scenario 9), are 63% (306 m to 188 m), 56% (291 m to 186 m), 71% (220 m to 129 m), and 72% (1012 m to 589 m), for the mean, standard deviation, median, and 95th percentile value, respectively. This is due to the sparser road network in rural areas, which increase the chance that the route mapping filter selects the correct road segments as most likely. The trajectory on the highway (scenario 10) is accurately reconstructed because the roads surrounding the highway have lower speed limits causing these (incorrect) candidate paths to lag behind and eventually be discarded in the route mapping algorithm. Note that this is only true if there is no traffic congestion.

6.6.5 Impact of smartphone usage

The shortest location update time or highest update rate occurs when a user is walking in an urban environment while actively using his or her smartphone, i.e., through an application that sends or receives data over the mobile network on a regular basis (scenario 2). In this case, there are 234 updates during the entire trajectory, which corresponds to a location update every 21 s or every 32 m on average. Note that the update rate for this *best case scenario* is not as high as most positioning algorithms for cellular networks are validated on. Location update rates of 0.5 s [12] and 10 s [7] are reported in related work, by using forced messages or synthesized validation data. These can thus be considered as not being realistic. Three trajectories are traveled for both smartphone usage modes (scenarios 1–6). The trajectories in an urban environment on foot and by bike are identical

and yield similar performances for the streaming and standby mode (scenarios 1–4). The higher location update rate has a negligible impact due to the limited speed for these modes of transportation. The trajectory by car shows a significant improvement for higher location update rates (standby vs. streaming). The accuracy improves by 41% (306 m to 217 m), 106% (291 m to 141 m), 10% (220 m to 200 m), and 117% (1012 m to 467 m), for the mean, standard deviation, median, and 95th percentile value, respectively.

6.6.6 Impact of mode of transportation

The trajectories done on foot and by bike yield similar accuracies as long as the environment is the same. The trajectories done by car perform worse in urban environments but better in rural environments as discussed in Section 6.6.4. It is to be noted that the proposed MoT estimator achieved an accuracy of 78% when a moving window of the last 5 min was used. Although this accuracy could be improved on the validation data by using a longer window, this will not always be the case, e.g., if the MoT changes during a scenario from walking to biking, a shorter window is recommended to detect the changes more quickly. Furthermore, the overall mean and median accuracy remained similar (192 m and 165 m vs. 183 m and 164 m) if the route mapping filter was provided with the correct MoT at each location update. This is because a wrong MoT estimation for a location update does not automatically result in a worse accuracy, e.g., when it is erroneously labeled as cycling while the user was actually driving at a slow speed due to traffic congestion.

6.7 Conclusion

In this chapter, a technique for outdoor location tracking of all mobile users residing on a cellular network, is presented. The proposed approach does not depend on GPS or prior training data and does not require any cooperation on the mobile side or changes to the network side. The topology and available measurements of a cellular network are used as input for the proposed AMT algorithm (named after antenna, map, and timing information). An additional route mapping filter is applied to ensure realistic, physically possible, trajectories. The inputs for this route mapping filter are the user's location history, enriched open map data (road infrastructure, maximum speed limits, type of road, and one-way street information), and a mode of transportation estimator to improve the assumed corresponding maximum speed. The novel AMT location tracking algorithm is implemented

in Apache Spark to support fast cluster-computing, runs completely on the network side, is confirmed to execute in real-time for more than a million users in parallel, and outperforms state-of-the-art particle filters. The experimental validation is done in urban and rural environments, near Ghent and Antwerp, with trajectories on foot, by bike, and by car, while a user's smartphone was used in standby and streaming mode. Best performances were obtained in urban environments with median accuracies up to 112 m. It is shown that the mode of transportation, smartphone usage, and environment impact the accuracy and that the proposed AMT location tracking algorithm is more robust and outperforms existing techniques with relative improvements up to 88%, 85%, and 57% compared to a cell-ID, a centroid, and a particle filter with map information based location tracking technique, respectively.

References

- [1] Bharat Rao and Louis Minakakis. *Evolution of mobile location-based services*. Communications of the ACM, 46(12):61–65, 2003.
- [2] Linus Bengtsson, Xin Lu, Anna Thorson, Richard Garfield, and Johan Von Schreeb. *Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: a post-earthquake geospatial study in Haiti*. PLoS medicine, 8(8):e1001083, 2011.
- [3] OpenStreetMap contributors. *Planet dump retrieved from <https://planet.osm.org>*. ”<https://www.openstreetmap.org>. Accessed: 2017-02-12.
- [4] Nico Deblauwe. *GSM-based positioning: techniques and applications*. ASP/VUBPRESS/UPA, 2008.
- [5] Ismail Guvenc and Chia-Chin Chong. *A survey on TOA based wireless localization and NLOS mitigation techniques*. IEEE Communications Surveys & Tutorials, 11(3), 2009.
- [6] Po-Hsuan Tseng, Kai-Ten Feng, Yu-Chiun Lin, and Chao-Lin Chen. *Wireless location tracking algorithms for environments with insufficient signal sources*. IEEE Transactions on Mobile Computing, 8(12):1676–1689, 2009.
- [7] Avik Ray, Supratim Deb, and Pantelis Monogioudis. *Localization of LTE measurement records with missing information*. In Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on, pages 1–9. IEEE, 2016.
- [8] Mussa Bshara, Umut Orguner, Fredrik Gustafsson, and Leo Van Biesen. *Robust tracking in cellular networks using HMM filters and cell-ID measurements*. IEEE Transactions on Vehicular Technology, 60(3):1016–1024, 2011.
- [9] S. Coleri Ergen, H.S. Tetikol, M. Kontik, R. Sevlial, R. Rajagopal, and P. Varaiya. *RSSI-Fingerprinting-Based Mobile Phone Localization With Route Constraints*. Vehicular Technology, IEEE Transactions on, 63(1):423–428, Jan 2014.
- [10] Michael McGuire, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. *Data fusion of power and time measurements for mobile terminal location*. IEEE Transactions on Mobile Computing, 4(2):142–153, 2005.

- [11] Yongjiu Feng, Yan Liu, and Michael Batty. *Modeling urban growth with GIS based cellular automata and least squares SVM rules: a case study in Qingpu–Songjiang area of Shanghai, China*. Stochastic environmental research and risk assessment, 30(5):1387–1400, 2016.
- [12] Marco Anisetti, Claudio A Ardagna, Valerio Bellandi, Ernesto Damiani, and Salvatore Reale. *Map-based location and tracking in multipath outdoor mobile networks*. IEEE Transactions on Wireless Communications, 10(3):814–824, 2011.
- [13] Reza Monir Vaghefi and R Michael Buehrer. *Cooperative RF pattern matching positioning for LTE cellular systems*. In Personal, Indoor, and Mobile Radio Communication (PIMRC), 2014 IEEE 25th Annual International Symposium on, pages 264–269. IEEE, 2014.
- [14] Robert Margolies, Richard Becker, Simon Byers, Supratim Deb, Ritwik Jana, Simon Urbanek, and Chris Volinsky. *Can you find me now? Evaluation of network-based localization in a 4G LTE network*. In INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, pages 1–9. IEEE, 2017.
- [15] Ayon Chakraborty, Luis E Ortiz, and Samir R Das. *Network-side positioning of cellular-band devices with minimal effort*. In Computer Communications (INFOCOM), 2015 IEEE Conference on, pages 2767–2775. IEEE, 2015.
- [16] *Propagation Delay*. <http://www.telecomhall.com/analyzing/coverage-with-propagation-delay-pd-and-timing-advance-ta/gsm-wcdma-lte.aspx>. Accessed: 2010-08-09.
- [17] Mohamed Ibrahim and Moustafa Youssef. *CellSense: An accurate energy-efficient GSM positioning system*. IEEE Transactions on Vehicular Technology, 61(1):286–296, 2012.
- [18] Torbjörn Wigren. *Adaptive enhanced cell-ID fingerprinting localization by clustering of precise position measurements*. IEEE Transactions on Vehicular Technology, 56(5):3199–3209, 2007.
- [19] Emiliano Trevisani and Andrea Vitaletti. *Cell-ID location technique, limits and benefits: an experimental study*. In Mobile computing systems and applications, 2004. WMCSA 2004. Sixth IEEE workshop on, pages 51–60. IEEE, 2004.
- [20] Jun Wang, Paulo Urriza, Yuxing Han, and Danijela Cabric. *Weighted centroid localization algorithm: theoretical analysis and distributed*

- implementation*. IEEE Transactions on Wireless Communications, 10(10):3403–3413, 2011.
- [21] Paul A Zandbergen. *Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning*. Transactions in GIS, 13:5–25, 2009.

Chapter 7

Conclusions and future work

This final chapter presents the overall conclusions based on the accomplished work in this dissertation, and proposes some opportunities for future research.

7.1 Conclusions

The focus of this dissertation was on indoor and outdoor location tracking systems with an emphasis on signal strength based systems, human body shadowing, radio map optimization, outdoor map matching, lane detection, and cellular networks.

Chapter 2 presented a real-time indoor location tracking system based on the Viterbi principle and semantic data. The system was evaluated by both simulations and an extensive experimental validation in a real office environment. The simulations confirmed that the proposed location tracking system was more robust against measurement noise, especially for networks with smaller node densities, e.g., a simulated noise level of 10 dB resulted in improvements of 70.9%, 52.8%, 54.1%, and 42.1% in mean accuracy compared to a fingerprinting technique, Kalman filter, particle filter, and particle smoother, respectively. In the experimental validation, an average median accuracy below 2 m was obtained over nine test trajectories with a total length of 783 m, in an office building that has 57 nodes and measures 90 m by 17 m (covering over 1500 m²). Furthermore, it was shown that the grid size has a huge impact on the execution time, required computational power, and memory usage, which is important to work in real-time on low cost portable devices.

Chapter 3 presented novel techniques to reduce the effects of human body shadowing on a tracking algorithm's performance. Two methods are found to be effective: the first one combines multiple mobile tags and the second one compensates explicitly for the human body shadowing caused by the user that is being tracked. Both methods can be independently combined, resulting in the most accurate performance. The first approach exploits the measured signal strengths from multiple mobile nodes, placed on different parts of a human body. Combining their measured signal strengths allows reducing the variation caused by the user's body and hence limits its influence. The second method used a compensation model based on three-dimensional electromagnetic simulations with a human phantom and relies on the orientation of a user's tag towards the infrastructure nodes. Using three instead of one body-worn tag resulted in a mean accuracy improvement of 31.1%, in an office building that has 48 nodes and measures 90 m by 17 m (covering over 1500 m²). Compensating for the user's orientation, further improved this result with 10.0%.

Chapter 4 presented an unsupervised learning technique to construct and optimize model-based radio maps or fingerprint databases for indoor positioning systems, e.g., to make the radio map more accurate or to automatically cope with changes in an office layout. The proposed technique does not rely on time-consuming measurement campaigns, device calibrations, or additional inertial measurement units, that are power consuming. Instead, it used an initial radio map based on a theoretical path loss model, unlabeled training data, a self-calibration method, and a route mapping filter. The premise of this approach was that the differences between real measurements and reference values, derived from a model-based radio map, tend to be correlated per room and access point. It was shown by measurements and simulations that the discrepancies between reference fingerprints and real measurements could be learned in various scenarios, based on the random walks by a typical person. An experimental validation on a testbed in a large office building, measuring 41 m by 27 m (covering over 1100 m²) and that has 35 nodes, confirmed the simulations. The highest relative improvement was 28.6% after unsupervised learning with only 15 min of unlabeled training data.

Chapter 5 presented a fast, memory-efficient, and worldwide map matching algorithm based on raw geographic coordinates and enriched open map data. The proposed algorithm combined the Markovian behavior and the shortest path aspect while taking into account the type and direction of all road segments, information about one-way traffic, maximum allowed speed per road segment, and driving behavior. Furthermore, a lane detection algorithm based on accelerometer readings and traffic lane information from the

open map data, that self-adapts to different driving behaviors, was added on top of the map matching algorithm. An experimental validation consisting of 12 trajectories on foot, by bike, and by car, showed the efficiency and accuracy of the proposed algorithms. The average F1-scores and median errors of the map matching algorithm were 99.1 % and 2.24 m, 99.5 % and 1.88 m, and 100.0% and 2.16 m, for the walks, bike rides, and car routes, respectively. Two trajectories with accelerometer data were used to evaluate the lane detection algorithm with F1-scores of 98.2% and 75.1% for the lane change detection, which resulted in the correctly estimated lane 99.2% and 86.8% of the time.

Chapter 6 presented a technique for outdoor location tracking of all mobile users residing on a cellular network. The proposed approach does not depend on prior training data and does not require any cooperation on the mobile side or changes to the network side. The topology and available measurements of a cellular network were used as input for the proposed AMT algorithm (named after antenna, map, and timing information). The map matching filter of Chapter 5 is applied to ensure realistic, physically possible, trajectories. The experimental validation is done in urban and rural environments, near Ghent and Antwerp, with trajectories on foot, by bike, and by car, while a user's smartphone was used in standby and streaming mode. It was shown that the mode of transportation, smartphone usage, and environment impact the accuracy and that the proposed AMT location tracking algorithm was more robust and outperformed existing techniques with relative improvements up to 88%, 85%, and 57% compared to a cell-ID, a centroid, and a particle filter with map information based location tracking technique, respectively.

7.2 Future work

Directions for future work are to study the effect of different indoor environments, e.g., offices with a narrow layout, industrial rooms, or open indoor hall spaces, on the performance of the proposed location tracking algorithms. Moreover, the impact of test and training data with multiple, simultaneously active users, with different mobile devices and mobility patterns, covering multiple floors, and the influence of access point location uncertainty can be looked into. Furthermore, different ranging techniques can be combined, e.g., RSS, AoA, and TDoA, to acquire a more robust and accurate performance. Another interesting direction is to assess the impact of walking behavior and human morphologies on the body shadowing compensation models and to apply the approach on objects by adapting the compensation models. Future work on the topic of radio map maintenance,

is to experimentally verify the ability to recover from physical changes in the environment and the effect of the proposed unsupervised learning technique on measurement-based fingerprint databases. Although the map matching algorithm has support for worldwide geographic data, it would be good to evaluate the performance in different geographic areas to gain insight in the effect of national road network patterns and topographies. Lastly, the proposed cellular location tracking algorithm can be adapted and applied to 4G LTE and 5G mobile networks, where further improvements are expected thanks to the more accurate timing information and the higher node densities.