



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Helena Leppäkoski

Novel Methods for Personal Indoor Positioning



Julkaisu 1330 • Publication 1330

Tampereen teknillinen yliopisto. Julkaisu 1330
Tampere University of Technology. Publication 1330

Helena Leppäkoski

Novel Methods for Personal Indoor Positioning

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB219, at Tampere University of Technology, on the 27th of October 2015, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2015

ISBN 978-952-15-3596-3 (printed)
ISBN 978-952-15-3612-0 (PDF)
ISSN 1459-2045

ABSTRACT

Currently, people are used to getting accurate GNSS based positioning services. However, in indoor environments, the GNSS cannot provide the accuracy and availability comparable to open outdoor environments. Therefore, alternatives to GNSS are needed for indoor positioning. In this thesis, methods for pedestrian indoor positioning are proposed. With these novel methods, the mobile unit performs all the required positioning measurements and no dedicated positioning infrastructure is required.

This thesis proposes novel radio map configuration methods for WLAN fingerprinting based on received signal strength measurements. These methods with different model parameters were studied in field tests to identify the best models with reasonable positioning accuracy and moderate memory requirements. A histogram based WLAN fingerprinting model is proposed to aid IMU based pedestrian dead reckoning that is obtained using a gyro and a 3-axis accelerometer, both based on MEMS technology. The sensor data is used to detect the steps taken by a person on foot and to estimate the step length and the heading change during each step.

For the aiding of the PDR with WLAN positioning, this thesis proposes two different configurations of complementary extended Kalman filters. The field tests show that these configurations produce equivalent position estimates. Two particle filters are proposed to implement the map aided PDR: one filter uses only the PDR and map information, while the other uses also the WLAN positioning. Based on the field tests, map aiding improves the positioning accuracy more than WLAN positioning. Novel map checking algorithms based on the sequential re-selection of obstacle lines are proposed to decrease the computation time required by the indoor map matching. To present the map information, both unstructured and structured obstacle maps are used. The feasibility of the proposed particle filter algorithms to real time navigation were demonstrated in field tests.

PREFACE

The research presented in this thesis has been carried out under the supervision of Prof. Jarmo Takala and Dr. Jussi Collin in the Department of Computer Systems and its successor Department of Pervasive Computing at Tampere University of Technology.

I would like to thank my supervisor Prof. Jarmo Takala for his professional insight, guidance, and support during the work. I would also like to thank Dr. Jussi Collin for his technical advice and ideas. I especially recognize the invaluable efforts of my pre-examiners Prof. Washington Yotto Ochieng and Dr. Mohammad Zahidul Hasan Bhuiyan for providing constructive comments. I would also like to thank Prof. Gwo Giun (Cris) Lee and Dr. Mohammad Zahidul Hasan Bhuiyan for agreeing to be the opponents in the public examination of my thesis.

The other experts that I would like to thank for advice and for sharing their knowledge are Prof. Robert Piché, Dr. Simo Ali-Löyhty, Prof. Jari Nurmi, Dr. Elena Simona Lohan, Prof. Ruizhi Chen, Dr. Ling Pei, and Dr. Stephan Sand. My special thanks go to Dr. Terho Jussila for the additional review of my thesis.

My present and former colleagues deserve special thanks for their friendship, helpful assistance, and stimulating discussions. In particular, I would like to thank Alejandro Rivero-Rodriguez, Henri Nurminen, Xiaolong Zhang, Juha Ala-Luhtala, Philipp Müller, and Dr. Matti Raitoharju, and especially my closest colleagues during the research work of this thesis: Arto Perttula, Saija Tikkinen, Dr. Martti Kirkko-Jaakkola, Dr. Pavel Davidson, Olli Pekkalin, Jussi Parviainen, Jayaprasad Bojja, and Francescantonio Della Rosa.

This research was funded by Department of Computer Systems, Finnish Funding Agency for Technology and Innovation (TEKES) under the projects “3D Personal Navigation and Location-Based Service for World Exposition 2010” (3D-NAVI-EXPO) and “Smart Integration for Rescue Teams” (EURIPIDES

SINETRA), and EU FP7 under the project “Galileo Ready Advanced Mass Market Receiver” (GRAMMAR).

Finally, I express my deepest gratitude to my family and friends for their support, encouragement, and understanding.

Lempäälä, September 2015

Helena Leppäkoski

TABLE OF CONTENTS

<i>Abstract</i>	i
<i>Preface</i>	iii
<i>List of Figures</i>	ix
<i>List of Tables</i>	xiii
<i>Abbreviations</i>	xv
<i>Symbols</i>	xix
<i>1. Introduction</i>	1
1.1 Scope and Objective of the Research	2
1.2 Main Contributions	3
1.3 Author's Contribution	4
1.4 Thesis Outline	5
<i>2. Preliminaries</i>	7
2.1 WLAN Fingerprinting	7
2.1.1 Radio Map	9
2.1.2 Pattern Matching	11
2.1.3 Probabilistic Model	12
2.2 Pedestrian Dead Reckoning	14
2.2.1 Step Detection	16
2.2.2 Step Length Estimation	18
2.3 Indoor Map	19

2.4	Data Fusion Using Bayesian Filtering	21
2.4.1	Kalman Filter	22
2.4.2	Extended Kalman Filter	24
2.4.3	Complementary Filter	25
2.4.4	Particle Filtering	29
2.5	Assessment of Positioning Accuracy	30
2.6	Summary of Adopted Methods	32
3.	<i>WLAN Positioning</i>	33
3.1	Related Work	34
3.1.1	Signals for Indoor Positioning	34
3.1.2	Measurements for WLAN Positioning	35
3.1.3	WLAN Positioning Methods	36
3.1.4	WLAN Fingerprinting	37
3.2	Accuracy Analysis	40
3.2.1	Data	42
3.2.2	Preprocessing for Comparisons of Radio Map Configurations	44
3.2.3	Effect of Bin Configuration	45
3.2.4	Benefit of Direction Information in Radio Map	47
3.2.5	Effect of Combining Measurements from Adjacent Transmitters	51
3.2.6	Normalization of WLAN RSS	54
3.3	Discussion	60
4.	<i>Aiding PDR with Indoor Map and WLAN Positioning</i>	65
4.1	Related Work	66
4.1.1	Motion Sensors and Dead Reckoning	66

4.1.2	Integration of Navigation Systems	67
4.1.3	Pedestrian Indoor Navigation	68
4.2	Methods for Data Processing and Fusion	71
4.2.1	PDR Preprocessing	71
4.2.2	Complementary Extended Kalman Filter	76
4.2.3	Particle Filter for Map Aided Navigation	82
4.2.4	Map Information	87
4.2.5	Distributed Indoor Positioning System	91
4.3	Performance Analysis	92
4.3.1	Data	92
4.3.2	Comparison of Different Combinations of Data Sources	94
4.3.3	Distributed Indoor Positioning System	100
4.3.4	Map Processing and Computation Time	104
4.4	Discussion	107
4.4.1	Results on Aided PDR	108
4.4.2	Future Developments	109
5.	Conclusions	111
5.1	Main Results	111
5.2	Future Work	113
	<i>Bibliography</i>	115

LIST OF FIGURES

2.1	RSS averages from two access points.	9
2.2	Phases of the fingerprinting based positioning: (a) Calibration and (b) Estimation.	10
2.3	Examples of PDF approximations using histograms.	11
2.4	Dead reckoning in two dimensions. Integer $k \geq 1$	15
2.5	Detection of steps from acceleration norm.	18
2.6	Complementary filter in feedforward and feedback configurations.	27
2.7	Complementary filter in embedded reference trajectory (ERT) configuration.	28
3.1	Floor plan of the library and the locations of calibration points (CP) and test points (TP).	42
3.2	Floor plan of the office building and the locations of calibration and test points.	43
3.3	Edges of six histogram bins, evenly and unevenly distributed.	45
3.4	Positioning errors with different bin configurations (number of bins, even and uneven bin widths) computed with different algorithms (ML, MEE) and radio maps (rmap1, rmap2, rmap3).	46
3.5	Effect of direction information in radio maps (rmap1, rmap2, rmap3). Positioning errors with different numbers of bins, ML and MEE algorithms, and uneven and even bin widths.	48

3.6	kNN pattern recognition with sample means: effect of direction information in radio maps on the average error distance (EMEAN), root mean square error (RMSE), and the 95th percentile of positioning error (E95).	49
3.7	Comparison of pattern recognition and probabilistic algorithms: pattern recognition using 1 and 5 nearest neighbors, ML with 2 and 8 bins, and MEE algorithm using 2, 3, and 8 bins, both ML and MEE with uneven bin widths.	50
3.8	Correlation coefficients between the RSS from different MACs. . .	52
3.9	Effect of combining signals from correlating sources: ML and MEE algorithms with bins distributed unevenly (ML1, MEE1) and evenly (ML2, MEE2), and pattern recognition with 2 – 5 NN.	53
3.10	Mobile1 and Mobile2 measurements compared. Unit conversion of Mobile1: $RSS = -RSSI$	56
3.11	Radio map data and independent test measurements collected using the same device.	57
3.12	Positioning using measurements of Mobile1 and radio maps based on different devices.	58
3.13	Effect of the size of the calibration data set to the positioning accuracy of Mobile1, when radio maps were obtained using Mobile1 and Mobile2.	59
4.1	Step length calibration using 10 test data sets to obtain c_{sc} and b . Observations shown with circles.	72
4.2	State diagram of the step detection algorithm. The function outputs the ‘Step detected’ status and the values of $t_{stepStart}$ and $t_{stepEnd}$ when the state is stepEnd or nextStep. The current time is denoted by t_i . The notation is described in Table 4.1.	73
4.3	Block diagram of the signal processing system including the PDR preprocessing of the sensor data and particle filter to perform the PDR state propagation and measurement update using WLAN positioning and indoor map matching.	84

4.4	Choice of the walls to be included in the crossing check: the walls to be checked are indicated by thick lines, other walls shown with thin lines.	89
4.5	System architecture of the distributed positioning system.	91
4.6	WLAN based position estimates shown with <i>triangles</i> . Transitions between consecutive estimates shown with <i>narrow dotted lines</i> , the true track with <i>thick dotted line</i> , and obstacles (rectangles) with <i>light solid lines</i>	95
4.7	Unaided pedestrian dead reckoning.	95
4.8	Complementary EKF using PDR and WLAN data.	96
4.9	Particle filter using PDR and map information.	97
4.10	Particle filter using PDR, WLAN positioning, and map information.	97
4.11	Mapping from the estimated track (red line) to the true track (green thick line) shown by the blue line segments connecting the tracks.	98
4.12	Track of field test #5.1 with control points. PDR: unaided PDR estimate, PF: particle filter, circles: control points.	101
4.13	Track of field test #5.2 with control points. PDR: unaided PDR estimate, PF: particle filter, circles: control points.	101
4.14	Estimation errors at control points. PDR: unaided PDR estimate, PF: average position of particles, 2σ : $2\times$ RMS error estimate based on the coordinate variances of the particles.	102
4.15	Example track of real time demonstration.	103
4.16	Run time of the particle filter processing as a function of the width of the cells in map grid.	105
4.17	Test data used to evaluate the computation times with sequential line re-selection.	106

LIST OF TABLES

3.1	Comparison between 5NN and MEE algorithms	63
4.1	Events that activate the state transitions, threshold parameters, and memory variables of the step detection algorithm described in Fig. 4.2	74
4.2	Data sets for performance evaluation of map aided indoor positioning	92
4.3	Approximated distance errors (m) using different algorithms	99
4.4	Approximated heading errors (degrees) using different algorithms	99
4.5	Distance errors (m) at control points	102
4.6	Comparison between line selection algorithms	106
4.7	Summary of accuracy results from Sections 3.2 and 4.3	107

ABBREVIATIONS

6DOF	6 Degree-Of-Freedom
AGNSS	Assisted GNSS
AM	Amplitude Modulation
AOA	Angle of Arrival
AP	Access Point
API	Application Programming Interface
CDF	Cumulative Distribution Function
CID	Cell Identifier
CP	Calibration Point
DFT	Discrete Fourier Transformation
DR	Dead Reckoning
EKF	Extended Kalman Filter
ERT	Embedded Reference Trajectory
ESA	European Space Agency
FBC	Feedback Configuration
FCC	Federal Communications Commission
FFC	Feedforward Configuration
FFT	Fast Fourier Transformation

FOC	Full Operational Capability
GLONASS	Globalnaya Navigatsionnaya Sputnikovaya Sistema, GNSS operated by the Russian Aerospace Defence Forces
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HLF	Hyperbolic Location Fingerprinting
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR	Infra Red
ISM	Industrial, Scientific and Medical
KF	Kalman Filter
kNN	k -Nearest Neighbors
LED	Light-Emitting Diode
LBS	Location Based Services
LOS	Line-of-Sight
MAC	Medium Access Control
MATLAB	MATrix LABoratory, a programming environment for technical computation and visualization. Registered trademark of the MathWorks, Inc.
MEE	Minimization of Expected (distance) Error
MEMS	Microelectromechanical Systems
ML	Maximum Likelihood
MMSE	Minimum Mean-Square Error
MU	Mobile Unit

NLOS	Non-Line-of-Sight
NN	Nearest Neighbor
NNSS	Nearest Neighbor in Signal Space
PDF	Probability Density Function
PDR	Pedestrian Dead Reckoning
PF	Particle Filter
RBF	Rank Based Fingerprinting
RF	Radio Frequency
RFID	Radio Frequency Identification
RMS	Root Mean Square
RMSE	Root Mean Square Error
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
RTLS	Real-Time Location Systems
RTT	Round Trip Time
SMC	Sequential Monte Carlo
STD	Standard Deviation
SoOP	Signals of Opportunity
TDOA	Time Difference of Arrival
TETRA	Terrestrial Truncated Radio
TOA	Time of Arrival
TP	Test Point
UKF	Unscented Kalman Filter

UWB	Ultra Wide Band
WLAN	Wireless Local Area Network
ZUPT	Zero-Velocity Update

SYMBOLS

\leftarrow	Assignment operator
$a_n(t)$	Acceleration norm at time t
$a_v(t)$	Vertical component of the acceleration at time t
$a_x(t), a_y(t), a_z(t)$	Components of the measured acceleration at time t
b	Constant offset
c_{sc}	Scaling factor
$d(\mathbf{v}_1, \mathbf{v}_2)$	Distance between vectors \mathbf{v}_1 and \mathbf{v}_2
δ_k	Error distance of the k th position sample
$\delta x_k, \delta y_k$	Coordinate errors in the k^{th} sample
$\delta \mathbf{x}_k$	State vector of the CEKF feedback model
$\delta \psi_k$	Error of heading in the k^{th} sample
$\delta(\cdot)$	Dirac delta function
Δs_k	Estimated step length (distance traveled) during the k^{th} step
Δs_k^*	True step length (distance traveled) during the k^{th} step
$\Delta \psi_k$	Estimated heading change during the k^{th} step
$\Delta \psi_k^*$	True heading change during the k^{th} step
Δt_k	Length of the k^{th} step interval
E05	5th percentile of error

E75	75th percentile of error
E90	90th percentile of error
E95	95th percentile of error
EMAX	Maximum error
EMIN	Minimum error
EMEAN	Average error
$E[\cdot]$	Expectation
f_g	Sampling frequency of gyro measurements
$\mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$	State transition function from time instance t_{k-1} to t_k
F_{k-1}	State transition matrix from time instance t_{k-1} to t_k
g	Gravitational acceleration
G_{k-1}	Matrix that relates the deterministic input vector at time t_{k-1} to the state vector at time t_k
$\mathbf{h}_k(\mathbf{x}_k)$	Time variant measurement function at t_k
H	Constant measurement matrix
H_k	Time variant measurement matrix at t_k
I	Identity matrix
k	Index of the step or Number of nearest neighbors
K_k	Time variant Kalman gain at t_k
l	Identifier of the location: index or coordinate pair
\hat{l}	Identifier of the estimated location: index or coordinate pair
\mathcal{L}	A set that contains all the possible locations
n_{AP}	Number of APs in radio map

n_B	Number of bins in histogram
n_{CP}	Number of CPs of radio map
$n_{\Delta s_k}$	Error of the PDR based step length estimate
$n_{\Delta \psi_k}$	Error of the PDR based heading change estimate
n_S	Number of samples
n_{S_l}	Number of measurement vector samples in location l
n_u	Dimension of deterministic input vector
n_x	Dimension of process state vector
n_z	Dimension of measurement vector
N	Number of particles
N_o	Number of obstacle lines
q	Exponent of the signal pattern in step length equation
Q_k	Covariance matrix of process noise w_k
ω_i	Angular rate (i^{th} sample)
P_k	Covariance of the updated state
P_k^-	Covariance of the predicted state
$p_j(k)$	Signal pattern j computed for the k^{th} step
$p(\cdot)$	Probability mass function
$p(\cdot \cdot)$	Conditional probability
$P(\cdot)$	Probability of an event
$\pi(\cdot)$	Proposal distribution
R	Constant covariance matrix of measurement noise v_k
R_k	Time variant covariance matrix of measurement noise v_k

rmap1	Radio map that includes a separate fingerprint entry for each measurement direction
rmap2	Radio map where CP data from all directions are lumped to one fingerprint entry
rmap3	Radio map where CP data from all directions are lumped to one fingerprint entry but uses only one fourth of the all available CP data
RMSE	Root mean square position error
σ^2	Variance of a random variable
$\sigma_{\Delta s}^2$	Variance of the PDR based step length estimates
$\sigma_{\Delta \psi}^2$	Variance of the PDR based heading change estimates
σ_W^2	Variance of the WLAN fingerprinting based coordinate estimates
t	Time
$t_e(k)$	End time of the k^{th} step
t_k	Time of the k^{th} filtering step
$t_s(k)$	Start time of the k^{th} step
\mathbf{u}_k	Deterministic (known) input vector
\mathbf{v}_k	Measurement noise vector
\mathbf{w}_k	Process noise vector
$w_k^{(j)}$	Weight of the j^{th} particle at t_k
x_k	Estimated x coordinate after the k^{th} step
\mathbf{x}_k	The n_x -dimensional process state vector at time t_k
x_k^*	True x coordinate after the k^{th} step
$\hat{\mathbf{x}}_k$	Estimate of the state after the measurement update at t_k

$\mathbf{x}_k^{(j)}$	State of the j^{th} particle at t_k
$\hat{\mathbf{x}}_k^-$	Prediction of the state based on the measurements up to t_{k-1} but without the knowledge about the measurement at t_k
x_{\min}, x_{\max}	Minimum and maximum values of x-coordinates of particles
$x_{m,k}$	The m^{th} element of the vector \mathbf{x} at t_k
$\mathbf{x}_{m:n,k}$	Elements from m to n of the vector \mathbf{x} at t_k
$\hat{\mathbf{x}}_{\text{MMSE}_k}$	MMSE estimate of the state at t_k
y_k	Estimated y coordinate after the k^{th} step
y_k^*	True y coordinate after the k^{th} step
y_{\min}, y_{\max}	Minimum and maximum values of y-coordinates of particles
Ψ_k	Estimated heading angle after the k^{th} step
Ψ_k^*	True heading angle after the k^{th} step
\mathbf{z}	Measurement vector
$\mathbf{z}_{1:k}$	All the measurement vectors up to the time t_k
z_i	The element in the RSS measurement vector corresponding to the i^{th} AP
\mathbf{z}_k	Measurement vector at time t_k
\mathbf{z}_{FP_l}	Pattern vector of location l
$z_{\text{FP}_l,i}$	The element in the pattern vector of location l corresponding to the i^{th} AP
Z_l	Matrix containing the RSS calibration measurements from location l
$z_{i,j}$	Measured RSS value, j^{th} sample from i^{th} AP

1. INTRODUCTION

During the last two decades, after the Global Positioning System (GPS) reached its Full Operational Capability (FOC) (Misra and Enge, 2006), the importance of various kinds of Location Based Services (LBS) relying on positioning and navigation capabilities have increased tremendously. The existing GPS based positioning and navigation applications range from consumer products, such as car navigation or diverse map, geotagging, and navigation applications in smart phones, through professional tools that enable, e.g., cost-effective cartography and accurate positioning of various kinds of vehicles on air, roads, and waterways, to very precise scientific observation systems, such as the systems monitoring the movements of tectonic plates or variations in the earth's rotation.

After GPS, other Global Navigation Satellite Systems (GNSS) have emerged: the FOC of the Russian GLONASS was restored on 2011 (Gibbons, 2012), and GALILEO, the GNSS program of the EU and the ESA has reached its in-orbit-validation capability (Falcone et al., 2013) and is proceeding towards FOC (Inside GNSS, 2014). The Chinese BeiDou Satellite Navigation system already provides navigation services regionally in Asia-Pacific Region, and China is planning to extend it to a global system by expanding its constellation with more satellites in orbits that provide worldwide coverage (Weirong, 2010).

Although the increasing number of GNSS with the growing number of navigation satellites in the sky will improve the availability, coverage, and continuity of the GNSS based positioning service to some extent even in severe urban canyons, the performance will not be sufficient for many navigation applications in urban environments (Ji et al., 2010). Especially in indoor environments, GNSS based positioning signals are known to suffer from serious attenuation or even total blockage: a GNSS receiver has poor visibility to satellites and low signal-to-noise ratio. Together with significant multipath effects on GNSS signals, these make the GNSS based indoor po-

sitioning inaccurate. Therefore, even the increasing multitude of GNSS without any additional aid cannot provide accurate positioning indoors. Due to these effects, even though Assisted GNSS (AGNSS) effectively improves the indoor performance of satellite positioning in the terms of availability and coverage, it is difficult to achieve accuracy better than some tens of meters indoors (Mautz, 2012).

While GNSS and AGNSS provide positioning solution for outdoor environments, the optimal strategy for indoor positioning is still an open issue. However, indoor positioning is currently gaining commercial interest (Lessin, 2013; Panzarino, 2013; Krulwich, 2013; Moyer, 2014; Dennehy, 2014). It is also a public safety issue. Nowadays it is very likely that also the emergency calls made indoors are wireless calls from mobile handsets, which are not associated with one fixed location or address. In USA, the Federal Communications Commission (FCC) has proposed to update E911 regulations to require carriers to be able to locate emergency calls that are made indoors (Partyka, 2014). Therefore, alternatives to GNSS are needed for indoor positioning.

1.1 Scope and Objective of the Research

This thesis is concerned with positioning and navigation methods required for pedestrian indoor positioning. The scope of the thesis is on the methods and algorithms that do not assume GNSS based positioning to be available indoors, and on methods that do not require any dedicated positioning infrastructure to be installed indoors. Moreover, the scope is restricted to mobile applications, where the measurements required for positioning are carried out solely by the mobile unit; e.g., collaborative positioning where several mobile units share their measurements, or network based methods where infrastructure elements perform the measurements, are not considered in this thesis. The main focus is on positioning based on WLAN fingerprints, Dead Reckoning (DR) sensors and their signal processing, and the fusion of the measurements available from these sources with indoor map information.

The objective of this thesis is to find practical models for systems and algorithms needed for personal indoor navigation, to provide workable solutions to cope with imperfections of measurement devices, and to develop methods to combine position related information from several sources that have complementary strengths and weaknesses.

Among the DR based navigation approaches, this thesis concentrates only on pedestrian dead reckoning, where the distance traveled is estimated using step detection and without double integration of the acceleration. The availability of sensor based dead reckoning estimates in the map aided indoor navigation is assumed, which allows the complementary filter formulation of the data fusion algorithms. The benefit of this is to avoid the modeling of the user movement as a random process, and the only needed assumptions regarding the user movement are the pedestrian mode of motion and the inability of the pedestrian to walk through walls.

The thesis concentrates on the following questions: (1) how to construct the WLAN radio map model that provides accurate positioning results efficiently with respect to computational load and required memory, (2) how to cope with the fact that different mobile devices measure the RSS differently, (3) how to efficiently combine information from dead reckoning sensors, indoor map, and WLAN for pedestrian indoor positioning, and (4) finding suitable sensor error models to be used in the data fusion.

1.2 Main Contributions

The main contributions of the thesis can be summed up as follows:

- Novel methods and models for accurate indoor positioning using pedestrian dead reckoning obtained using sensors based on microelectromechanical systems and fused together with WLAN signals and indoor map
- Novel methods to decrease the computational load of map checks in particle filters
- Novel methods to combine the information sources using complementary filter structure in extended Kalman filter (EKF) and particle filter
- Novel histogram configurations of radio maps for WLAN fingerprinting
- Normalization method for measurements of received WLAN signal strength to allow different devices to be used for position estimation

1.3 Author's Contribution

The contributions of the author include design and implementation of the algorithms for pedestrian indoor navigation and positioning, design of measurement campaigns for sensor and signal modeling, design of field tests to verify the performance of the proposed methods, and analysis and reporting of the results. In addition to this thesis, the author's contributions are reported in (Leppäkoski et al., 2013, 2012, 2010, 2009a,b; Leppäkoski and Takala, 2007), where the author was the main author of the publications, and in (Perttula et al., 2009, 2014), to the contents of which the author had a significant contribution.

The author introduced the Complementary Extended Kalman Filter (CEKF) used to combine the PDR with WLAN positioning for indoor navigation as well as the CEKF based process model used in the particle filter for data fusion of PDR and WLAN positioning with indoor map information. The author carried out the implementation together with the tests and analyses for the performance verification of these algorithms (Leppäkoski et al., 2013, 2012) and the study on the complementary Kalman filter configurations (Leppäkoski and Takala, 2007). The design and MATLAB implementation of PF and map matching algorithms as well as the design of the PDR preprocessing algorithm were carried out by the author, who also took part into the planning and implementation of the field tests and analysis of the test results (Perttula et al., 2014). While these publications include results on decreasing the computational load of the map processing for unstructured obstacle maps, this thesis includes also results on speeding up the map computations required for structured obstacle map.

In the results reported in (Leppäkoski et al., 2009a,b; Perttula et al., 2009; Leppäkoski et al., 2010), algorithms to generate the WLAN radio maps and to perform the fingerprint based positioning were needed in order to analyze the accuracy of the developed methods. The design and MATLAB implementation of these algorithms were carried out by the author. The modifications of the histogram configurations in the radio maps for WLAN fingerprinting were proposed and their performance evaluation was carried out by the author (Leppäkoski et al., 2009a, 2010). For the normalization of WLAN signal strength measurements the author proposed a novel approach and took part in the planning of the measurement campaign and the analysis of the results (Leppäkoski et al., 2009b). In the study on the effect of the grid point density of the

radio maps to the accuracy of WLAN positioning, the author took part in the design of the measurement campaign and the analysis of the measurements (Perttula et al., 2009). This thesis includes unpublished results regarding the kNN method and its comparisons with ML and MEE algorithms.

1.4 Thesis Outline

Chapter 2 presents the known positioning and navigation methods and algorithms that are used as the basis to develop the indoor positioning algorithms presented in this thesis. These preliminaries include: WLAN fingerprinting using radio maps based on pattern matching and probabilistic models; pedestrian dead reckoning; utilization of indoor maps; data fusion based on Bayesian filtering, including Kalman and extended Kalman filtering, complementary filtering, and particle filtering; and assessment of positioning accuracy.

Chapter 3 addresses WLAN based positioning using WLAN fingerprints. It reviews the published results related to positioning based on WLAN and fingerprinting, describes in detail the data and test method used to analyze WLAN fingerprinting models. The results on experiments regarding the bin configuration of the radio map, the effect of direction information in the radio map, the effect of combining correlating WLAN sources, and the normalization among different WLAN devices are presented. Finally the results of accuracy analysis are discussed.

Chapter 4 starts with the state-of-the-art presentation on map aided indoor navigation including the review on motion sensors and dead reckoning, integration of navigation systems, and pedestrian dead reckoning. This is followed by the description of the methods used in this thesis to utilize the map information and for fusing map with PDR estimates and WLAN positioning. Performance analysis based on field tests is given, including comparison between unaided WLAN positioning, unaided PDR, PDR aided by WLAN positioning, PDR aided with indoor map, and PDR aided with both the indoor map and the WLAN positioning. Finally the results on the speeding up the computations required for on-line processing of map information are given, followed by the discussion of the results.

Chapter 5 concludes the thesis. It summarizes the main results of the thesis and points out directions for future research.

2. PRELIMINARIES

This Chapter presents well known positioning and navigation methods. First a review on the models and algorithms suitable for positioning based on WLAN fingerprinting is given, followed by the description of methods for pedestrian dead reckoning based on inertial sensors. Then the usage of map information for indoor positioning is reviewed. Finally, the common Bayesian filtering algorithms including Kalman Filter, Extended Kalman Filter, and particle filter, used in this thesis for data fusion, are summarized.

2.1 *WLAN Fingerprinting*

WLAN positioning methods based on Received Signal Strength (RSS) measurements have been reported on numerous papers, e.g., (Bahl and Padmanabhan, 2000; Battiti et al., 2002; Prasithsangaree et al., 2002; Roos et al., 2002; Smailagic and Kogan, 2002; Wallbaum and Wasch, 2004). To obtain the WLAN RSS measurement, a Mobile Unit (MU) scans the available WLAN channels. Infrastructure Access Points (AP) periodically emit beacon frames, which include their Medium Access Control (MAC) addresses, i.e., their unique identifiers (IEEE, 2012). Using the beacon frames, the MU associates its RSS measurements to the MAC addresses of the listened APs before it carries out the position estimation. As the MUs can obtain the RSS measurements by passive listening, without connecting to the APs, the number of MUs that one AP can serve, is not restricted.

Fingerprinting approaches use mathematical models to relate the measured RSS levels from different APs directly to the location of the MU (Bahl and Padmanabhan, 2000); these models are called radio maps (Youssef et al., 2003; Wallbaum and Diepolder, 2005). In this way, the fingerprinting actually makes use of the location dependent characteristics of radio signal propagation.

Often the fingerprinting approaches use radio maps that are defined experimentally (Bahl and Padmanabhan, 2000; Roos et al., 2002; Smailagic and Kogan, 2002; Castro et al., 2001; Youssef et al., 2003), but also semi-analytical approaches are proposed (Bahl and Padmanabhan, 2000; Wallbaum and Wasch, 2004). In semi-analytical method the radio map is computed from the wall information obtained from the floor plan and signal propagation model with attenuation terms for obstacles. The attenuation properties of the materials are taken into account by using experimentally determined wall attenuation factors (Bahl and Padmanabhan, 2000; Wallbaum and Wasch, 2004). However, the experimental radio map provides better accuracy; for example, Bahl and Padmanabhan (2000) reported average positioning errors 4.3 m and 2.9 m for path loss based and experimental fingerprint models, respectively.

The experimental radio maps are based on off-line collected data from several locations that sufficiently cover the area where positioning service is to be performed. The procedure for radio map generation is often called calibration or training, referring to calibration or training of the experimental model; the required data is called calibration or training data, respectively. The locations where the calibration data is collected are called calibration points (CP). In estimation phase, new measurement vectors are compared with the information stored in the radio map to infer the MU position.

Compared with other RSS-based methods, fingerprinting algorithms are considered more robust against the signal propagation fluctuations generated by environment characteristics. This is because fingerprinting makes use of the location-dependent variation of RSS by assuming that the environments have unique signal propagation characteristics and each location can be associated with a unique tuple of RSS values (Hoshen, 2000; Wallbaum and Diepolder, 2005). As an example, averages of RSS from two APs are shown in Fig. 2.1. The RSSs were measured in the library of Tampere University of Technology in the locations shown with squares. It is clear that the RSS does not decrease smoothly as the distance grows. The known disadvantages of the fingerprinting approaches include the following: the collection of calibration data is laborious and time consuming; the fingerprint databases get outdated due to environment changes such as the relocations of furniture or APs and therefore regular updating is needed; the environmental dynamics including the varying numbers and positions of people should be similar in both the calibration and the estimation phases.

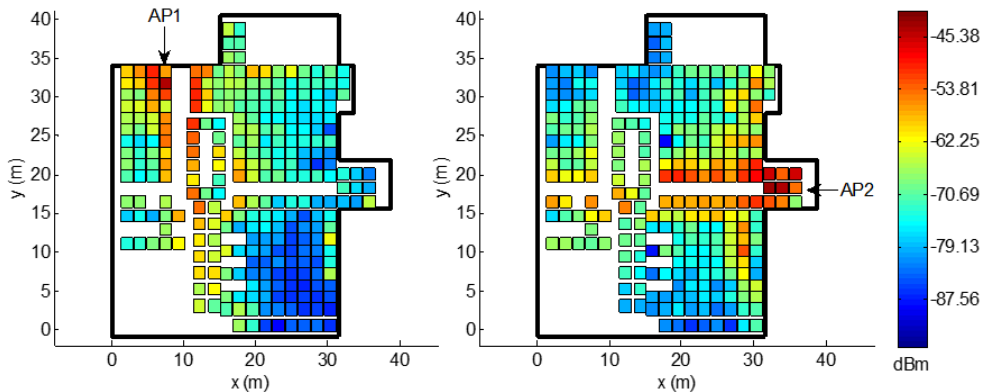


Fig. 2.1. RSS averages from two access points.

Location fingerprinting has also been proposed using other signals than WLAN RSS. For example, Bluetooth positioning based on RSSI fingerprints was studied by Pei et al. (2010). Moghtadaiee et al. (2011) also proposed fingerprinting approach based on FM radio signals. The potential of fingerprints based on magnetic field was studied by Storms and Raquet (2009) and Li et al. (2012); IndoorAtlas offers free mobile phone applications to create magnetic indoor map and to use it for navigation (IndoorAtlas).

2.1.1 Radio Map

WLAN fingerprinting consists of two phases: in the calibration phase, off-line-collected RSS data is used to generate a radio map, and in the estimation phase new RSS measurement vectors are related with the information stored in the radio map. This process is illustrated in Fig. 2.2. For each CP, the radio map contains the known coordinates (or other suitable location identifier) of the CP together with RSS features extracted from RSS measurements collected in the CP. The CP coordinates together with the extracted features is called a fingerprint.

The RSS features reported in the literature to have been used for positioning with WLAN fingerprints include, for example, the sample mean of RSS measurements (Bahl and Padmanabhan, 2000; Prasithsangaree et al., 2002; Leppäkoski et al., 2009a) and approximations of the Probability Density Functions (PDF) of the RSS samples (Roos et al., 2002; Youssef et al., 2003; Perttula et al., 2009; Leppäkoski et al., 2009a,

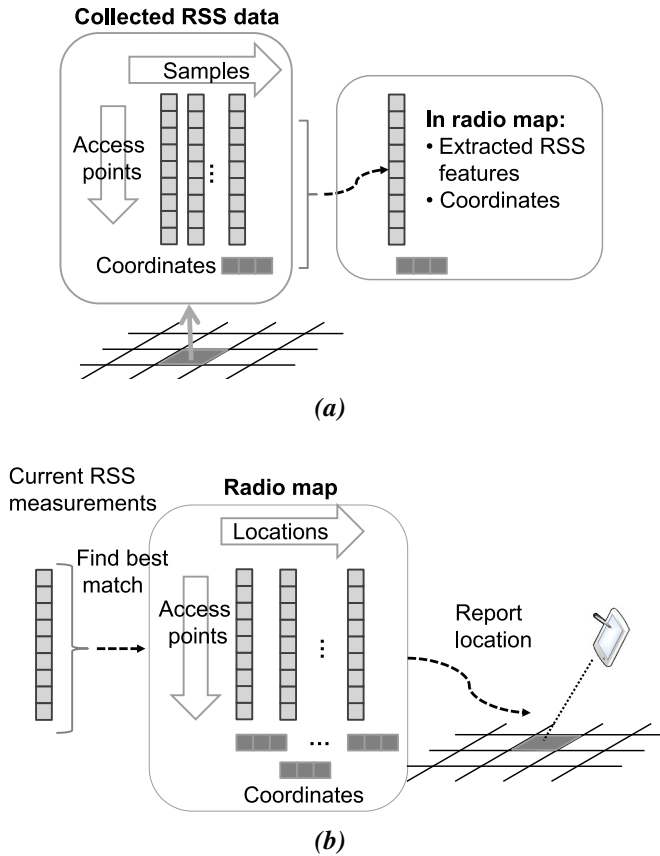


Fig. 2.2. Phases of the fingerprinting based positioning: (a) Calibration and (b) Estimation.

2010). In each CP, the sample means or PDFs are computed separately for each AP. Location fingerprinting methods can be categorized in pattern matching and probabilistic algorithms. The fingerprint patterns stored into radio map could be individual measured RSS vectors, but more commonly the information of several measured RSS vectors are summarized to a pattern vector - sample mean of several RSS vectors is used, e.g., in (Bahl and Padmanabhan, 2000; Prasithsangaree et al., 2002). A more complex radio map model with perhaps less intuitive interpretation can be obtained using neural network (Battiti et al., 2002).

With probabilistic algorithms, the information of calibration data is summarized to PDFs. For each CP, the radio map contains the PDF of the measured RSS for each AP. The PDFs can be approximated using, e.g., kernel functions (Roos et al., 2002) or histograms (Castro et al., 2001; Roos et al., 2002; Youssef et al., 2003). The histogram approximates the PDF using a piecewise constant function, where the range

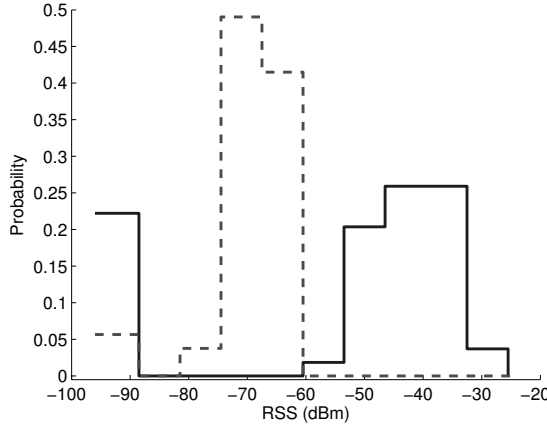


Fig. 2.3. Examples of PDF approximations using histograms.

of the random variable is divided into non-overlapping bins; an example is shown in Fig. 2.3. If the number of bins is small enough, the memory requirement with histograms is significantly lower compared to kernel based approximations. Even more compressed models have been proposed, where the PDF is approximated using parametric functions (Honkavirta et al., 2009; Koski et al., 2010a,b). In the system design, there is a trade-off between the accuracy and the memory compression.

With histogram based PDFs, position estimation algorithms using Maximum Likelihood (ML) principle (Castro et al., 2001; Youssef et al., 2003), or Minimization of Expected (distance) Error (MEE) (Roos et al., 2002) are proposed. The granularity of estimates using ML algorithm is determined by the density of calibration point grid, whereas the estimate by MEE algorithm can interpolate between CP locations.

2.1.2 Pattern Matching

For pattern matching, the calibration measurements from the CPs are collected into vectors. The vector elements are the RSS measurements associated with different MAC addresses, so that the order of MAC addresses is fixed. In calibration phase, the RSS values $z_{i,j}$ measured in location l are collected into a matrix

$$Z_l = \begin{bmatrix} z_{1,1} & \cdots & z_{1,n_{S_l}} \\ \vdots & \ddots & \vdots \\ z_{n_{AP},1} & \cdots & z_{n_{AP},n_{S_l}} \end{bmatrix} \quad (2.1)$$

where n_{S_l} is the number of measurement vector samples in location l and n_{AP} is the number of APs. The location variable l represents discrete locations. It may be expressed either by coordinates, index, or some other identifier. The elements of the pattern vectors $\mathbf{z}_{FP_l} = [z_{l,1}, \dots, z_{l,n_{AP}}]^T$ for locations $l = 1, \dots, n_{CP}$ are obtained by taking averages of the rows of Z_l :

$$z_{FP_{l,i}} = \frac{1}{n_{S_l}} \sum_{j=1}^{n_{S_l}} z_{i,j}. \quad (2.2)$$

Thus the size of the radio map is $n_{CP} \cdot n_{AP}$ elements. In estimation phase, new measurement vectors $\mathbf{z} = [z_1, \dots, z_{n_{AP}}]^T$ are compared with pattern vectors \mathbf{z}_{FP_l} stored in the radio map. The comparison is based on their distances in signal space. Prasithsangaree et al. (2002) proposed the use of generalized weighted distance L_p for the comparison. One common choice for the distance measure is L_2 with equal weights, i.e., the normal Euclidean distance (Bahl and Padmanabhan, 2000):

$$d(\mathbf{z}, \mathbf{z}_{FP_l}) = \sqrt{\sum_{i=1}^{n_{AP}} (z_i - z_{FP_{l,i}})^2}. \quad (2.3)$$

Using these, the position can be estimated by finding the nearest neighbor (NN) in signal space, i.e., the pattern vector, which minimizes the distance:

$$\hat{l} = \arg \min_l (d(\mathbf{z}, \mathbf{z}_{FP_l})). \quad (2.4)$$

From this algorithm there exists a version, where instead of the best matching pattern vector, 2 or more best matching patterns, i.e., k -nearest neighbors (kNN), are taken into account, and position estimate is computed as the average of their positions. Prasithsangaree et al. (2002) reported that increasing k in the kNN method improves the accuracy of the algorithm at least to $k = 3$, Bahl and Padmanabhan (2000) observed the same and that the accuracy starts to deteriorate when $k > 4$.

2.1.3 Probabilistic Model

In histogram based approximation of PDF, the signal range is divided into n_B bins when the continuous or fine-resolution discrete RSS scale becomes discrete scale with coarse resolution. The value n_B is a design parameter of the algorithm. It has an effect to the obtainable positioning accuracy, memory requirement of the radio map,

and computational load of the position estimation. If the bin configuration, i.e., the bin edges in RSS scale, is the same for all APs and CP locations, the radio map with n_B bins requires memory for $n_{CP} \cdot n_{AP} \cdot n_B$ elements.

A histogram based radio map stores marginal distributions $p(\mathbf{z}|l)$ for each CP location l , i.e., the conditional probabilities that the measured RSS vector \mathbf{z} can be observed at location l . In estimation phase, the conditional probabilities are employed to calculate the posterior probability $p(l|\mathbf{z})$, i.e. the probability of being located at l , given the measured RSS values \mathbf{z} .

The posterior probabilities of the locations can be computed using Bayes' Theorem (Roos et al., 2002):

$$p(l|\mathbf{z}) = \frac{p(\mathbf{z}|l)p(l)}{p(\mathbf{z})} = \frac{p(\mathbf{z}|l)p(l)}{\sum_{l' \in \mathcal{L}} p(\mathbf{z}|l')p(l')} \quad (2.5)$$

where $p(l)$ is the prior probability of being at location l . For snap-shot type of positioning, where prior information of the location is not available or not used, a non-informative uniform distribution is used. The set \mathcal{L} contains all the possible locations and $p(\mathbf{z})$ is the probability of the measurement vector \mathbf{z} over all locations; $p(\mathbf{z})$ does not depend on location and can be treated as a normalizing constant.

The *Maximum Likelihood (ML) estimate* is obtained by finding the location l that maximizes the likelihood $p(\mathbf{z}|l)$:

$$\hat{l} = \arg \max_l p(\mathbf{z}|l). \quad (2.6)$$

Assuming independence of observations z_i , i.e., $P(Z_1 = z_1, \dots, Z_{n_{AP}} = z_{n_{AP}}) = \prod_{i=1}^{n_{AP}} P(Z_i = z_i)$, where $P(\cdot)$ is the probability of an event, the ML estimate (2.6) can be computed using the following (Youssef et al., 2003):

$$\hat{l} = \arg \max_l \left(\prod_{i=1}^{n_{AP}} p(z_i|l) \right). \quad (2.7)$$

As $p(\mathbf{z})$ does not depend on l , and if the non-informative prior (equal $p(l)$ for all l) is assumed, then based on (2.5), the ML estimate (2.6) maximizes also the posterior probability $p(l|\mathbf{z})$.

If the location variable is numerical, which is the case when the location variable consists of coordinates, the *Minimization of the Expected squared location Error*

(MEE) is obtained with MEE estimate (Roos et al., 2002):

$$\hat{l} = E[l|\mathbf{z}] = \sum_{l' \in \mathcal{L}} l' p(l'|\mathbf{z}), \quad (2.8)$$

where $E[\cdot|\cdot]$ is the conditional expectation. Assuming equal $p(l)$ for all l and independence of the observations z_i , the posterior probabilities $p(l|\mathbf{z})$ in (2.8) can be computed using Bayes' theorem:

$$p(l|\mathbf{z}) = \frac{\prod_{i=1}^{n_{\text{AP}}} p(z_i|l)}{\sum_{l' \in \mathcal{L}} (\prod_{i=1}^{n_{\text{AP}}} p(z_i|l'))}. \quad (2.9)$$

Compared to the ML algorithm, this algorithm has the advantage that it allows the estimate to interpolate between the CPs. Its disadvantage is that the nominator in (2.9) needs to be evaluated, which increases the computational load in the estimation phase.

2.2 Pedestrian Dead Reckoning

Dead reckoning (DR) is a navigation technique where the position is estimated relative to the starting point by keeping track of the direction and distance traveled on each section of the route, i.e., a new position estimate is computed by adding a newly measured displacement to the previous position estimate (Misra and Enge, 2006). The principle is illustrated in Fig. 2.4. Inertial sensors are well known devices for providing the information on the direction and the distance traveled (Titterton and Weston, 2004; Groves, 2008).

The process of keeping track of position and direction of travel using inertial sensors is called mechanization. A 6 Degree-Of-Freedom (6DOF) Inertial Measurement Unit (IMU) consists of a tri-axial accelerometer for measuring the specific force in three dimensions and a tri-axial gyro for measuring the angular rates in three dimensions. The accelerometer is unable to observe the acceleration caused by the gravitational field but it observes the supporting force working against the gravitational field. However, it cannot distinguish between this supporting force and the total acceleration in space. It provides measurements of the difference between the acceleration in space and the gravitational acceleration; this quantity is called specific force. The traditional Inertial Navigation System (INS) mechanization includes the following tasks: 1) integration of the outputs of gyros to obtain the attitude of the system in the desired

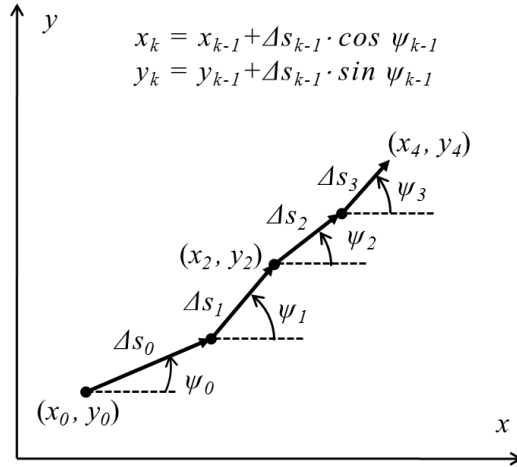


Fig. 2.4. Dead reckoning in two dimensions. Integer $k \geq 1$.

coordinate reference frame, 2) using the obtained attitude of the system, transformation of the specific force measurements to the chosen reference frame, 3) computing the local gravity in the chosen reference frame and adding it to the specific force to obtain the device acceleration in space, 4) if required by the chosen reference frame, the Coriolis correction is applied, 5) integration of the acceleration to obtain the velocity and the position of the device (Titterton and Weston, 2004).

In the double-integration of accelerations, even a small error in acceleration measurement yields a large position error drift in the output. Moreover, because the gravity compensation of accelerations requires the coordinate transformation, any error in gyroscope output causes errors in the transformed accelerations. As the coordinate transformation is obtained by integrating the gyro outputs, the gyro errors produce a position error which increases with time cubed. Therefore the gyro performance is very critical in INS implementations. IMUs with high performance gyros are available at high price. For example, the price of a tactical grade IMU varies from 5,000 to 20,000 USD, and it can be used stand-alone for inertial navigation for only a few minutes. A position solution accurate for one hour can be obtained by using a marine grade IMU, which costs about one million USD (Misra and Enge, 2006).

As the accurate INS mechanization requires very high-quality and expensive sensor units, the developers of mass-market applications are looking for solutions where multiple integration of sensor errors can be avoided. Pedestrian Dead Reckoning (PDR) is one way to reduce the effects of accelerated accumulation of sensor errors.

Two different approaches can be found in inertial sensor-based PDRs: systems with foot-mounted sensors, and systems with sensors placed elsewhere, for example sensors attached on waist or torso. In both approaches, the heading is estimated by integrating angular velocity measured with gyroscopes. However, the distances are estimated using different techniques. With shoe mounted sensors the traditional INS mechanization with Zero-Velocity Update (ZUPT) can be used (Elwell, 1999; Foxlin, 2005). In the ZUPT, the algorithm uses the stance phase between the steps, when the foot stays stable for a short time, to reset the inertial errors. When sensors are not placed on foot, the stance phase seen by the sensor is shorter and the estimation of the inertial errors becomes less accurate. In these cases, instead of integration of the accelerations, the displacement or the speed of the walk is estimated from the periodical acceleration waveform produced by pedestrian movements. The speed can be estimated either from the main frequency of the periodic signal or by detecting individual steps and estimating their lengths and durations from the acceleration waveform. This information along with the estimated heading is used to propagate the estimate of user position. It can be shown that PDR mechanization is superior to the traditional INS mechanization for a person on foot (Mezentsev et al., 2005). The main drawback of PDR is its limitation to one motion mode; the mechanization works only when walking, while traditional INS works without any assumptions about the user motion. In this thesis, only PDR systems using torso mounted sensors are considered in details.

In PDR using torso mounted sensors, the estimation of the distance traveled usually consists of two tasks. In step detection, the accelerometer signal is analyzed to find the instances where a new step starts and where it ends. In step length estimation the distance traveled during the time interval between the start and end of the step is estimated. In this thesis in the context of PDR, the step is defined as the displacement of feet between two consecutive foot prints.

2.2.1 Step Detection

The occurrence of a step can be easily detected from the signal pattern of the vertical acceleration component (Levi and Judd, 1996; Meriheinä, 2007). However, this approach is sensitive to the orientation errors of the sensor unit, as it assumes that one axis is aligned with vertical or that transformation to vertical is known. Other

possibility it to compute the magnitude of the measured acceleration vector, i.e. the norm of acceleration (Käppi et al., 2001).

The signal pattern varies according to where the user attaches the sensor unit (Ladetto, 2000). Typical choices to wear the sensor unit are on the belt, e.g., on the side of the user or on lower back, or onto upper parts of the torso, e.g., attached to the shoulder strap of a backpack or placing it in a chest pocket. Also with a shoe mounted sensor unit the estimation of the distance traveled can be based on the analysis of the acceleration waveform (Stirling et al., 2003).

Step detection is often based on the detection of signal peaks (Ladetto, 2000) or crossings of the signal with its average (Käppi et al., 2001) or some other reference level (Meriheinä, 2007). For example, step detection from acceleration norm, without requiring knowledge of the orientation of the 3D-accelerometer or its alignment with the vertical, consists of the following phases:

1. Low pass filtering and resampling the signal; sampling frequency in the range 20–25 Hz is high enough.
2. Computation of the Euclidean norm of the current acceleration sample, i.e.,

$$a_n(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}, \quad (2.10)$$

where $a_n(t)$ is the acceleration norm and $a_x(t)$, $a_y(t)$, and $a_z(t)$ are the low-pass filtered components of the measured acceleration. Within the sensor fabrication tolerances, the measured components are orthogonal to each other but the orientation of the components with respect to the vertical is not fixed.

3. The step starts $t_s(k)$ are detected by observing the g-crossings, i.e., the instances when the acceleration norm crosses the acceleration caused by gravity. It can also be required that the g-crossing is followed by a rise rate and a peak height that exceed the preset limits and that the time between the current and previous g-crossings is long enough.
4. The step end $t_e(k)$ is considered to be found when the next step starts or when a predefined time, considered as the maximum duration of one step, has passed after the start of the current step.

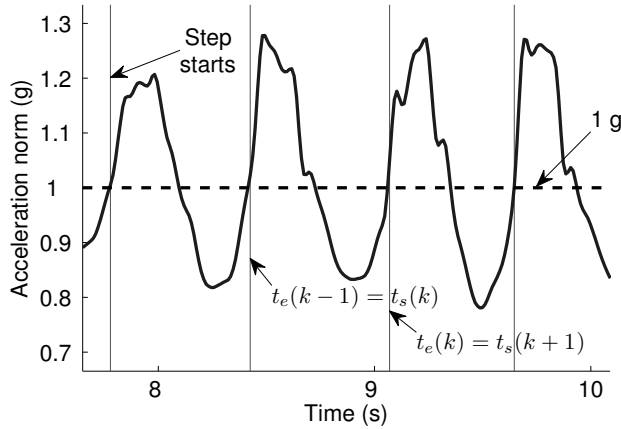


Fig. 2.5. Detection of steps from acceleration norm.

An example with acceleration norm and the detected step starts is shown in Fig. 2.5. The data for the figure was recorded using a sensor unit that was attached to the belt and positioned to the back of the test walker.

2.2.2 Step Length Estimation

There are two main categories for methods to estimate step length. The first category includes models that are based on the biomechanical principles while the models in the second category are based on empirical relationships between acceleration signal pattern and step length. With the biomechanical models certain user related parameters, such as leg length are needed in addition to the empirically determined scaling parameters (Jahn et al., 2010). In empirical models, acceleration norm $a_n(t)$ or vertical acceleration component $a_v(t)$ are typically used for step length estimation. The signal patterns that have good correlation with step length include the following:

- main frequency of the periodical signal (Levi and Judd, 1996; Ladetto, 2000) which can be obtained using FFT or some other DFT methods or by computing the inverse of step duration determined by step detection algorithm

$$p_1(k) = 1 / (t_e(k) - t_s(k)) \quad (2.11)$$

- variance of the acceleration signal over a time window comparable to some

step durations (Ladetto, 2000), e.g. variance over one step

$$p_2(k) = \text{var}(a_n(t)), \quad t_s(k) \leq t < t_e(k) \quad (2.12)$$

$$p_3(k) = \text{var}(a_v(t)), \quad t_s(k) \leq t < t_e(k) \quad (2.13)$$

- integral of the absolute value acceleration norm where local gravity has been subtracted, integrated over the duration of the step (Käppi et al., 2001)

$$p_4(k) = \int_{t_s(k)}^{t_e(k)} |a_n(t) - g| dt \quad (2.14)$$

- difference between the maximum and minimum acceleration of a detected step (Jahn et al., 2010)

$$p_5(k) = \max a_n(t) - \min a_n(t), \quad t_s(k) \leq t < t_e(k) \quad (2.15)$$

$$p_6(k) = \max a_v(t) - \min a_v(t), \quad t_s(k) \leq t < t_e(k) \quad (2.16)$$

Also the use of combinations of these signal patterns has been proposed (Ladetto, 2000; Jahn et al., 2010), as well as slightly different patterns from these (Meriheinä, 2007). The step length model often includes at least one empirically determined parameter. In many cases a non-linear function, such as raising to a power or extraction of root, has to be applied to the signal pattern. A general model for step length is

$$\Delta s_k = c_{sc} p_j(k)^q + b \quad (2.17)$$

where Δs_k is the distance traveled during the k^{th} step, p_j is the signal pattern computed for the step, b is the constant offset and c_{sc} is the scaling factor that relates Δs_k with $p_j(k)^q$, q is the exponent to be applied on the signal pattern, and the subscript j refers to the signal pattern type, e.g., one of the patterns given in (2.11) – (2.16). An example comparing these signal patterns in step length estimation is presented in Collin et al. (2013).

2.3 Indoor Map

The use of map information is a common practice in car navigation and similar principles have also been proposed for indoor positioning. However, the use of an indoor

map for pedestrian navigation differs from the way the street maps are used in car navigation. The characteristic dimensions are different in indoor and outdoor maps: the roads in outdoor environment are characterized by small width and large length while the width-length ratio of the rooms inside the buildings is typically much larger (Glanzer et al., 2009). In car navigation, the roads represent the possible locations of the car, and the task of the positioning algorithm is to use some clever method to force the position estimate to the most probable road segment (Quddus et al., 2007). In indoor navigation, instead of defining possible routes the indoor map gives information about impossible movements: the positioning algorithm uses information about walls and obstacles that the pedestrian cannot walk through.

Two main approaches have been used to present the indoor map or floor plan information in pedestrian navigation algorithms. The first one is called here obstacle model, which defines the locations of walls and obstacles, which are used to constrain the position of the MU (Wang et al., 2007; Widyawan et al., 2008; Beauregard et al., 2008; Woodman and Harle, 2008). The second approach transforms the indoor map information to a form similar to the representation of road network; the information about the walls and obstacles is used to compute a node-link model (Gilliéron et al., 2004; Evennou et al., 2005).

The obstacle model can be unstructured or structured. An unstructured obstacle model includes just a list of the walls and obstacles that are presented by line segments defined by the coordinates of their starting and ending nodes; an example can be found e.g. in (Leppäkoski et al., 2012). Evennou et al. (2005) presents an algorithm, where instead of the line segments, the obstacles are defined using pixels.

In a structured obstacle model, the walls are grouped as rooms: a room is a closed polygon, where polygon edges represent both the walls and passages such as doors and staircases. Each of the polygon lines is defined to be an obstacle or a passage (Woodman and Harle, 2008). The structured obstacle model has benefits: if the room where the transition of the MU begins is known, it is possible to decrease the number of walls to be processed when checking whether the transition is possible or not. With a structured model it is also possible to associate alphanumeric data with the lines or the polygons representing rooms, staircases, or doors, which makes it easier to present the positioning information to the user in a relevant form; for example, the user may be more interested in the room number of her location than the geographic coordinates of the room.

However, the wall check algorithm is more complicated when the structured obstacle model is used: instead of simply checking the crossings of the MU transition with all the possible walls, this algorithm has to proceed the checks in a conditional sequence. The checking starts from the polygon lines of the room including the initial location. If polygon lines are not crossed in the transition of the MU, the map constraints are not violated and the MU must be still located in the same room. If a polygon line defining an obstacle is crossed, the transition is declared impossible. If a polygon line defining a passage is crossed, the transition is possible and the MU has moved to the other room. Now also the polygon lines of the new room need to be checked; the sequence of obstacle and passage line checks has to be continued until no obstacle or passage lines are crossed (the room where the transition ends has been found) or an obstacle line is crossed (indicating an impossible movement) (Woodman and Harle, 2008).

In a node-link model, the links are the route segments where the MU can move only along the segment, turn around, or stop. A link begins from a node and ends to another node. The junctions and intersections, such as doors, are defined as nodes: if there is more than one link ending to the same node, the node offers to the MU a possibility to change from one link to another. The elevators and staircases are presented as vertical links. In the model, the coordinates of the nodes need to be defined, and for the links the properties such as horizontal or vertical need to be defined. The node-link model can be obtained e.g. by computing a Voronoi diagram, i.e., a set of edges that are equidistant to all the walls (Evennou et al., 2005).

2.4 Data Fusion Using Bayesian Filtering

Usually there is no single source of data that alone could provide a perfect and complete knowledge about the problem of interest; rather the information must be obtained indirectly from sources of imperfect and incomplete data. The goal of data fusion is to combine or integrate data from various sources to provide improved description of the process of interest (Durrant-Whyte, 2001). Estimation problems in navigation and positioning belong to this category, where data fusion is usually needed to obtain the required positioning accuracy.

In data fusion, the modeling of the uncertainties of both the process and the measurements are necessary. The framework of Bayesian filtering, building on the probability

theory, provides an extensive set of methods for describing and manipulating uncertainty. The following subsections present Kalman filter, some of its variants, and particle filter, which in this thesis are used to perform the Bayesian filtering task.

2.4.1 Kalman Filter

Since the invention of Kalman Filter (KF) in 1960 (Kalman, 1960), the KF and algorithms derived from it have been in wide use in the field of navigation. The discrete-time KF provides means to estimate dynamic, time-varying quantities that can be presented using discrete time state space equations.

The KF and its derivation is presented in many text books, e.g., in Sorenson (1966); Anderson and Moore (1979); Maybeck (1979); Mendel (1995); Brown and Hwang (1997); Bar-Shalom and Li (1998); Kailath et al. (2000); Grewal and Andrews (2001). In this thesis, the following formulation and assumptions are followed: the state equation describing the dynamics of a linear system is

$$\mathbf{x}_k = F_{k-1}\mathbf{x}_{k-1} + G_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad k = 1, \dots \quad (2.18)$$

where \mathbf{x}_k is the n_x -dimensional process state vector at time t_k , \mathbf{u}_k is the n_u -dimensional known input vector, F_{k-1} is the $n_x \times n_x$ -dimensional state transition matrix from time instance t_{k-1} to t_k , and G_{k-1} is the $n_x \times n_u$ -dimensional matrix that relates the deterministic inputs at time t_{k-1} to the state vector at time t_k . The n_x -dimensional process noise vector \mathbf{w}_k is assumed to be a white sequence of random variables, i.e., that are zero-mean and uncorrelated timewise. The covariance matrix of \mathbf{w}_k , denoted by Q_k , is assumed to be known, thus

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k. \end{cases} \quad (2.19)$$

The measurement (or observation) equation is

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k, \quad k = 1, \dots \quad (2.20)$$

where \mathbf{z}_k is the n_z -dimensional measurement vector at time t_k , H_k is the $n_x \times n_z$ -dimensional measurement matrix, and \mathbf{v}_k is the n_z -dimensional measurement noise vector, which is assumed to be white sequence with known $n_z \times n_z$ -dimensional covariance R_k and to have zero cross-covariance with \mathbf{w}_k , i.e.,

$$E[\mathbf{v}_k \mathbf{v}_i^T] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \quad (2.21)$$

$$E [\mathbf{w}_k \mathbf{v}_i^T] = 0, \quad \text{for all } k \text{ and } i. \quad (2.22)$$

In equations (2.18)–(2.21) the matrices F_k , G_k , Q_k , H_k , and R_k are assumed to be known and they can be time varying, i.e., the equations are able to describe time varying system with nonstationary noises (Bar-Shalom and Li, 1998). Often the term $G_{k-1} \mathbf{u}_{k-1}$ is omitted in (2.18); this is typical when the control actions are not known to the signal processing system where the KF is used to interpret the measurements. However, this term can be used in the formulation of a complementary filter configuration used in this thesis and is, therefore, included here.

The Minimum Mean-Square Error (MMSE) estimator, i.e., a linear filter that minimizes the mean-square error $P_k = E [(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T]$, where $\hat{\mathbf{x}}_k$ is the updated estimate produced by the filter, can be written using the following equations:

$$\text{State propagation (prediction):} \quad \hat{\mathbf{x}}_k^- = F_{k-1} \hat{\mathbf{x}}_{k-1} + G_{k-1} \mathbf{u}_{k-1} \quad (2.23)$$

$$\text{Covariance propagation:} \quad P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1} \quad (2.24)$$

$$\text{Gain computation:} \quad K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (2.25)$$

$$\text{Measurement update of the state:} \quad \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - H_k \hat{\mathbf{x}}_k^-) \quad (2.26)$$

$$\text{Covariance of the updated state:} \quad P_k = (I - K_k H_k) P_k^- \quad (2.27)$$

where $\hat{\mathbf{x}}_k^-$ is the best prediction (*prior estimate*) of the state without the knowledge about the measurement at t_k , P_k^- is the covariance of the predicted state, and I is identity matrix. The filter is started from the initial conditions

$$\text{Initial estimate:} \quad \hat{\mathbf{x}}_0 \quad (2.28)$$

$$\text{Uncertainty of the initial estimate:} \quad P_0. \quad (2.29)$$

The covariance update in (2.27) is known as the standard form. The other form

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (2.30)$$

is known as the *stabilized form* (Mendel, 1995) or as *Joseph form* (Grewal and Andrews, 2001). Compared to (2.27), (2.30) is less sensitive to numerical errors and maintains better the positive definiteness of the computed P_k . In addition, (2.27) is valid only for optimal gain while (2.30) is valid also for suboptimal gains (Brown and Hwang, 1997). The equations (2.23)–(2.29) present the KF algorithm in covariance filter form. An alternative form for this is information filter where the inverse of

the covariance is propagated instead of the covariance. The information filter is useful especially in situations when the initial uncertainty is large. For both covariance and information filters there exist also algorithms based on square root factorization methods to improve the numerical stability of the filters (Kailath et al., 2000; Grewal and Andrews, 2001). These are useful when the filter is implemented using a computer with limited-precision arithmetics.

For a system described by (2.18)–(2.22), the KF algorithm (2.23)–(2.29) is the best *linear* MMSE estimator. In addition, if all the noises and the initial state are Gaussian random variables, the KF is the optimal MMSE estimator among all possible filters (Brown and Hwang, 1997; Bar-Shalom and Li, 1998). Under the Gaussian assumption the estimate $\hat{\mathbf{x}}_k$ by (2.26) is also the conditional mean $E[\mathbf{x}_k|\mathbf{z}_{1:k}]$, where the random variable \mathbf{x} at t_k is conditioned on the measurement stream $\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_k$, denoted as $\mathbf{z}_{1:k}$. For the Gaussian variables, the conditional mean is also the point where the density function $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ gets its maximum (Brown and Hwang, 1997).

2.4.2 Extended Kalman Filter

Many practical dynamical estimation problems are nonlinear. Therefore, the linear techniques presented in Section 2.4.1 cannot be directly applied to solve these. In general, the implementation of the optimal nonlinear filter is infeasible, and filter designers have to settle with suboptimal solutions (Bar-Shalom and Li, 1998). Many estimation problems are nonlinear but ‘smooth’ in the sense that although the state dynamics and the relation between the state and the measurement are nonlinear, they are still approximately linear for small perturbations in the values of the state variables (Grewal and Andrews, 2001). A common suboptimal filter for this kind of nonlinear problems is Extended Kalman Filter (EKF). It is based on the linearization of the system and measurement models by using the first order series expansion.

For EKF, the following state and measurement models are assumed:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \quad k = 1, \dots \quad (2.31)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad k = 1, \dots \quad (2.32)$$

where noise properties are described by (2.19), (2.21), and (2.22).

The EKF algorithm has the similar look as the KF algorithm given in (2.23)–(2.27). However, the state propagation and measurement update are obtained using the non-

linear models (2.31) and (2.32), and the linear approximations of F_{k-1} and H_k needed in the covariance and gain computations are obtained by evaluating the partial derivatives of \mathbf{f}_{k-1} and \mathbf{h}_k with respect to the state \mathbf{x} . In EKF, the prior estimate $\hat{\mathbf{x}}_k^-$ serves as the reference trajectory around which the linearization is performed.

$$\text{State propagation (prediction): } \hat{\mathbf{x}}_k^- = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \quad (2.33)$$

$$\text{Linearization of } \mathbf{f}: F_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}} \quad (2.34)$$

$$\text{Linearization of } \mathbf{h}: H_k = \left. \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \quad (2.35)$$

$$\text{Measurement update of the state: } \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)) \quad (2.36)$$

It can be noticed that in (2.33)–(2.36), the filter keeps track of the total estimates $\hat{\mathbf{x}}$ rather than the incremental quantities $\Delta \hat{\mathbf{x}}$ typical in linearized systems. This is allowed by choosing the predicted state $\hat{\mathbf{x}}_k^-$ as the linearization point, which reduces the incremental $\Delta \hat{\mathbf{x}}_k$ to zero (Mendel, 1995; Brown and Hwang, 1997). The covariance propagation and the computation of the gain and covariance of the updated state are obtained using (2.24), 2.25), and (2.27), respectively.

2.4.3 Complementary Filter

Kalman filter (KF) has been applied in wide variety of integrated navigation systems for fusing information from diverse types of sensors (Brown and Hwang, 1997). Integration of inertial measurements with radio navigation is a well-known example of such systems (Kaplan, 1996; Parkinson and Spilker, 1996; Brown and Hwang, 1997; Farrell and Barth, 1998; Groves, 2008). The main reasons to integrate redundant measurement information from separate sensor systems are measurement errors and temporary availability breaks of some measurements. The integration brings benefits especially when the error dynamics of the systems are different as they are in radio navigation and inertial navigation. In radio navigation systems - such as satellite positioning or WLAN based positioning - the noise is quite significant but the error is bounded. Due to the relatively large random errors the ability of the radio navigation to track dynamic changes is poor. The navigation relies on external signal sources; breaks in the availability of the signal are possible, the signal is prone to interference, and the availability of the signals is geographically limited – even the GNSS, which

are built to have global coverage on the earth, suffer from poor availability indoors and in urban canyons. In inertial navigation, the sensor noise is relatively small and the system is able to track dynamic changes accurately, but its long term accuracy is poor as the errors can grow without bounds. The sensors are self-contained, they do not need any external fields and are practically immune to interference, and their operation is unconstrained geographically.

When a Kalman filter in non-complementary form is used to integrate positioning information from various sources, all the measurements are fed to the filter as measurement updates. The prediction of the state is obtained using a motion model, which describes the motion of the user as random process, i.e., the a priori statistics of the driving input (2.19) need to be known. In personal positioning applications, motion characteristics of the user may vary a lot. A proper model suitable for all possible motion states of the user is not easy to find. An attempt to use one motion model to cover all the possible motion states may lead to a solution that does not describe well any of the motion states. Multiple model approaches have been proposed to solve this problem (Syrjärinne and Saarinen, 1999) but these approaches lead to a more complex algorithm and the tuning of model change probabilities.

When complementary filtering is applied in integrated navigation systems, only the noise parts of the signals are modeled as random processes. For the total dynamical quantities, such as position and velocity, the a priori statistics are not needed (Brown, 1972-73). A complementary filter has the advantage that only the noise part of the signals is filtered whereas the true dynamic signal passes the filter without any lags. The principles and benefits of complementary filtering are described, e.g., in (Brown, 1972-73; Brown and Hwang, 1997; Farrell and Barth, 1998).

Complementary filter configurations for integrated navigation systems can be divided in three main categories: feedforward (FFC) and feedback configurations (FBC) and configuration with embedded reference trajectory (ERT). The configurations differ in their hardware architectures; the conceptual block diagrams of these are shown in Figs. 2.6–2.7.

The Inertial Navigation System (INS) in the FFC and FBC (Fig. 2.6) includes an Inertial Measurement Unit (IMU) and a navigation processor to produce position and velocity outputs from the measurements of IMU. In these filters, the output of the inertial system is compared with aiding positioning sources, such as radio navigation.

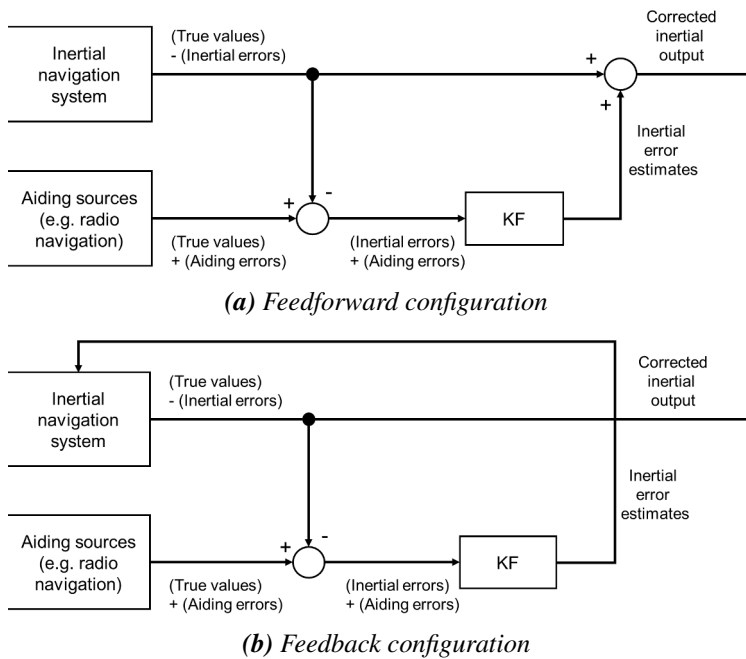


Fig. 2.6. Complementary filter in feedforward and feedback configurations.

The inertial errors are estimated by the KF and then used to correct the inertial system. As in these configurations the filter estimates only inertial errors, the filters are also known as error state space filters or indirect filters (Maybeck, 1979). The main difference between the FFC and FBC is the phase where the corrections to inertial system are applied: in FFB, the corrections are available for the navigation processor before it computes its outputs, while in FFC the corrections are added to the inertial output after the navigation processor. A practical difference in their implementations is state reset: to avoid the errors being integrated twice, the error states in the FBC filter are set to zero always when error estimates are transferred to the navigation processor.

With nonlinear problems, the KF needs to be replaced by EKF, and the filter needs the reference trajectory input for the linearization, which is obtained from the output of the INS. An important difference between FFC and FBC arises here: in the FBC, the INS has corrected its outputs with the previous error estimates from the filter, while in FFC the INS errors may be much larger as the corrections are not connected back to the INS. Without the feedback corrections the growth of the inertial errors is unbound and the reference trajectory used for linearization can drift far from the true trajectory,

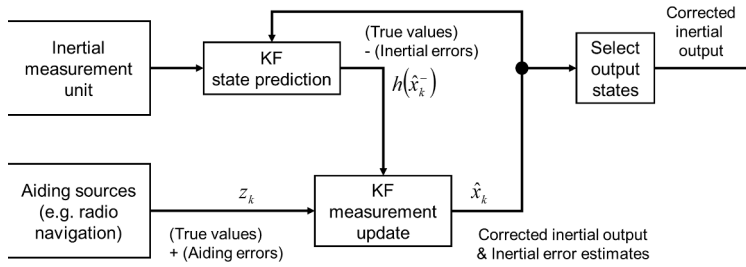


Fig. 2.7. Complementary filter in embedded reference trajectory (ERT) configuration.

which increases the approximation errors in F_{k-1} , H_k , and finally affects the estimate $\hat{\mathbf{x}}_k$ though the gain K_k . Therefore, FBC suits better than FFC for nonlinear filtering problems.

In the third complementary filter configuration, shown in Fig. 2.7, the reference trajectory computation is embedded in the KF. Differently from the usual KF implementations, where all the available measurement information is processed in the measurement update step of the filter, in ERT configuration, some measurements are used to predict the state estimate while others provide the normal measurement update of the state. The outputs of IMU provide information about the changes of state (e.g., position, velocity, and attitude) - therefore, they can be used directly to predict the state. These transition related measurements are modeled as a sum of the true value and noise, when they fit into the form of (2.18), where the deterministic input \mathbf{u}_{k-1} represents the true value and \mathbf{w}_{k-1} the noise. In ERT, the filter states include the total dynamical states. If the system includes slowly varying error components, such as biases or drifts, augmented states can be used to describe these non-white noise processes – similarly as with the FFC and FBC. Similarly as in FBC, in ERT the estimated effect of the noise history is taken into account in the prior estimate $\hat{\mathbf{x}}_k^-$, i.e., in the reference trajectory the large errors are mitigated, which makes also this configuration suitable for nonlinear problems. Leppäkoski and Takala (2007) showed that for a certain linear model all the three CKF configurations give equal estimation results.

2.4.4 Particle Filtering

Sometimes the characteristics of the estimation problem do not allow the successful use of KF or EKF – the models may be highly nonlinear and then the ‘smoothness’ assumption for EKF does not hold, or the noise distributions may be difficult, e.g., with multimodal distributions the estimated mean may not give sufficient characterization of the state of the process. This deficiency has led to the development of new state-space processors that are based on the idea that “it is easier to approximate a distribution than to approximate an arbitrary nonlinear function of transformation” (Candy, 2009). Unscented Kalman Filter (UKF) (Julier and Uhlmann, 2004) and particle filters (Arulampalam et al., 2002) are examples of these new processor algorithms.

The basic idea in using particle filter is to use random number generator to simulate the effect of the state driving noises (2.19) to the state variables (2.31). Instead of a single state estimate and its variance, the posterior PDF is approximated using a particle cloud, which consists of N weighted state estimates, carried by distinct estimators and their weights $\left\{ \left(\mathbf{x}_k^{(j)}, w_k^{(j)} \right), j = 1, \dots, N \right\}$. Using particles, the discrete approximation of the posterior density at k can be written as

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{j=1}^N w_k^{(j)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(j)}) \quad (2.37)$$

where $\mathbf{z}_{1:k}$ includes all the measurements up to the sample instance k and $\delta(\cdot)$ is the Dirac delta function. Theoretically, as the number of particles increases, the particle distribution approaches the posterior probability density function and the particle filter approaches the optimal Bayesian estimate (Arulampalam et al., 2002).

As in Kalman filters, also in particle filters the operation consists of state propagation and measurement update steps. In the state propagation, the states of the particles are predicted by drawing samples for the time instance k from the proposal distribution π :

$$\mathbf{x}_k^{(j)} \sim \pi\left(\mathbf{x}_k^{(j)} | \mathbf{x}_{k-1}^{(j)}, \mathbf{z}_{1:k-1}\right) \quad (2.38)$$

where $\mathbf{z}_{1:k-1}$ includes all the measurements before the sample instance k . The measurement update in the particle filter is performed by updating the weights $w_k^{(j)}$ of the

particles using the measurement likelihood:

$$w_k^{(j)} = w_{k-1}^{(j)} \frac{p(\mathbf{z}_k | \mathbf{x}_k^{(j)}) p(\mathbf{x}_k^{(j)} | \mathbf{x}_{k-1}^{(j)})}{\pi(\mathbf{x}_k^{(j)} | \mathbf{x}_{k-1}^{(j)}, \mathbf{z}_{1:k-1})}, \quad (2.39)$$

where $p(\mathbf{z}_k | \mathbf{x}_k^{(j)})$ is the probability of the measurement \mathbf{z}_k given the state $\mathbf{x}_k^{(j)}$. The probability $p(\mathbf{z}_k | \mathbf{x}_k^{(j)})$ is computed using the measurement equation (2.20) and the properties of measurement noise (2.21). After the normalization of the weights, i.e., $w_k^{(j)} \leftarrow w_k^{(j)} / \sum_{j=1}^N w_k^{(j)}$, the MMSE estimate of the state can be computed as the weighted average the particle states:

$$\hat{\mathbf{x}}_{\text{MMSE}_k} = \sum_{j=1}^N w_k^{(j)} \mathbf{x}_k^{(j)}. \quad (2.40)$$

The posterior distribution is approximated by resampling from the proposal distribution (2.38) so that the weight of the particle (2.39) defines the probability of a particle to be included into the set approximating the posterior distribution. In a bootstrap type of particle filter, the resampling is performed at every time-step, after which the weights all the particles are set equal.

2.5 Assessment of Positioning Accuracy

Several error measures can be used to assess the accuracy of positioning. Here, Root Mean Square Error (RMSE) is considered as it is a commonly used performance measure in the navigation and positioning community (Misra and Enge, 2006, Section 6.1.4). The two dimensional RMS position error is computed using

$$RMSE = \sqrt{\frac{1}{n_S} \sum_{k=1}^{n_S} (\delta x_k^2 + \delta y_k^2)} \quad (2.41)$$

where δx_k and δy_k are the errors of x and y coordinates of the position estimate samples, and n_S is number of the samples. In WLAN positioning related literature, the various statistics of error distance

$$\delta_k = \sqrt{\delta x_k^2 + \delta y_k^2} \quad (2.42)$$

are often used as a basis of comparisons (e.g., Bahl and Padmanabhan, 2000; Roos et al., 2002; Youssef et al., 2003). The commonly used statistics include the average

and various percentiles, from which the 95th percentile (E95) is the most commonly used. Often the Cumulative Distribution Function (CDF) of the error distance δ_k is also presented.

If the positioning error is assumed to follow a multivariate normal distribution, the coordinate errors δx and δy are assumed to be uncorrelated, zero-mean, and identically distributed with variance σ^2 , the error distance is Rayleigh distributed: $\delta \sim \text{Rayleigh}(\sigma)$. The median (50th percentile), mean, and the RMS value of a *Rayleigh*(σ) distributed random variable are the following (ITU, 2007):

$$E50 = \sigma \sqrt{\ln(4)} = 1.1774 \sigma, \quad (2.43)$$

$$EMEAN = \sigma \sqrt{\pi/2} = 1.2533 \sigma, \quad (2.44)$$

$$RMSE = \sqrt{2\sigma^2} = 1.4142 \sigma. \quad (2.45)$$

In practical error evaluation problems the true values are not known, and the error is estimated using a reference that is assumed to be more accurate than the estimate. The total uncertainty regarding the estimated parameter is the sum of the uncertainties: $\sigma_{tot}^2 = \sigma_{est}^2 + \sigma_{ref}^2$ where σ_{tot}^2 is the total uncertainty of the estimate, σ_{est}^2 is the mean square error $RMSE^2$ computed from the estimate and the reference, and σ_{ref}^2 is the uncertainty regarding the reference. In the following Chapters, σ_{ref}^2 is not taken into account in error evaluations due to its small effect. If the difference between the estimate and the reference is 1.0 m and the uncertainty in determining the reference value is 0.1 m, then $\sigma_{tot} = 1.005$. If σ_{ref}^2 is neglected in the evaluation of the error estimate, the introduced relative error in the error estimate is 0.5%. Considering the tests carried out in this thesis, these example figures are conservative: the error of the reference is at least ten times smaller than the average differences between the reference and the estimate.

In addition to the accuracy, important aspects to the navigation performance are also availability, continuity, and integrity, especially in safety critical applications (Prasad and Rugieri, 2005; Pullen, 2008). Integrity is the ability of the navigation system to provide timely warnings to users when the system should not be used to navigation, i.e., when the accuracy does not meet its limits. In the navigation systems developed in this thesis, integrity related information could be derived from the posterior covariances of the position provided by fusion filters. The feasibility of the covariance for integrity monitoring requires the consistency of the filter, i.e., that the estimates

are unbiased and the actual mean square error matches the covariance calculated by the filter (Bar-Shalom and Li, 1998). Continuity means the capability of the navigation system to perform its function without nonscheduled interruptions during the intended operation. Availability of a navigation system is the long-term average of the probability that the accuracy, integrity, and continuity conditions are simultaneously met. This thesis concentrates mainly on accuracy and does not further elaborate these other performance characteristics.

2.6 Summary of Adopted Methods

The following approaches presented in this Chapter are chosen for further research. In Chapter 3, WLAN fingerprinting is studied using average RSS patterns and kNN estimation (2.1)–(2.4) and probabilistic ML and MEE algorithms (2.7)–(2.9) using histogram approximations.

In Chapter 4, methods for aided PDR are studied. The PDR is based on step detection (Fig. 2.5) and step length estimation based on step frequency (2.11). The aiding information include indoor map presented using both unstructured and structured obstacle models and WLAN positioning based on MEE algorithm. To implement the PDR aiding, nonlinear filtering with particle filters and EKFs in both feedback (Fig. 2.6b) and ERT (Fig. 2.7) configurations are used. The EKF is implemented using state propagation and linearization (2.33)–(2.35), covariance propagation and gain computation (2.24)–(2.25), measurement update (2.36), and covariance update in the standard form (2.27).

3. WLAN POSITIONING

This Chapter addresses indoor positioning using WLAN RSS fingerprints. The goal is to find a WLAN positioning algorithm suitable for the use on a smart phone. The work was restricted to algorithms, where the MU performs the required signal strength measurements and calculates the position estimate. While the mobile application sets the accuracy requirements to the positioning, the smart phone as the implementation platform sets restrictions to the computational complexity and the memory usage of the algorithm. The performance of the probabilistic algorithms studied here has been demonstrated on smart phones (Perttula et al., 2009; Pei et al., 2009).

For fingerprint based positioning, a radio map model is needed. The model includes knowledge about the WLAN radio signal properties as a function of the location. The radio map is prior information that can be made available to the MU, e.g., by using wireless internet connection. There are several choices and parameters of the model that significantly affect the performance of the positioning. No clear-cut general rules exist yet for the best choices of the model and its parameters. These have to be decided based on the requirements of the positioning application and available resources.

When an adequate radio map for fingerprint based positioning in a certain area is obtained, it is desirable that the same radio map can be used by all devices that are to position themselves. One challenge related to radio maps is the fact that different MUs hear the RF signals differently, mainly due to the differences in their hardware composition, e.g., their antenna size and placement. Therefore, methods to normalize the RSS observations of different devices to a common scale are needed.

In this Chapter, the results of accuracy analysis and comparisons among different model parameters and RSS normalization are presented and discussed.

3.1 Related Work

Several information sources alternative to GNSS have been proposed for absolute positioning in indoor environments. These include positioning systems that rely on special positioning infrastructure installed into buildings and Signals of Opportunity (SoOP), such as the signals of Wireless Local Area Network (WLAN).

3.1.1 Signals for Indoor Positioning

Examples of the special infrastructure dedicated to positioning include pseudolites, i.e., pseudo satellites that are installed on the ground and use RF signal with a structure very similar to navigation satellites (Wang, 2002; Kuusniemi et al., 2012), transmitters of different kinds of signals, such as Infra Red (IR) badges (Want et al., 1992), ultra sound beacons (McCarthy et al., 2006), RFID tags (Fu and Retscher, 2009), and combination of ultra sound beacons and Radio Frequency (RF) signals (Priyantha, 2005). The usage of indoor lighting based on Light-Emitting Diodes (LED) modulated to transmit optical signal that allows the identification of the light source has also been proposed for indoor positioning Lou et al. (2012).

SoOP are signals that originally are intended for purposes other than navigation but from which position related information can be extracted. It is desired that a SoOP can be exploited for navigation without affecting their primary users (Eggert and Raquet, 2004). Generally, there are more SoOP available in urban environments than in rural areas (Raquet et al., 2007).

Several broadcast communication signals have been proposed for SoOP positioning, e.g., AM radio signals (Hall et al., 2002), digital television signals (Rabinowitz and Spilker, 2004), analog television signals, (Eggert and Raquet, 2004), digital audio broadcasting (Palmer et al., 2011), and mobile TV based on the DVB-SH standard (Thevenon et al., 2011). The accuracies of these systems range from 15 m to more than 100 m, which are not sufficient for many indoor positioning applications. Shorter range RF communication technologies available for positioning include Zig-Bee (Blumenthal et al., 2007), Ultra Wide Band (UWB) (Sahinoglu et al., 2008), Bluetooth (Kotanen et al., 2003a; Hallberg et al., 2003), and WLAN (also known as Wi-Fi) (Bahl and Padmanabhan, 2000).

Although some commercial applications exist, e.g., positioning systems using bluetooth (9solutions; Quuppa) and UWB (Time Domain; Ubisense), neither UWB nor ZigBee have become common in everyday mobile devices, and although Bluetooth is currently integrated into many devices, typically the positions of these are not known accurately enough to allow their signals to be used for the positioning of other mobile units nearby.

Contrary to Bluetooth, for WLAN communication there is nowadays stationary infrastructure hardware available almost everywhere in built environments such as public, commercial or residential buildings. Compared to Bluetooth, WLAN signals are generally more suitable for positioning as the WLAN networks include stationary units, APs, and because they have larger coverage area.

3.1.2 Measurements for WLAN Positioning

WLAN offers several observables applicable for positioning. Detailed descriptions of these methods can be found in, e.g., (Liu et al., 2007; Gezici, 2008; Gu et al., 2009). Time of Arrival (TOA) measurements between the MU and APs require time synchronization between the AP and the MU. This requires two-way messaging between the MU and the AP (Günther and Hoene, 2005; Izquierdo et al., 2006), adds traffic to the network and restricts the number of possible MUs served by one AP.

For Time Difference of Arrival (TDOA) measurements between the APs and the MU, the APs need to be synchronized with each other, which requires a special messaging scheme between them (Günther and Hoene, 2005; Izquierdo et al., 2006; Golden and Bateman, 2007; Yamasaki et al., 2005; Exel et al., 2010). As the orders of magnitudes TDOA are below microseconds, they need to be estimated on hardware layer, which is not supported by all WLAN hardware. In Differential Time Difference of Arrival (DTDOA), the need to synchronize APs is avoided by installing reference nodes with known distances and Line-of-Sight (LOS) to a master reference node which can be an AP (Nur et al., 2012, 2013). Also this method requires extra hardware components to be used in addition to the standard WLAN devices.

Angle of Arrival (AOA) measurements require antenna arrays to provide angle information. For example, in the system described by Wong et al. (2008), the arrays in both the transmitter and the receiver ends consisted of four antennas equally spaced

at 2.65 cm. However, this method is sensitive to multipath and Non-Line-Of-Sight (NLOS) propagation and also requires complex processing.

WLAN positioning methods based on Received Signal Strengths (RSS) and its variant Received Signal Strength Indicator (RSSI) have been reported on numerous papers, e.g., (Bahl and Padmanabhan, 2000; Battiti et al., 2002; Prasithsangaree et al., 2002; Roos et al., 2002; Smailagic and Kogan, 2002; Wallbaum and Wasch, 2004). Typically, in mobile devices these are available through Application Programming Interfaces (API) of standard WLAN services without any extra hardware. The MU can obtain the RSS by passive listening, without connecting to the AP. For outdoor environments with poor GNSS reception, the WLAN based positioning services are already available (e.g., Skyhook).

3.1.3 *WLAN Positioning Methods*

For producing WLAN RSS based position estimates there exist several types of algorithms. A detailed taxonomy of the methods can be found in (Kjærgaard, 2007). The simplest of these is the Cell Identifier (CID) method, where the position estimate of the MU is the position of the AP to which the MU is connected (Hodes et al., 1997). This is also the least accurate method, as the estimation accuracy is determined by the distances between the APs. An improved CID can be obtained by using a list of the APs and their RSSs that the MU listened (Hermersdorf, 2006; di Flora and Hermersdorf, 2008). In this method, the effect of measurement noise can be decreased using RSS based weighting and counting to increase the reliability of the highly granular positioning results. This also allows position to be estimated without connection to the AP, which is required by the original CID.

In algorithms based on path loss model and trilateration, the path loss model is used to convert the RSS measurements to distance estimates. Based on these distances and the known positions of the APs, the trilateration method is used to estimate MU position (Bahl and Padmanabhan, 2000; Kotanen et al., 2003b; Nurminen et al., 2012). Usually the parameters of the path loss model are experimentally defined. In indoor environments the path loss modeling is complicated by NLOS propagation and attenuation caused by walls and other structures. Even people cause significant fluctuations to the RSS (Bahl and Padmanabhan, 2000). This makes the simple path loss models too inaccurate in real life situations. To overcome this problem, the

performance of triangulation can be enhanced using other methods, such as pattern recognition (Smailagic and Kogan, 2002) and probabilistic filtering (Wallbaum and Wasch, 2004; Nurminen et al., 2012).

Fingerprinting approaches are introduced to overcome the difficulties of indoor path loss modeling. Instead of trying to apply a general model to relate RSS with distance, they use mathematical models to relate the measured RSS levels from different APs directly to the location of the MU (Bahl and Padmanabhan, 2000); these models are called radio maps. In this way, the fingerprinting actually makes use of the location dependent characteristics of radio signal propagation. Most often the radio map is determined experimentally. Several different models, model structures, and algorithms utilizing the models have been proposed for the radio maps (Bahl and Padmanabhan, 2000; Roos et al., 2002; Honkavirta et al., 2009; Liu et al., 2010). The choice of the model type, structure, and size determine the suitable positioning algorithm and affect the obtainable accuracy, computational complexity, the memory required to store the radio map, and also the amount of time and work required to collect the experimental data for the radio map. In addition, the way how the data is collected has an effect on the quality of the radio map.

The position estimation methods can be further categorized as snap-shot methods and filtering methods. The snap-shot methods use only the latest set of available RSS samples to infer the current position of the MU, without using the information from the past RSS measurements. Methods to enhance these basic snap-shot estimates by utilizing the measurement history with different kinds of filtering methods have been presented in many papers, e.g., (Wallbaum and Wasch, 2004; Ciurana et al., 2007; Nurminen et al., 2012).

3.1.4 WLAN Fingerprinting

Pattern recognition based WLAN fingerprinting is proposed in (Bahl and Padmanabhan, 2000; Prasithsangaree et al., 2002) while neural network based fingerprinting is proposed in (Battiti et al., 2002). Probabilistic fingerprinting are proposed in (Castro et al., 2001; Roos et al., 2002; Youssef et al., 2003). In these works, the PDF of the RSS is approximated using histograms or kernel functions. Fingerprinting with the PDFs approximated using parametric functions have also been proposed (Honkavirta et al., 2009; Koski et al., 2010a,b).

Bahl and Padmanabhan (2000) introduced a pattern matching based fingerprinting that utilizes the sample mean as the summarized signal pattern stored in the radio map and the principle of the kNN in signal space to estimate the position. They observed 6% reduction in the median position errors when increasing k from 1 to 5: 2.94 m with NN and 2.75 m with 5NN. In these tests, they had distinct radio map entries for each measurement directions. Bahl and Padmanabhan (2000) also observed that the measurement direction has a significant effect on the measured RSS. They tested the effect of measurement orientation by summarizing the calibration data of one CP over all the four measurement directions by computing first the RSS averages separately for each directions and then saving the maximum of the four RSS averages to the radio map. This treatment decreased the median positioning error by 9% with NN. With the combined directions they obtained the best accuracy with 3NN; the median error was decreased by 28% from the 2.94 m. The test area in (Bahl and Padmanabhan, 2000) consisted of narrow corridors.

Roos et al. (2002) compared the accuracy of the histogram based algorithm with the NN algorithm. With these algorithms, they obtained the average errors 2.76 m and 3.71 m, respectively. The fingerprints in their NN algorithm were the individual calibration measurements without summarizing samples from one AP in one CP. The reporting in (Roos et al., 2002) does not mention the measurement directions and their treatment in the tests, as well as not the number of histogram bins used in their comparison.

As the measurements campaigns for obtaining the radio maps are time consuming, it is important to solve the problem of RSS normalization between different devices, i.e., the possibility to use radio map based on measurements of one device to estimate position using measurements of another device. The normalization problem has been addressed in recent works: Kjærsgaard (2011) and Machaj et al. (2011) proposed methods to avoid the normalization, and Laoudias et al. (2013) proposes a normalization method based on the existing samples in the radio map and a small set of calibration samples by the new device.

Kjærsgaard (2011) proposed Hyperbolic Location Fingerprinting (HLF) to avoid the normalization between heterogeneous WLAN devices. In HLF, instead of absolute RSS values, the fingerprints and positioning measurements are based on RSS ratios between two APs; these RSS ratios factor out the constant differences in RSS between different devices. The drawback of this method is that instead of each individual

AP, the parameters are required for the possible pairs of APs, which leads to the increase of the radio map size. Machaj et al. (2011) avoids the normalization by using Rank Based Fingerprinting (RBF): instead of RSS values, the fingerprints consist of RSS based ranks of APs, because the rank information is invariant to any monotonic increasing transformations such as bias and scale.

Laoudias et al. (2013) found that in both HLF and RBF, the omitting of the RSS level information causes accuracy degradation. Instead of these methods, they propose a self-calibration method where a new mobile client self-calibrates itself against the fingerprints stored in the radio map. The self-calibration is based on the RSS histograms of ten sample scans by the new device and the RSS histogram derived from the radio map data. The challenge with this method is the quality of the self-calibration samples – if the samples are collected at a location where time variant, local radio channel anomalies distort the RSS distribution severely, it is possible that the obtained calibration parameters are not valid. However, the results of Laoudias et al. (2013) on the RSS bias correction among the devices are consistent with the RSS normalization results presented in this thesis.

Despite the advancements in normalization methods between heterogeneous WLAN devices, the differences between the devices set significant limitations on WLAN based positioning. One reason for this fundamental problem is the lack of standardization: the existing standards do not require the device manufacturers to measure and report the WLAN RSS in a standard way. Lui et al. (2011) found in their tests with 17 different devices that some of the devices have either very small slope in their RSS versus logarithmic distance curve or spurious, unexplainable temporal variations in their reported RSS, which both mean that these devices are not suitable for any positioning methods relying on location or distance dependence of the RSS levels. However, they can still be useful in methods based on coverage areas (Koski et al., 2010b).

In this thesis, the fingerprinting based on WLAN RSS was chosen for the better accuracy of the fingerprinting and the easy availability of the RSS measurements. They are available from the existing wireless communication infrastructure and they can be obtained by passive listening of the APs. In the mobile devices they are available through the standard APIs. Only the snap-shot methods are considered in order to be able to analyze the effect of the measurement modeling in isolation from the filter performance. The filtering and utilization of the past measurements in the

context of multisensor data fusion are handled in Chapter 4.

Both the pattern matching and histogram based fingerprinting algorithms were implemented and compared. These methods were tested in an open library hall. The calibration data from all CPs included samples measured from four directions. For both the methods, the radio maps where the directions had separate entries were compared to radio maps where the data from different orientations were combined. For the pattern matching the data from orientations were combined using averaging, instead of the maximum used in (Bahl and Padmanabhan, 2000). With the histogram based method, the data from the orientations were combined to one histogram. With the pattern matching method the comparisons were made using different k values of kNN, and for histogram based methods the comparisons were made with several numbers of bins.

This thesis also studies the problem of RSS normalization among different devices. These results on RSS normalization were originally published in (Leppäkoski et al., 2009b). In the proposed approach, the RSS offset parameter for the normalization was determined in positioning tests, where the RSS values in the radio map were corrected with the offset parameter before using the radio map in the position estimation using the RSS measurements of another device. The offset value that provided the best accuracy was chosen as the normalization parameter. Later, the normalization problem has been addressed also by other researchers (Kjærgaard, 2011; Machaj et al., 2011; Laoudias et al., 2013).

3.2 Accuracy Analysis

In indoor positioning with WLAN fingerprints, the positioning accuracy is affected by the amount of RSS data details that are stored into the radio map. It has been found that the CP grid density has an effect on the accuracy (Perttula et al., 2009); the mean positioning error grows gradually but clearly as the distance between calibration points grows. It has also been found that when probabilistic methods are used for position estimation and the RSS distribution in a calibration point is approximated using a histogram, the number of histogram bins affects positioning accuracy as well as the way how the information about the measurement direction is treated in the radio map (Leppäkoski et al., 2009a). However, answers to some questions still remain unclear:

- What is the number of histogram bins after which the positioning accuracy does not improve any more even if the number of bins is still increased?
- Is it beneficial to have a separate bin for missing samples, i.e. samples with measurement value below the minimum of correct RSS measurement range?
- Does it improve the positioning accuracy if the information of the measurement directions is included into the radio map?
- If there are transmitters very close to each other so that RSS measurements from them are correlated, is it beneficial to combine the correlating signals in the radio map and the estimation?
- Are the previously mentioned effects similar for histogram based methods and the kNN based pattern recognition?

The tasks of finding proper parameters for radio propagation model or creating radio maps are complicated by the fact that different WLAN receiver implementations commonly see and report the signal strengths differently. Partly this is due to unpredictable noise; partly it is caused by the different designs of WLAN antennas and receivers. The important question is how the propagation model parameters or radio map defined using one WLAN device can be utilized when estimating position with another WLAN device. This is essential especially considering the radio maps for fingerprinting methods, because the experimental phase for creating the radio map is laborious and time consuming. It would be of great benefit if a radio map created using one WLAN receiver could be utilized in positioning with many different types of WLAN devices.

In the next subsections, the effect of the model parameters used in the radio maps and the RSS normalization among different devices are studied. The algorithms described in Section 2.1 for the radio map generation and the position estimation were implemented to carry out the tests and comparisons. The data sets used in the analysis are described in Section 3.2.1. The effects of radio map configurations to the accuracy and memory requirements of the radio map are studied in Sections 3.2.2–3.2.5 and the normalization of the WLAN RSS is studied in Section 3.2.6.

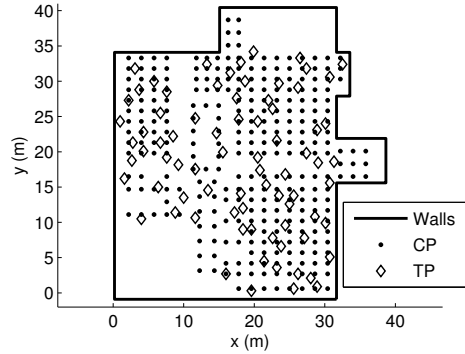


Fig. 3.1. Floor plan of the library and the locations of calibration points (CP) and test points (TP).

3.2.1 Data

In the following analyses two different data sets were used. The first one was collected at the library of Tampere University of Technology using one mobile device and the second one was collected in Tietotalo building at Tampere University of Technology using three different mobile devices simultaneously. From the data, the radio maps and position estimates were computed using the equations given in Section 2.1. All the computations were carried out using MATLAB.

Data Set #1: Library

The radio map configurations were studied using data that was collected at the library of Tampere University of Technology with a mobile unit (Nokia N800 Internet Tablet) carried by a person. The measured RSS data files were transferred to a PC for offline analysis with MATLAB. The data contains measurements from 287 CPs and 77 test points (TPs). During the data collection in both the CPs and the TPs, the user was standing holding the WLAN device and blocked the signals from behind. Data was collected during day time when people used the library normally. The effects of other people were random disturbances in the collected data. Calibration data were measured from four directions in each CP. Measurement time for each direction in a CP was about 30 s, during which on average 13 samples were obtained. The TP data was collected from only one direction that varied randomly from point to point, and the average number of samples from each TP was 10. The coordinates of the CPs

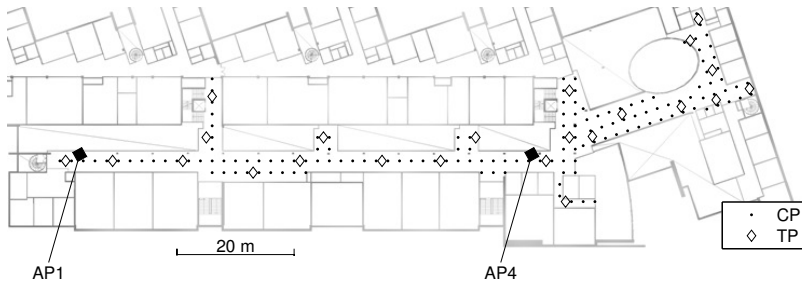


Fig. 3.2. Floor plan of the office building and the locations of calibration and test points.

and TPs and the measurement directions were manually recorded and saved with the measured RSS samples. In total, the calibration and test data sets contain 15372 and 791 RSS vector samples. The test area and the CP and TP locations are shown in Fig. 3.1. The separation between the CP grid lines varied between 1.5 and 2.1 m.

Data Set #2: Office building

To study the normalization of the RSS measurements, a measurement campaign was conducted in the main corridor of the second floor of Tietotalo building at Tampere University of Technology. The floor plan of the area with the CP and TP positions is shown in Fig. 3.2. The number of measured CPs and TPs were 146 and 24, respectively. The grid between CPs was 2 m.

In each point the data was simultaneously collected using three devices: Nokia N95 8GB Symbian smartphone (Mobile1), Nokia N800 Internet Tablet (Mobile2), and laptop PC, General Dynamics GoBook XR-1 with Windows XP (PC). During the measurements, the devices were placed on a wheeled cart so that the user was not blocking the signals to the devices. Data was collected during day time when people moved on the area normally. The effects of other people were random disturbances in the collected data. In each device, the data logger software scanned regularly the available WLAN channels and saved the measured RSS values of the listened channels together with their unique MAC addresses into a file. The coordinates of the measurement points were manually recorded and saved with the measured RSS samples.

In Mobile1 the data logging was implemented using the WLAN API of S60 3rd Edition FPI operating system. In Mobile2, the data logger software was written in C++

and it used an open source software `iwlist` for WLAN access point scanning. With the PC, the WLAN scans were performed using Network Stumbler, a freeware for analyzing the wireless networks. From all the devices, the measured RSS data files were transferred to a PC for offline analysis with MATLAB.

3.2.2 Preprocessing for Comparisons of Radio Map Configurations

To compare the effects of the several design choices, i.e., number of histogram bins, separate bin for missing values, direction information in radio map, and combining of the signals from adjacent transmitters, these options were varied and several different radio maps were computed from the CP data. Radio maps were generated for several numbers of bins, both uneven and even bin widths, and three different treatments of the direction information. The TP data was used to estimate positions with these radio maps and the obtained estimates were compared with the true coordinates saved in the TP data. The true coordinates were used to compute the error distances to assess the accuracy. Finally the positioning accuracies obtained with the different radio maps and algorithms were compared with each other.

The effect of measurement direction was studied by generating three radio maps where the direction information was treated differently. One radio map (*rmap1*) included the direction information such that separate fingerprints (radio map entries) were generated for each measurement direction in each CP. In the second radio map (*rmap2*), the direction information was omitted, thus all the measurements from the same CP regardless of the measurement direction were combined to one fingerprint. The resulting radio map had a size of only a fourth of the size of the first radio map. On the other hand, now there were four times larger amount of calibration data available for each radio map entry.

To analyze the effect of combining samples from all directions in isolation from the sample size for a fingerprint, the third radio map (*rmap3*) was created, where only one fourth of the samples from each direction were combined together to compute one fingerprint. This yields approximately the same calibration set size for each fingerprint as was available in *rmap1*, where each direction was treated separately.

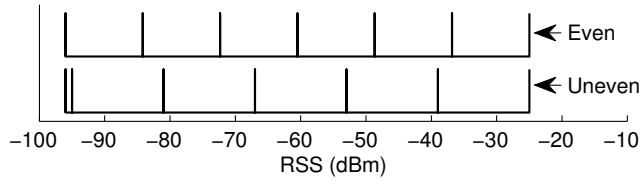


Fig. 3.3. Edges of six histogram bins, evenly and unevenly distributed.

3.2.3 Effect of Bin Configuration

To study the effect of the number of bins, radio maps were generated using 2 to 15 histogram bins. To investigate the effect of separate bin for missing samples, the number of bins test was performed in two different ways. For each number of bins, there was one bin configuration with even bin widths where the RSS minimum -96 dBm, which was used to indicate the missing samples, was classified to the same bin with other small values. Another configuration was defined to have one narrow bin for the minimum RSS value while the rest of the bins had equal widths. An example of the used bin widths for six bins is shown in Fig. 3.3. The results of the bin configuration tests are shown in Fig. 3.4, where average error distance (EMEAN), RMS positioning error (RMSE), the 95th percentile of error distance (E95), and maximum error distance (EMAX) are plotted for all the test cases. The results are presented for all the three treatments of direction information (*rmap1*, *rmap2*, *rmap3*) to illustrate the fact that the bin configuration results do not depend significantly on how the direction information is used.

According to the test results, the RMS position error as well as the average error distance decrease as the number of bins increases until it gets the value seven or eight, after which the errors stop decreasing (Figs. 3.4.a, 3.4.b). The number of bins has very little impact, if any, on the maximum errors (Fig. 3.4.d). From the plots of the 95th error percentile it can be observed that with ML algorithm, the errors clearly drop as a function of the number of bins, but the effect is not so significant with MEE algorithm (Fig. 3.4.c). The results in Fig. 3.4 also show that it depends on the number of bins whether a separate bin for missing samples brings a benefit or not: if the number of bins is six or less, the positioning errors are smaller with uneven bin widths where there is a separate bin for RSS minimum. With numbers of bins seven

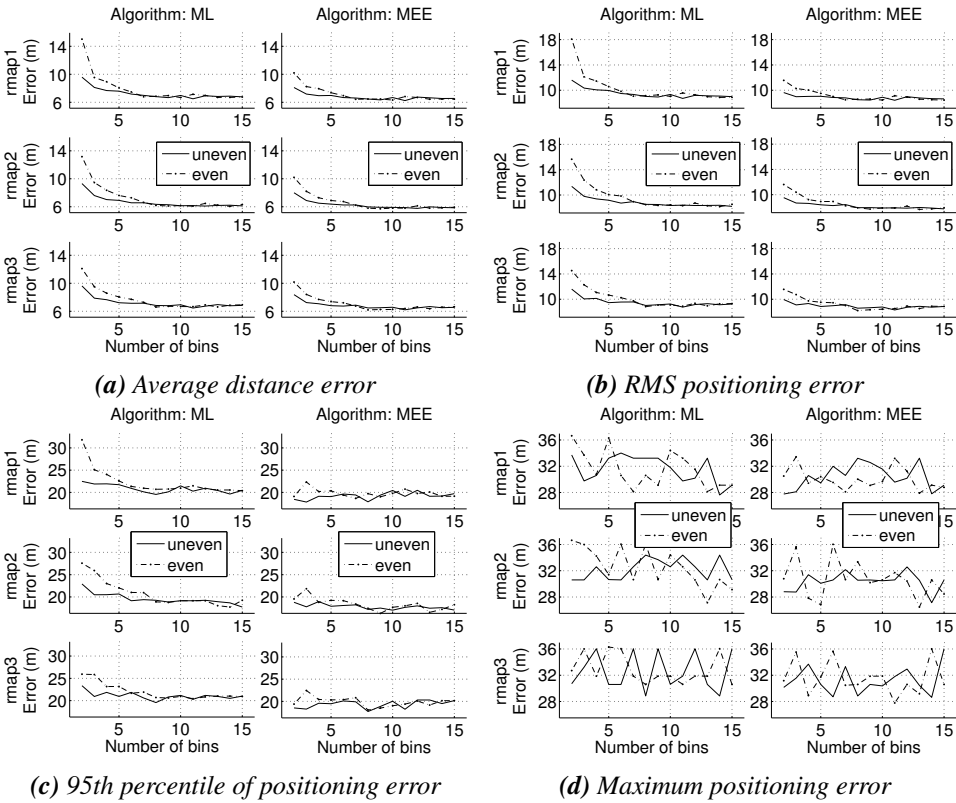


Fig. 3.4. Positioning errors with different bin configurations (number of bins, even and uneven bin widths) computed with different algorithms (ML, MEE) and radio maps ($rmap1$, $rmap2$, $rmap3$).

or more, the differences between even and uneven bin widths are very small, and it seems to vary randomly whether the positioning accuracy is better using uneven or even bin widths. The effect of the number of bins on the accuracy is more significant with even bin widths, as the errors with fewer bins are larger with this configuration. From Fig. 3.4 it can also be seen that in general, the MEE algorithm is more accurate than the ML algorithm.

To conclude the results on bin configurations, the main findings are summarized as follows:

- The positioning accuracy of the histogram based algorithms improve when the number of histogram bins increases until it reaches 8
- With less than 6 histogram bins, the accuracy can be improved if, instead of

having all bins with equal widths, there is one narrow bin for the minimum RSS.

- MEE algorithm is generally more accurate than the ML algorithm

3.2.4 Benefit of Direction Information in Radio Map

To test the effect of the direction information, three radio maps were compared: *rmap1* includes a separate fingerprint for each direction, therefore, less data per one fingerprint than in *rmap2*, where CP data from all directions were lumped to one fingerprint, therefore containing the largest amount of data per fingerprint. In *rmap3* the CP data from all directions were lumped to one fingerprint, but only 25 % of the data was used for radio map. Therefore it contains the same amount of data per one fingerprint as *rmap1*. Position estimates using both the histogram based methods and the pattern recognition method were computed using these radio maps.

The test results using histogram based radio maps where the direction information is treated differently are shown in Fig. 3.5. In these figures, results are shown for both the ML and the MEE algorithms and for configurations with uneven and even bin widths.

The average distance error, RMS error and 95th error percentile are presented in Figs. 3.5.a-c. The performance of *rmap2* is equal or better than the performance of *rmap1* in most cases. However, with numbers of bins 2 and 3 the difference between *rmap1* and *rmap2* is almost zero and starts to grow with increasing numbers of bins. Throughout the figures the performance of *rmap3* is comparable with *rmap1*; which one is better, varies with the number of bins, but the variation appears to be random. Hence the crucial factor to explain the differences between the accuracies provided by the radio maps is the sample size used to compute one fingerprint. On average, there are 13 samples per direction in CP data for estimating the radio map parameters. With the small numbers of bins, the amount of data is enough to estimate separate parameters for each direction. With larger numbers of bins, the number of required model parameters increases resulting in decreased model accuracy with the restricted amount of data. Now the combining of the directions decreases the number of parameters to be estimated and allows more calibration data for each parameter. Thus with larger numbers of bins and restricted amount of calibration data, having a separate

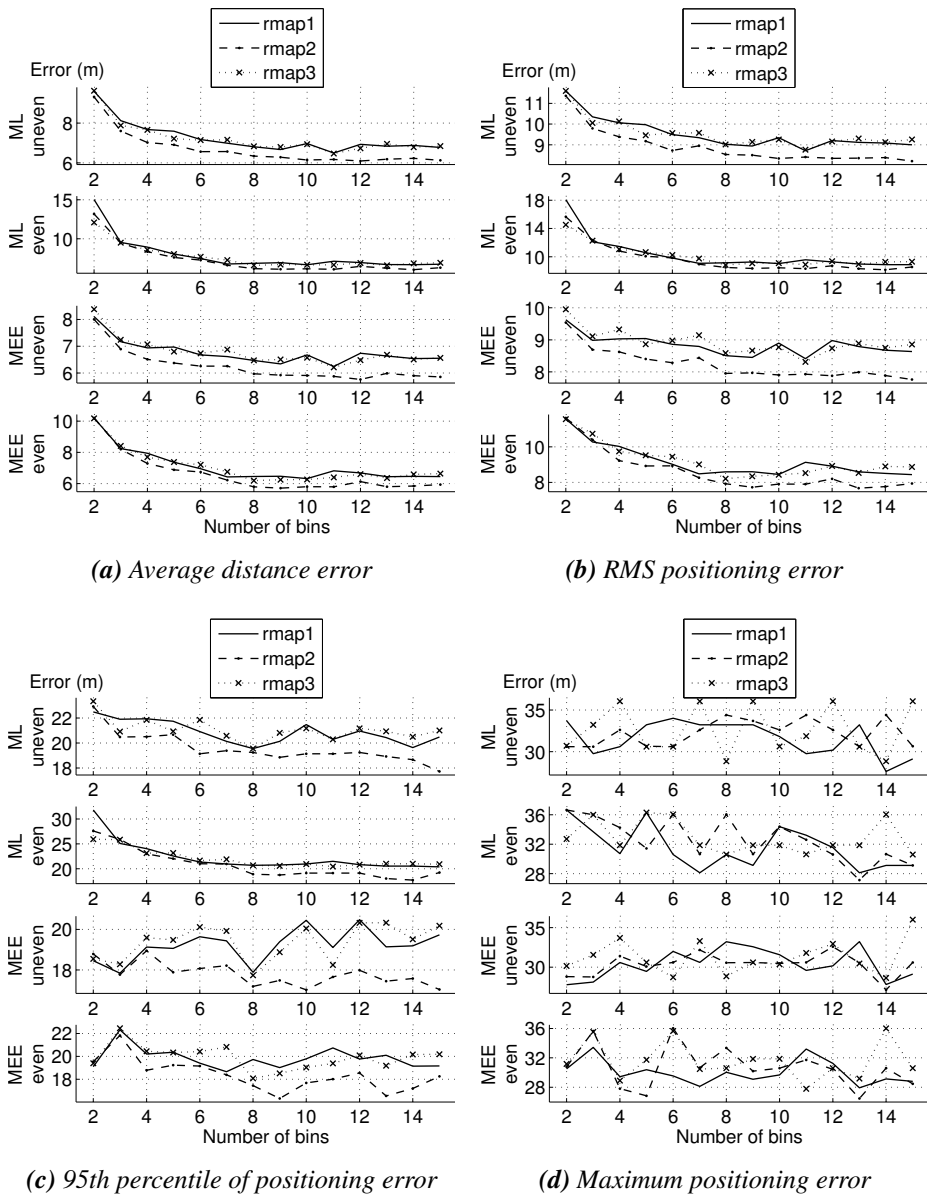


Fig. 3.5. Effect of direction information in radio maps (*rmap1*, *rmap2*, *rmap3*). Positioning errors with different numbers of bins, ML and MEE algorithms, and uneven and even bin widths.

fingerprint for each calibration measurement directions does not improve accuracy, but increases the radio map size.

As with bin configuration tests, neither treatment of the direction information has ef-

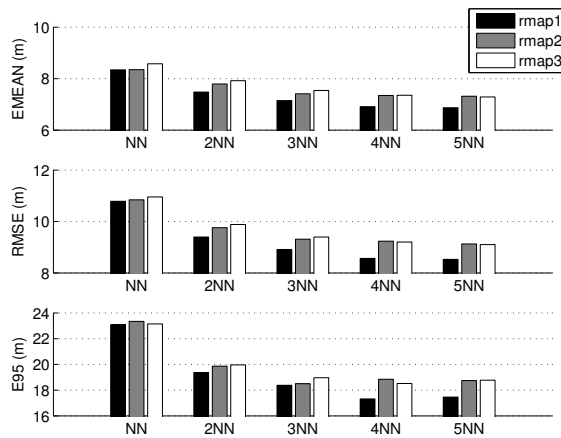


Fig. 3.6. *kNN* pattern recognition with sample means: effect of direction information in radio maps on the average error distance (EMEAN), root mean square error (RMSE), and the 95th percentile of positioning error (E95).

fect on maximum position error (Fig. 3.5.d); also here the superiority of radio maps seems to behave randomly as the number of bins changes. The effect is similar regardless of the positioning algorithm and the bin configurations.

The effect of different treatments of direction information in *kNN* pattern recognition with sample means is illustrated in Fig. 3.6. Average error distance, RMS error, and the 95th percentile of positioning error are shown for the three radio maps when the number of nearest neighbors taken into account in position estimation varies from 1 to 5. For all error measures and numbers of NN, *rmap1* gives the most accurate results. With NN ($k = 1$), the differences between the radio maps are small, less than 0.3 m. In the most cases the accuracy with *rmap2* is better than with *rmap3* but these differences are also smaller than 0.3 m. With respect to direction information, pattern recognition algorithms differ from probabilistic models: best accuracy is obtained with a radio map where the measurements from different directions are used to form separate fingerprints. This does not contradict with the interpretation of the direction results of Fig. 3.5: the direction information in the radio map decreased the positioning accuracy with large numbers of model parameters. With *kNN* the number of the required parameters is only one per direction and therefore it can be determined accurately with the available calibration data. With all the radio maps, the average error distance and RMS error decrease as the number of nearest neighbors

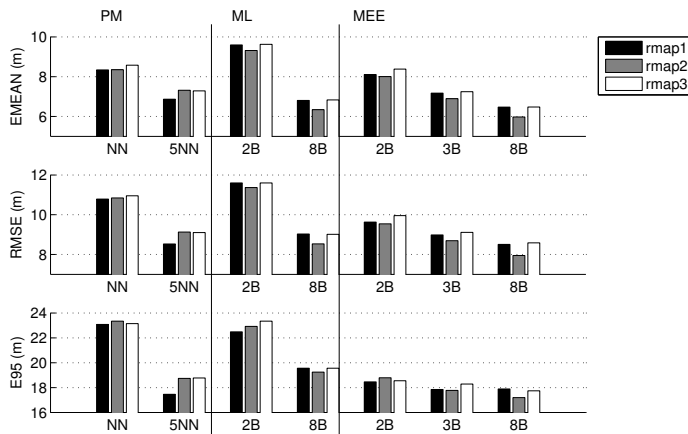


Fig. 3.7. Comparison of pattern recognition and probabilistic algorithms: pattern recognition using 1 and 5 nearest neighbors, ML with 2 and 8 bins, and MEE algorithm using 2, 3, and 8 bins, both ML and MEE with uneven bin widths.

used in the estimation increases; with the 95th error percentile, the effect is not clear with 4NN and 5NN. With kNN the number of radio map parameters is not affected by the number of NNs used.

To compare all the implemented algorithms, the average error distance, RMS error, and the 95th percentile of positioning error obtained using pattern recognition with 1 and 5 NN, ML with 2 and 8 unevenly spaced bins, and MEE using 2, 3, and 8 unevenly spaced bins are shown in Fig. 3.7. The histograms with small numbers of bins is chosen to the comparison because with two bins, the probabilities of RSS belonging to these bins can be expressed using one parameter only, as the other probability can be obtained by subtracting the first from the probability 1. In this case the memory size requirement of the radio map is the same as in pattern recognition with sample means; with three bins, the memory size is already two times as large as with pattern recognition.

From Fig. 3.7 (and also from Figs. 3.4 and 3.5) it is clear that the MEE algorithm has better accuracy than the ML. An interesting observation is that the MEE algorithm with two bins is more accurate than pattern recognition with NN. However, the accuracy of the pattern recognition can be improved by taking more NNs into account in the position estimation. With five NNs and *rmap1*, the accuracy is comparable with ML using eight bins or MEE using three bins, both using *rmap2*. In pattern

recognition, the change from NN to 5NN does not increase the radio map size, but the complexity of the computations increases.

In these tests, the size of the *rmap1* for pattern recognition is one half of the *rmap2* with eight bins. Using five NNs in pattern recognition yields accuracy almost comparable with the MEE with eight bins, as the differences in the errors are less than 1 m. For most applications, trading off this small accuracy degradation for reduced memory requirement is acceptable. The good performance of 5NN with respect to MEE and 8 bins may be caused by the restricted amount of calibration data. However, increasing the amount of the data may not be feasible in practical implementations due to the high cost of the data collection.

To conclude the results on the direction information in radio maps, the main findings are summarized as follows:

- The best accuracy with histogram based algorithms was obtained by combining the calibration samples measured from different directions in the radio map
- With sample means as fingerprints and pattern matching with kNN, the positioning accuracy is better if there are separate fingerprints for each measurement direction
- With small number of model parameters, better accuracy is obtained with separate fingerprints for different directions
- With larger number of model parameters, the accuracy is better if the RSS information from all directions is combined one fingerprint

3.2.5 Effect of Combining Measurements from Adjacent Transmitters

In the test area, the infrastructure WLAN access points contain four antennas each. Therefore, the measured data contains groups of four MAC-addresses, where the RSS measurements within each group show high correlation due to the mutual proximity of their sources. Fig. 3.8 shows the correlation coefficients computed for the RSS data from different MACs where the higher correlations within the groups of four MACs can clearly be observed. On the other hand, there is quite large amount of missing samples in measurement sets (Leppäkoski et al., 2009a). One idea for mitigating the

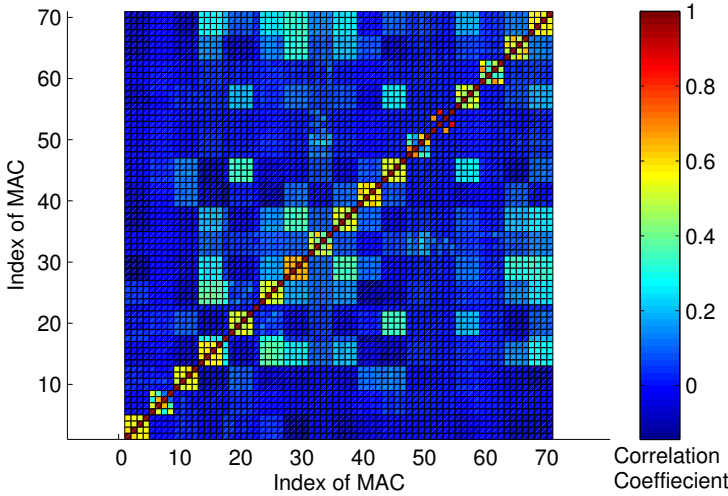


Fig. 3.8. Correlation coefficients between the RSS from different MACs.

problem of missing samples is to combine the measurements from the correlating sources before the computations, both in the radio map and in the estimation.

The correlations between the RSS measurements from adjacent emitters were attempted to be utilized to mitigate the problem of missing samples and to decrease the size of the radio map. Five alternative methods were used to combine the four signals from the same AP: *mean* (mean of the signals), *mean2* (mean of the signals where the minimum values below or equal to -96 dBm were excluded from the data), *median* (median of the signals), *median2* (median of the signals where the minimum values were excluded from the data), and *max* (maximum of the signals). Radio maps were computed for sample mean based pattern recognition and two different bin configurations, i.e., eight bins using both even and uneven bin widths. For ML and MEE, the *rmap2* was used, i.e., the calibration samples from all measurement directions were combined to one radio map entry, as this provided the best accuracy in the previously described tests (Figs. 3.4, 3.5, and 3.7). For pattern recognition, the *rmap1* with separate fingerprint entries for the four measurement directions was used, as it was the most accurate for pattern recognition (Figs. 3.6 and 3.7).

Position estimates using these radio maps were computed using both the bin configurations and sample mean based pattern recognition with four different kNN numbers.

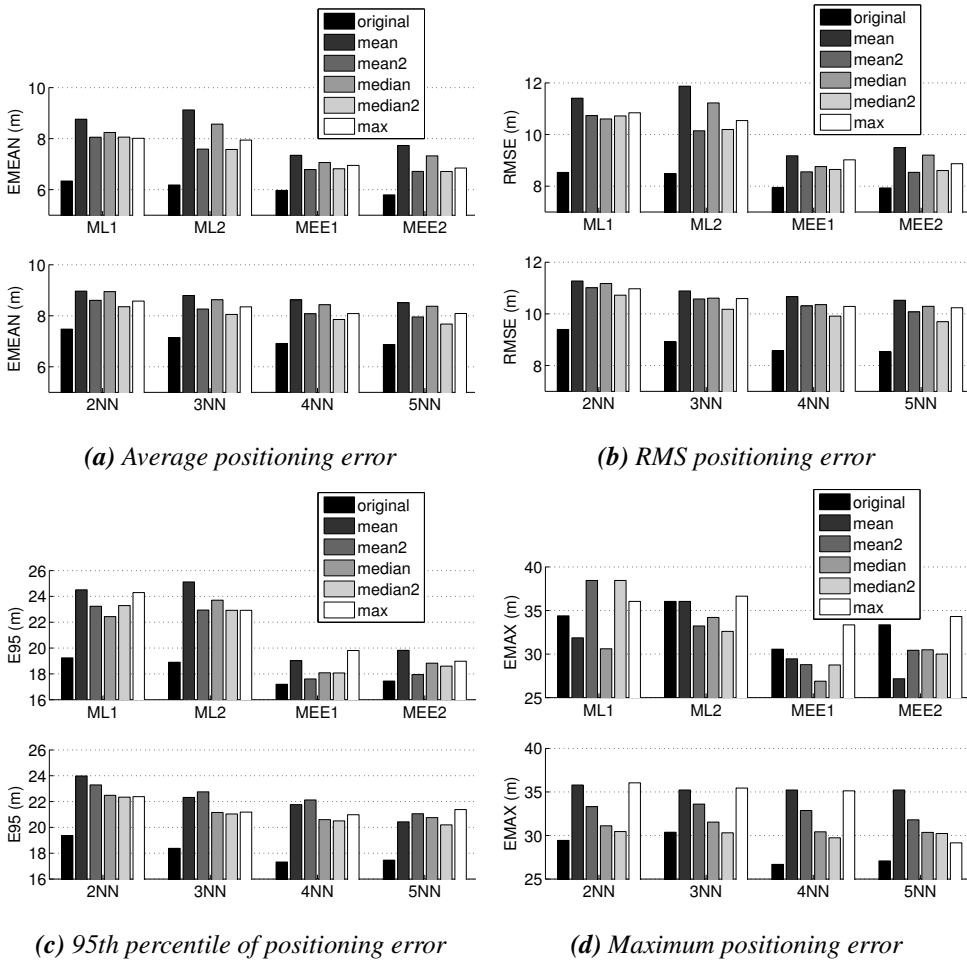


Fig. 3.9. Effect of combining signals from correlating sources: ML and MEE algorithms with bins distributed unevenly (ML1, MEE1) and evenly (ML2, MEE2), and pattern recognition with 2 – 5 NN.

The results of combined signals together with the original results without source combinations are shown in Fig. 3.9, where average error, RMS error, 95th percentile of position error, and maximum error are shown. Based on Figs. 3.9a, 3.9b, and 3.9c, it is clear that in most cases the original approach without signal combinations is the most accurate, yielding always smaller position error than the combined versions. This can be observed with all the radio maps, all three algorithms, all bin configurations and all numbers on NN. However, in Fig. 3.9d some of the combined signals provide smaller maximum errors than the uncombined version, i.e., the combination of the signals provides slightly improved robustness. With ML algorithm the median

of the correlated sources provides smaller error than uncombined signals with both bin configurations, and with MEE improved maximum error is obtained using all the combination functions except maximum. With pattern recognition the combination of sources does not decrease even the maximum error.

With MEE algorithm the accuracy of the combination using *mean2* is almost as good as with the original signals; the RMSE and E95 are only about 0.6 m larger than with uncombined signals. The difference is so small that in most cases the accuracy degradation could be traded off for the decreased radio map size and improved robustness that is obtained by using the combined signal sources. However, in the terms of RMSE, E95 and EMAX, the accuracy is on about the same level as can be obtained using pattern recognition with 4 or 5 NNs and uncombined signals. With 4NN and 5NN, the radio map size is two times as large as the radio map size when histograms with eight bins and four combined signals are used: for one AP, the *rmap1* includes one fingerprint parameter for each calibration measurement direction, and these for each of the uncombined signals from four transmitters, i.e., 16 parameters in total. At the same time, in histogram with eight bins using *rmap2* and combined signals from one AP, there are only eight parameters.

From the results shown in Fig. 3.9, and considering the required radio map size, it can be concluded that if the WLAN infrastructure includes multiple transmitters in the same AP device, the MEE algorithm with eight bins and combining the calibration samples from all the measurement directions in one CP and the samples from all the transmitters of one AP is the most reasonable choice.

To conclude the results on combining of the measurements from adjacent correlating transmitters, the main findings are summarized as follows:

- Combining of the correlating transmitters decreases the accuracy
- The savings obtained by decreasing the radio map size by 75% cost 1–2 m in average error

3.2.6 Normalization of WLAN RSS

This Section addresses the problem of normalizing WLAN RSS measurements. The goal is to make the radio map models obtained using one WLAN device applicable

to position estimation with another WLAN device. The proposed method was tested using a laptop PC and two hand-held devices. The normalization testing was based on the comparison of the position estimation accuracy of two cases: in the first, the estimation is based on measurements of the same device as was used to create the radio map, and in the second, the estimation is based on normalization and the measurements from different devices than what was used to create the radio map models. Probabilistic fingerprinting algorithms were used in the positioning phase.

An interesting possibility is the case where the models are created using the measurements of devices that are able to update their RSS readings more frequently, while the estimation is carried out using the device where RSS updates are limited to slower sampling. The time saving in modeling would be significant if the radio map model generated using the data of faster device can also be used in estimation phase on the slower measuring devices.

Data Preprocessing

Most of the standard 802.11 devices transmit power at roughly 20 dBm and can receive power all the way to -96 dBm, which can be considered as the minimum of correct RSS measurements (Bardwell, 2002). During the tests, it was found that sometimes a measurement device reported a RSS value that indicates weaker signal than the minimum -96 dBm, and sometimes the RSS was not reported at all. In such a case, the RSS is replaced by a predefined number, which is out of the normal measurement range, to indicate that the sample is actually missing.

In each CP, the data collection time was approximately five minutes with each device. The number of received signal samples varied between devices as their average scan intervals were different; they were approximately 8-10 s, 2.3 s, and 1.0 s for Mobile1, Mobile2 and the PC, respectively.

In both CPs and TPs, the data was collected only in one direction, which varied over the area. The TP data was collected in the same directions as data from nearby CPs. In TPs, the data collection time was only one minute with each device. Both the PC and Mobile2 report the received signal strengths in dBms, while Mobile1 reports Received Signal Strength Indicator (RSSI) values, which are positive and where higher value means weak signal. The first task was to find suitable model for

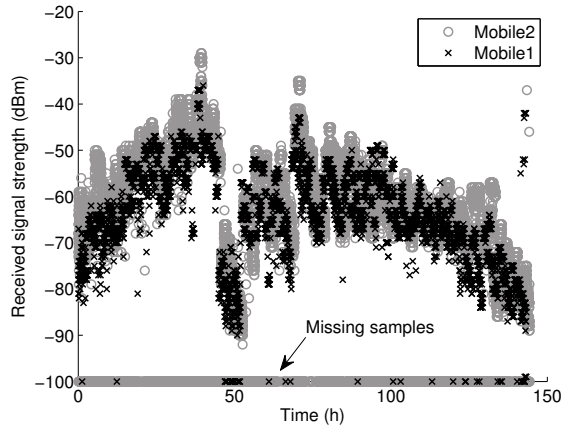


Fig. 3.10. *Mobile1 and Mobile2 measurements compared. Unit conversion of Mobile1: $RSS = -RSSI$.*

conversion between these RSS dBm and RSSI values. The following equation was used to convert RSSI values to RSS values:

$$RSS = -RSSI + b. \quad (3.1)$$

Without the offset in (3.1), i.e., when $b = 0$, the time domain observations of RSS values revealed significant differences between the devices. In Fig. 3.10, all the measurements by the both mobiles from one AP and from all the CPs, i.e., when the measurement location varies, are shown. The RSSs collected by Mobile2 are almost all the time 5-10 units higher than RSSs of Mobile1, obtained using (3.1) with $b = 0$. On the other hand, the number of missing samples is larger with Mobile2.

Estimation Results

The main interest was to find out how much the positioning accuracy changes when the radio map is created using the measurements of the PC or Mobile2, while the measurements for positioning estimation are done using Mobile1. To study how well a radio map that is computed using measurements of one device can be used for position estimation with RSS measurements from other device, we first evaluated the position accuracies using the same devices for radio map generation and position estimation. The radio maps were generated for Mobile1, Mobile2 and PC measurements using all the calibration data, and estimation tests were done using all test point sam-

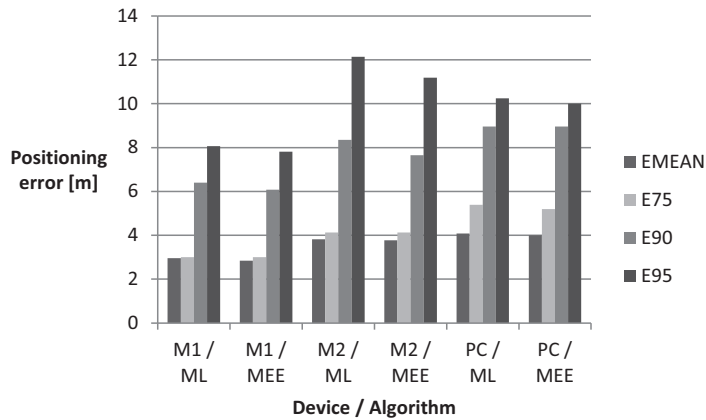


Fig. 3.11. Radio map data and independent test measurements collected using the same device.

ples of the corresponding devices. No conversions were made to measurements, i.e., the radio map of Mobile1 and the estimation were based on reported RSSI values directly, while the computations with Mobile2 and PC data were conducted using RSS measurements.

With each device, position was estimated using both ML and MEE algorithms. From estimated positions, we computed estimation errors, i.e., distances between true and estimated positions. From these, four error measures were computed: the mean value of estimation errors and 75th, 90th, and 95th percentiles of position errors. The results are graphically illustrated in Fig. 3.11. With respect to all the error measures, the best performance is obtained with Mobile1: mean errors below 3 m and 95th percentiles approximately 8 m. With respect to mean values and the lower error percentiles, Mobile2 performs better than PC, while PC has the lowest 95th percentile. In most cases, MEE algorithm yields smaller errors than ML. However, the differences are small, from a couple of centimeters to some tens of centimeters.

The applicability of Mobile2 or PC based radio map for position estimation with Mobile1 data was studied by generating radio maps from the calibration point measurements of Mobile2 and PC, estimating position with these radio maps and test point measurements of Mobile1, and finally computing statistics from the obtained position estimates. Both ML and MEE algorithms were used for estimation. The number of histogram bins for both algorithms was seven.

Before the generation of radio maps, the RSS measurements of Mobile2 and PC were

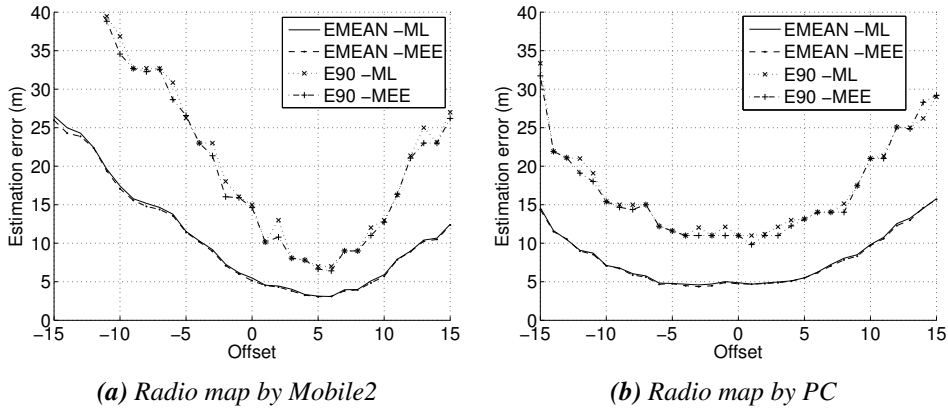


Fig. 3.12. Positioning using measurements of Mobile1 and radio maps based on different devices.

converted to RSSI values using

$$RSSI = -RSS + b. \quad (3.2)$$

The radio maps were generated using several different values of offset b , and the performance of the estimation was analyzed for each case. The mean errors and 90th percentile of error were computed for each test and both estimation algorithms. The results from tests with Mobile2 based radio map are shown in Fig. 3.12a, while Fig. 3.12b contains the results from tests with PC based radio map.

From Fig. 3.12 it can be observed that with the positioning measurements of Mobile1, the radio map by Mobile2 gives better estimation accuracy than the radio map by the PC. The smallest mean error with radio map of Mobile2 is approximately 3 m, at the same level as with Mobile1's own radio map. With the radio map by PC, the smallest mean error is about 5 m – this is even higher than the mean error 4 m obtained by using the measurements of the PC both in the radio map and in the estimation. With Mobile2 based radio map the optimum value of offset b is between 5 and 6. With PC based radio map, the range of b that yields errors that are very close to the minimum values is between -6 and 3: the minimum mean error is obtained with $b = -3$ and the minimum E90 is obtained with $b = 1$. The shapes of the error vs. offset value curves are different for Mobile2 and the PC; with the PC the offset range, which yields errors that are close to the minimum, is wider.

In the results shown in Figs. 3.11 and 3.12, the samples for radio map were taken from the whole 5 min span of the calibration data. However, to find out how well the

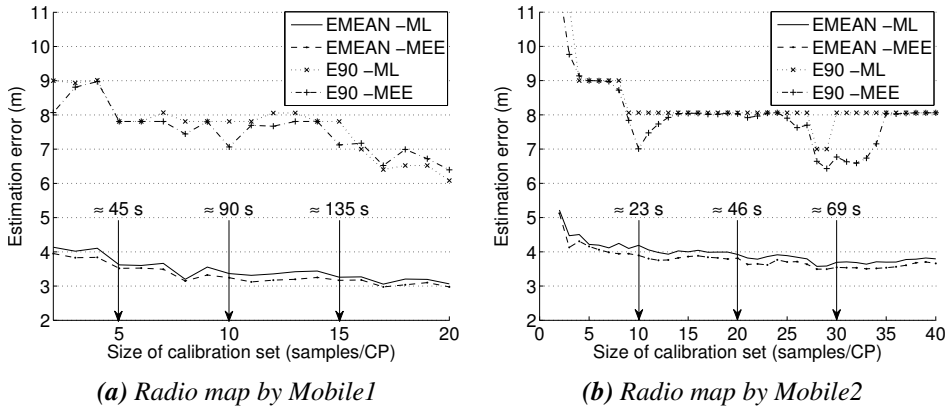


Fig. 3.13. Effect of the size of the calibration data set to the positioning accuracy of Mobile1, when radio maps were obtained using Mobile1 and Mobile2.

faster sampling of Mobile2 could be utilized, the effect of the size of the calibration data set was studied by computing the radio maps separately from the calibration data of Mobile1 and Mobile2 using different numbers of calibration samples per CP. The radio map by Mobile2 was calibrated for the use with measurements of Mobile1 by applying (3.2) with $b = 6$. The results of the performance evaluation using the test data of Mobile1 in the positioning phase are shown in Fig. 3.13.

In Fig. 3.13a, where both the radio map and position estimation were obtained using Mobile1, the positioning accuracy gradually improves as the number of calibration samples increases. To reach the levels of 3 m in the average error and 6 m in the 90th percentile of error obtained using all the calibration measurements (shown in Fig. 3.11), about 20 samples are required, equivalent of 180 s of data collection time. When radio map is based on Mobile2 and estimation is based on Mobile1 (Fig. 3.13b), in 30 samples or 70 s, the accuracy reaches the same level as which was obtained when using both calibration and test data of Mobile2, shown in Fig. 3.11. However, even with 40 samples and 92 s, the accuracy does not reach the same level as obtained using all the calibration data from the 5 min measurement period, shown in Fig. 3.12b. On the other hand, as this test does not include the results of calibration data collection times between 92 s and 300 s, we cannot make confident conclusion about whether the accuracy level close to the optimum shown in Fig. 3.12b could still be obtained with calibration data collection significantly shorter than 300 s.

As a conclusion, it is shown that normalization can be found between the devices

reported in the tests. Using a radio map based on 5 min calibration sets by Mobile2 that was normalized for Mobile1, the positioning accuracy with the test data of Mobile1 was comparable to the accuracy obtained by Mobile1 test data and radio map based on 3 min calibration data by Mobile1 itself. However, using only 1.5 min of the Mobile2 calibration data, the accuracy was only on the same level as it was when using both radio map and test data obtained with Mobile2. With positioning data of Mobile1, better accuracy can be achieved with the radio map obtained using Mobile2 than with the radio map obtained using the PC. The accuracy obtainable using normalized radio map depends on both the devices: the device that was used to obtain the radio map as well as on the device used to measure the test data. If the minor degradation of the accuracy is acceptable, then the time savings obtained by using normalization of the existing radio map is significant when compared to the situation when the calibration measurements for the radio map has to be repeated by all different device types for which the positioning is needed.

To conclude the results on the RSS normalization among different devices, the main findings are summarized as follows:

- The RSS normalization between different devices can be obtained using an offset correction
- The positioning with a normalized radio map obtained using another device can be as accurate as the positioning with the radio map generated using the same device that is used for positioning

The following questions regarding the normalization were not studied: 1) how many common CPs between the devices are needed in order to obtain adequate normalization; 2) how accurately the measurement points with different devices need to match. The cost of the normalization decreases as the number of required common CPs and the required CP localization accuracy decreases. Therefore more research is needed on these questions.

3.3 Discussion

In this Chapter, effects of several design parameters of fingerprinting based WLAN positioning were studied and compared. The evaluations were based on the obtained

positioning accuracy and memory requirements of the algorithms. The memory requirement is also an indicator of the computational complexity of the estimation phase, as the number of required unit operations is directly proportional to the stored model parameters.

Two histogram based algorithms, ML and MEE, were studied and compared. In the comparisons, the MEE algorithm mostly was more accurate than the ML algorithm. In the studies regarding the histogram bin configurations, it was found that the accuracy of the position estimation improves when the number of bins increases. However, when the number of bins reaches 8, the accuracy does not improve any more. It was also found that with numbers of bins less than 6, the accuracy is better, if instead of having all bins with equal widths, there is one narrow bin for the minimum RSS of the receiver and the other bins are wider and have equal widths.

In the tests using *rmap1* (distinct radio map entries for each measurement directions) the EMEAN and RMSE with 5NN were 6.87 m and 8.52 m, respectively, and the improvements from NN were 18% and 21%. The improvements are clearly larger than reported by Bahl and Padmanabhan (2000) for the similar setup but the errors are on higher level. Possible factors that cause differences in accuracy levels include e.g. the differences in the properties of the hardware and the radio propagation channel. To the latter, the shape of the test area has a significant contribution: in (Bahl and Padmanabhan, 2000), the test area consisted of narrow corridors, while in this thesis, the kNN method was tested in a more open library hall. In (Leppäkoski et al., 2009a), the positioning errors in the test area consisting mainly corridors were only about 60% of the errors in hall-like test area. A similar difference in accuracy levels can be observed also in the results of histogram based algorithms presented in Sections 3.2.3 and 3.2.4 with hall-like test area and in 3.2.6 with a corridor-like test area, the latter providing clearly better accuracy.

The measurements from different orientations were combined in *rmap2* using averaging. With NN the averaging over the directions improved the accuracy very slightly, in the EMEAN by 0.01 m, and in RMSE by 0.06 m, which corresponds to less than 1% decrease. With 5NN, the accuracy with the combined orientations of *rmap2* degrades: compared to the results with *rmap1* (without averaging), the EMEAN and RMSE increase by 0.5 m to 0.6 m, i.e., in both cases the errors are 7% larger with *rmap2*. In Bahl and Padmanabhan (2000) the best accuracy with the combined directions was obtained using 3NN, when the accuracy improvement was clear. However,

the results regarding the effect of the orientation in this thesis cannot directly be combined with the results by Bahl and Padmanabhan (2000), as different functions were used to summarize the RSS in these studies. With the histogram based algorithms, the best accuracy was obtained using *rmap2*, where the measurements in one CP, from one AP, and from all measurement directions were combined to one histogram.

A kNN algorithm, which used averaging to summarize the RSS measurements, was compared with histogram based algorithms. Among kNN with $k = 1, \dots, 5$; the best accuracy was obtained with $k = 5$, and this algorithm was compared with histogram based ML and MEE algorithms using several different bin configurations. The most important results are collected into Table 3.1, where the following algorithms are presented:

- 5NN using *rmap1*: the most accurate kNN implementation studied in this thesis
- MEE algorithm using 8 bins with uneven bin widths, as MEE was more accurate than ML, and increasing the number of bins from 8 did not significantly improve the accuracy any more
- MEE algorithm using 3 bins with uneven bin widths: uneven bin width is used because with small number of bins it gives better accuracy than even bin widths, and with number of bins larger than 3, the accuracy improvement as a function of the number of bins starts to settle
- MEE algorithm using 8 bins with uneven bin widths and combining in radio map the measurements from correlating sources – in our case there were the 4 adjacent transmitters included in the same AP device

From the Table 3.1 it can be seen that the EMEAN with the best histogram based implementation was about 1 m smaller than with 5NN algorithm. In RMSE and E95 the differences between 5NN and MEE are smaller than in EMEAN, and with EMAX the sign of the difference changes: the 5NN has smaller maximum errors than the MEE implementations. Because in Table 3.1 the 5NN uses *rmap1* with separate entries for the measurement directions and MEE uses *rmap2* with combined directions, the MEE implementation required only twice the number of parameters required by 5NN. It is also observed, that the number of parameters required by the

Table 3.1. Comparison between 5NN and MEE algorithms

Algorithm	5NN	MEE	MEE	MEE
Number of bins	n/a	3	8	8
Radio map	<i>rmap1</i>	<i>rmap2</i>	<i>rmap2</i>	<i>rmap2</i>
APs combined	–	–	mean2	–
Parameters/CP/AP	4	3	2	8
EMEAN (m)	6.9	6.9	6.8	6.0
RMSE (m)	8.6	8.7	8.6	8.0
E95 (m)	17.5	17.8	17.6	17.2
EMAX (m)	27.1	28.8	28.8	30.6

histogram can be decreased below the number required by 5NN by using less bins or by combining the highly correlated APs in the radio map, and still the accuracy is on about the same level as with the 5NN algorithm. The results in the Table 3.1 can be summarized as follows:

- With 2–4 parameters/CP/AP, the EMEAN is 6.9 m; with 8 parameters/CP/AP, it is 6.0 m
- With 2–4 parameters/CP/AP, the RMSE is 8.6 m; with 8 parameters/CP/AP, it is 8.0 m
- The benefit of doubling the radio map size is only less than 1 m

The main objective in the RSS normalization was to decrease the time required to generate radio maps by using radio map data of one device also in the positioning phase of other devices. It was shown that the normalization can be found, and with some devices, the positioning with normalized radio map can be as accurate as with the radio map generated by the same device that is used in the positioning phase. However, in this work the offset parameter for the conversion was obtained in positioning tests using a large amount of data in search of the correct parameter. A procedure requiring significantly fewer data samples for the parameter estimation is needed to save time.

Scalability of the fingerprinting methods was not addressed in this thesis. However, this is an important issue, as it is obvious that the limits exist to the size of the radio map that can be handled in a smart phone. Youssef et al. (2003) proposed a clustering

of the locations in a radio map and selecting the radio map portion to be used in the estimation by using the set of the strongest hearable APs as a cluster key. As the MAC addresses of the APs are unique, these can be used to globally search the required radio map data to be downloaded to the phone. The smart phones can also have a coarse estimate of their locations from the cellular network base station they are connected to or from their previous available GNSS based position, which could also be used as the key to find the local radio map.

4. AIDING PDR WITH INDOOR MAP AND WLAN POSITIONING

Several information sources alternative to GNSS have been proposed for pedestrian indoor positioning. A DR system based on inertial MEMS sensors is ideal for indoor navigation in many aspects: the inertial sensors are self-contained, they do not need any external fields for their operation and they cannot be disturbed or spoofed by external sources. The MEMS sensors are also small-sized, light-weight, and have relatively low power consumption. However, the errors in unaided PDR systems grow without bounds. To limit the error growth, other sources of position information need to be combined with DR, e.g., WLAN or Bluetooth signals can be used to obtain absolute position estimates and map based navigation aiding can be used to constrain the PDR track.

All the mentioned approaches for indoor positioning have their strengths and weaknesses. Generally the combination of several sources brings better performance than a single source alone. There are also situations where radio network based positioning is not practical. The radio network infrastructure is vulnerable to fire, power outages and collapses, which could make radio network based positioning useless when positioning fire fighters, first responders, police etc. in emergency conditions. Therefore it is important to have also a possibility to navigate by using a PDR aided with a map only.

Different variations of algorithms, mainly based on Bayesian filtering, have been proposed for fusing the indoor navigation data from diverse types of sensors and signals in different combinations. While the general principles of PDR, WLAN positioning, indoor map, and data fusion algorithms are presented in Section 2, this Chapter reviews their applications in pedestrian and indoor navigation.

In this Chapter, models and algorithms are proposed for combining information from low-cost MEMS inertial sensors, indoor map, and WLAN signals for pedestrian indoor navigation. Several field tests show the potential of the proposed methods. Re-

sults of the field tests are presented where MEMS based PDR estimates were fused with WLAN RSS based positioning, indoor map information and both of them. Complementary extended Kalman filter (CEKF) was used to fuse together WLAN RSS based position estimates and PDR estimates based on an inertial sensor unit including a heading gyro and three-axis accelerometer. Particle filters were used to combine the PDR data with map information and WLAN position or map information only. In the particle filters, the state model was the same as in the CEKF, i.e., the PDR estimates were used to predict the new particle states.

One goal of this work was to design a filter and measurement processing that is suitable also for real time position estimation; this was demonstrated with a distributed indoor positioning system including inertial measurements and map matching.

4.1 Related Work

In addition to the absolute positioning methods described in Section 3.1, the alternatives to GNSS for indoor positioning include also relative positioning methods, such as Dead Reckoning (DR) based on sensors that observe motion and direction. However, to improve the long term accuracy of DR, it is usually combined with some absolute positioning method.

4.1.1 Motion Sensors and Dead Reckoning

Even with the most accurate WLAN RSS fingerprinting approaches large occasional positioning errors are possible, especially indoors. The situation can be improved using filtering to mitigate the effect of large errors; even better results can be obtained using a filtering scheme that utilizes the information about the motion characteristics of the MU. The precise motion characteristics are generally unknown to the filter designer. This can be taken into account by describing the MU motion using stochastic motion models (Syrjärinne and Saarinen, 1999).

Another option to take into account the motion characteristics of the MU is to measure the MU movements using motion sensors, such as accelerometers or odometers to obtain the distance traveled and gyroscopes or electromagnetic compasses to obtain the direction of the travel. With these sensors, DR algorithm can be used to update

the information about the previous known position with new measurements on the direction and the distance traveled (Groves, 2008). Accelerometers, gyroscopes and electromagnetic compasses based on MEMS technology are light-weight and low-power devices and, therefore, well-suited for pedestrian navigation (Levi and Judd, 1996). Inertial sensors, i.e., accelerometers and gyroscopes have a special property: they are self-contained, i.e., they do not require any external signals or fields for their operation and external fields cannot interfere with them, and they perform equally well both indoors and outdoors (Titterton and Weston, 2004).

Pedestrian dead reckoning takes into account the special characteristics of the movements of a person on foot and provides better accuracy than the DR based on the traditional INS mechanization (Mezentsev et al., 2005). When a foot mounted sensor units are used in PDR, the traditional INS mechanization can be used to estimate the distance traveled. During the stance phase between the steps, the algorithm performs Zero-Velocity Update (ZUPT) to reset the inertial errors (Elwell, 1999; Foxlin, 2005). When sensors are not placed on foot, the ZUPT cannot be used. In these cases, the distance traveled can be estimated from the periodical acceleration waveform produced by pedestrian movements (Levi and Judd, 1996; Ladetto, 2000; Käppi et al., 2001; Meriheinä, 2007).

DR type of data is also available in vision based navigation using camera cells. For example, the processing of image sequences can be used to provide DR information to aid some absolute positioning method (Veth, 2011) and odometry based position estimate can be aided using a vision system to find the landmarks that are expected to be visible from the current position of the robot (Borenstein et al., 1997). The vision systems require high computational power and still are not capable of autonomous navigation (Ruotsalainen, 2013).

4.1.2 Integration of Navigation Systems

With an ideal DR, the MU motion could be perfectly known. In the real world, the sensor readings always suffer from measurement errors to some extent and the initial estimates include some uncertainty. In a DR system, the accuracy of the position estimate is affected by the accumulating effects of the errors in the initial estimates of the position and heading and the sensor and modeling errors. Due to the accumulation of the errors, the DR position error can grow without bounds. Therefore, the operation

time of an unaided DR is limited. This problem can be mitigated by integrating with a DR system some external position information, such as WLAN positioning, so that the errors can be periodically corrected using the external source. Also maps provide position information suitable for correcting DR errors. The use of map information is common practice in car navigation (Quddus et al., 2007), and similar principles can also be applied in indoor positioning (Evennou et al., 2005; Wang et al., 2007; Widyawan et al., 2008; Woodman and Harle, 2009; Bhuiyan et al., 2012). The map provides constraints to the position and can be used to correct DR errors when they cause the MU track to violate the map constraints.

Data fusion algorithms are used to combine or integrate information from several sources. For data fusion in position and navigation applications, Bayesian filtering algorithms are commonly used, such as Kalman Filter (KF) or its nonlinear variant Extended Kalman Filter (EKF) (Sorenson, 1966), and Sequential Monte Carlo (SMC) based methods, such as particle filters (Arulampalam et al., 2002). When DR data is available, the other sources are usually redundant, and in these cases complementary filters can be used (Brown, 1972-73; Brown and Hwang, 1997). Applications of CEKF to vehicular navigation have been presented, e.g., by Yang et al. (2000) and Qi and Moore (2002).

In the data fusion algorithms, mathematical models are needed: process models that describe how the new filter states depend on the previous ones, measurement models that describe how the measurements depend on the filter states, and stochastic models that describe the uncertainties regarding the initial states, state propagation, and measurement errors. The basic filtering algorithms can be found easily from text books (e.g., Brown and Hwang, 1997; Grewal and Andrews, 2001; Candy, 2009), but the modeling is left to the implementers: how to choose the filter states and measurements, what real world effects to include into the model, which less significant effects can be ignored, how to mathematically express the relations between the states, measurements, and the uncertainties involved, and how to choose the adjustable parameters of the model. Also the implementation details need consideration, e.g., the issues related to the computational load.

4.1.3 Pedestrian Indoor Navigation

Gabaglio (2001) proposed a CEKF in feedback configuration for pedestrian naviga-

tion using a single-axis gyro, two-axis accelerometer, and GPS receiver. It can be noticed that the authors of the reports do not necessarily mention that their filter is a complementary filter; however, the complementary filter structure can be inferred from the way how the filter states and its measurement inputs are chosen. Frank et al. (2009) proposed an indoor pedestrian positioning system including a CEKF in ERT configuration to combine WLAN fingerprinting and foot mounted sensor unit and EKF with ZUPT to produce the PDR estimate.

Davidson and Takala (2013) proposed a body-mounted 6DOF IMU for pedestrian navigation in a CEKF feedback configuration where the traditional strap-down mechanization of the IMU was used to produce the reference trajectory. This was corrected using the velocity estimates obtained with step detection and step length estimation. However, the heading cannot be corrected using this method.

The PDR combined with map matching but without particle filters has also been proposed. (Gilliéron et al., 2004) proposed a body-mounted PDR unit that was aided with node-link based map-matching using point-to-point and point-to-link matching methods. Fang et al. (2005) proposed a system of a body-mounted sensor unit and radio network based positioning, where the user can assist the positioning by executing a manual map-matching through the graphical user interface of the system. Glanzer et al. (2009) proposed system that combines the track generated using a foot-mounted IMU with the building information model consisting of polygons defining rooms, staircases, doors, and transitions between them. The authors state that their method is not based on particle filtering or any other known map matching method but they do not describe the method in more detail.

The PDR combined with WLAN positioning and node-link based map-aiding with particle filters has been proposed by e.g. Khan and Syrjärinne (2013). Evennou et al. (2005) proposed the fusion of WLAN fingerprinting and node-link based map-aiding by using particle filter without motion sensors. Indoor positioning systems based on WLAN positioning and map-aiding particle filters without a PDR or with only a partial PDR are proposed also by Wang et al. (2007) and Widyawan et al. (2008). The former proposed a system with step lengths obtained from a sensor unit but without sensor based heading estimate, while the latter proposed a system using motion models without sensors.

Beauregard et al. (2008) and Woodman and Harle (2008) proposed very similar sys-

tems to the one used in this thesis, based on particle filter and map-aiding with obstacle models. However, in their systems, foot-mounted sensor units were used. Although the foot-mounted PDR has some advantages, such as easier tuning and algorithm development (Groves et al., 2007), it also has its disadvantages: due to its position it is vulnerable to mechanical knocks and it is difficult to protect unless it is not embedded into the structure of the shoe, in which case the power supply and maintenance of the unit becomes difficult. The foot-mounted unit is also more susceptible to shocks, which significantly increase the gyro errors (Davidson and Takala, 2013).

The systems with similar components as in the one used in this thesis, including a body-mounted PDR and map-aiding particle filter with obstacle map were proposed by Woodman and Harle (2008), Kemppi et al. (2010), Kirkko-Jaakkola et al. (2013), and Nurminen et al. (2013). In these works, the methods for decreasing the computational load of the map checks were not considered, except in (Nurminen et al., 2013), where MATLAB's MEX-files were used to speed up the computation in the most critical parts of the algorithm.

The initialization of the navigation states, such as the coordinates and the heading, has a significant effect on the performance of the navigation filter. If the initial information is known to be inaccurate, the diagonal elements of the initial covariance of the EKF are set to large values and in particle filter, the initial values are drawn from distributions with large variances. In this case, the convergence of the filter states towards their true values is slow. Therefore, methods for decreasing the initial uncertainty has been discussed by many researches, e.g., Woodman and Harle (2009); Kemppi et al. (2010); Kirkko-Jaakkola et al. (2013); Nurminen et al. (2013). RF-based methods to initialize the particle filters are used in (Woodman and Harle, 2009) and (Kemppi et al., 2010); in (Kirkko-Jaakkola et al., 2013) the filter is initialized indoors using a high- sensitivity GNSS receiver.

In this thesis, WLAN positioning and map matching were used to limit the PDR error growth. The well-developed Bayesian approach in the form of particle filters was chosen to allow automatic map-aiding and map-aiding without WLAN positioning was also studied. To present the map information, an obstacle model is preferred over the node-link model. Although the node-link model requires less particles and is computationally less demanding, in PDR integration the latter has the advantage that it allows the estimates to use all the available space: with the obstacle model only

the impossible transitions are prohibited while in node-link model all the estimates are forced to the most probable routes. The map matching without PDR was not considered, because the PDR allows the map-aided indoor navigation even without the aid of a radio network based positioning. Due to its mechanical robustness and the ease of its maintenance, the body-mounted sensor unit was chosen to provide the PDR information.

The computational load of the map checks was decreased by developing novel algorithms for map checks. With these, the obstacle lines that are outside the area where the collisions with the particles are possible, were left out of the map checks. A CEKF fusing WLAN positioning and PDR estimates from a body-mounted IMU was compared to map matching particle filters. The alternative CEKF configurations, i.e., the ERF and the feedback configurations, were also compared with each other. The availability of the methods to decrease the initial uncertainty of navigation states was assumed: the GPS was used in one test, while in others the errors of the initial values from their true values and the initial variances were simulated based on the assumed outdoor-to-indoor entrance scenario.

4.2 Methods for Data Processing and Fusion

This Section describes the methods used in the performance evaluations of the map aided indoor navigation. The emphasis is on the implementation details and modifications to the general methods described in Section 2.

4.2.1 PDR Preprocessing

In order to use step detection based PDR, the step length estimate is needed. In this thesis, the step length estimate was computed using a linear model relating the step frequency, i.e., the inverse of step duration (2.11) to the step length. With this signal pattern, the exponent $q = 1$ was used in model (2.17):

$$\Delta s_k = \frac{c_{sc}}{t_e(k) - t_s(k)} + b. \quad (4.1)$$

To obtain the calibration parameters c_{sc} and b , ten sets of walking data were collected in a straight corridor using an accelerometer triad. The straight path of a known

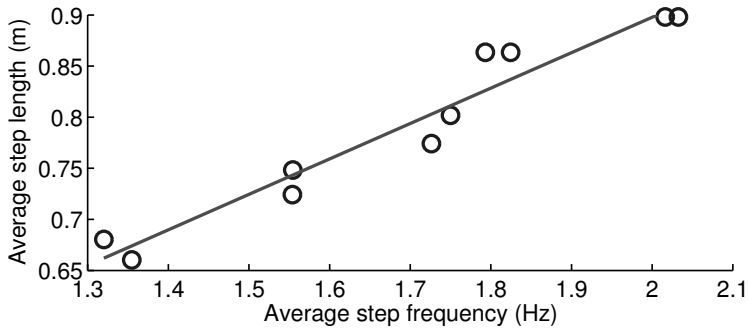


Fig. 4.1. Step length calibration using 10 test data sets to obtain c_{sc} and b . Observations shown with circles.

length was walked ten times. As it is known that the step length is also a function of the walking speed (Levi and Judd, 1996), the walking speed was varied in the test walks in order to collect step samples with different step lengths. The walker tried to adjust the walking speed to normal, slower than normal, slow, faster than normal, and fast; each of the speeds were used in two test walks.

With the data, the steps were detected from the acceleration norms and step intervals were determined. Using step intervals averaged over each walk, the number of detected steps per walk, and the known length of total traveled distance per walk, a linear fit can be found between average step frequencies and step lengths of each test data set, as shown in Fig. 4.1. These parameters can be used in real-time to estimate step length. In regular walking, this gives fairly good estimates and other methods, e.g., map matching, can be used to fix small errors.

For the distributed indoor positioning system the algorithms for step detection and step length estimation, described in Section 2.2, were needed in real time. The state-machine description of the implemented step detection algorithm is shown in Fig. 4.2. In the figure, the binary states are indicated by circles. In the states drawn with dotted double line, the algorithm produces output, and in the state drawn with solid double line, all the memory variables are zeros. The possible state transitions are indicated by the arrows connecting the states. The text associated to the arrows consists of two parts: before the slash (/) is shown the event or the condition that activates the state transition and after the slash the actions related to the state transition are listed. In this algorithm, the actions are changes to the memory variables.

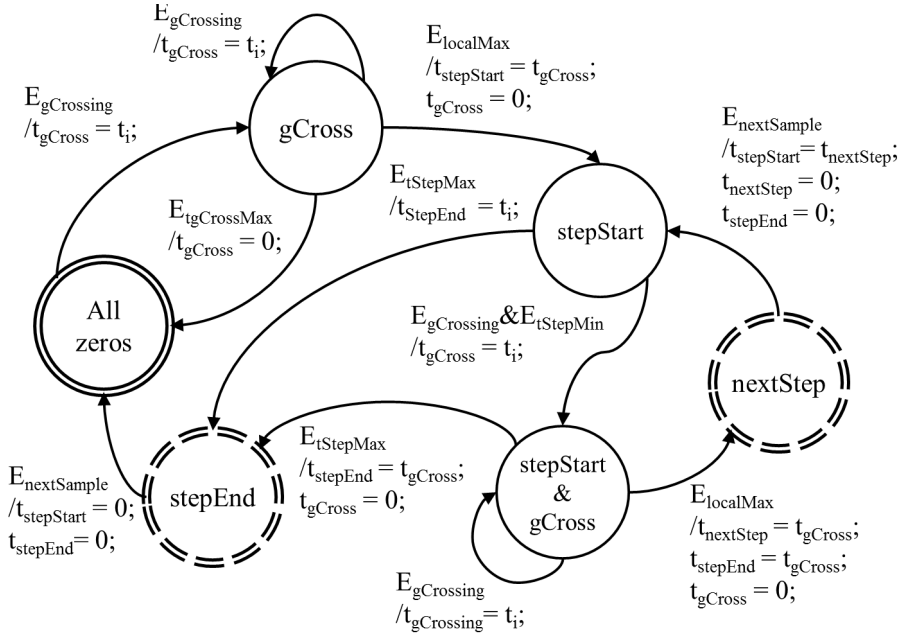


Fig. 4.2. State diagram of the step detection algorithm. The function outputs the ‘Step detected’ status and the values of $t_{stepStart}$ and $t_{stepEnd}$ when the state is stepEnd or nextStep. The current time is denoted by t_i . The notation is described in Table 4.1.

The events that activate the state transitions, the required threshold parameters, and the memory variables are described in Table 4.1. The step detection algorithm includes constraints for proper steps, such as thresholds for minimum time between g-crossings (t_{trMin}) and minimum value for peak acceleration in detected step (a_{trMin}). The algorithm considers g-crossings as possible new step starts and checks the constraints to accept or reject the start. The end of the current step is found in two situations: when the start of the next step is accepted, or when the start of the next step does not follow the current step in the predefined time frame (t_{trMax}). The latter occurs when the user stops walking. The algorithm outputs both the start time $t_s(k) \leftarrow t_{stepStart}$ and the end time $t_e(k) \leftarrow t_{stepEnd}$ of the k^{th} step either when the start of the next step is detected or when step end is concluded from the detection timeout. The symbols used in Fig. 4.2 and Table 4.1 are defined as follows: t_i is the measurement time of the current sample, a_i is the norm of the filtered acceleration at t_i , g is the gravitational acceleration, and t_{trMax} , t_{trMin} , and a_{trMin} are the threshold parameters of the algorithm. Variables t_{gCross} , $t_{stepStart}$, $t_{nextStep}$, and $t_{stepEnd}$ include the memory of the algorithm.

Table 4.1. Events that activate the state transitions, threshold parameters, and memory variables of the step detection algorithm described in Fig. 4.2

$E_{gCrossing}$	g-crossing found on rising signal: $a_{i-1} < g \leq a_i$
$E_{t_{gCrossMax}}$	timeout, time from g-crossing exceeds limit: $t_i - t_{gCross} > t_{trMax}$
$E_{t_{StepMax}}$	timeout, time from step start exceeds limit: $t_i - t_{stepStart} > t_{trMax}$
$E_{t_{StepMin}}$	new step start possible: $t_i - t_{stepStart} \geq t_{trMin}$
$E_{l_{localMax}}$	found peak high enough: $a_{i-2} < a_{i-1} \ \& \ a_{i-1} \geq a_i \ \& \ a_{i-1} \geq a_{trMin}$
$E_{nextSample}$	new sample a_{i+1} at t_{i+1} available
t_{trMax}	maximum duration of a step
t_{trMin}	minimum duration of a step
a_{trMin}	peak minimum during walk
t_{gCross}	time of g-crossing
$t_{stepStart}$	time of step start
$t_{nextStep}$	time of the next step start
$t_{stepEnd}$	time of detected step end

The heading change estimation relies also on the step detection; in order to use the estimated heading change and step length together in the dead reckoning process, their sampling intervals must be the same. Therefore, the results of the step detection are used to obtain the integration limits for the angular rate ω ; to estimate the heading change $\Delta\Psi_k$:

$$\Delta\Psi_k = \frac{1}{f_g} \sum_i \omega_i, \quad \left(\frac{t_s(k)}{f_g} \leq i < \frac{t_e(k)}{f_g} \right) \quad (4.2)$$

where f_g is the sampling frequency of the gyro measurements and i is the sample counter.

After obtaining the heading change during a step using (4.2), the PDR preprocessing algorithm outputs it together with the step length. However, gyro output needs to be integrated also during the stops between the intervals of continuous walk. For example, in order to walk through a closed door, the user usually needs to stop in order to open the door, and the opening action typically causes the user body and the body mounted gyro to turn slightly, after which the walking can continue before the body angle has fully bounced back from the door opening. If the gyro readings during this stop are not taken into account, a permanent error is introduced into the heading estimate. Therefore, the implemented algorithm restarts the integration of gyro outputs after each step end, and continues integration even if a step start does

not follow immediately. After a stop, when the algorithm eventually detects the start of a new step, it outputs zero step length and the heading change integrated over the stop time; this does not affect the total distance traveled, but ensures that the heading estimate is up to date.

In the PDR mechanization in two dimensions (see Fig. 2.4), the position estimate is computed by starting from the initial coordinates, x_0 , y_0 , and the initial heading angle ψ_0 . As the DR method is not able to determine absolute positions, these initial estimates have to be determined using alternative positioning methods, such as radio navigation or satellite based positioning.

In unaided PDR, the heading and horizontal coordinates can be propagated using

$$\psi_k = \psi_{k-1} + \Delta\psi_{k-1} \quad (4.3a)$$

$$x_k = x_{k-1} + \Delta s_{k-1} \cos(\psi_{k-1}) \quad (4.3b)$$

$$y_k = y_{k-1} + \Delta s_{k-1} \sin(\psi_{k-1}). \quad (4.3c)$$

In (4.3) the estimates based on step detection and step length estimation are available at step intervals Δt_k ; Δt_k is the time span between t_k and t_{k+1} . Generally the step interval is time-variant as it is affected by the walking style and the speed of the pedestrian.

In this thesis, the estimated step length and the heading changes are also used in the the fusion filter, i.e., CEKF or particle filter, to keep track of the accumulating total heading and coordinates. The initial heading and coordinates are defined as the initial conditions of the filter.

As the estimation of the distance traveled is based on the shape of the acceleration signal rather than the integration of the signal, the algorithm is quite robust against the usual accelerometer bias or scale factor errors. However, gyro errors are more critical. When the sensor unit is attached to the torso, the sensitive axis of the gyro is often tilted from the vertical, which is seen as a scale factor error in the gyro output. The tilt also varies slightly during the walk movement. In this thesis the average of this error was estimated and used to correct the heading change estimates. The tilt correction needs to be estimated once for each user, because even when the unit is attached to the same position on the torso, the tilts are different on different users. In the systems implemented in this thesis, the correction of the effects of the other gyro errors, such as bias and noise were left to the fusion filter.

4.2.2 Complementary Extended Kalman Filter

In this thesis, the complementary filter is used to integrate PDR estimate with WLAN positioning. As the model is non-linear, EKF is used, and due to the complementary filter formulation, the resulting filter is CEKF. The CEKF can be implemented in two different forms: it can be formulated as a feedback configuration, sometimes called an error state KF, or as an Embedded Reference Trajectory (ERT) form. The models for both the alternatives are derived in the following.

Embedded Reference Trajectory Configuration

The process model for the CEKF is based on the DR equations (4.3). If the initial state $[\Delta\psi_{k-1}, x_{k-1}, y_{k-1}]^T$ was perfectly known and the heading change and the step length estimated without errors, the system state after next step could be determined accurately using (4.3). However, the PDR outputs are corrupted by errors and the initial states are uncertain to some extent. As the errors of the PDR outputs are difficult to predict and they vary in time, they are modeled as white noise processes. They can be expressed as the sums of their true values and their errors:

$$\Delta\psi_k = \Delta\psi_k^* + n_{\Delta\psi_k} \quad (4.4a)$$

$$\Delta s_k = \Delta s_k^* + n_{\Delta s_k} \quad (4.4b)$$

where $\Delta\psi_k^*$ and Δs_k^* are the true values of the heading change and the step length, respectively, and $n_{\Delta\psi_k}$ and $n_{\Delta s_k}$ are the errors in their PDR based estimates.

By expressing the true (but unknown) DR outputs using the known estimated DR outputs and their unknown errors, the PRD equations can be written as

$$\psi_k = \psi_{k-1} + \Delta\psi_{k-1} - n_{\Delta\psi_{k-1}} \quad (4.5a)$$

$$x_k = x_{k-1} + \Delta s_{k-1} \cos(\psi_{k-1}) - n_{\Delta s_{k-1}} \cos(\psi_{k-1}) \quad (4.5b)$$

$$y_k = y_{k-1} + \Delta s_{k-1} \sin(\psi_{k-1}) - n_{\Delta s_{k-1}} \sin(\psi_{k-1}). \quad (4.5c)$$

By denoting

$$\mathbf{x}_k = \begin{bmatrix} \psi_k \\ x_k \\ y_k \end{bmatrix}, \mathbf{u}_k = \begin{bmatrix} \Delta\psi_k \\ \Delta s_k \end{bmatrix}, \mathbf{w}_k = - \begin{bmatrix} 1 & 0 \\ 0 & \cos(\psi_k) \\ 0 & \sin(\psi_k) \end{bmatrix} \begin{bmatrix} n_{\Delta\psi_k} \\ n_{\Delta s_k} \end{bmatrix}, \quad (4.6)$$

the propagation of the state can be expressed a function of \mathbf{u}_{k-1} and \mathbf{w}_{k-1} . The equations (4.5) written in the form of the EKF state model (2.31) are

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} 1 & 0 \\ 0 & \cos(x_{1,k-1}) \\ 0 & \sin(x_{1,k-1}) \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \quad k = 1, \dots \quad (4.7)$$

where the noise variance $E[\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T] = \mathbf{Q}_{k-1}$. The m^{th} element of the vector \mathbf{x} at t_k is denoted by $x_{m,k}$. The input \mathbf{u}_{k-1} is deterministic, as its values are perfectly known once estimated by the PDR preprocessing algorithm. However, these known values differ from the true values. This is taken into account by the noise input \mathbf{w}_{k-1} . The sine and cosine terms make the noise model adaptive to the heading.

The state prediction (2.33) of the filter is

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \quad (4.8)$$

where

$$\mathbf{G}_{k-1} = \begin{bmatrix} 1 & 0 \\ 0 & \cos(\hat{x}_{1,k-1}) \\ 0 & \sin(\hat{x}_{1,k-1}) \end{bmatrix}. \quad (4.9)$$

For covariance prediction, (4.8) is linearized using (2.34) to obtain the approximation of the state transition matrix:

$$\mathbf{F}_{k-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\sin(\hat{x}_{1,k-1})u_{2,k-1} & 1 & 0 \\ \cos(\hat{x}_{1,k-1})u_{2,k-1} & 0 & 1 \end{bmatrix}. \quad (4.10)$$

The variance \mathbf{Q}_{k-1} is approximated as functions of the variance of step length noise $\sigma_{\Delta s}^2$, the variance of heading change noise $\sigma_{\Delta \psi}^2$, and the estimated heading $\hat{x}_{1,k-1}$:

$$\mathbf{Q}_{k-1} = \mathbf{G}_{k-1} \begin{bmatrix} \sigma_{\Delta \psi}^2 & 0 \\ 0 & \sigma_{\Delta s}^2 \end{bmatrix} \mathbf{G}_{k-1}^T = \begin{bmatrix} \sigma_{\Delta \psi}^2 & 0 & 0 \\ 0 & \cos^2(\hat{x}_{1,k-1})\sigma_{\Delta s}^2 & \sin(\hat{x}_{1,k-1})\cos(\hat{x}_{1,k-1})\sigma_{\Delta s}^2 \\ 0 & \sin(\hat{x}_{1,k-1})\cos(\hat{x}_{1,k-1})\sigma_{\Delta s}^2 & \sin^2(\hat{x}_{1,k-1})\sigma_{\Delta s}^2 \end{bmatrix}. \quad (4.11)$$

The variances $\sigma_{\Delta s}^2$ and $\sigma_{\Delta \psi}^2$ are adjusted to match with the observed uncertainty in the heading change and the step length estimated from the IMU measurements. The

approximation of the heading as a constant during a step causes some errors and the possible sensor unit tilts during the walk cause small misscalings to the angular rate measurements of the vertical axis gyro. Therefore, in the filtering algorithm the heading change uncertainty $\sigma_{\Delta\Psi}^2$ has to be larger than effect of the pure sensor errors. In $\sigma_{\Delta s}^2$ the main contributors are the modeling errors in the step detection and step length estimation.

In this filter, the measurement input is $\mathbf{z}_k = [x_{W_k}, y_{W_k}]^T$ where x_{W_k} and y_{W_k} are the x and y coordinates estimated using WLAN fingerprints, respectively. The same variance σ_W^2 is used for both the coordinates. The measurement equation is

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k \quad (4.12)$$

where

$$H = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad E[\mathbf{v}_k \mathbf{v}_k^T] = R = \begin{bmatrix} \sigma_W^2 & 0 \\ 0 & \sigma_W^2 \end{bmatrix}.$$

No linearization is needed for (4.12) as it is already linear. The derived CEKF algorithm is summarized in Algorithm 4.1. The pseudocode is written using the syntax described in (Brito, 2009) and comments are indicated by enclosing them between $/*$ and $*/$. The filter is started from the initial estimate $\hat{\mathbf{x}}_0$ and the initial covariance P_0 , which are set according to the best available estimate about the initial position and heading and the uncertainty of their initial information.

Feedback Configuration

To derive the equations for the CEKF in feedback configuration, also known as the error state filter, the heading and the coordinates are expressed as the sums of DR estimates and estimation errors

$$\Psi_k^* = \Psi_k + \delta\Psi_k \quad (4.13a)$$

$$x_k^* = x_k + \delta x_k \quad (4.13b)$$

$$y_k^* = y_k + \delta y_k \quad (4.13c)$$

where Ψ_k^* , x_k^* , and y_k^* are the true heading and coordinates, Ψ_k , x_k , and y_k are the estimates obtained by the DR algorithm running on the change estimates $\Delta\Psi_k$ and Δs_k defined in (4.4), and $\delta\Psi_k$, δx_k , and δy_k are the errors in the DR estimates. In

Algorithm 4.1 CEKF in embedded reference trajectory configuration

-
- 1) Initialize state $\hat{\mathbf{x}}_0$, variance P_0 , step counter $k = 0$
 - 2) Wait until new estimate from PDR or WLAN positioning available
 - 3) **if** New PDR estimates \mathbf{u}_{k-1} available (step detected) **then**
 - 4) $k \leftarrow k + 1$ /* step counter */
 - 5) Prediction
 - Predict state using (4.8)
 - Linearize state equation using (4.10)
 - Compute state noise covariance using (4.11)
 - Predict state covariance: $P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$
 - 6) **else if** New WLAN positioning estimate available **then**
 - 7) Measurement update
 - Compute the gain $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$
 - Update the state $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - H_k \hat{\mathbf{x}}_k^-)$
 - Update the state covariance $P_k = (I - K_k H_k) P_k^-$
 - 8) **end if**
 - 9) Go to line 2
-

the filter, the knowledge on the error states $\delta\psi_k$, δx_k , and δy_k is used to correct the DR estimates. Thus the algorithm is split into two parts: DR algorithm based on the incomplete sensor information and error state filter that provides corrections to the DR estimates.

By using (4.13), (4.5) can be rewritten as

$$\psi_k^* = \psi_{k-1} + \delta\psi_{k-1} + \Delta\psi_{k-1} - n_{\Delta\psi_{k-1}} \quad (4.14a)$$

$$x_k^* = x_{k-1} + \delta x_{k-1} + (\Delta s_{k-1} - n_{\Delta s_{k-1}}) \cos(\psi_{k-1} + \delta\psi_{k-1}) \quad (4.14b)$$

$$y_k^* = y_{k-1} + \delta y_{k-1} + (\Delta s_{k-1} - n_{\Delta s_{k-1}}) \sin(\psi_{k-1} + \delta\psi_{k-1}). \quad (4.14c)$$

Using the first order Taylor's series expansion the sine and cosine terms in (4.14) can be approximated as

$$\cos(\psi_{k-1} + \delta\psi_{k-1}) \approx \cos(\psi_{k-1}) - \sin(\psi_{k-1}) \delta\psi_{k-1} \quad (4.15a)$$

$$\sin(\psi_{k-1} + \delta\psi_{k-1}) \approx \sin(\psi_{k-1}) + \cos(\psi_{k-1}) \delta\psi_{k-1}. \quad (4.15b)$$

By substituting (4.15) into (4.14) and rearranging the terms the following is obtained:

$$\begin{aligned} x_k^* &\approx x_{k-1} + \Delta s_{k-1} \cos(\psi_{k-1}) + \delta x_{k-1} - \Delta s_{k-1} \sin(\psi_{k-1}) \delta \psi_{k-1} \\ &\quad - \cos(\psi_{k-1}) n_{\Delta s_{k-1}} + \sin(\psi_{k-1}) \delta \psi_{k-1} n_{\Delta s_{k-1}} \end{aligned} \quad (4.16a)$$

$$\begin{aligned} y_k^* &\approx y_{k-1} + \Delta s_{k-1} \sin(\psi_{k-1}) + \delta y_{k-1} + \Delta s_{k-1} \cos(\psi_{k-1}) \delta \psi_{k-1} \\ &\quad - \sin(\psi_{k-1}) n_{\Delta s_{k-1}} - \cos(\psi_{k-1}) \delta \psi_{k-1} n_{\Delta s_{k-1}}. \end{aligned} \quad (4.16b)$$

Now (4.14a) and (4.16) can be expressed as two recursive systems containing the DR part and error state part, as proposed in (4.13). The dead reckoning part contains all the terms without error and noise variables:

$$\psi_k = \psi_{k-1} + \Delta \psi_{k-1} \quad (4.17a)$$

$$x_k = x_{k-1} + \Delta s_{k-1} \cos(\psi_{k-1}) \quad (4.17b)$$

$$y_k = y_{k-1} + \Delta s_{k-1} \sin(\psi_{k-1}). \quad (4.17c)$$

The error state equations are formed from the terms the error and noise variables:

$$\delta \psi_k = \delta \psi_{k-1} + n_{\Delta \psi_{k-1}} \quad (4.18a)$$

$$\begin{aligned} \delta x_k &= \delta x_{k-1} - \Delta s_{k-1} \sin(\psi_{k-1}) \delta \psi_{k-1} \\ &\quad - \cos(\psi_{k-1}) n_{\Delta s_{k-1}} + \sin(\psi_{k-1}) \delta \psi_{k-1} n_{\Delta s_{k-1}} \end{aligned} \quad (4.18b)$$

$$\begin{aligned} \delta y_k &= \delta y_{k-1} + \Delta s_{k-1} \cos(\psi_{k-1}) \delta \psi_{k-1} \\ &\quad - \sin(\psi_{k-1}) n_{\Delta s_{k-1}} - \cos(\psi_{k-1}) \delta \psi_{k-1} n_{\Delta s_{k-1}}. \end{aligned} \quad (4.18c)$$

By denoting $\delta \mathbf{x}_k = [\delta \psi_k, \delta x_k, \delta y_k]^T$, (4.18) can be written as

$$\begin{aligned} \delta \mathbf{x}_k &= \begin{bmatrix} 1 & 0 & 0 \\ -\Delta s_{k-1} \sin(\psi_{k-1}) & 1 & 0 \\ \Delta s_{k-1} \cos(\psi_{k-1}) & 0 & 1 \end{bmatrix} \delta \mathbf{x}_{k-1} \\ &+ \begin{bmatrix} 1 & 0 \\ 0 & -\cos(\psi_{k-1}) + \sin(\psi_{k-1}) \delta x_{1,k-1} \\ 0 & -\sin(\psi_{k-1}) - \cos(\psi_{k-1}) \delta x_{1,k-1} \end{bmatrix} \begin{bmatrix} n_{\Delta \psi_{k-1}} \\ n_{\Delta s_{k-1}} \end{bmatrix} \end{aligned} \quad (4.19)$$

In the feedback configuration of the complementary filter for DR system, the error states get non-zero values only after measurement update. After that the error states are used to correct the DR estimate and then immediately set back to zeros, i.e., $\delta \mathbf{x}_{k-1} = 0$. Therefore the last terms in (4.18b) and (4.18c) and in the coefficients of

$n_{\Delta s_{k-1}}$ in (4.19) are zeros and can be omitted. The error state system (4.19) conforms with the KF state propagation model (2.18) with adaptive, time variant system matrix and noise:

$$\delta \mathbf{x}_k = F_{k-1} \delta \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (4.20)$$

where

$$F_{k-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\Delta s_{k-1} \sin(\psi_{k-1}) & 1 & 0 \\ \Delta s_{k-1} \cos(\psi_{k-1}) & 0 & 1 \end{bmatrix} \quad (4.21)$$

and

$$\mathbf{w}_{k-1} = \begin{bmatrix} 1 & 0 \\ 0 & -\cos(\psi_{k-1}) \\ 0 & -\sin(\psi_{k-1}) \end{bmatrix} \begin{bmatrix} n_{\Delta \psi_{k-1}} \\ n_{\Delta s_{k-1}} \end{bmatrix}. \quad (4.22)$$

In the feedback configuration of the CEKF, the prediction part (4.20) of the EKF is applied only to the state covariance. The covariance is propagated using (2.24). The covariance Q_{k-1} of the state noise is computed using

$$\begin{aligned} Q_{k-1} &= E [\mathbf{w}_{k-1} \mathbf{w}_{k-1}^T] \\ &= \begin{bmatrix} \sigma_{\Delta \psi}^2 & 0 & 0 \\ 0 & \cos^2(\psi_{k-1}) \sigma_{\Delta s}^2 & \sin(\psi_{k-1}) \cos(\psi_{k-1}) \sigma_{\Delta s}^2 \\ 0 & \sin(\psi_{k-1}) \cos(\psi_{k-1}) \sigma_{\Delta s}^2 & \sin^2(\psi_{k-1}) \sigma_{\Delta s}^2 \end{bmatrix}. \end{aligned} \quad (4.23)$$

The measurement input of the filter is the difference between the WLAN based position estimate and the DR estimate obtained using (4.17). Taking into account (4.13), the difference can be written as

$$\begin{aligned} \mathbf{z}_k &= \begin{bmatrix} x_{w_k} - x_k \\ y_{w_k} - y_k \end{bmatrix} = \begin{bmatrix} x_k^* + n_{x_{w_k}} - (x_k^* - \delta x_k) \\ y_k^* + n_{y_{w_k}} - (y_k^* - \delta y_k) \end{bmatrix} = \begin{bmatrix} \delta x_k + n_{x_{w_k}} \\ \delta y_k + n_{y_{w_k}} \end{bmatrix} \\ &= H \begin{bmatrix} \delta \psi_k \\ \delta x_k \\ \delta y_k \end{bmatrix} + \mathbf{v}_k \end{aligned} \quad (4.24)$$

where H and \mathbf{v}_k are the same as in (4.12). The derived algorithm for CEKF in feedback configuration is summarized in Algorithm 4.2.

Algorithm 4.2 CEKF in feedback configuration

-
- 1) Initialize PDR estimates Ψ_0 , x_0 , y_0 , state variance P_0 , step counter $k = 0$
 - 2) Wait until new estimate from PDR or WLAN positioning available
 - 3) **if** New PDR estimates $\Delta\Psi_{k-1}$ and Δs_{k-1} available (step detected) **then**
 - 4) $k \leftarrow k + 1$ /* step counter */
 - 5) PDR propagation
 - Propagate estimates using (4.17)
 - Linearize state equation using (4.21)
 - Compute state noise covariance using (4.23)
 - Propagate state covariance: $P_k^- = F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1}$
 - 6) **else if** New WLAN positioning estimate available **then**
 - 7) Measurement update
 - Compute \mathbf{z}_k using (4.24)
 - Compute the gain $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$
 - Update the state (correction to PRD) $\delta \hat{\mathbf{x}}_k = K_k \mathbf{z}_k$
 - Update the state covariance $P_k = (I - K_k H_k) P_k^-$
 - 8) Apply estimated correction to PDR estimate

$$\begin{bmatrix} \Psi_k \\ x_k \\ y_k \end{bmatrix} \leftarrow \begin{bmatrix} \Psi_k \\ x_k \\ y_k \end{bmatrix} + \delta \hat{\mathbf{x}}_k$$
 - 9) $\delta \hat{\mathbf{x}}_k^- \leftarrow \mathbf{0}$ /* Reset error states */
 - 10) **end if**
 - 11) Go to line 2
-

4.2.3 Particle Filter for Map Aided Navigation

For indoor map aiding of DR, the general EKF is not applicable. The EKF approach assumes measurements which can take several values on a certain range, such that the filter can compare the actual measurement to its predicted value and use the difference to correct the estimate. With indoor maps the available information is of different

type: for a given MU transition, we can conclude from the wall information whether a certain transition is possible or not, i.e., we have an on/off type of measurement. From this, it may be possible to conclude whether the estimate needs to be corrected, but it is very hard to say how much and to which direction. Now there is also a discontinuity at the point of linearization, and the partial derivative needed for the linearization of the measurement equation (2.35) does not exist. Therefore the covariance cannot be predicted (Julier and Uhlmann, 1996).

A solution is to use particle filters: instead of one filter, several distinct particles can simultaneously try to estimate the same quantities. The states of the particles are driven by the same process model, but their states differ due to their simulated noise inputs. Now each of the particles can probe the walls individually and get the 'transition impossible' or 'transition possible' result depending on whether the transition crosses the wall or not. The particles with impossible transition will not survive into the next iteration cycle; the algorithm will replace them by picking replacements from the group of particles with feasible transitions. The positions of the particles in this group provide an approximate distribution of the position of the MU, and statistical functions such as mean, median, or mode can be used to obtain the position estimate of the MU. In this thesis, a bootstrap particle filter (Candy, 2009) is used for map aided navigation. The block diagram of the particle filter algorithm is shown Fig. 4.3.

Propagation of Particles

The particles are propagated using an equation similar to (4.5) in CEKF, except that now the noise components of the PDR outputs are simulated using a random number generator and then added to the particle states. The proposal distribution (2.38) is obtained using the simulated noise sequences $n_{\Delta\psi}$ and $n_{\Delta s}$ together with the PDR outputs, step length Δs_{k-1} and the heading change $\Delta\psi_{k-1}$, estimated using (4.1) and (4.2), respectively. The samples from the proposal distribution are obtained by

$$\mathbf{x}_k^{(j)} = \mathbf{x}_{k-1}^{(j)} + \begin{bmatrix} \Delta\psi_{k-1} + n_{\Delta\psi}^{(j)} \\ \left(\Delta s_{k-1} + n_{\Delta s}^{(j)} \right) \cos x_{1,k-1}^{(j)} \\ \left(\Delta s_{k-1} + n_{\Delta s}^{(j)} \right) \sin x_{1,k-1}^{(j)} \end{bmatrix} \quad (4.25)$$

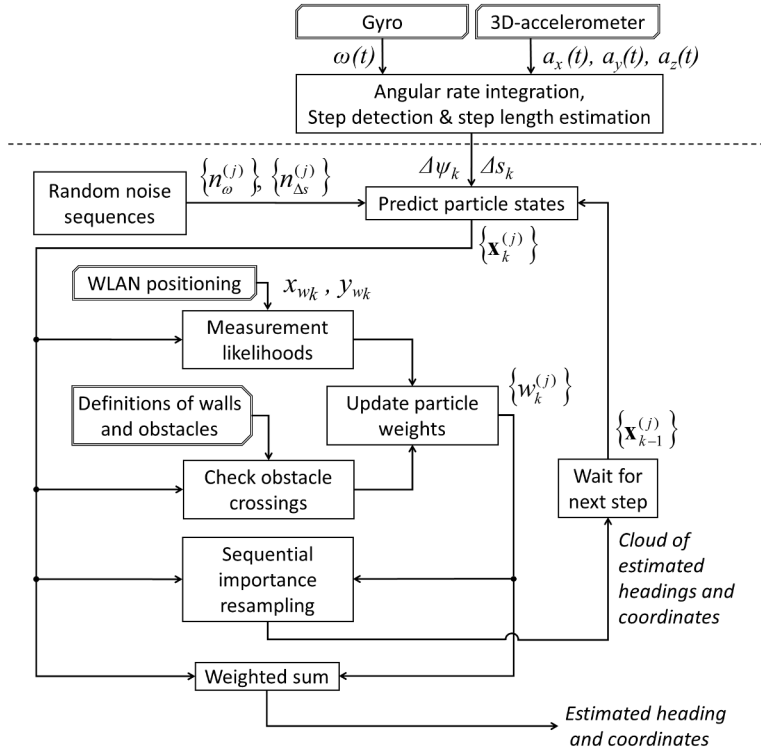


Fig. 4.3. Block diagram of the signal processing system including the PDR preprocessing of the sensor data and particle filter to perform the PDR state propagation and measurement update using WLAN positioning and indoor map matching.

where the elements of the state vectors $\mathbf{x}_k^{(j)}$ are heading, x-coordinate, and y-coordinate. If the structured map model is used, the room identifier of particle location is included into the particle state vector as the fourth element. The superscript j is the index of the particle; the subscript k is the index of the detected step and grows with time. The state variable $x_{1,k}^{(j)}$ is the first element of the state of particle (j), i.e., its heading estimate. The variances of the noise sequences introduced through $n_{\Delta\psi}$ and $n_{\Delta s}$ are adjusted to match with the observed uncertainty in $\Delta\psi$ and Δs .

Measurement Updates and Resampling

The measurement update is performed in two different cases: when new WLAN position estimates become available or when new PDR estimates are ready. In WLAN based measurement update the coordinates of the WLAN based position estimates are

assumed to be independent and identically Gaussian distributed with variance σ_W^2 . In bootstrap filter the resampling is performed at every time step and the weights $w_k^{(j)}$ are proportional to the likelihoods $p(\mathbf{z}_k | \mathbf{x}_k^{(j)})$:

$$w_k^{(j)} = \exp \left\{ -\frac{\left(x_{2,k}^{(j)} - x_{W_k}\right)^2 + \left(x_{3,k}^{(j)} - y_{W_k}\right)^2}{2\sigma_W^2} \right\} \quad (4.26)$$

where $x_{2,k}^{(j)}$ and $x_{3,k}^{(j)}$ are the x and y coordinates of the particles, x_{W_k} and y_{W_k} are the WLAN based coordinate estimates, and σ_W^2 is the uncertainty of x_{W_k} and y_{W_k} . Before the next particle state propagation, importance resampling is performed. For the resampling, the weights are normalized:

$$w_k^{(j)} \leftarrow \frac{w_k^{(j)}}{\sum_{j=1}^N w_k^{(j)}} \quad (4.27)$$

where N is the number of particles. In the resampling after WLAN updates, N particles are sampled with replacement from the population of $\{\mathbf{x}_k^{(j)}\}$. The probability of the j^{th} particle to be selected and continue to the next propagation step is $w_k^{(j)}$.

After new PDR estimates, the particle likelihoods are based on the map based obstacle checks. This weight update is performed every time the algorithm has propagated particles using (4.25) and the PDR information. The obstacle information from the map fits easily into particle filtering. As it is impossible to walk through obstacles, the probability of a particle transition that intersects the line segment defined by obstacle line coordinates is very small. A measurement update with binary likelihood is used:

$$w_k^{(j)} = p(\mathbf{z}_k | \mathbf{x}_k^{(j)}) = \begin{cases} 0, & \text{if the particle crosses an obstacle} \\ 1 & \text{otherwise.} \end{cases} \quad (4.28)$$

If a particle goes through an obstacle, its weight goes to zero and it dies; only the particles with feasible transition history survive. After the obstacle based weight update, the remaining particles are resampled so that the number of particles is returned to the original. This choice to let the particles that survived the obstacle checks to continue directly to the next propagation step without the risk of not getting selected in sampling with replacement is adopted from the field of genetic algorithms. In genetic algorithms, a principle of elitism may be used, when a certain number of the best

members of the population are always kept to the next generation without selection, crossover, and mutations among them (Jang et al., 1997). As the obstacle checking is computationally the heaviest part of the filter, the elitism is applied to all the survived particles to ensure that no diversity among this group will be lost. If WLAN based estimates are not available at all, then (4.26) and (4.27) are omitted and only the map based measurement update (4.28) is performed.

Initialization of Particles

In the initialization of the particle filter, the states of the particles ($\mathbf{x}_0^{(j)}$) are given values that are generated from the distribution with the statistical properties (mean and covariance) best reflecting the available knowledge about the initial position and heading. If the initial information is inaccurate, the initial coordinates and heading are drawn from distributions with large variances, and if the knowledge is fairly accurate, the variances are small. For example, if we only know that the user is inside a building, the initial coordinates can be spread all over the building floor plan and the initial headings are spread to cover the whole circle (360°). On the other hand, if the user has just entered the building, and the GPS coordinates of the user device are known few seconds earlier, we may conclude that the user coordinates indoors are near the entrance closest to the last GPS coordinates and the standard deviations of the initial coordinate states can be set to few meters. If the device also has tracked the heading of the user all the way from outdoors to indoors, we can also have quite good estimate of the initial heading indoors, and the standard deviation of the initial heading can be set to some tens of degrees. In the tests performed in this thesis, this outdoor/indoor transition scenario in the initialization is assumed. The particle initialization for navigation system is explained in more detail in (Kirkko-Jaakkola et al., 2013).

For large buildings, a modest number of particles (e.g. some hundreds) is not large enough to cover the whole area of the building and all possible directions adequately. One solution would be adding more particles to the initial cloud and then gradually decrease the number to its normal value, but that could induce significant problems with the computational load, especially in real time applications, causing the wall check processing of the first steps to take much longer time than it takes for the user to walk the next steps. Therefore, methods for decreasing the initial uncertainty are

needed. These may include a positioning method that is capable of giving coordinate estimates indoors with reasonable accuracy, i.e., with accuracy level of a room and its neighbors (Kirkko-Jaakkola et al., 2013), or the utilization of radio navigation aids prior to indoor entrance. A reliable means for detecting the environmental context of the user, especially the separation between indoor and outdoor environments, is essential for the utilization of the prior outdoor location in particle initialization when entering indoors; methods for indoor/outdoor detection have been described, e.g., in (Eronen et al., 2013) and (Zhou et al., 2012).

Sometimes a re-initialization of the particles is needed during a particle filter run. Due to the unmodeled effects, such as small errors in the map data or a possible small sliding of the sensor unit with respect to the user's body, it occurs that due to the error all the particles get trapped into a dead end. In order to allow the particles to escape the dead end, in these situations the propagation step (4.25) is re-iterated from $\mathbf{x}_{k-1}^{(j)}$ (when the particles were not yet trapped) with noise variances significantly larger than $n_{\Delta\psi}$ and $n_{\Delta\delta}$.

4.2.4 Map Information

In a particle filter using an obstacle map, the obstacle check is based on the computation of the intersection of the line segment that define a wall or an obstacle and the line segment that define the transition of a particle. In the crossing checks the x and y coordinates of the particles are used, defined by the 2nd and the 3rd particle states, i.e., $x_{2,k}^{(j)}$ and $x_{3,k}^{(j)}$, respectively. With the structured obstacle map the 4th state $x_{4,k}^{(j)}$, defining the room identifier of the particle location is also needed. The room identifier $x_{4,k}^{(j)}$ is not changed in the prediction equation (4.25). Its value is updated only in the obstacle check performed using sequential re-selection described in Algorithm 4.3. The particle transition is defined by the line segment that connects the previous and current particle positions, i.e., $\mathbf{x}_{2:3,k-1}^{(j)}$ and $\mathbf{x}_{2:3,k}^{(j)}$, respectively. The symbol $\mathbf{x}_{2:3,k}^{(j)}$ denotes the vector containing only the 2nd and the 3rd elements of $\mathbf{x}_k^{(j)}$. If the intersection exists, the algorithm interprets it as an obstacle crossing.

The obstacle crossing checks are the most computationally intensive part of the positioning algorithm, as the total number of possible obstacle crossings is $N \times N_o$, where N is the number of particles and N_o is the number of obstacle lines. For good performance, we would like to have as many particles in the filter as possible, but

increasing N also increases the computational load and risks the real-time operation, as the time required for particle computations must not exceed the duration of one step. Fortunately, not all of these N_o lines need to be checked in order to detect all the actual obstacle crossings; to keep the number of obstacle checks as small as possible and still have enough particles for robustness of the filter, the set of the lines to be checked can be reduced using an obstacle line selection logic. The selection logic depends on the data structure where the map data is stored.

When the map information is presented with an unstructured obstacle model, the data structure is a simple array consisting of the coordinates of the start and end nodes of each obstacle line, without any defined dependencies between the lines. In the structured obstacle map used in this thesis, the data structure for a room included the room identifier, obstacle lines of the room, and passage lines of the room, where each passage line is associated with the identifier of the room where the passage leads to. In this thesis, two approaches were tested to reduce the number of walls to be checked when using unstructured obstacle model. With structured obstacle one model was tested.

Line Grouping with Unstructured Obstacle Model

In this approach, the obstacle lines are divided into groups, each group including obstacle lines close to each other. For the grouping of the obstacle lines, square-shaped cells with defined cell widths were used. Instead of checking crossings with all the possible line segments, the algorithm first searches for each of the particles the line groups that cover the area where the particle transition happens, and checks the line crossings only with the line segments of these groups (Leppäkoski et al., 2013).

Line Re-Selection with Unstructured Obstacle Model

In this approach, the set of the lines to be checked is re-selected after each IMU step update (Perttula et al., 2014), without line grouping in advance. With the line grouping the lines to be checked were selected separately for each particle. The computational load can be further decreased by making a common selection of lines for the whole particle cloud. In this method, the set of the walls to be checked is re-selected for each PDR step update. Only the obstacle lines that are inside or cross

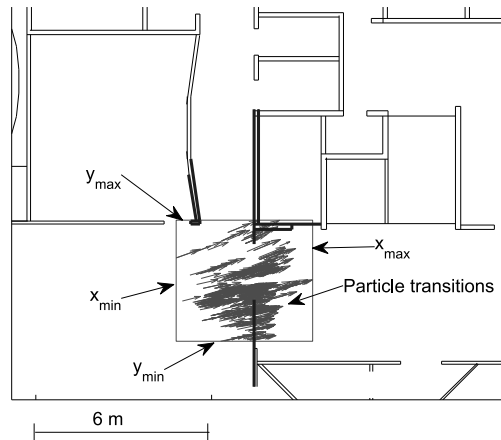


Fig. 4.4. Choice of the walls to be included in the crossing check: the walls to be checked are indicated by thick lines, other walls shown with thin lines.

the box limited by x_{min} , x_{max} , y_{min} , and y_{max} are selected for checking. The limits are the minimums and maximums of the x and y coordinates of all the current and previous particle positions (see Fig. 4.4).

Sequential Line Re-Selection with Structured Obstacle Model

With structured obstacle model, the obstacle lines are already grouped based on the rooms they border. Using this type of map, the obstacle checking can be restricted to the rooms where the particle transition happens. In small office rooms this works well. However, for example with long corridors visible in Fig. 4.17, the polygon including all the surrounding walls and doors may consist of many lines that are not located near the current particle transitions. Using re-selection to exclude these distant lines from the checks can reduce the computational load significantly.

The simple line selection logic sufficient for unstructured obstacle model requires modifications when a structured obstacle model is used. It is difficult to choose the lines to be checked at one time, as before the line checking only the rooms where the particle transitions start are known, but the other rooms visited by the particles during their transitions are not known yet. Therefore, similarly as the obstacle checking in general, this model follows the sequence of rooms along the particle transition. At the same time when checking the obstacle lines in a room based sequence, the lines

Algorithm 4.3 Sequential line re-selection with structured obstacle model

Require: Particle state predictions $\left\{ \left(\mathbf{x}_k^{(j)}, w_{k-1}^{(j)} \right), j = 1, \dots, N \right\}$
after k^{th} step done

Require: Limits $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$ defining the box B computed

$n_R \leftarrow 0$ /* Initialize the number of elements in structure arrays
 R (room ids), W (obstacle lines), and D (passage lines) */

for $j=1$ **to** N **do** /* Particles loop */

$dx \leftarrow$ Line segment defining transition from $\mathbf{x}_{2:3,k-1}^{(j)}$ to $\mathbf{x}_{2:3,k}^{(j)}$

$m \leftarrow 1$ /* First room */

$lastRoomAlongTransitionChecked \leftarrow \mathbf{false}$ /* Initialize status */

repeat /* 'Rooms along transition' loop */

if $(n_R = 0)$ **or** $\left(x_{4,k}^{(j)} \text{ not in } R(1:n_R) \right)$ **then** /* Add current
 room to temporary structure */

$n_R \leftarrow n_R + 1$

$R(n_R) \leftarrow x_{4,k}^{(j)}$ /* Room id */

$W(n_R) \leftarrow$ Obstacle lines of $R(n_R)$ inside or crossing B

$D(n_R) \leftarrow$ Passage lines of $R(n_R)$ inside or crossing B

end if

if dx crosses a passage line included in $D(1:n_R)$ **then**

$x_{4,k}^{(j)} \leftarrow$ The id of the room connected to the crossed passage
 line

$m \leftarrow m + 1$ /* Next room along the particle transition */

else if dx crosses an obstacle line in $W(1:n_R)$ **then**

$w_k^{(j)} \leftarrow 0$ /* Update particle weight */

$lastRoomAlongTransitionChecked \leftarrow \mathbf{true}$

else

$w_k^{(j)} \leftarrow w_{k-1}^{(j)}$ /* Update particle weight */

$lastRoomAlongTransitionChecked \leftarrow \mathbf{true}$

end if

until $lastRoomAlongTransitionChecked$

end for

to be checked are collected into a data structure. The process including the sequential line re-selection and line checks is described in Algorithm 4.3; this algorithm uses

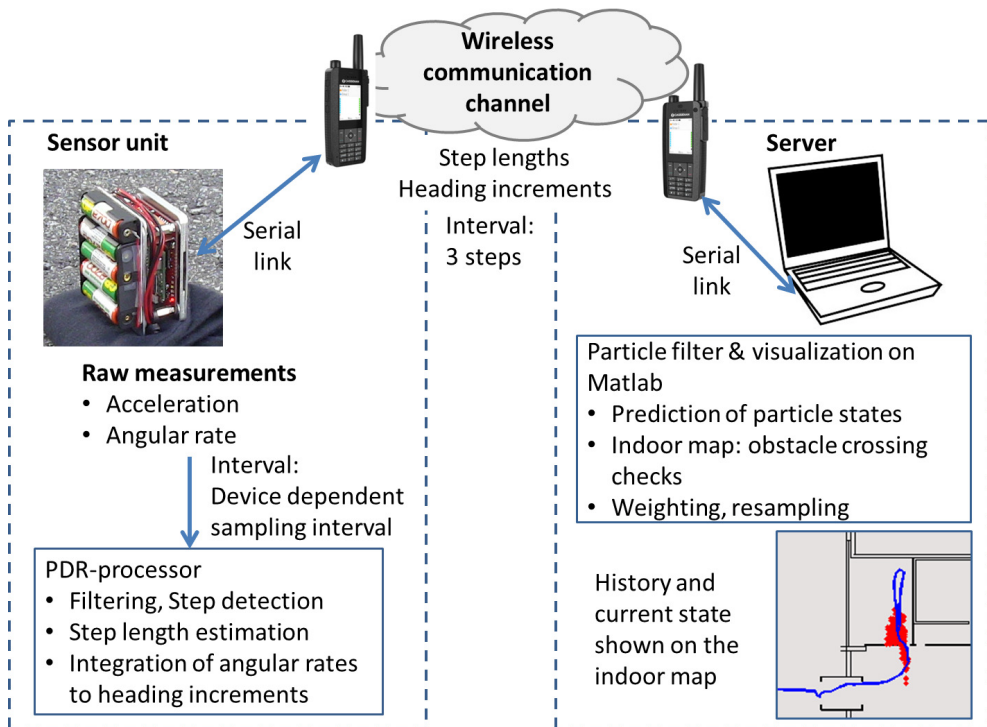


Fig. 4.5. System architecture of the distributed positioning system.

and updates also $x_{4,k}^{(j)}$, the room identifier of the j^{th} particle.

4.2.5 Distributed Indoor Positioning System

The signal processing system described in Fig. 4.3 is well-suited for a distributed positioning system. The estimation of the PDR outputs is computationally relatively light, and significantly less data is transferred from PDR to particle filter than from the sensors to the PDR estimation.

A microcontroller based system was used in real time tests to perform the PDR pre-processing, while the particle filter processing and location display were performed on a server. The PDR outputs were sent over a wireless link to the server. The organization of the different functions are shown in Fig. 4.5. The goal of the system architecture is to process the sensor data efficiently on the sensor unit while keeping the amount of data transmissions as low as possible.

For data transmission, the estimation outputs of three steps, i.e., three step lengths

Table 4.2. *Data sets for performance evaluation of map aided indoor positioning*

Data set	WLAN positioning	Map type	Test duration	Track length	Distributed system
#3	Yes	Unstructured	8.8 min	480 m	No
#4	No	Structured	7.5 min	480 m	No
			6.7 min	380 m	
#5	No	Structured	8.5 min	400 m	Yes
			9.0 min	420 m	
#6	No	Unstructured	6.0 min	300 m	Yes

and the corresponding heading changes, were packed into a single message. Because each data transmission requires architecture specific metadata, e.g., packet header, and handshaking procedures, this arrangement allowed the reduction of the overhead of the metadata transmission and handshaking and still kept the delay in the position display within acceptable limits. In regular walking, this produces a message once in 1.8 s on average, with standard deviation of 0.3 s and minimum 0.6 s in the message intervals.

On the server, the packet of the estimates from three steps is given as an input to the particle filter. The filter processes the data step by step using (4.25) and (4.28) to allow accurate position estimation also when the user is skirting a corner. As the sensor unit transmits the step data only after every three steps, in the worst case the estimate shown on the server may be three steps delayed from the actual user position.

4.3 Performance Analysis

The performance of the proposed indoor navigation methods were evaluated using data from test walks. In this Section the tests and their results are described.

4.3.1 Data

Four data sets were used for performance evaluations of the proposed algorithms. The data was collected from six test walks; the main characteristics of the test data and the walks are listed in Table 4.2. In all of the tests, the accelerations and the

angular rate were measured using the MEMS based IMU described in (Murata Electronics). It includes Murata SCC1300-D04 chip with 1-axis gyroscope and 3-axis accelerometer. The measurement ranges of the sensors are $\pm 300^\circ/s$ and $\pm 6g$, and their sensitivities are $\frac{1}{18}^\circ/s$ and $\frac{1}{650}g$. This sensor was selected as it had the lowest bias instability among the available MEMS gyroscopes. The sensors were read and the PDR preprocessing computations were performed at 200Hz sampling frequency.

In addition to the sensors, the IMU contains also 32MHz 8-bit microcontroller (Atmel ATxmega128A1) for data processing. In off-line tests, the microcontroller sampled the sensor data and stored it to the memory card, from which the data was read to a PC and processed there with MATLAB running the PDR preprocessing and data fusion functions, i.e., the CEKF or the particle filters. In real time tests with the distributed navigation system it took care of the measurement sampling, PRD pre-processing, creating data package after every third detected step, and sending the package to the server that uses MATLAB to run the particle filter. During the test walks, the sensor unit was attached to the back of the test walker and aligned so that the sensitive axis of the gyro was vertical, i.e., it was able to measure heading changes.

The test walk of data set #3 was conducted in the library of the Tampere University of Technology, where the test route consisted of four loops in the library. This is the only data set that includes also WLAN positioning data. In the library, only the outer walls of the building totally block radio signals, but the inner walls that border the library hall and the book shelves in there cause either strong attenuations or non-line-of-sight conditions for the radio signal propagation. On the other hand, as the book shelves are obstacles that the pedestrian cannot walk through, they provide useful map information for the particle filter. The obstacle model with this data was unstructured, and the PDR sensor data was collected using the microcontroller of the IMU and the PDR processing was performed off-line on PC.

The test walks of data sets #4 and #5 were conducted in the Tietotalo building of Tampere University of Technology. For this building, a structured obstacle map was available. The IMU measurements of the data set #4 were stored on the memory card of the IMU and post-processed on a PC, while in the data set #5 the PDR processing was carried out in the IMU and only the estimated step lengths and heading changes were transferred to the PC. However, in these tests the PRD estimates were transferred through a wired link. The data set #6 was a real time demonstration, where the PDR estimates were transferred to the server PC over wireless channel provided

by Terrestrial Truncated Radio (TETRA) network. For the wireless communication, both the IMU and the server were each connected to a handset equipped with a TETRA radio. The data from this test includes screen shots of the server display showing the user track and a video (TUT insnavgroup, 2013); numerical data was not recorded.

4.3.2 Comparison of Different Combinations of Data Sources

In this Section, the navigation performance using different combinations of sensor information is compared: the estimates by WLAN positioning only, PDR only, PDR and WLAN positioning fused together with CEKF, PDR and map information fused with particle filter, and finally PDR, WLAN positioning, and map information all fused with particle filter. The comparisons are performed using the test data set #3.

WLAN Positioning

The WLAN signal strengths were collected using a mobile handset, which produces WLAN scan results at 2.3 s intervals. The position estimates were computed using the MEE algorithm described in Section 2.1.3 with radio map including seven evenly distributed histogram bins for RSS PDF approximation (see Section 3.2.3). The position estimates computed from WLAN signal strengths that were collected along the test walk are shown in Fig. 4.6. From the figure we can see that the estimates are not evenly spread along the route, but rather concentrated in the center of the library area. The RMSE position error is about 10 m.

PDR Positioning

The result of the unaided PDR estimate is shown in Fig. 4.7. It can be seen that during the first loop the traveled distance gets longer and the heading starts to get distorted. After the first loop, the three following loops seem to be quite similar in size and orientation. The maximum distance error is 5 m and the absolute heading error is mostly below 23° . The mean heading error is -9° ; this counterclockwise bias is clearly visible in the estimated track.

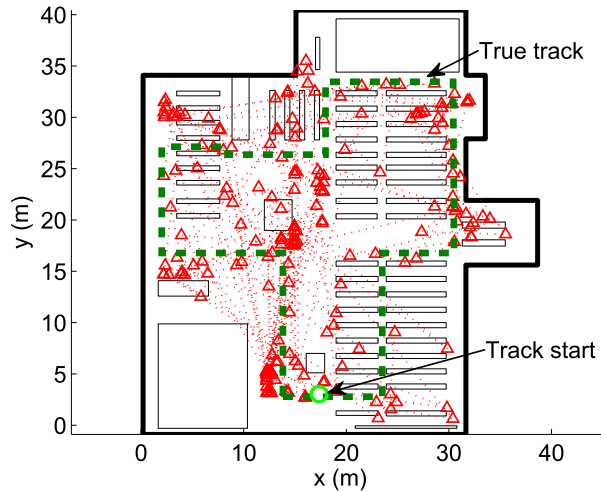


Fig. 4.6. WLAN based position estimates shown with triangles. Transitions between consecutive estimates shown with narrow dotted lines, the true track with thick dotted line, and obstacles (rectangles) with light solid lines.

WLAN and PDR Positioning with CEKF

The result of CEKF processing of the PDR and WLAN based position estimates is shown in Fig. 4.8. The CEKF was initialized with the same initial heading and coordinates as the unaided PDR estimate. It can be seen that CEKF can correct some of

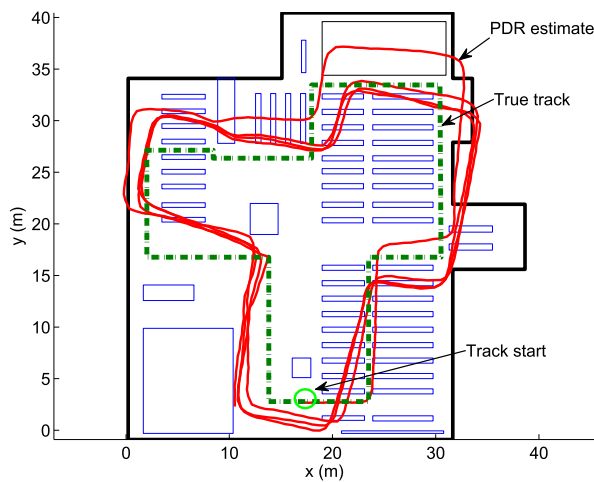


Fig. 4.7. Unaided pedestrian dead reckoning.

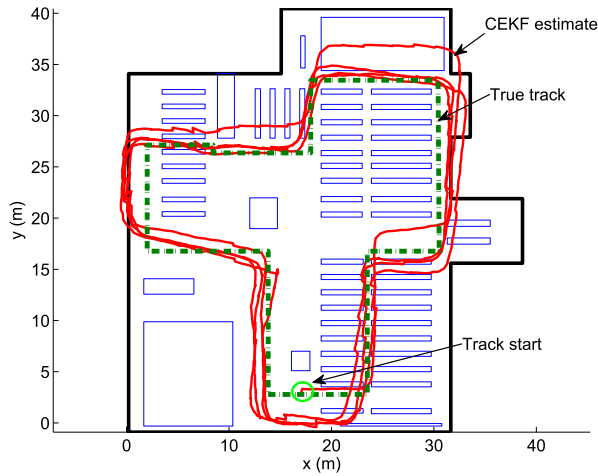


Fig. 4.8. Complementary EKF using PDR and WLAN data.

the skewness in PDR loops. The maximum distance error is 4 m; the average heading error is -1.1° , which is significantly lower than with PDR only. The CEKF estimates were computed using both the ERT and the feedback configurations, described in Section 4.2.2. The order of magnitude of the differences between the estimates by these configurations was 10^{-12} m, which can be considered as the effect of numerical inaccuracy of the computations. It can be concluded that for this nonlinear navigation model, the ERT and feedback configurations are practically the same.

Map Aided PDR through Particle Filtering

In the first particle filter test, the filter was used to fuse PDR and map information, while in the second test, it was used to fuse also WLAN estimates with PDR and map information. The number of particles used in the tests was 500. The particle states were initialized with the same initial values as the CEKF. The results of the tests are shown in Figures 4.9 and 4.10. The plotted particle filter track is the Minimum Mean Square Estimate (MMSE) computed using (2.40) from particle positions at each sampling instance. The maximum distance errors of both particle filters are less than 4 m. The average distance errors and RMSE are smaller than with CEKF, and with the second particle filter (PF2, shown in Fig. 4.10) these errors are slightly smaller than with the first particle filter (PF1, shown in Fig. 4.9). The mean heading error with PF1 is -1.9° while with PF2 it is -0.6° . The estimated track of PF2

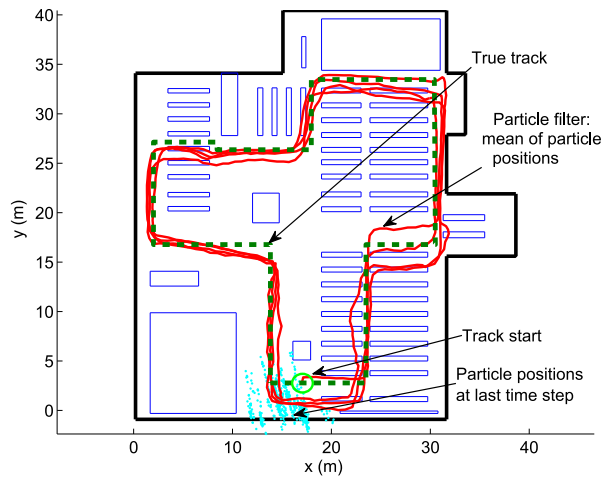


Fig. 4.9. Particle filter using PDR and map information.

seems to follow the true track better than the track of PF1. Therefore, the WLAN estimates are able to improve the particle filtering result. However, the improvement in accuracy is not significant, especially given the effort required to build the radio maps. On the other hand, the WLAN positioning could improve the robustness of the algorithm by serving as a fall-back system, which is valuable if the particle filter needs to be re-initialized during an indoor navigation mission.

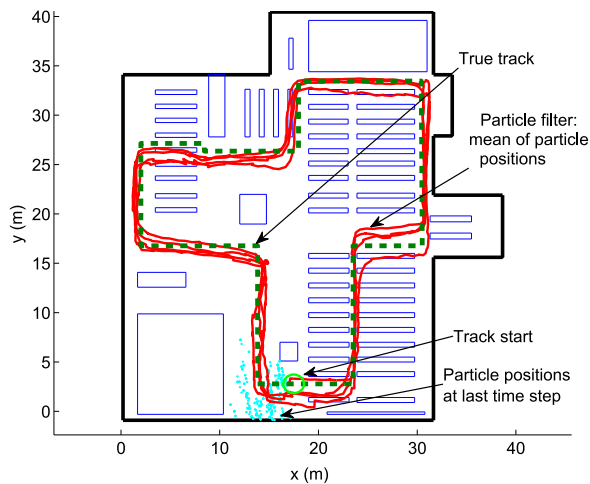


Fig. 4.10. Particle filter using PDR, WLAN positioning, and map information.

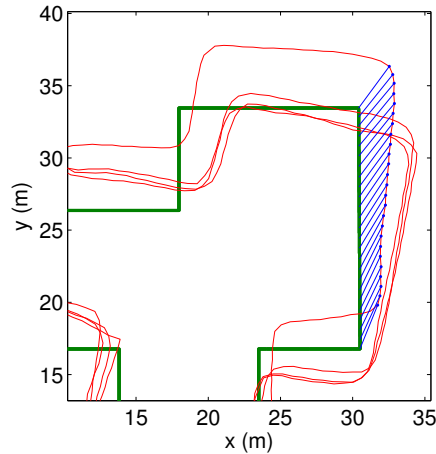


Fig. 4.11. Mapping from the estimated track (red line) to the true track (green thick line) shown by the blue line segments connecting the tracks.

Approximated Error Statistics

Because the true track of the test presented in this section (Figures 4.6-4.10) is not time tagged, the mapping of the points on the estimated track to the true track cannot be performed in a straight forward way. However, the turns of the track are clearly detectable in the estimated tracks of PDR, CEKF, and particle filters (Figures 4.7-4.10). This enables an approximate, shape matching based mapping between the tracks. For the points between two consecutive turns on an estimated track, the mapping to the true track is interpolated between the corresponding edges of the true track as a function of the estimated distance traveled between the turns. An example of the mapping is shown in Fig. 4.11. This method is not directly applicable to the track based on WLAN positioning, because its high noise content makes it impossible to reliably detect the turns. However, as both the WLAN and the PDR track are time tagged and synchronized, mapping from the WLAN track to the true track can be obtained by interpolating the PDR-to-true-track mapping with the time tags. The error statistics obtained like this can bring forth the cumulative effects of the errors in the estimated step length and the heading which transform length or direction of the track segments. However, it cannot detect the errors in the parameters of individual steps.

The approximated statistics of the distance errors are shown in Table 4.3. The values in the table conform to the visual observations from Figures 4.6-4.10. Clearly the

Table 4.3. *Approximated distance errors (m) using different algorithms*

	WLAN	PDR	CEKF	PF1	PF2
EMEAN	7.1	3.1	1.9	1.4	1.3
RMSE	9.8	3.3	2.1	1.6	1.5
E95	21.6	5.0	3.4	2.5	2.5
EMAX	32.8	6.6	3.8	2.9	3.9

errors of the WLAN based track are larger than with the other methods. From the PDR column to the right, i.e., adding more information sources to the estimator, the errors decrease. Although the average error and the RMSE of the particle filter with WLAN positioning (PF2) are smaller than these errors with the particle filter without WLAN (PF1), the maximum error with PF2 is still larger than with PF1.

The approximated statistics of the heading errors are shown in Table 4.4. The angles are computed from the estimated tracks. As with the error distances, also with the heading errors the absolute mean values decrease as more information sources are taken into account by the algorithm. With all the algorithms the mean error is negative, biasing the estimated tracks counterclockwise. This is clearly visible in the PDR track (Fig. 4.7). As the other algorithms are also using the PDR information, they inherit the problem, although they are able to mitigate it significantly. The WLAN updates introduce fast instantaneous heading changes to the estimated track, which can be observed from the larger standard deviations with CEKF and PF2 when compared with PDR and PF1. This is also visible in the RMSE errors. With PDR, 90 percent of the heading errors are within 30° range, with PF1 the 90 percent range is almost 40° , and with CEKF and PF2 it is almost 60° . The maximum absolute errors

Table 4.4. *Approximated heading errors (degrees) using different algorithms*

	PDR	CEKF	PF1	PF2
EMIN	-43.2	-85.5	-35.4	-156.3
E05	-23.0	-19.4	-19.4	-23.9
EMEAN	-9.0	-1.1	-1.9	-0.6
E95	7.1	37.7	16.7	35.9
EMAX	37.4	175.6	59.1	90.0
RMSE	13.2	21.3	11.8	19.3
STD	9.7	21.3	11.6	19.3

are relatively large, especially with CEKF and PF2. However, these are occasional situations that do not persist over many steps. This can be concluded from the Figures 4.7-4.10, where the estimated tracks do not propagate long distances to totally wrong directions.

4.3.3 Distributed Indoor Positioning System

To analyze the accuracy of the indoor navigation system, two tests were carried out to obtain the data set #5, where the routes included control points with known coordinates. In test #5.1 there were 33 control point stops while in test #5.2 the number was 35. The sensor unit sent the step lengths and heading changes in packages of three steps to the PC, which was carried by the user. The particle filter processing was carried out in the PC and the estimation results were saved to a file; from the particle filter results, the average coordinates of the particles were recorded together with the variances of the particle coordinates. The unaided PDR estimate was also computed from the step lengths and heading changes. The tracks of the tests are shown in Figs. 4.12 and 4.13. The total length of the tracks was about 400 m; the first track took about 8.5 min and the second about 9 min to walk.

A field showing the number of the current step was added to the particle filter display. During the test walks, the user stopped at the control points, and the step number shown in the screen was manually recorded. The recorded step numbers made it possible to afterwards identify the saved estimates of the particle filter and unaided PDR at the control points. The positioning errors were obtained by computing the error distances

$$d_{pf} = \sqrt{(x_{pf} - x_{cp})^2 + (y_{pf} - y_{cp})^2}$$

$$d_{pdr} = \sqrt{(x_{pdr} - x_{cp})^2 + (y_{pdr} - y_{cp})^2}$$

where d_{pf} and d_{pdr} are the positioning errors of particle filter and unaided PDR, respectively, (x_{pf}, y_{pf}) and (x_{pdr}, y_{pdr}) the coordinate estimates by the both methods, and (x_{cp}, y_{cp}) the true coordinates of the control points. The errors are shown in Fig. 4.14. The estimation errors can be also compared with the positioning uncertainty 2σ assessed by the particle filter itself. It is computed as $2\sigma = 2\sqrt{\sigma_x^2 + \sigma_y^2}$, where σ_x^2 and σ_y^2 are the coordinate variances of the particles recorded at the control points. The error statistics are shown in Table 4.5.

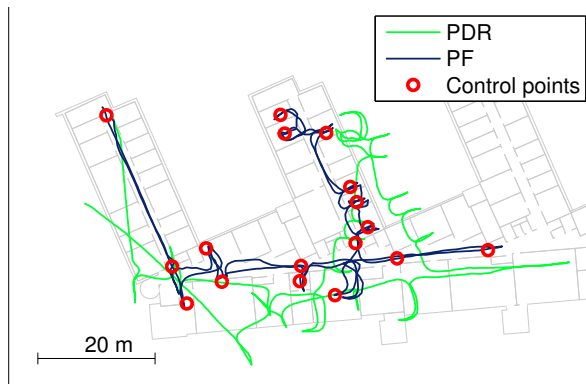


Fig. 4.12. Track of field test #5.1 with control points. PDR: unaided PDR estimate, PF: particle filter, circles: control points.

From the Figs. 4.12–4.14 it is clear that the error of the unaided PRD estimate grows fast when compared to the error of the particle filter; the map constraints realized through the particle filter significantly reduce the positioning error. The average distance errors of the PDR estimates are 9.9 m and 10.3 m in tests #5.1 and #5.2, respectively, while with the particle filter the average distance errors are 1.4 m and 1.7 m. When comparing the position error of the particle filter to the uncertainty indicator 2σ , it can be observed that the uncertainty assessment of the filter increases when the actual error increases.

Comparing the unaided PDR tracks in Figs. 4.7, 4.12, and 4.13, it can be observed

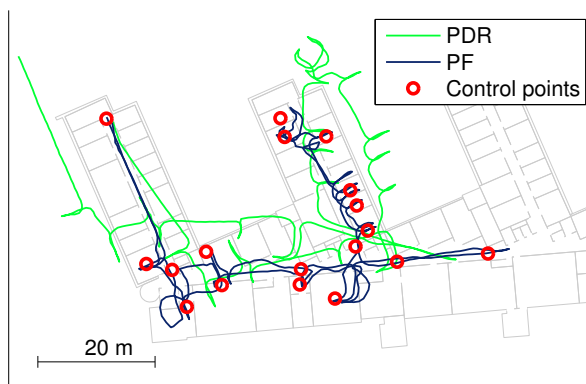


Fig. 4.13. Track of field test #5.2 with control points. PDR: unaided PDR estimate, PF: particle filter, circles: control points.

Table 4.5. Distance errors (m) at control points

	Test# 5.1		Test# 5.2	
	PDR	PF	PDR	PF
EMEAN	9.9	1.4	10.3	1.7
RMSE	10.1	1.6	10.9	1.9
E95	14.8	3.2	16.2	3.9
EMAX	15.5	3.5	18.4	4.1

that the heading bias behaves differently in these. In Fig. 4.7, the bias is almost consistent, while in Figs. 4.12 and 4.13 the bias and its direction varies more. This may be due to the differences in the test tracks. In Fig. 4.7, all the turns are 90° in size while the tracks in Figs. 4.12 and 4.13 include also turns of 180° . In these full turns, the user's movements are larger. This poses two risks: 1) the instantaneous

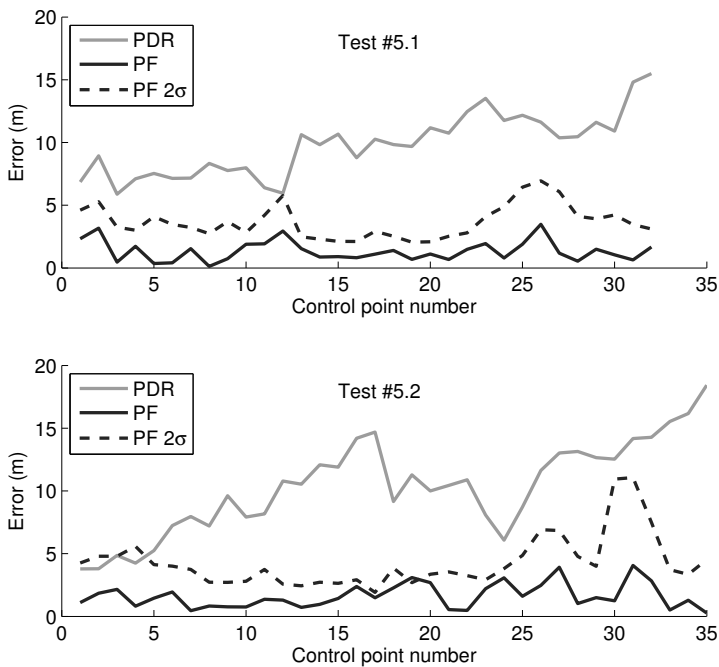


Fig. 4.14. Estimation errors at control points. *PDR*: unaided PDR estimate, *PF*: average position of particles, 2σ : $2 \times \text{RMS}$ error estimate based on the coordinate variances of the particles.

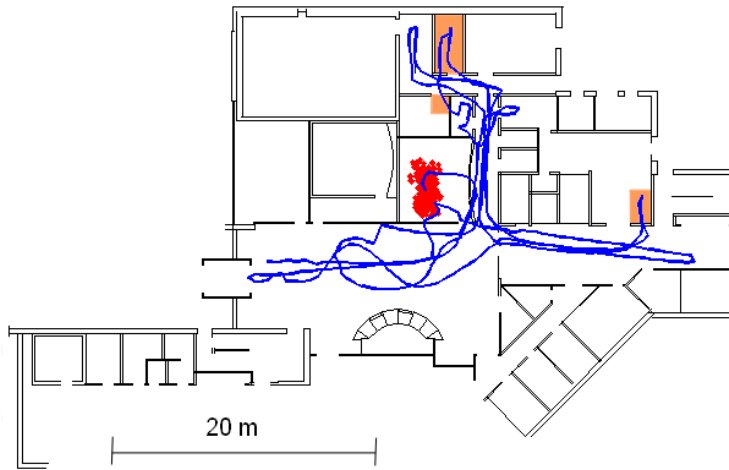


Fig. 4.15. Example track of real time demonstration.

angular rate may occasionally exceed the limits of the angular rate range of the gyro; 2) in larger movements, the sliding of the sensor unit with respect to the user body is more probable. These both result in sudden heading error changes that become visible after the turns.

The real time demonstration of the prototype navigation system was given indoors. The initial position in the front door of the building was obtained with the aid of GPS. The test person with the sensors attached to his belt walked around an office building in hallway, corridors, stairs, and several rooms. During the walk the sensor unit transmitted its PDR outputs to the server PC located in a meeting room.

Fig. 4.15 shows a track of a test route for a 6-minute walk. The starting point is on the left side at entrance door to hallway and the end point is in the middle of the figure in the meeting room where the red particle cloud is located. The shown track is the average of particles. A couple of times the test track seems to go through the walls. This happens because also the particles which die in the near future, i.e., particles that are not on the correct track, affect the average as well as the "proper" particles. There were three points where most of the particles were in a wrong room – these locations are indicated with the orange background color in the map. However, these particles died soon and the rest of the position track remained reliable. A video of the demonstration session is available in (TUT insnavgroup, 2013).

4.3.4 Map Processing and Computation Time

Three different methods were developed to decrease the computation time of obstacle crossing checks, described in Section 4.2.4: two methods for unstructured and one method for structured obstacle model. The speeding up of the algorithms for the obstacle crossing checks is based on screening out the unnecessary checkings that would not have any possibility to change the positioning results. Therefore, in all the tests regarding the computation time, the actual position estimates obtained with the modified faster algorithms were equal to the estimates obtained with the basic map processing without the processing time savings. In the following, the results are reported.

Line Grouping with Unstructured Obstacle Model

To group the obstacle lines, shown e.g. in Fig. (4.10), square-shaped cells were used. The groupings were formed with different cell widths, growing in 1 m steps from 2 to 41 m. To find the best cell width for the grouping, the particle filter using PDR, WLAN and map information was run in MATLAB to process all the measurement data of the data set #3. The run times of the computations were recorded, and the results are shown in Fig. 4.16. From the figure it can be concluded that the map data processing is the fastest when the cell width is 14 m. The grouping of map data clearly brings benefits: the run time without grouping, i.e., with the cell width 41 m when all the obstacle lines belong to the same group, is more than 400 s, while with the best grouping the run time is only 150 s.

For comparison, the computation times of other methods presented in this paper are listed here: position computation using WLAN data only takes 1 s, PRD preprocessing computations to detect steps, to estimate the step lengths, and to integrate the angular rates takes about 15 s for the whole data set, DR processing from step lengths and delta headings takes less than 0.001 s, and CEKF processing from step lengths, delta headings, and WLAN based positions takes about 0.2 s. The particle filter run times given in Fig. 4.16 do not include the run times of the necessary preprocessing phases, i.e., step detection, step length estimation, angular rate integration, or WLAN positioning.

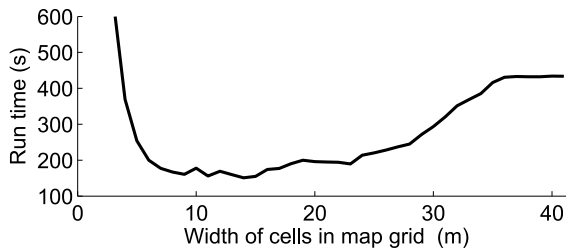


Fig. 4.16. Run time of the particle filter processing as a function of the width of the cells in map grid.

Line Re-Selection with Unstructured Obstacle Model

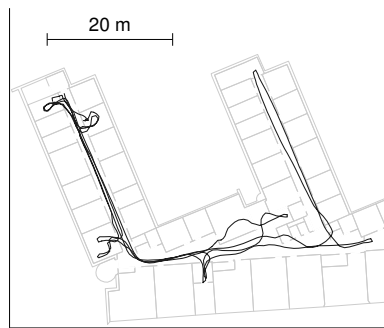
The line re-selection method was tested using the same data that was used in line grouping tests. Using the re-selection of line sets after each step, the computation time of the particle filter processing for PDR, WLAN and map information took 110 s; without WLAN information the time for particle filter processing was 104 s. Compared to the fastest line grouping requiring 150 s with both WLAN and map based positioning in use, the re-selection provides 26.7 % decrease in the computation time. This improvement provided extra margin for the real time operation in distributed indoor positioning system. Using this wall selection logic and running the MATLAB implementation of the particle filter on a laptop PC, it was found that a filter with 500 particles can perform all the needed computations in real-time between the consecutive IMU step data inputs.

Sequential Line Re-Selection with Structured Obstacle Model

The effect of the sequential line re-selection on the computation times with structured obstacle model was evaluated by comparing the sequential line re-selection, described in Algorithm 4.3, with the basic obstacle check algorithm, described in Section 2.3. In the latter, only the room information is used as the line selection criterion, i.e., all the polygon lines associated with the rooms along the particle transition are checked, while in the former only the lines which are inside or cross the 2D rectangular limits of the particle cloud positions are checked from the rooms that are visited along the particle transition. The algorithms were compared using the data set #4, comprising of two test walks, shown on the map in Fig. 4.17. In the tests, particle filters with 500 particles were used to process PDR and map data; WLAN



(a) Test route #4.1



(b) Test route #4.2

Fig. 4.17. Test data used to evaluate the computation times with sequential line re-selection.

positioning was not used.

The results of the computation time comparisons are shown in Table 4.6 where also the durations of the actual test walks are recorded as well as the number of steps detected by the PDR preprocessing. The number of steps is also the number of times the obstacle checks are performed for the particle cloud with these data. It can be

Table 4.6. Comparison between line selection algorithms

Test route	Data		Line selection algorithm and computation time		
	Duration of walk	Number of steps	Based on room id only	Sequential re-selection	Relative improvement
#1	441 s	661	526 s	107 s	79.6 %
#2	403 s	552	280 s	82 s	70.7 %

seen that with these data, the sequential re-selection of lines reduces significantly the computation time compared to the line selection based on the room only: the relative improvement is 70 % – 80 %. Comparing the computation times with the durations of the test walks it can also be noticed that without this improvement, on the test route #4.1 the map based particle filtering would not be able to operate in real time. It is also clear that the computation times of the test routes are not proportional to the numbers of steps on the routes: the number of obstacle line checks increases, if the variance of particle cloud increases, especially in the direction where there are more walls. The relative improvement in computation time obtained with sequential re-selection with test route #4.1 is larger than with test route #4.2 because on the route #4.1 the relative proportion of long corridors is higher.

4.4 Discussion

In this Chapter, the methods for combining information from inertial sensors, indoor map, and WLAN signals for pedestrian indoor navigation were proposed. The target applications of the methods include location based services and tracking of a user in an office environment or in open indoor spaces such as libraries or retail stores where fixed obstacles, such as shelves, divide the space into corridors. In these applications, room level accuracy or accuracy of 2–3 m is required and the processing speed of the algorithms must be high enough to allow real time processing.

Table 4.7. Summary of accuracy results from Sections 3.2 and 4.3

Data fusion algorithm	Information sources	Environment	User static / moving	Average distance error
unaided	WLAN	Office	Static	3.0–4.0 m
unaided	WLAN	Library	Static	6.0 m
unaided	WLAN	Library	Moving	7.1 m
unaided	PDR	Library	Moving	3.1 m
CEKF	PDR, WLAN	Library	Moving	1.9 m
Particle filter	PDR, map	Library	Moving	1.4 m
Particle filter	PDR, map, WLAN	Library	Moving	1.5 m
unaided	PDR	Office	Moving	≤ 10.3 m
Particle filter	PDR, map	Office	Moving	≤ 1.7 m

4.4.1 Results on Aided PDR

Table 4.7 summarizes the accuracies obtained in the field tests when different algorithms and different measurement sources have been used. The following observations can be made from these figures:

- The WLAN positioning for a static user is more accurate than for a moving user
- In the library, unaided PDR is more accurate than WLAN positioning, while in the office environment the WLAN positioning is more accurate
- Compared to the accuracies of the unaided PDR or WLAN positioning, combining these two using CEKF improves the accuracy
- As an aiding source for PDR, indoor map improves the accuracy more than WLAN positioning
- The average distance errors in the aided PDR algorithms meet the accuracy requirements

In the field tests, the comparisons of the different combinations of the available sensor information show that both the map information and WLAN signals can be used to improve the PDR estimate based on inertial sensors. In the comparison of the two alternative configurations of the CEKF, both configurations produced the same results with the nonlinear navigation model used in this thesis.

To enable the real-time operation of the system, the computational load caused by the map matching had to be decreased. The speeding up of the map checking algorithms was based on avoiding unnecessary line checks. With the developed algorithms, the computation times needed for the map checks were significantly reduced and the real-time functionality of the algorithms was demonstrated in the field tests. The speeding up of the map processing did not affect the position estimates. In this thesis, the speeding up of the map processing was performed only by improving the algorithms. In future, the speed could be further improved by implementing the computations using a programming language faster than MATLAB.

The inertial sensor unit used in these tests performed relatively well even as an unaided PDR system. However, fusing it with either WLAN positioning or with map

information improves accuracy. With map aided PDR the maximum position errors were below 4 m and average distance errors less than 2 m in the tests both in the library and in the office environment. The quality of the WLAN position data used in the library tests seemed to be quite poor: the average distance error was 7.1 m. Furthermore, the error for a moving user in the library is about 1 m larger than the error for a static user, which in turn is 2–3 m larger than the average error for the static user in office environment. Still the WLAN based position estimate in the library included some useful information for the data fusion filter. The WLAN based positioning was also complementary with map information: map information is relatively useless in open areas, where walls and obstacles cannot guide the particles, while in areas with high density of obstacles this information is frequently available and useful. Just the opposite, in areas dense with obstacles there are lot of disturbances present in WLAN signals which distort the positioning, while in open areas the quality of WLAN based position estimate is better. However, due to the special characteristics of the test area, it is difficult to generalize the results of this system directly to other types of environments without new field tests.

4.4.2 Future Developments

In the WLAN based likelihood computations of the particle filter, the WLAN based positioning was taken into account as estimated coordinates. This resembles loosely coupled integration architectures in integrated navigation systems including a GNSS receiver. Here, a tighter integration could be implemented using the approximated conditional probabilities of WLAN RSS stored in the radio map to determine the particle likelihoods. However, this approach would require some mechanism to map the particle positions with continuous range to the radio map CPs with discrete locations. The algorithm should also take into account the fact that the probabilities in the radio map are approximations and in addition to this, may include errors due to the temporal variations of the radio environment.

The accuracy of the map aided PDR, proposed in this Chapter, is sufficient also for providing reference position for example when collecting the calibration samples for fingerprinting with WLAN signals (Woodman and Harle, 2009) or magnetic field. On the other hand, when the fingerprint database exists, the fingerprints can be used to aid the initialization of the particle filter by providing a coarse initial distribution

of the position (Kemppi et al., 2010). This ability is also useful if the particles need to be re-initialized during the estimation (Nurminen et al., 2013).

Although the results of the map aided PDR were promising, the system has limitations. The usage of only one gyro does not allow real time tilt corrections of angular rate, the motion mode is restricted to normal walk, and orientation change of the gyro with respect to the walking person causes a bias change in heading. The first issue can be solved by using a 3-axis gyro. To mitigate the problems caused by other motion modes, such as climbing stairs or taking sidesteps, using a 3-axis gyro and combining the step detection based PDR and 6DOF mechanization of the IMU can be used (Davidson and Takala, 2013). The effects of large changes in gyro orientation are difficult to correct even for particle filter, where re-initialization may be required. A signal processing method to detect the orientation of the sensor unit with respect to the direction of the walk also exists (Parviainen and Collin, 2014). The models for step detection and step length estimation should also be verified with the larger group of users with varying properties, e.g. heights and ages.

In this work, the good quality of the gyro allowed the simplification of the process model by neglecting the gyro bias from the state vector. In the tests, the particle filter was able to take care of the effect of the remaining small bias component. However, the gyro bias state may be required with gyros with different specifications. The measurement range of the gyro used in the tests could be a limiting factor, as it is known that a pedestrian can perform very quick turns, where the instantaneous angular speed may exceed $300^\circ/s$.

In the testing of the indoor positioning method a single floor problem was assumed. With the used hardware, the system could have problems in multi-floor positioning tasks, especially if the floor plans of different floors do not differ significantly, and if the WLAN positioning is not available. Adding a barometer to the system would solve this problem.

5. CONCLUSIONS

In this thesis, methods for pedestrian indoor positioning were studied. With the proposed methods, the mobile unit performs all the required positioning measurements. The methods consist of WLAN fingerprinting based positioning, dead reckoning using gyro and accelerometers, indoor map, and the filtering algorithms for combining these to improve the positioning accuracy.

In WLAN fingerprinting, received signal strength measurements were used and two different types of algorithms were used for position estimation. One was based on the received signal strength averages stored in the radio map and pattern matching using kNN while the other was based on probabilistic models with histogram approximations of signal strength distributions. The dead reckoning was implemented using MEMS based sensor unit including a gyro and a 3-axis accelerometer. These were used to estimate the step length and the heading change during each step and these estimates were used to propagate the position estimate. The estimates of the step length and the heading change were computed in variable intervals which were determined by using step detection based on the acceleration signal pattern.

The WLAN positioning was combined with pedestrian dead reckoning using complementary extended Kalman filters in two different configurations. Particle filters were used to fuse the pedestrian dead reckoning with indoor map information, WLAN positioning, or both of them. For indoor map matching, both unstructured and structured obstacle maps were used.

5.1 *Main Results*

With map aided PDR navigation, the typical accuracy was couple of meters. The maximum position errors were below 4 m and RMS errors less than 2 m in the tests both in the library and in the office environment. In the library, where WLAN posi-

tioning was also available, the integration of WLAN with map aided PDR improved the overall accuracy, although the maximum error did not decrease significantly. In the library, the map information was complementary to WLAN: in open areas where there was no obstacles that could correct the estimates, the accuracy of WLAN was better. In the areas dense with obstacles the accuracy of WLAN positioning decreased, but the map corrections were available.

Three methods were tested for decreasing the computational load caused by the map matching. In field tests, map checks based on sequential re-selection of obstacle lines after each PDR step were used. The method was shown to decrease the computational load enough to allow real time navigation with 500 particles in the particle filter. The required computation speed was obtained with both the unstructured and structured obstacle maps.

Complementary EKF was used to combine the PDR with the WLAN positioning without map constraints. Algorithms for the two alternative complementary filter configurations were derived and implemented. For the process and measurement models used in this thesis, the estimates produced by both configurations were the same within the numerical accuracy of the computations.

Field tests on WLAN positioning were performed to determine the radio map configurations with reasonable positioning accuracy and moderate memory requirements. It was found that from two histogram based algorithms, the MEE algorithm was generally more accurate than the ML algorithm. The positioning accuracy improves when the number of histogram bins increases until it reaches 8. After that, increasing the number of bins does not improve the accuracy but increases the memory requirements of the radio map. With less than 6 histogram bins, the accuracy can be improved if, instead of having all bins with equal widths, there is one narrow bin for the minimum RSS. The best accuracy with histograms was obtained by combining the calibration samples measured from different directions in the radio map. The number of parameters for the histograms can be further decreased, if APs are correlating and they can be combined into one histogram. Using sample means as fingerprints and pattern matching with kNN, the positioning accuracy is better if there are separate fingerprints for each measurement direction.

In the WLAN positioning tests, the RSS normalization between different devices was obtained by determining the RSS offset for the radio map that provides the best posi-

tioning accuracy when RSS measurements of another device are used for positioning. It was also found that the positioning with a normalized radio map obtained using another device can be as accurate as the positioning with the radio map generated using the RSS measured by the same device that is used for positioning.

In this thesis, the following assumptions and simplifications were made: 1) the motion modes of the user are either normal walk or static – the other modes, such as climbing stairs or taking sidesteps, were not considered; 2) in the PDR processing, it was assumed that the orientation of the sensor unit with respect to the user body is fixed; 3) single floor navigation problem was assumed, methods for multi-floor navigation were not studied; 4) in the state models of the CEKF and particle filters no sensor errors or biases were included; 5) in fusion filters, the same WLAN device was used both to collect the radio map data and to obtain the RSS measurements in the positioning phase.

5.2 Future Work

Although this thesis provides promising results regarding the methods for pedestrian indoor positioning, further research is still required to develop these into reliable applications. In WLAN based positioning, compromises with the positioning accuracy may be required to obtain a method, where the scalability of radio map and the RSS normalization are reliably obtained using heterogeneous user devices.

For the data fusion of WLAN based positioning in the particle filter, a tighter integration model could be developed. In the likelihood computations of the measurement update, the tighter integration could use directly the approximated RSS probabilities conditioned to the location, stored in the histograms of the radio map.

In the PDR processing, a 3-axis gyro could be used to provide more accurate tilt corrections to the heading estimation. It would also allow running a 6DOF INS mechanization in parallel to the step detection based estimation. This combination could be used to detect unusual motion modes, such as climbing or side steps. To improve the accuracy of the estimation of the traveled distance, models to estimate the step lengths in more unusual motion modes are also needed. For multifloor positioning, a reliable floor detection is needed. This can be obtained for example by using barometer.

The optimal choices of methods and their parameters depend on the requirements of the application, and it is possible that the integrated indoor positioning system benefit from the deploying of some emerging new positioning techniques. These new techniques include, e.g., magnetic fingerprinting and generating fingerprint databases through crowdsourcing.

BIBLIOGRAPHY

- 9solutions. Homepage of the company. Referred 12 Feb 2015. [Online]. Available: <http://9solutions.com/en/>
- B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, Inc., 1979.
- M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM 2000*, vol. 2, Tel Aviv, Israel, Mar. 2000, pp. 775–785.
- Y. Bar-Shalom and X.-R. Li, *Estimation & Tracking: Principles, Techniques and Software*. Ybs Publishing, 1998.
- J. Bardwell. (2002, Nov.) Converting signal strength percentage to dBm values. Referred 12 Feb 2015. [Online]. Available: http://madwifi-project.org/attachment/wiki/UserDocs/RSSI/Converting_Signal_Strength.pdf?format=raw
- R. Battiti, T. L. Nhat, and A. Villani, "Location-aware computing: a neural network model for determining location in wireless LANs," University of Trento, Italy, Tech. Rep. DIT-02-0083, Feb. 2002.
- S. Beauregard, Widyawan, and M. Klepal, "Indoor PDR performance enhancement using minimal map information and particle filters," in *IEEE/ION PLANS 2008*, Monterey, CA, USA, May 2008, pp. 141–147.
- M. Z. H. Bhuiyan, H. Kuusniemi, L. Chen, L. Ruotsalainen, L. Pei, R. Guinness, and R. Chen, "Utilizing building layout for performance optimization of a multi-sensor fusion model in indoor navigation," in *Proc. ICL-GNSS 2012*, Starnberg, Germany, Jun. 2012, pp. 1–6.

- J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted centroid localization in ZigBee-based sensor networks," in *Proc. Conference or Symposium or Workshop*, Alcala de Henares, Spain, Oct. 2007, pp. 1–6.
- J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *Journal of Robotic Systems*, vol. 14, no. 4, pp. 231–249, Apr. 1997.
- R. Brito. (2009, Aug.) The algorithms bundle. File: algorithms.pdf. Referred 12 Feb 2015. [Online]. Available: <http://www.ctan.org/tex-archive/macros/latex/contrib/algorithms/>
- R. G. Brown, "Integrated navigation systems and kalman filtering: A perspective," *NAVIGATION*, vol. 19, no. 4, pp. 355–362, Winter 1972-73.
- R. Brown and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 3rd ed. John Willey & Sons, Inc., 1997.
- J. Candy, *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods*. John Willey & Sons, Inc., 2009.
- P. Castro, P. Chiu, T. Kremenek, and R. Muntz, "A probabilistic room location service for wireless networked environments," in *Proc. UbiComp '01*, Atlanta, GA, USA, Oct. 2001, pp. 18–34.
- M. Ciurana, S. Cugno, and F. Barcel-Arroyo, "WLAN indoor positioning based on TOA with two reference points," in *Proc. WPNC'07*, Hannover, Germany, Mar. 2007, pp. 23–28.
- J. Collin, P. Davidson, M. Kirkko-Jaakkola, and H. Leppäkoski, "Inertial sensors and their applications," in *Handbook of Signal Processing Systems*, S. S. Bhat-tacharyya, E. F. Deprettere, R. Leupers, and J. Takala, Eds. Springer New York, 2013.
- P. Davidson and J. Takala, "Algorithm for pedestrian navigation combining IMU measurements and gait models," *Gyroscopy and Navigation*, vol. 4, no. 2, pp. 79–84, 2013.

- K. Dennehy. (2014, Feb.) Indoor location has major growing pains, but big upside. GPS World, Wireless LBS Insider. Referred 12 Feb 2015. [Online]. Available: <http://gpsworld.com/indoor-location-has-major-growing-pains-but-big-upside/>
- C. di Flora and M. Hermersdorf, "A practical implementation of indoor location-based services using simple WiFi," *Journal of Location Based Services*, vol. 2, no. 2, pp. 87–111, Jun. 2008.
- H. Durrant-Whyte, "Multi sensor data fusion," Australian Centre for Field Robotics, The University of Sydney, Australia, Course notes, Jan. 2001.
- R. J. Eggert and J. F. Raquet, "Evaluating the navigation potential of the NTSC analog television broadcast signal," in *Proc. ION GNSS-2004*, Long Beach, CA, USA, Sep. 2004, pp. 2436–2446.
- J. Elwell, "Inertial navigation for the urban warrior," in *Proc. SPIE Conference on Digitalization of the Battlespace IV*, Orlando, FL, USA, Apr. 1999, pp. 196–204.
- A. Eronen, J. Leppänen, J. T. Collin, J. Parviainen, and J. Bojja, "Method and apparatus for determining environmental context utilizing features obtained by multiple radio receivers," United States Patent Application US20130053069 A1, 2013.
- F. Evennou, F. Marx, and E. Novakov, "Map-aided indoor mobile positioning system using particle filter," in *Proc. IEEE WCNC 2005*, vol. 4, New Orleans, LA, USA, Mar. 2005, pp. 2490 – 2494.
- R. Exel, G. Gaderer, and P. Loschmidt, "Localisation of wireless LAN nodes using accurate TDoA measurements," in *Proc. IEEE WCNC 2010*, Sydney, Australia, Apr. 2010, pp. 1–6.
- M. Falcone, S. Binda, E. Breeuwer, J. Hahn, E. Spinelli, F. Gonzalez, G. López Risueño, P. Giordano, R. Swinden, G. Galluzzo, and A. Hedquist, "Galileo on its own: First position fix," *Inside GNSS magazine*, vol. 8, no. 2, pp. 50–53,71, Mar./Apr. 2013. [Online]. Available: <http://www.insidegnss.com/magazine>
- L. Fang, P. J. Antsaklis, L. A. Montestruque, M. B. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Kouroulis, M. Haenggi, M. Xie, and X. Xie, "Design of a wireless assisted pedestrian dead reckoning system - the NavMote experience," *IEEE Trans. Instrumentation Measurement*, vol. 54, no. 6, pp. 2342–2358, Dec. 2005.

- J. Farrell and M. Barth, *The Global Positioning System and Inertial Navigation*. The McGraw-Hill Companies, Inc., 1998.
- E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Computer Graphics and Applications, IEEE*, vol. 25, no. 6, pp. 38–46, 2005.
- K. Frank, B. Krach, N. Catterall, and P. Robertson, "Development and evaluation of a combined WLAN & inertial indoor pedestrian positioning system," in *Proc. ION GNSS 2009*, Savannah, Georgia, USA, Sep. 2009, pp. 538–546.
- Q. Fu and G. Retscher, "Active RFID trilateration and location fingerprinting based on RSSI for pedestrian navigation," *The Journal of Navigation*, vol. 62, pp. 323–340, Apr. 2009.
- V. Gabaglio, "Centralised Kalman filter for augmented GPS pedestrian navigation," in *Proc. ION GPS 2001*, Salt Lake City, Utah, USA, Sep. 2001, pp. 312–318.
- S. Gezici, "A survey on wireless position estimation," *Wireless Personal Communications*, vol. 44, no. 3, pp. 263–282, Feb. 2008.
- G. Gibbons. (2012, Mar.) Munich summit charts progress of GPS, GLONASS, Galileo, Beidou GNSSes. Inside GNSS News. Referred 12 Feb 2015. [Online]. Available: <http://www.insidegnss.com/node/2981>
- P.-Y. Gilliéron, D. Büchel, I. Spassov, and B. Merminod, "Indoor navigation performance analysis," in *Proc. ENC-GNSS 2004*, Rotterdam, Netherlands, may 2004, pp. 1–9.
- G. Glanzer, T. Bernoulli, T. Wießflecker, and U. Walder, "Semi-autonomous indoor positioning using MEMS-based inertial measurement units and building information," in *Proc. Workshop on Positioning, Navigation and Communication*, Hannover, Germany, March 19 2009, pp. 135–139.
- S. A. Golden and S. S. Bateman, "Sensor measurements for Wi-Fi location with emphasis on time-of-arrival ranging," *IEEE Trans. on Mobile Computing*, vol. 6, no. 10, pp. 1185–1198, Oct. 2007.
- M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 2nd ed. John Wiley & Sons, Inc., 2001.

- P. D. Groves, G. W. Pulford, C. A. Littlefield, D. L. J. Nash, and C. J. Mather, "Inertial navigation versus pedestrian dead reckoning: Optimizing the integration," in *Proc. ION GNSS 2007*, Fort Worth, TX, USA, Sep. 2007, pp. 2043–2055.
- P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008.
- Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *Communications Surveys Tutorials, IEEE*, vol. 11, no. 1, pp. 13–32, 2009.
- A. Günther and C. Hoene, "Measuring round trip times to determine the distance between WLAN nodes," in *Proc. Networking 2005*, Waterloo, Canada, May 2005, pp. 1–12.
- T. D. Hall, C. C. C. III, and P. N. Misra, "Radiolocation using AM broadcast signals: Positioning performance," in *Proc. ION GPS-2002*, Portland, OR, USA, Sep. 2002, pp. 921–932.
- J. Hallberg, M. Nilsson, and K. Synnes, "Positioning with bluetooth," in *Proc. ICT 2003*, vol. 2, Papeete, Tahiti, French Polynesia, Mar. 2003, pp. 954–958.
- M. Hermersdorf, "Indoor positioning with a WLAN access point list on a mobile device," in *Proc. WSW'2006 at SenSys'2006*, Boulder, CO, USA, Oct. 2006, pp. 1–5.
- T. D. Hodes, R. H. Katz, E. Servan-Schreiber, and L. Rowe, "Composable ad-hoc mobile services for universal interaction," in *Proc. MobiCom '97*, Budapest, Hungary, Sep. 1997, pp. 1–12.
- V. Honkavirta, T. Perälä, S. Ali-Löytty, and R. Piché, "A comparative survey of WLAN location fingerprinting methods," in *Proc. WPNC'09*, Hannover, Germany, Mar. 2009, pp. 243–251.
- J. Hoshen, "Personal locator services emerge," *Spectrum, IEEE*, vol. 37, no. 2, pp. 41–48, Feb. 2000.
- IEEE, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11, 2012.

- IndoorAtlas. Homepage of the company. Referred 12 Feb 2015. [Online]. Available: <https://www.indooratlas.com/>
- Inside GNSS. (2014, Dec.) Errant Galileo FOC satellite signal helps provide first positioning. Inside GNSS. Latest News December 17, 2014. Referred 8 Aug 2015. [Online]. Available: <http://www.insidegnss.com/node/4337>
- ITU, *Probability distributions relevant to radiowave propagation modelling*, ITU Recommendation ITU-R P.1057-2, 2007.
- F. Izquierdo, M. Ciurana, F. Barceló, J. Paradells, and E. Zola, "Performance evaluation of a TOA-based trilateration method to locate terminals in WLAN," in *1st International Symposium on Wireless Pervasive Computing*, Phuket, Thailand, Jan. 2006, pp. 1–6.
- J. Jahn, U. Batzer, J. Seitz, L. Patino-Studencka, and J. Gutiérrez Boronat, "Comparison and evaluation of acceleration based step length estimators for handheld devices," in *Proc. Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, Zurich, Switzerland, sep 2010, pp. 1–6.
- J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1997.
- S. Ji, W. Chen, X. Ding, Y. Chen, C. Zhao, and C. Hu, "Potential benefits of GPS/GLONASS/GALILEO integration in an urban canyon – Hong Kong," *The Journal of Navigation*, vol. 4, pp. 681–693, Oct. 2010.
- S. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Robotics Research Group, Department of Engineering Science, University of Oxford, UK, Tech. Rep., Nov. 1996.
- S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
- T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice-Hall, Inc., 2000.
- R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transaction of the ASME, Journal of Basic Engineering*, pp. 35–45, Mar. 1960.
- E. Kaplan, *Understanding GPS: Principles and Applications*. Artech House, 1996.

- J. Käppi, J. Syrjärinne, and J. Saarinen, "MEMS-IMU based pedestrian navigator for handheld devices," in *Proc. ION GPS 2001*, Salt Lake City, UT, USA, Sep. 2001, pp. 1369–1373.
- P. Kemppi, T. Rautiainen, V. Ranki, F. Belloni, and J. Pajunen, "Hybrid positioning system combining anglebased localization, pedestrian dead reckoning and map filtering," in *Proc. IPIN 2010*, Zürich, Switzerland, Sep. 2010, pp. 1–7.
- M. I. Khan and J. Syrjärinne, "Investigating effective methods for integration of buildings map with low cost inertial sensors and Wifi-based positioning," in *Proc. IPIN 2013*, Montbéliard, France, Oct. 2013, pp. 1–8.
- M. Kirkko-Jaakkola, J. Collin, and J. Takala, "Using building plans and self-contained sensors with GNSS initialization for indoor navigation," in *Proc. IEEE VTC 2013 Spring*, Dresden, Germany, Jun. 2013, pp. 1–5.
- M. B. Kjærgaard, "A taxonomy for radio location fingerprinting," in *Proc. LoCA'07*, Oberpfaffenhofen, Germany, Sep. 2007, pp. 139–156.
- M. B. Kjærgaard, "Indoor location fingerprinting with heterogeneous clients," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 31–43, Feb. 2011.
- L. Koski, R. Piché, V. Kaseva, S. Ali-Löyty, and M. Hännikäinen, "Positioning with coverage area estimates generated from location fingerprints," in *Proc. WPNC 2010*, Dresden, Germany, Mar. 2010, pp. 99 – 106.
- L. Koski, T. Perälä, and R. Piché, "Indoor positioning using WLAN coverage area estimates," in *Proc. IPIN 2010*, Zurich, Switzerland, Sep. 2010, pp. 1 – 7.
- A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. Hämäläinen, "Experiments on local positioning with bluetooth," in *Proc. ITCC 2003*, Las Vegas, NV, USA, Apr. 2003, pp. 297–303.
- A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. Hämäläinen, "Positioning with IEEE 802.11b wireless LAN," in *Proc. IEEE PIMRC 2003*, vol. 3, Beijing, China, Sep. 2003, pp. 2218–2222.
- B. Krulwich. (2013, Apr.) Understanding apple buying WiFiSLAM and what's next. Seeking Alpha. Referred 12 Feb 2015. [Online]. Available: <http://seekingalpha.com/article/1320101-understanding-apple-buying-wifislam-and-what-s-next>

- H. Kuusniemi, M. Z. H. Bhuiyan, M. Ström, S. Söderholm, T. Jokitalo, L. Chen, and R. Chen, "Utilizing pulsed pseudolites and high-sensitivity GNSS for ubiquitous outdoor/indoor satellite navigation," in *Proc. IPIN 2012*, Sydney, Australia, Nov. 2012, pp. 1–7.
- Q. Ladetto, "On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering," in *Proc. ION GPS 2000*, Salt Lake City, UT, USA, Sep. 2000, pp. 1735–1740.
- C. Laoudias, R. Piché, and C. G. Panayiotou, "Device self-calibration in location systems using signal strength histograms," *Journal of Location Based Services*, vol. 7, no. 3, pp. 165–181, Aug. 2013.
- H. Leppäkoski and J. Takala, "Analysis of complementary Kalman filter configurations for navigation systems," in *Proc. ENC-GNSS 2007*, Geneva, Switzerland, May 2007, pp. 1365–1376.
- H. Leppäkoski, S. Tikkinen, A. Perttula, and J. Takala, "Comparison of indoor positioning algorithms using WLAN fingerprints," in *Proc. ENC-GNSS 2009*, Naples, Italy, May 2009, pp. 1–11.
- H. Leppäkoski, S. Tikkinen, A. Perttula, and J. Takala, "Normalization of signal strength measurements for WLAN based indoor positioning," in *Proc. ION GNSS 2009*, Savannah, GA, USA, Sep. 2009, pp. 494–502.
- H. Leppäkoski, S. Tikkinen, and J. Takala, "Optimizing radio map for WLAN fingerprinting," in *Proc. UPINLBS*, Kirkkonummi, Finland, Oct. 2010, pp. 1–8.
- H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals," in *Proc. ICASSP 2012*, Kyoto, Japan, Mar. 2012, pp. 1569–1572.
- H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals," *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 287–296, Jun. 2013.
- J. E. Lessin. (2013, Mar.) Apple acquires indoor location company WifiSLAM. The Wall Street Journal's Digits blog. Referred 12 Feb 2015. [Online]. Available: <http://blogs.wsj.com/digits/2013/03/23/apple-acquires-indoor-location-company-wifislam/>

- R. W. Levi and T. Judd, "Dead reckoning navigational system using accelerometer to measure foot impacts," U.S. Patent 5 583 776, 1996.
- B. Li, T. Gallagher, A. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?" in *Proc. IPIN 2012*, Sydney, Australia, Nov. 2012, pp. 1–9.
- H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.
- J. Liu, R. Chen, L. Pei, W. Chen, T. Tenhunen, H. Kuusniemi, T. Kröger, and Y. Chen, "Accelerometer assisted robust wireless signal positioning based on a hidden Markov model," in *Proc. IEEE/ION PLANS 2010*, Indian Wells, CA, USA, May 2010, pp. 488–497.
- P. Lou, H. Zhang, X. Zhang, M. Yao, and Z. Xu, "Fundamental analysis for indoor visible light positioning system," in *Proc. ICC 2012*, Beijing, China, Aug. 2012, pp. 59–63.
- G. Lui, T. Gallagher, B. Li, A. Dempster, and C. Rizos, "Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization," in *Proc. ICL-GNSS 2011*, Tampere, Finland, Jun. 2011, pp. 53–57.
- J. Machaj, P. Brida, and R. Piché, "Rank based fingerprinting algorithm for indoor positioning," in *Proc. IPIN 2011*, Guimarães, Portugal, Sep. 2011, pp. 1–6.
- R. Mautz, "Indoor positioning technologies," ETH Zurich, Switzerland, Habilitation Thesis, Feb. 2012.
- P. S. Maybeck, *Stochastic Models, Estimation, and Control*, ser. Mathematics in Science and Engineering. Academic Press, 1979, vol. 1.
- M. McCarthy, H. L. Muller, A. Calway, and R. E. Wilson, "Position and velocity recovery from independent ultrasonic beacons," in *Proc. EUSIPCO 2006*, Florence, Italy, Sep. 2006, pp. 1–5.
- J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Prentice-Hall, Inc., 1995.

- U. Meriheinä, "Method and device for measuring the progress of a moving person," U.S. Patent 7 962 309, 2007.
- O. Mezentsev, J. Collin, and G. Lachapelle, "Pedestrian dead reckoning – a solution to navigation in GPS signal degraded areas," *Geomatica*, vol. 59, no. 2, pp. 175–182, 2005.
- P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Ganga-Jamuna Press, 2006.
- V. Moghtadaiee, A. Dempster, and S. Lim, "Indoor localization using FM radio signals: A fingerprinting approach," in *Proc. IPIN 2011*, Guimarães, Portugal, Sep. 2011, pp. 1–7.
- B. Moyer. (2014, Jul.) And then there was one: InvenSense scoops up Movea, Trusted Positioning. EE Journal, feature article. Referred 12 Feb 2015. [Online]. Available: <http://www.eejournal.com/archives/articles/20140714-fusion>
- Murata Electronics, "SCC1300-D04 - Combined gyroscope and 3-axis accelerometer with digital SPI interfaces," Datasheet, Doc.Nr. 82113100, rev. D.
- K. Nur, S. Feng, C. Ling, and W. Ochieng, "Application of the improved FOCUSS for arrival time estimation (IFATE) algorithm to WLAN high accuracy positioning services," in *Proc. UPIN 2012*, Helsinki, Finland, Oct. 2012, pp. 1–8.
- K. Nur, S. Feng, C. Ling, and W. Ochieng, "Integration of GPS with a WiFi high accuracy ranging functionality," *Geo-spatial Information Science*, vol. 16, no. 3, pp. 155–168, 2013.
- H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization," in *Proc. IPIN 2013*, Montbéliard, France, Oct. 2013, pp. 1–10.
- H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E.-S. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks," in *Proc. IPIN 2012*, Sydney, Australia, Nov. 2012, pp. 1–9.

- D. Palmer, T. Moore, C. Hill, M. Andreotti, and D. Park, "Radio positioning using the digital audio broadcasting (DAB) signal," *The Journal of Navigation*, vol. 64, pp. 45–59, Jan. 2011.
- M. Panzarino. (2013, Mar.) What exactly WiFiSLAM is, and why apple acquired it. The Next Web. Referrer 12 Feb 2015. [Online]. Available: <http://thenextweb.com/apple/2013/03/26/what-exactly-wifislam-is-and-why-apple-acquired-it/>
- B. W. Parkinson and J. J. J. Spilker, *Global Positioning System: Theory and Applications, Volume II*. American Institute of Aeronautics and Astronautics, Inc., 1996.
- J. Partyka. (2014, Feb.) FCC ready for indoor location rules for 911 calls. GPS World, Wireless LBS Insider. Referred 12 Feb 2015. [Online]. Available: <http://gpsworld.com/fcc-ready-for-indoor-location-rules-for-911-calls/>
- J. Parviainen and J. Collin, "Method, apparatus and computer program product for orienting a smartphone display and estimating direction of travel of a pedestrian," United States Patent Application PCT/US2013/022075, 2014.
- L. Pei, R. Chen, Y. Chen, H. Leppäkoski, and A. Perttula, "Indoor/outdoor seamless positioning technologies integrated on smart phone," in *Proc. Advances in Satellite and Space Communications, 2009. SPACOMM 2009.*, Colmar, France, Jul. 2009, pp. 141–145.
- L. Pei, R. Chen, J. Liu, H. Kuusniemi, T. Tenhunen, and Y. Chen, "Using inquiry-based bluetooth RSSI probability distributions for indoor positioning," *Journal of Global Positioning Systems*, vol. 9, no. 2, pp. 122–130, 2010.
- A. Perttula, H. Leppäkoski, S. Tikkinen, and J. Takala, "WLAN positioning on mobile phone," in *Proc. IAIN World Congress*, Stockholm, Sweden, Oct. 2009, pp. 1–10.
- A. Perttula, H. Leppäkoski, M. Kirkko-Jaakkola, P. Davidson, J. Collin, and J. Takala, "Distributed indoor positioning system with inertial measurements and map matching," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 11, pp. 2682–2695, 2014.
- R. Prasad and M. Rugieri, *Applied Satellite Navigation Using GPS, GALILEO, and Augmentation Systems*. Artech House Publishers, 2005.

- P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, "On indoor position location with wireless LANs," in *Proc. IEEE PIMRC 2002*, vol. 2, Lisbon, Portugal, Sep. 2002, pp. 720–724.
- N. B. Priyantha, "The Cricket indoor location system," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, Jun. 2005. [Online]. Available: <http://nms.csail.mit.edu/papers/index.php?detail=132>
- S. Pullen. (2008, Sep.) What are the differences between accuracy, integrity, continuity, and availability, and how are they computed? Inside GNSS. GNSS Solutions. Referred 31 Aug 2015. [Online]. Available: <http://www.insidegnss.com/auto/sepoct08-gnssolutions.pdf>
- H. Qi and J. B. Moore, "Direct Kalman filtering approach for GPS/INS integration," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 2, pp. 687–693, Apr. 2002.
- M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C*, vol. 15, no. 5, pp. 312–328, oct 2007.
- Quuppa. Homepage of the company. Referred 12 Feb 2015. [Online]. Available: <http://quuppa.com/>
- M. Rabinowitz and J. J. Spilker, Jr., "Augmenting GPS with television signals for reliable indoor positioning," *NAVIGATION*, vol. 51, no. 4, pp. 269 – 282, Winter 2004-2005 2004.
- J. F. Raquet, M. M. Miller, and T. Q. Nguyen, "Issues and approaches for navigation using signals of opportunity," in *Proc. ION NTM 2007*, San Diego, CA, USA, Jan. 2007, p. 10731080.
- T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, Jul. 2002.
- L. Ruotsalainen, "Vision-aided pedestrian navigation for challenging GNSS environments," Ph.D. dissertation, Tampere University of Technology, Finland, Nov. 2013. [Online]. Available: <http://URN.fi/URN:ISBN:978-951-711-303-8>

- Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge University Press, 2008.
- Skyhook. Homepage of the company. Referred 12 Feb 2015. [Online]. Available: <http://www.skyhookwireless.com/>
- A. Smailagic and D. Kogan, "Location sensing and privacy in a context-aware computing environment," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 10–17, Oct. 2002.
- H. W. Sorenson, "Kalman filtering techniques," in *Advances in Control Systems*, C. T. Leondes, Ed. Academic Press Inc., 1966, vol. 3, pp. 219–292.
- R. Stirling, J. Collin, K. Fyfe, and G. Lachapelle, "An innovative shoe-mounted pedestrian navigation system," in *Proceedings of European Navigation Conference GNSS*, 2003, pp. 110–5.
- W. F. Storms and J. F. Raquet, "Magnetic field aided indoor navigation," in *Proc. ENC-GNSS 2009*, Naples, Italy, May 2009, pp. 1–9.
- J. Syrjärinne and J. Saarinen, "An evaluation of motion model structures within the IMM frame using range only measurements," in *Proc. of Int. Conference on Artificial Intelligence, Vol. I*, Las Vegas, NV, USA, Jun. / Jul. 1999, pp. 254–260.
- P. Thevenon, D. Serant, O. Julien, C. Macabiau, M. Bousquet, L. Ries, and S. Corazza, "Positioning using mobile TV based on the DVB-SH standard," *NAVIGATION*, vol. 58, no. 2, pp. 71–90, Summer 2011.
- Time Domain. Homepage of the company. Referred 12 Feb 2015. [Online]. Available: <http://www.timedomain.com>
- D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. The Institution of Engineering and Technology, 2004.
- TUT insnavgroup. (2013, Mar.) Sinetra demonstration video. Video. Referred 12 Feb 2015. [Online]. Available: <http://www.youtube.com/watch?v=ItA8Tj09Wzk>
- Ubisense. Homepage of the company. Referred 12 Feb 2015. [Online]. Available: <http://www.ubisense.net/en>

- M. J. Veth, "Navigation using images, a survey of techniques," *NAVIGATION*, vol. 58, no. 2, pp. 127–139, 2011.
- M. Wallbaum and S. Diepolder, "Benchmarking wireless LAN location systems," in *Proc. IEEE WMCS'05*, Munich, Germany, Jul. 2005, pp. 42–51.
- M. Wallbaum and T. Wasch, "Markov localization of wireless local area network clients," in *Wireless On-Demand Network Systems*, ser. Lecture Notes in Computer Science, R. Battiti, M. Conti, and R. Lo Cigno, Eds. Springer Berlin Heidelberg, 2004, vol. 2928.
- H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. Hanebeck, "WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors," in *Proc. WPNC'07*, Hannover, Germany, Mar. 2007, pp. 1–7.
- J. Wang, "Pseudolite applications in positioning and navigation: Progress and problems," *Journal of Global Positioning Systems*, vol. 1, no. 1, pp. 48–56, 2002.
- R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, Jan. 1992.
- Z. Weirong. (2010, May) Beidou satellite navigation system to cover whole world in 2020. China Military Online English Edition. Referred 12 Feb 2015. [Online]. Available: http://eng.chinamil.com.cn/news-channels/china-military-news/2010-05/20/content_4222569.htm
- Widyawan, M. Klepal, and S. Beauregard, "A novel backtracking particle filter for pattern matching indoor localization," in *MELT '08: Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, San Francisco, CA, USA, Sep. 2008, pp. 79–84.
- C. Wong, R. Klukas, and G. Messier, "Using WLAN infrastructure for angle-of-arrival indoor user location," in *IEEE VTC 2008-Fall*, Calgary, Canada, Sep. 2008, pp. 1–5.
- O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proc. 10th Int. Conf. on Ubiquitous Computing*, Seoul, Korea, Sep. 2008, pp. 114–123.

- O. Woodman and R. Harle, “RF-based initialisation for inertial pedestrian tracking,” in *Proc. Pervasive '09. The 7th International Conference on Pervasive Computing*, Nara, Japan, May 2009, pp. 238–255.
- R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato, “TDOA location system for IEEE 802.11b WLAN,” in *Proc. IEEE WCNC 2005*, vol. 4, New Orleans, LA, USA, Mar. 2005, pp. 2338–2343.
- Y. Yang, J. Farrell, and M. Barth, “High-accuracy, high-frequency differential carrier phase GPS aided low-cost INS,” in *Proc. of the IEEE PLANS 2000*, San Diego, CA, USA, Mar. 2000, pp. 148–155.
- M. Youssef, A. Agrawala, and A. Udaya Shankar, “WLAN location determination via clustering and probability distributions,” in *Proc. IEEE PerCom 2003*, Fort Worth, TX, USA, Mar. 2003, pp. 143–150.
- P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen, “IODetector: A generic service for indoor outdoor detection,” in *Proc. ACM SenSys 12*, Toronto, Ontario, Canada, Nov. 2012, pp. 113–126.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3596-3
ISSN 1459-2045